UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN**

**Promotion, Integration, Management and
Processing of Critical Inpatients' Open Big
Data Repositories**

PhD Thesis

Miguel Ángel Armengol de la Hoz

Madrid, 2020

Departamento de Tecnología Fotónica y Bioingeniería
Grupo de Bioingeniería y Telemedicina
E.T.S.I. de Telecomunicación

UNIVERSIDAD POLITÉCNICA DE MADRID

**Promotion, Integration, Management and Processing
of Critical Inpatients' Open Big Data Repositories**

TESIS DOCTORAL

Autor: **Miguel Ángel Armengol de la Hoz**

Directores: **Enrique J. Gómez Aguilera,**
Julio A. Mayol Martínez

Madrid, 2020

TRIBUNAL

Tribunal nombrado por el Magnífico y Excelentísimo Sr. Rector de la Universidad Politécnica de Madrid, el día 21 de Julio de 2020.

Presidente:	Dña. Elena Hernando Pérez	Doctor Ingeniero de Telecomunicación Catedrático de Universidad Universidad Politécnica de Madrid
Vocales:	Mr. Rodrigo Octavio Deliberato	MD, PhD Research Affiliate Massachusetts Institute of Technology
	D. Miguel Sánchez García	Doctor en Medicina Jefe de Servicio. Medicina Intensiva Hospital Clínico San Carlos
	Maria J. Martin-Bautista	Doctora en Informática Catedrático de Universidad Universidad de Granada
Secretario:	Dª. Ernestina Menasalvas Ruiz	Doctor Ingeniero Informático Catedrático de Universidad Universidad Politécnica de Madrid
Suplentes:	Mr. Mattieu Komorowski	MD, PhD Senior Lecturer Imperial College London
	Dª. María Luaces	Doctor en Medicina Directora Unidad de Innovación Hospital Clínico San Carlos

El Presidente

Los Vocales

El Secretario

ACKNOWLEDGMENTS

I wish to express my sincere appreciation to my PhD supervisors: Professor Enrique J. Gómez Aguilera, who convincingly guided and encouraged me to be precise and do the right thing even when the road got rough: without his persistent help, the goal of this project would have never been concluded and Professor Julio Mayol Martinez whose assistance was a critical milestone in the completion of this project.

I am also thankful to the physical and technical contribution of the Laboratory for Computational Physiology - Critical Data from the Massachusetts Institute of Technology, the Department of Anesthesia, Critical Care and Pain Medicine from Beth Israel Deaconess Medical Center at Harvard Medical School is truly appreciated. My colleagues from the Albert Einstein Hospital at São Paulo and Hospital Clínico San Carlos at Madrid Intensive Care Units: Dr. Rodrigo O. Deliberato, Dr. Matthieu Komorowski, Dr. Antonio Núñez and Miguel Sánchez. Without their support, confidence, know-how and resources, this project could not have reached its aim.

I would like to pay my special regards to Dr. Leo Anthony Celi, the assistance that he supplied providing invaluable access to both clinical and technical contacts, critical data resources and, above all, his medical research knowledge and trust during the development of this dissertation proved to be crucial in reaching a satisfactory outcome.

And finally, I also want to acknowledge the support and great love of my family, my girlfriend Claudia; my mother, Manuela; my sister, Teresa and my brother, Juanma. They kept me running on and this work would not have been possible without their tremendous support.

INDEX

TRIBUNAL.....	II
ACKNOWLEDGMENTS.....	II
INDEX.....	III
FIGURE INDEX	1
TABLE INDEX	2
SUMMARY	3
RESUMEN.....	5
I. INTRODUCTION	7
I.1. Decision Based Medicine in the Intensive Care Unit and the Operating Room	9
I.2. Open data, collaborative research and the reproducibility crisis.....	9
I.2.1. What is a datathon?	9
I.3. Big Data and Machine Learning.....	10
I.4. Thesis outline	11
II. RESEARCH HYPOTHESIS AND OBJECTIVES.....	14
II.1. Research Hypothesis	16
II.1.1. General Hypothesis	16
II.1.2. Specific Hypotheses.....	16
II.2. Objectives	18
II.2.1. General Objective	18
II.2.2. Specific Objectives	18
III. FRAMEWORK FOR DEVELOPING THE PHD THESIS.....	21
III.1. Promotion of Critical Inpatients' Open Big Data Repositories.....	24
III.2. Integration and management of Critical Inpatients' Open Big Data Repositories	25
III.3. Processing of Critical Inpatients' Open Big Data Repositories	25
IV. ASSESSING TEAM EFFECTIVENESS AND AFFECTIVE LEARNING IN A DATATHON	28
IV.1. Introduction and problem statement.....	30
IV.2. Methods	30
IV.2.1. Study design and participants selection	30
IV.2.2. Questionnaire and data collection	31
IV.2.3. Statistical analysis.....	33
IV.3. Results.....	33
IV.4. Discussion.....	34
V. FIRST BRAZILIAN DATATHON IN CRITICAL CARE.....	38
V.1. Introduction and problem statement.....	40
VI. BIG DATA AND MACHINE LEARNING IN CRITICAL CARE: OPPORTUNITIES FOR COLLABORATIVE RESEARCH	42
VI.1. Introduction and problem statement.....	44
VI.2. Collaborative work.....	46
VI.3. Databases of critically ill patients.....	47
VI.4. Initiatives in Spain.....	47
VI.5. Critical Care Datathon Madrid	48
VI.6. What the future holds and conclusions	50
VII. BIG DATA ANALYSIS AND MACHINE LEARNING IN INTENSIVE CARE UNITS	51
VII.1. Introduction and problem statement.....	53

VII.2. Methods	55
VII.2.1. Data input	55
VII.2.2. Processing of the data in vivo	56
VII.2.3. Data storage	56
VII.2.4. What is Big Data?	57
VII.2.5. What is Machine Learning?	58
VII.2.6. Difficulties and strategies for applying these techniques in a Department of Intensive Care Medicine	61
VII.2.7. Shared databases	61
VII.2.8. Natural language processing	63
VII.2.9. Retrospective Big Data analysis versus randomized clinical trials	64
VII.2.10. Ethical and legal issues	64
VIII. SEVERITY OF ILLNESS SCORES MAY MISCLASSIFY CRITICALLY ILL OBESE PATIENTS	67
VIII.1. Introduction and problem statement	69
VIII.2. Methods	69
VIII.2.1. Data Sources	69
VIII.2.2. Study Population	70
VIII.2.3. Study Variables	70
VIII.2.4. Statistical Analyses	70
VIII.3. Results	72
VIII.4. Discussion	75
IX. DEVELOPING WELL-CALIBRATED ILLNESS SEVERITY SCORES FOR DECISION SUPPORT IN THE CRITICALLY ILL	79
IX.1. Introduction and problem statement	81
IX.2. Methods	82
IX.2.1. Data source	82
IX.2.2. Ethical approval	82
IX.2.3. Cohort feature extraction	82
IX.2.4. Training and validation split	84
IX.2.5. Modeling procedures	84
IX.2.6. Model performance evaluation	85
IX.2.7. Statistical analysis	86
IX.2.8. Software	86
IX.2.9. Reporting summary	86
IX.2.10. Data availability	86
IX.2.11. Code availability	86
IX.3. Results	87
IX.3.1. Cohort overview	87
IX.3.2. Linear models	90
IX.3.3. Sequential models	92
IX.3.4. Gradient boosting machine	94
IX.4. Discussion	95
X. TEMPORAL TRENDS IN CRITICAL CARE OUTCOMES IN UNITED STATES MINORITY SERVING HOSPITALS	99
X.1. Introduction and problem statement	101
X.2. Methods	101
X.2.1. Data Source	101
X.2.2. Exposure	102
X.2.3. Outcomes	102
X.2.4. Variables	102

X.2.5.	Analysis.....	102
X.3.	Results.....	103
X.3.1.	Usage and characteristics of minority serving hospitals	103
X.3.2.	Temporal trends in critical care mortality.....	105
X.3.3.	Sensitivity analyses of mortality	107
X.3.4.	Temporal trends in ICU and hospital lengths of stay	108
X.3.5.	Temporal trends in delay to ICU transfer in patients admitted from the emergency department 110	
X.4.	Discussion.....	110
XI.	INCIDENCE AND RISK MODEL DEVELOPMENT FOR SEVERE TACHYPNEA FOLLOWING TERMINAL EXTUBATION	112
XI.1.	Introduction and problem statement.....	114
XI.2.	Methods	115
XI.2.1.	Data Source	115
XI.2.2.	Study population	115
XI.2.3.	Data Elements.....	115
XI.2.4.	Outcomes	116
XI.2.5.	Statistical Analysis.....	116
XI.3.	Results.....	117
XI.4.	Discussion.....	121
XII.	ADMISSION DISORIENTATION AND RISK OF CODE STATUS CHANGE.....	124
XII.1.	Introduction and problem statement.....	126
XII.2.	Methods	126
XII.2.1.	Data Source	126
XII.2.2.	Exposure.....	127
XII.2.3.	Outcome	127
XII.2.4.	Analysis.....	127
XII.3.	Results.....	127
XII.4.	Discussion.....	129
XIII.	ALTERED CONSCIOUSNESS MEDIATES THE EFFECT OF HYPERNATREMIA BUT NOT HYPONATREMIA, ON MORTALITY	130
XIII.1.	Introduction and problem statement.....	132
XIII.2.	Methods	132
XIII.2.1.	Data Source	132
XIII.2.2.	Exposures	133
XIII.2.3.	Outcome	133
XIII.2.4.	Variables.....	133
XIII.2.5.	Statistics	133
XIII.3.	Results.....	134
XIII.3.1.	Baseline characteristics.....	134
XIII.4.	Discussion.....	137
XIV.	ASSOCIATION OF INTRAOPERATIVE HYPOTENSION WITH POSTOPERATIVE STROKE, ACUTE KIDNEY INJURY, AND DEATH IN PATIENTS HAVING CARDIAC SURGERY WITH CARDIOPULMONARY BYPASS	139
XIV.1.	Introduction and problem statement.....	140
XIV.2.	Methods	140
XIV.2.1.	Study design and participants	140
XIV.2.2.	Anesthesia management.....	142
XIV.2.3.	Hemodynamic monitoring	142

XIV.2.4.	Vasopressor-inotrope treatment	142
XIV.2.5.	BP threshold for IOH	142
XIV.2.6.	IOH during CPB	143
XIV.2.7.	Study Outcomes	143
XIV.2.8.	Statistical Analysis	143
XIV.3.	Results.....	144
XIV.3.1.	Baseline characteristics.....	144
XIV.3.2.	Postoperative major adverse events (MAE).....	145
XIV.3.3.	IOH duration, AUC-MAP < 65mmHg and vasopressor-inotrope dose.....	146
XIV.3.4.	IOH and vasopressor-inotrope dose association with MAEs.....	147
XIV.3.5.	Fraction of IOH during CPB and its association with MAEs	151
XIV.4.	Discussion.....	153
XV.	MACHINE LEARNING MODELS WITH PREOPERATIVE RISK FACTORS AND INTRAOPERATIVE HYPOTENSION PARAMETERS PREDICT MORTALITY FOLLOWING CARDIAC SURGERY	156
XV.1.	Introduction and problem statement.....	158
XV.2.	Methodology	159
XV.2.1.	Study design and participants	159
XV.2.2.	Perioperative management	159
XV.2.3.	Factors used as predictors in machine learning models	159
XV.2.4.	IOH definition and characterization	160
XV.2.5.	Vasopressor-Inotrope dose	160
XV.2.6.	Primary outcome.....	161
XV.2.7.	Model training and validation	161
XV.2.8.	Model performance	162
XV.2.9.	Data imputation	162
XV.2.10.	Statistical Analysis.....	163
XV.3.	Results.....	163
XV.3.1.	Baseline characteristics.....	163
XV.3.2.	Postoperative mortality	165
XV.3.3.	IOH and vasopressor-inotrope dose.....	166
XV.3.4.	Predictive ability of models.....	166
XV.4.	Discussion.....	170
XVI.	ASSOCIATION OF CHLORIDE ION AND SODIUM-CHLORIDE DIFFERENCE WITH ACUTE KIDNEY INJURY AND MORTALITY IN CRITICALLY ILL PATIENTS	172
XVI.1.	Introduction and problem statement.....	174
XVI.2.	Materials and methods	174
XVI.2.1.	Study population and data source	174
XVI.2.2.	Study variables	175
XVI.2.3.	Statistical Analyses	175
XVI.3.	Results.....	176
XVI.3.1.	Participant characteristics	176
XVI.3.2.	Univariate analyses of primary and secondary outcomes	179
XVI.3.3.	Multivariable analyses	183
XVI.4.	Discussion.....	184
XVI.5.	Conclusions	186
XVII.	CONCLUSIONS AND FUTURE WORKS	187
XVII.1.	Publications	189
XVII.2.	Research hypotheses verification.....	190

XVII.2.1.	General Hypothesis	190
XVII.2.2.	Specific Hypothesis	191
XVII.3.	Future Works	196
XVII.3.1.	Social aspects: Promotion of Critical Inpatients' Open Big Data Repositories	196
XVII.3.2.	Technological aspects: Integration and management of Critical Inpatients' Open Big Data Repositories	196
XVII.3.3.	Clinical aspects: Processing of Critical Inpatients' Open Big Data Repositories.....	197
REFERENCES		200
APPENDIX I: BIG DATA AND MACHINE LEARNING IN CRITICAL CARE: OPPORTUNITIES FOR COLLABORATIVE RESEARCH		222
APPENDIX II: BIG DATA AND MACHINE LEARNING IN INTENSIVE CARE UNITS		223
Programming languages.....		223
Table 1_Sup. Machine Learning Methodologies		224
APPENDIX III: SEVERITY OF ILLNESS SCORES MAY MISCLASSIFY CRITICALLY ILL OBESE PATIENTS		227
Code		227
Loading dataset.....		240
Data Transformation		240
Changing NA to 0 in the pressor_1stday column in the final dataset using the code below:		240
Correcting age		241
Adjusting height and BMI		241
Adjusting some variables like Sodium.....		241
Data exploration.....		241
Statistical analysis according to the variables distribution (Table 1)		241
Statistical analysis proportion comparisons (table 1)		242
QQ plots (to assess normality) for baseline labs on both obese and normal weight.....		244
QQ Plots for the difference (ICU-baseline)for obese and normal weight.....		244
Baseline comparison (lab results between normal weight and obese patients)		244
Creating the laboratory deviation column (ICU - baseline).....		245
Unadjusted Statistical analysis of the deviation.....		245
Creating log_bun , log_creatinine , deviation and t test.....		246
Merging the two files normal weight(final_2) with obese (final_4)		246
Adjusted laboratory deviation analysis		246
Creation of subset without any missing variable and analyzing the impact of the statistically significant deviations in the hospital mortality		249
Supplemental Digital Content		250
APPENDIX IV: DEVELOPING WELL-CALIBRATED ILLNESS SEVERITY SCORES FOR DECISION SUPPORT IN THE CRITICALLY ILL.....		258
Cohort Extraction		258
Environment		260
Materialized Views.....		260
Load Features		261
Inclusion / Exclusion.....		264
Cleaning, Formatting and Feature Engineering		267
Save Train/Test Split of Features and Label.....		271
Modeling		290
Model Analysis		290
Model Metrics		294
Models and Data		295

Sequential Model.....	296
Train Models	297
APPENDIX V: TEMPORAL TRENDS IN CRITICAL CARE OUTCOMES IN UNITED STATES MINORITY SERVING HOSPITALS	300
APPENDIX VI: INCIDENCE AND RISK MODEL DEVELOPMENT FOR SEVERE TACHYPNEA FOLLOWING TERMINAL EXTUBATION	301
Code	301
APPENDIX VII: ASSOCIATION OF INTRAOPERATIVE HYPOTENSION WITH POSTOPERATIVE STROKE, ACUTE KIDNEY INJURY, AND DEATH IN PATIENTS HAVING CARDIAC SURGERY WITH CARDIOPULMONARY BYPASS	393
STS (Society of Thoracic Surgeons) VERSION 2.81 Definitions for postoperative outcomes.....	394
Complications	394
Renal Failure	394
Stroke	394
Death	394
STS Dataset Notations.....	394
Code	395
APPENDIX VIII: MACHINE LEARNING MODELS WITH PREOPERATIVE RISK FACTORS AND INTRAOPERATIVE HYPOTENSION PARAMETERS PREDICT MORTALITY FOLLOWING CARDIAC SURGERY	495
STS (Society of Thoracic Surgeons) VERSION 2.81 Definitions for postoperative outcomes.....	495
Complications	495
Renal Failure	495
Stroke	495
Death	495
STS Dataset Notations.....	496
Extended description of ML models and their performance.....	497
Code	501
Data exploration.....	501
Dataset creation full.....	503
train_test_df_creation	511
dataset_creation_CPB	514
train_test_df_creation_CPB	517
dataset_creation_outsideCPB.....	519
train_test_df_creation_outsideCPB.....	523
STS Risk Performance.....	524
training_models_cpb.....	535
training_models_full.....	547
training_models_outsidecpb	558
APPENDIX IX ASSOCIATION OF CHLORIDE ION AND SODIUM-CHLORIDE DIFFERENCE WITH ACUTE KIDNEY INJURY AND MORTALITY IN CRITICALLY ILL PATIENTS	572
Tables.....	572
Figures.....	580
Code	581
Dataset creation and analysis (R+SQL)	581

FIGURE INDEX

Figure 1. Dataflow in intensive care medicine.	54
Figure 2. Typical structure of a data frame	58
Figure 3. One of the first examples of the application of BDA to clinical research.	65
Figure 4. Patient fluxogram.	72
Figure 5. Fluxogram. Cohort selection process	88
Figure 6. Reliability curves for the APACHE IVa and Logit models.....	91
Figure 7. Receiver operating characteristic and precision-recall curves.	92
Figure 8. Reliability curves for sequential models.....	93
Figure 9. Reliability curves for the extreme gradient boosting model.....	95
Figure 10. Incidence of Critical illness mortality	105
Figure 11. Adjusted hazard ratio (95% CI) of hospital mortality per additional calendar year of admission between 2006 and 2016, stratified by ethnicity.	107
Figure 12. Temporal trends in hospital and ICU length of stay in minority- and non-minority-serving hospitals.	109
Figure 13. Number of patients with post-extubation tachypnea by time.	117
Figure 14. Kaplan-Meier survival curves by anticipatory dosing of opiates pre-extubation.	121
Figure 15. Potential mechanistic linking hyponatremia with increased mortality.	135
Figure 16. Effect of consciousness on mortality rates in dysnatremia.....	136
Figure 17. Mediating effect of decreased consciousness on the association between sodium concentrations and mortality.	137
Figure 18. Flow chart presenting patient selection and analysis.....	141
Figure 19. Association between AUC-MAP < 65 mmHg, vasopressor-inotrope dose, and Composite Outcome for Renal Failure, Death and Stroke.	147
Figure 20. Association between intraoperative hypotension during CPB and outcome (composite outcome, renal failure, stroke & death).....	152
Figure 21. Flow chart presenting patient selection and analysis.	164
Figure 22. Area Under the Receiver Operating Characteristic curves for Calibrated versus non-calibrated eXtreme Gradient Boosting with outside-CPB phase data.	168
Figure 23. AUROC comparison.....	169
Figure 24. Calibration plot.	169
Figure 25. Incidence of acute kidney injury among stratified groups based on main strong ion difference.	180
Figure 26. Incidence of acute kidney injury among stratified groups based on serum chloride ion concentration.	181
Figure 27. ICU mortality among stratified groups based on mSID.	182
Figure 28. ICU mortality among stratified groups based on [Cl-].....	183

TABLE INDEX

Table 1. Datathon HIAE's results	32
Table 2. Means and Pearson's correlation coefficient matrix for simple correlations on teamwork variables.	34
Table 3. Pearson's correlation coefficient matrix for simple correlations on Affective Learning and Teamwork Effectiveness Leadership, Mean and Standard Deviation.	35
Table 4. Regression Models Affective Learning on Real Life engagement behavior outcome. ...	36
Table 5. Machine learning methodologies	46
Table 6. Steps in constructing a Machine Learning system.	60
Table 7. Aspects of clinical research studied using Big Data Analysis and Machine Learning techniques.	63
Table 8. Baseline Demographic Characteristics	74
Table 9. Laboratory Results at Baseline and Change in Results Between ICU and Baseline	75
Table 10. Multivariable Linear Regression of Laboratory Deviation	76
Table 11. General overview of the cohort features	90
Table 12. The XGB model hyperparameters as selected by ten-fold cross-validation XGB extreme gradient boosting.	94
Table 13. Baseline Patient and Hospital Characteristics	104
Table 14. Hazard Ratio (95% Confidence Interval) of Critical Illness Mortality per Admission Year Category Provided, with 2006–2008 Considered as Reference for All Analyses.	106
Table 15. Change (95% CI) in Length of ICU and Critical Illness Hospital Stay (in Days),.....	108
Table 16. Adjusted Change (95% CI) per Additional Calendar Year in the Length (in Days) of ICU and Critical Illness Hospital Stay according to Ethnicity.....	109
Table 17. Patient characteristics and their unadjusted association with severe tachypnea episodes within 1-hour, and 6-hours of extubation.	118
Table 18. Independent predictors* of severe tachypnea at 1-hour and 6-hours of PVW.....	120
Table 19. Patient and hospital characteristics.	128
Table 20. Adjusted association between sodium concentration, consciousness, and mortality in the critically ill.....	134
Table 21. Demographics and Clinical Characteristics stratified by Major Adverse Events (MAE)*	145
Table 22. Multivariable logistic regression model evaluating the association between intraoperative hypotension duration MAP <65mmHg, vasopressor-inotrope dose and Major Adverse Events (MAE)*	149
Table 23. Multivariable logistic regression model evaluating the association between intraoperative hypotension AUC-MAP <65mmHg, vasopressor-inotrope dose and Major Adverse Events (MAE)*	150
Table 24. Model evaluating the association between intraoperative hypotension during CPB and outcomes (composite, renal failure, stroke and death).	151
Table 25. Phase-specific factors used as predictors for machine learning models	160
Table 26. Baseline patient and surgical characteristics stratified by mortality	165
Table 27. Ability of models in predicting mortality based on specific phases of surgery.....	167
Table 28. Comparison of patients' (A) demographics and (B) hemodynamic and laboratory values based on main strong ion difference classification	178
Table 29. Comparison of outcomes based on main strong ion difference classification.....	179

SUMMARY

The results of the research work carried out in this doctoral thesis can be summarized in three main areas:

1. **Social aspects:** Promoting and boosting the use and integration of open clinical data repositories through datathons and scientific publications: in this sense we participated in several datathons in Asia, Europe, North America and South America. During the development of this dissertation we were able to overcome different organizational, bureaucratic, cultural and professional barriers in the field of study of the promotion of Large Open Data Warehouses of critical patients enhancing the transference of knowledge between the clinical-technical world and the society.

This work has led to the publication of two manuscripts so far.

2. **Technological aspects:** Define a basic layout that enhances the design of integrated, normalized and structured open and de-identified high-resolution clinical big data repositories.

In addition, we were able to conduct a review on how the integration and management of large open critical patient data repositories can help drive critical patient research by working closely with critical care unit members to address their unmet needs and focus on delivering better value to their patients by means of cutting edge technologies such as big data and machine learning.

This work has led to the publication of two manuscripts so far.

3. **Clinical aspects:** Using innovative statistical and machine learning techniques to mine data in order to answer relevant clinical questions about critical inpatients employing state-of-the-art knowledge. Finally, but most importantly, we address specific clinical issues in the two hospital departments where we focus our efforts:

- *Intensive Care Unit:* we helped gather more knowledge about the obesity paradox, we study and improved the way severity scores are developed while providing open access to our exceptionally performing score for decision support, we studied how critical care outcomes differ throughout the United States geography, we developed a model for severe tachypnea risk in terminal patients, we address admission disorientation and risk of code status change, we studied how GCS mediates the baseline sodium effect when predicting mortality, finally we studied the association of chloride ion and sodium-chloride difference with acute kidney injury and mortality in critically ill patients.
- *Operating Room:* we addressed the association of intraoperative hypotension with postoperative stroke, acute kidney injury and death in patients having cardiac surgery with cardiopulmonary bypass and in a more analytically advanced version of this project we trained Machine Learning models with preoperative risk factors and intraoperative hypotension parameters, so we were able to predict mortality following cardiac surgery.

This work led to the publication of eight manuscripts so far, but most importantly allowed us and our collaborating physicians to broaden the knowledge about critically ill patients and their treatment optimization. It also helped us acquire essential data analysis skills that now and in the foreseeable future will aid us tackle a wide variety of clinically relevant questions.

RESUMEN

Los resultados de la labor de investigación realizada en esta tesis doctoral pueden resumirse en tres áreas principales:

1. **Aspectos pedagógicos:** Promoción e impulso del uso e integración de repositorios de datos clínicos abiertos a través de datathons y publicaciones científicas: en este sentido hemos participado en varios datathons en Asia, Europa, América del Norte y América del Sur. Durante el desarrollo de esta tesis pudimos superar diferentes barreras organizativas, burocráticas, culturales y profesionales en el campo de estudio de la promoción de Grandes Depósitos de Datos Abiertos de pacientes críticos. Este trabajo ha llevado a la publicación de dos manuscritos hasta ahora.
2. **Aspectos tecnológicos:** Definición y revisión de una estructura básica que mejore el diseño de grandes repositorios de datos clínicos de alta resolución integrados, normalizados y estructurados, abiertos y anonimizados. Además, pudimos realizar un examen de la forma en que la integración y la gestión de grandes repositorios abiertos de datos de pacientes críticos puede ayudar a impulsar la investigación en este ámbito trabajando en estrecha colaboración con los miembros de la unidad de cuidados intensivos para atender sus necesidades no cubiertas y así poder centrarnos en ofrecer un mejor valor a sus pacientes mediante tecnologías de vanguardia, como son el big data y Machine Learning. Este trabajo ha llevado a la publicación de dos manuscritos hasta ahora.
3. **Aspectos clínicos:** Utilización de innovadoras técnicas estadísticas y de machine learning para extraer datos con el fin de responder a las preguntas clínicas pertinentes sobre los pacientes hospitalizados críticos empleando los conocimientos más avanzados.

Por último, pero más importante, abordamos cuestiones clínicas específicas en los dos departamentos del hospital en los que centramos nuestros esfuerzos:

- *Unidad de Cuidados Intensivos:* ayudamos a entender mejor la paradoja de la obesidad, estudiamos y mejoramos la forma en la que se elaboran las scores de gravedad al tiempo que proporcionamos acceso abierto a nuestro algoritmo de rendimiento excepcional para el apoyo de decisiones, estudiamos cómo difieren los tratamientos de los cuidados intensivos en toda la geografía de los Estados Unidos, desarrollamos un modelo para el riesgo de taquipnea grave en pacientes terminales, abordamos la desorientación en los ingresos y el riesgo de cambio de estado del nivel de intervenciones médicas que un paciente desea que se inicien si su corazón o su respiración se detienen, estudiamos cómo la escala de coma de Glasgow media el efecto del sodio basal al predecir la mortalidad, por último estudiamos la asociación del ion cloruro y la diferencia de cloruro de sodio con la lesión renal aguda y la mortalidad en los pacientes críticos.

- *Quirófano:* abordamos la asociación de la hipotensión intraoperatoria con el accidente cerebrovascular postoperatorio, la lesión renal aguda y la muerte en pacientes sometidos a cirugía cardíaca con baipás cardiopulmonar y en una versión más avanzada analíticamente de este proyecto entrenamos modelos de machine learning con factores de riesgo preoperatorios y parámetros de hipotensión intraoperatoria, para ser capaces de predecir la mortalidad después de la cirugía cardíaca.

Este trabajo condujo a la publicación de ocho manuscritos hasta el momento, pero lo más importante fue que nos permitió a nosotros y a los médicos con los que colaboramos ampliar el conocimiento sobre pacientes en estado crítico optimizar su tratamiento. También nos ayudó a adquirir habilidades esenciales de análisis de datos que ahora y en el futuro previsible nos ayudarán a abordar una amplia variedad de cuestiones clínicamente relevantes.

I. INTRODUCTION

I.1. Decision Based Medicine in the Intensive Care Unit and the Operating Room

In a fast-evolving world where computer prices have dropped and processing and storing power has increased exponentially, the possibilities and potential of healthcare data science seem to be unlimited. Even though the massive digitalization of information in the healthcare field has made it possible to collect and store an unprecedented quantity of data, the current state of medical knowledge does not provide the guidance to make most clinical decisions. In fact, only 10-20% of clinical decisions are evidence based (Darst et al., 2010).

Thus, there is a significant gap between our ability to collect, organize, store, and access data on patient care, on the one hand, and our ability to apply the data to patients as they are newly admitted to hospitals, on the other. Ideally, we would be able to provide healthcare that accurately reflects the state-of-the-art of our collected and available knowledge.

Among all the departments where we could begin to apply the cutting-edge technologies of data collection and information storage, Intensive Care Units (ICU) come most readily to mind. The patient admitted to the ICU is continuously being monitored from the moment of admission. When trying to tackle clinical problems that affect critical patients, we already have every single sensor we may need; therefore, we can collect high-resolution and reasonably well-structured data, so there are no extra wearables or devices needed. Thus, we have all the tools and raw materials we may need to start performing our job as data scientists or researchers. For all the above reasons the ICU is one logical place to start applying data science and machine learning.

The same activity of data collection occurs in the Operating Room (OR). In digitized hospitals such as the ones we have been working in during our stay in Boston since 2016 (Beth Israel Deaconess Medical Center and Massachusetts General Hospital), every single action the healthcare providers (anesthesiologists, surgeons, and nurses) carry out is recorded and time-stamped, resulting in high resolution databases of procedures, vital signs, and medications.

I.2. Open data, collaborative research and the reproducibility crisis.

In order to foster open-data initiatives and promote the creation of transparent algorithms, the scientific publications and projects that adhere to this dissertation make all the necessary code to analyze the addressed data publicly available. By making the data publicly available, we will help address the reproducibility crisis and improve how predictive models generalize regardless of where the patient being treated is from. Nonetheless, to be able to access open datasets and work in a multidisciplinary collaborative environment, we need events that can have an important impact in society and change the research culture such as datathons.

I.2.1. What is a datathon?

A datathon is a sprint-like event in which participants, as members of a team, collaborate intensively with their teammates to answer a clinical question using large clinical care datasets. The goal of the datathon is to have a fun, creative, and innovative event that brings together experts and novices, clinicians, engineers, statisticians, researchers and data scientists to learn more about data and discover new knowledge in an interdisciplinary atmosphere that in some cases culminates in scientific publications. Ultimately datathons are a key factor in the challenging task of promoting and boosting an innovative and collaborative culture around the use and integration of open clinical data repositories.

I.3. Big Data and Machine Learning

Data, which has been called the new oil, is valuable, but like oil, requires refining. If unrefined, we cannot make use of it. Oil needs to be transformed into gas, plastic, chemicals, etc. Analogously, data must be broken down and “cleaned,” so that it is possible to gather knowledge from it. This transformation and data wrangling phase is one of the main subjects around which this dissertation is centered. The huge amount of data our hyperconnected world is generating every fraction of a second has found a perfect ally with the emergence of Big-Data. Because of the new data-science techniques now available, healthcare research is not limited to statistical analysis (testing hypotheses with the support of mathematical methods) but, rather, can be involved in the task of generating knowledge and storing it in some usable form within the memory of a computer (Krumholz, 2014). This is what we call machine learning, a specific kind of Artificial Intelligence (AI).

Machine learning is the art of teaching computers how to learn by mean of algorithms. Rather than the generally held belief that machine learning can replace physicians, machine learning will be able to perform the tedious tasks so that physicians can focus on what they do best: taking care of the patient. Machine learning will not take physicians’ jobs away, but it will improve their routine by freeing them from the tedious part of their job: expending time on the computer. In the upcoming decades we are not going to witness a battle between clinicians and computers but a battle between clinicians who have access to machine learning and clinicians who lack this access.

Regardless of which machine learning or statistical method we are using in order to obtain useful knowledge from data by means of machine learning, there are 7 steps that must be followed (Yufeng, 2017):

- Gathering data
- Preparing that data
- Choosing a model
- Training
- Evaluation
- Hyperparameter tuning
- Prediction.

This dissertation is focused on the exploitation of cutting-edge techniques with the aim of bridging the gap between research and healthcare in order to solve real clinical problems by improving the

existing knowledge on the critical inpatient and the application of this knowledge to the patient's care.

I.4. Thesis outline

The dissertation is structured according to each of the steps proposed on the conceptual framework, and thus every main chapter will cover the methodological aspects and results obtained for the framework implementation proposed.

Chapter 1 has established the foundations of this PhD, focusing on the relevant aspects of mis training and assessment. Challenge areas have been identified, and justification for the research established. The conceptual framework that will be validated in this PhD has been presented.

Chapter 2 presents the hypotheses that drive this research, as well as the specific objectives to meet in this PhD.

Chapter 3 will present the framework for developing the PhD dissertation, talking about the institutions where the research was carried out.

Chapter 4 will address the study of team effectiveness and affective learning in a datathon, this chapter was based on the manuscript de Toledo Piza, F. M., Celi, L. A., Deliberato, R. O., Bulgarelli, L., de Carvalho, F. R. T., Rabello Filho, R., Armengol de La Hoz, M. Á.... & Kesselheim, J. C. (2018). Assessing team effectiveness and affective learning in a datathon. *International Journal of Medical Informatics*, 112, 40-44.

Chapter 5 is an overview of the first Brazilian datathon in critical care, this chapter was based on the manuscript Serpa Neto, A., Kugener, G., Bulgarelli, L., Rabello Filho, R., Armengol de La Hoz, M. Á., D. L., Johnson, A. E., ... & Ferraz, L. J. R. (2018). First Brazilian datathon in critical care. *Revista Brasileira de terapia intensiva*, 30(1), 6-8.

Chapter 6 reviews how tools such as big data and machine learning have tremendous opportunities for collaborative research in critical care, this chapter was based on the manuscript Organizing Committee of the Madrid 2017 Critical Care Datathon, Núñez Reiz A, Martínez Sagasti F, Álvarez González M, Blesa Malpica A, Martín Benítez JC, Nieto Cabrera M, Del Pino Ramírez Á, Gil Perdomo JM, Prada Alonso J, Celi LA, Armengol de la Hoz MÁ, Deliberato R, Paik K, Pollard T, Raffa J, Torres F, Mayol J, Chafer J, González Ferrer A, Rey Á, González Luengo H, Fico G, Lombroni I, Hernandez L, López L, Merino B, Cabrera MF, Arredondo MT, Bodí M, Gómez J, Rodríguez A, Sánchez García M. Big data and machine learning in critical care: Opportunities for collaborative research. *Med Intensiva*. 2019 Jan-Feb;43(1):52-57. English, Spanish. doi: 10.1016/j.medin.2018.06.002. Epub 2018 Aug 2. PMID: 30077427.

Chapter 7 explains how big data analysis and machine learning can constitute a revolution in intensive care units, this chapter was based on the manuscript Reiz, A., Sanchez Garcia, M., Armengol de La Hoz, M. Á. Big Data Analysis and Machine Learning in Intensive Care Medicine. *Medicina Intensiva* (2018).

Chapter 8 tackles how severity of illness scores may misclassify critically ill obese patients, this chapter was based on the manuscript Deliberato, R. O., Ko, S., Komorowski, M., Armengol de La Hoz, M. Á., Frushicheva, M. P., Raffa, J. D., ... & Stone, D. J. (2018). Severity of illness scores may misclassify critically ill obese patients. *Critical care medicine*, 46(3), 394-400.

Chapter 9 develops a well-calibrated illness severity score for decision support in the critically ill, this chapter was based on the manuscript Cosgriff C. V. , Celi L. A., Ko S., Sundaresan T., Johnson A. , Pollard T., Armengol de la Hoz M. Á., Kaufman A. R., Stonen D. J., Badawi O., Deliberato R. O. .Developing Well Calibrated Illness Severity Scores for Decision Support in the Critically Ill. *Nature Digital Medicine*.

Chapter 10 reviews temporal trends in critical care outcomes in united states minority serving hospitals, this chapter was based on the manuscript Danziger, J., Armengol de La Hoz, M. Á., Li, W., Komorowski, M., Deliberato, R. O., Rush, B. N. M., Mukamal, K.J., Leo Celi, Badawi, O. Temporal trends in critical illness outcomes in United States minority serving hospitals.

Chapter 11 studies the incidence and risk model development for severe tachypnea following terminal extubation, this chapter was based on the manuscript Fehnel, C. R. †, de la Hoz, M. A. †, Celi, L. A., Campbell, M. L., Hanafy, K., Nozari, A., ... & Mitchell, S. L. (2020). Incidence and risk model development for severe tachypnea following terminal extubation. *Chest*.

Chapter 12 studies how admission disorientation can have an impact on the risk of code status change, this chapter was based on the manuscript Armengol de La Hoz, M.Á., Celi, L.A., Gómez, E.J., Danziger, J.. Admission Disorientation and Risk of Code Status Change.

Chapter 13 goes over how altered consciousness mediates the effect of hypernatremia but not hyponatremia for mortality prediction, this chapter was based on the manuscript Armengol de La Hoz, M.Á. †, Hao, S., Celi, L.A., Gómez, E.J.*, Danziger, J.*. Altered Consciousness Mediates the Effect of Hypernatremia but Not Hyponatremia, On Mortality.

Chapter 14 evaluates the association of intraoperative hypotension with postoperative stroke, acute kidney injury, and death in patients having cardiac surgery with cardiopulmonary bypass, this chapter was based on the manuscript Armengol de La Hoz, M. Á. †, Rangasamy, V.†,Novack V., Brenes Bastos A., Nabel S., Ramachandran S.K & Subramaniam B. Increasing Vasopressor Dose used to treat Intraoperative Hypotension and not the Duration of Hypotension is Strongly Associated with Adverse Outcomes following Cardiac Surgeries.

Chapter 15 develops machine learning models using preoperative risk factors and intraoperative hypotension parameters in order to predict mortality following cardiac surgery, this chapter was based on the manuscript Armengol de la Hoz, M. Á.† , Fernandes M.†, Rangasamy V., Subramaniam B. Machine Learning Models with Preoperative Risk Factors and Intraoperative Hypotension Parameters Predict Mortality Following Cardiac Surgery.

Chapter 16 studies the association of chloride ion and sodium-chloride difference with acute kidney injury and mortality in critically ill patients, this chapter was based on the manuscript Kimera S. †, Armengol de La Hoz, M. Á. †, Raines, N.H. , Celi, L.A., Association of Chloride Ion and Sodium-Chloride Difference with Acute Kidney Injury and Mortality in Critically Ill Patients .

Chapter 17 addresses all conclusions each of the manuscripts dealt with going over validation of the hypotheses and future works.

† Indicates co-first authors, * Indicates co-senior authors.

II. RESEARCH HYPOTHESIS AND OBJECTIVES

II.1. Research Hypothesis

II.1.1. General Hypothesis

Our main hypothesis is that an effective strategy of promotion, integration, management and processing of critical inpatients' Open Big Data Repositories can overcome decisive social, technological and clinical barriers. This can contribute to advance our current knowledge about critical inpatients and will ultimately lead to the enhancement of their quality of life.

II.1.2. Specific Hypotheses

II.1.2.1. Social aspects: Promotion of Critical Inpatients' Open Big Data Repositories

- H#1: We hypothesized that a datathon can promote affective learning and effective teamwork and that we can study it using descriptive statistics.
- H#2: We hypothesized that effective teamwork during a datathon is associated with affective learning and that we can study it using multivariable regression models.

II.1.2.2. Technological aspects: Integration and management of Critical Inpatients' Open Big Data Repositories

- H#3: We hypothesized that using data governance, quality check, quality assurance, normalization, standardization and de-identification techniques performed on the repositories; the data we are going to analyze will meet the desired quality standards, standardization and anonymization requirements.
- H#4: Intensive care is an ideal environment for the use of Big Data Analysis (BDA) and Machine Learning (ML), due to the huge amount of information processed and stored in electronic format in relation to such care. We hypothesized these tools can improve our clinical research capabilities and clinical decision making.

II.1.2.3. Clinical aspects: Processing of Critical Inpatients' Open Big Data Repositories

- H#5: We hypothesized that within the same severity of illness category, obese patients may have inherently different degrees of deviation from their usual state during critical illness and that we can fit a model to prove it.

- H#6: While high-risk patients in ICU cohorts represent a distinct and identifiable group of patients, prior severity scores have been built with methods that are too simple and inflexible to learn to forecast mortality accurately across such heterogeneous groups; therefore we hypothesized that a more efficient severity score can be built using machine learning techniques.
- H#7: We hypothesized that there is a difference between hospitals which predominantly care for minority patients and non-minority hospitals regarding improvements in critical care outcomes and that we can study it using Cox proportional hazards models, multiplicative interactions, least squares regression, generalized estimating equations (GEE) with Poisson error distribution, log link function, and an exchangeable covariance structure.
- H#8: Little is known about factors associated with better symptom control among terminally ill patients undergoing this procedure. While evidence suggests lower rates of respiratory distress with gradual reduction of ventilator support versus immediate extubation it is not known whether administering analgesia/sedation pre-extubation relieves distress more effectively than giving these drugs only in response to observed symptoms we hypothesize we can study such relief using multivariable prediction models.
- H#9: While there are many factors that enter the decision to de-escalate intensity of critical illness care, the influence of mental status has not been previously directly measured. We hypothesized that disoriented patients were more likely to have code status change to Do not resuscitate/Intubate (DNR/I) or other type of care-de-escalation and that we can study such likelihood using multiplicative interactions.
- H#10: Although disorders of water balance have been associated with increased mortality, the mechanisms explaining the association are not clear. We hypothesized that decreased consciousness is a mediator on the association between dysnatremia and mortality and that we can study such mediated effect conducting a mediation analysis.
- H#11: Intra Operative Hypotension (IOH) (MAP <65 mmHg) even for only >10 minutes during CPB has been shown to increase the risk of stroke.²⁰ Vasopressors and inotropes are used to treat IOH during cardiac surgery. We believe there is an association between IOH and intraoperative vasopressor/inotrope use with postoperative outcomes.
- H#12: We hypothesized that incorporating adverse intraoperative factors that include Intraoperative hypotension (IOH) during and outside Cardiopulmonary Bypass (CPB) along with the preoperative risk factors into the machine learning models such as eXtreme gradient boosting or Random Forests will improve discrimination, specificity, sensitivity and positive predictive value for mortality prediction following cardiac surgery.
- H#13: Derangements of chloride ion concentration ($[Cl^-]$) have been shown to be associated with acute kidney injury (AKI) and other adverse outcomes. For a physicochemical approach, however, $[Cl^-]$ should be considered with sodium ion concentration ($[Na^+]$). We hypothesized there is an association of $[Cl^-]$ and the main strong ion difference ($mSID = [Na^+] - [Cl^-]$) during the first 24 hours after admission into intensive care unit (ICU) with the development of AKI and mortality.

II.2. Objectives

II.2.1. General Objective

Our aim is to employ different social, technical and methodological approaches in order to tackle specific and relevant clinical questions regarding critically ill inpatients to improve the knowledge we have about this high risk population group to ultimately be able to improve the outcomes of such patients in the Intensive Care Unit and the operating room so that we are able to bring better value to them. In order to do so we plan to study how the promotion of open big data repositories by means of events such as datathons and the integration and management of critical inpatients repositories can assist in the processing of these repositories using technologies such as machine learning and big data in order to solve relevant clinical questions and improve the existing knowledge on the critically ill both in the intensive care unit and the operating room.

II.2.2. Specific Objectives

II.2.2.1. Social aspects: Promotion of Critical Inpatients' Open Big Data Repositories

- O#1: We are going to explore the association between teamwork and learning in a health datathon.
- O#2: We are going to measure affective learning and investigate how it relates to teamwork.

II.2.2.2. Technological aspects: Integration and management of Critical Inpatients' Open Big Data Repositories

- O#3: We are going to ensure that the data analyzed meets specific quality standards, standardization and anonymization requirements through data governance, quality check, quality assurance, normalization, standardization and de-identification techniques performed on the repositories, detailed below in a conceptual manner and providing example code on how to carry out these tasks.
- O#4: We are performing a review about the foundations of Big Data analysis and Machine Learning and exploring possible applications in the critical care field from a clinical viewpoint.
- O#5: We are suggesting potential strategies to optimize Big Data analysis and Machine Learning.
- O#6: We are describing a new kind of hybrid healthcare-data science professional with a linking role between clinicians and data.

II.2.2.3. Clinical aspects: Processing of Critical Inpatients' Open Big Data Repositories

- O#7: Severity of illness scores rests on the assumption that patients have normal physiologic values at baseline and that patients with similar severity of illness scores have the same degree of deviation from their usual state. Prior studies have reported differences in baseline physiology, including laboratory markers, between obese and normal weight individuals, but these differences have not been analyzed in the ICU. We are going to compare deviation from baseline of pertinent ICU laboratory test results between obese and normal weight patients, adjusted for the severity of illness.
- O#8: We are going to compare a sequential modeling approach wherein an initial regression model assigns risk and all patients deemed high risk then have their risk quantified by a second, high-risk-specific, regression model to a logistic regression model and a sophisticated machine learning approach, the gradient boosting machine.
- O#9: We are going to examine whether improvements in critical care outcomes over the last decade extend to minority serving hospitals. Using a large repository of almost 1.1 million ICU admissions from hospitals across the United States, we are going to describe the temporal trends of critical illness outcomes according to hospital minority composition and whether these trends differed by ethnicity.
- O#10: We plan to describe the prevalence of severe tachypnea among patients who underwent PVW through development of multivariable prediction models.
- O#11: We are going to identify modifiable factors associated with the occurrence of severe tachypnea among patients who underwent PVW through development of multivariable prediction models.
- O#12: We are going to examine whether disoriented patients were more likely to have code status change to Do not resuscitate/Intubate (DNR/I) or other type of care-de-escalation.
- O#13: Given that disorders of water imbalance presumably associate with mortality through changes in neuronal size and function, we are going to examine the mediating effect of decreased consciousness on the association between dysnatremia and mortality.
- O#14: We aim to study the relationship between a) IOH (defined as a MAP <65 mmHg) and b) the fraction of total IOH occurring during CPB (fractionIOH-CPB) with a composite of three Major Adverse Events (stroke, acute kidney injury, and death) accounting for intraoperative vasopressor and inotrope dose in patients having cardiac surgery with cardiopulmonary bypass (CPB).
- O#15: In cardiac surgical patients, we aimed to explore, 1) the performance of commonly used machine learning models incorporating IOH and other intraoperative adverse factors for 30-day mortality prediction and 2) the performance of these models with IOH during and outside CPB phase for mortality prediction.

- O#16: We aim to study the association of $[\text{Cl}^-]$ and the main strong ion difference ($\text{mSID} = [\text{Na}^+] - [\text{Cl}^-]$) during the first 24 hours after admission into intensive care unit (ICU) with the development of AKI and mortality.

III. FRAMEWORK FOR DEVELOPING THE PHD THESIS

In order to carry out the **promotion of critical inpatients' open big data repositories** and the promotion of an innovative culture, it is necessary to conduct datathons where awareness about the importance of open data, collaborative research and transparent and generalizable algorithms can be raised among hospitals, academic institutions, clinicians, researchers and engineers.

Before proceeding with data analysis, the data engineer must perform the design of methods and procedures to perform the **integration and management of large open data repositories of critical inpatients** and thus ensure that the data meets specific quality standards, standardization and anonymization requirements through data governance, quality check, quality assurance, normalization, standardization and de-identification techniques performed on the repositories, detailed below in a conceptual manner and providing example code on how to carry out these tasks.

Finally, and only when the two previous steps have been followed, we will have the certainty that we have a reliable and quality high-resolution critical patient data repository with which you can answer specific clinical questions by the **processing of critical inpatients' open big data repositories**. Although machine learning is largely used in the traditional way in diagnosis (medical imaging, deep learning), treatment optimization will benefit considerably from automatic learning. At some point, hospitals in developing countries could also create their own databases. Researchers who cannot afford clinical trials can understand their own patient populations and treat them better, democratizing the creation and validation of new knowledge. In fact, a database of patients' health records contains the results of millions of mini experiments involving their patients.

The framework to develop this dissertation has been obtained thanks to the collaboration between the following institutions between the years 2016 and 2020:

- Biomedical Engineering and Telemedicine Group, Biomedical Technology Centre CTB, ETSI Telecomunicación, Universidad Politécnica de Madrid, Madrid, Spain.
- Unidad de Innovación, Instituto de Investigación Sanitaria San Carlos, Hospital Clínico San Carlos, Madrid, Spain.
- MIT Critical Data, Laboratory for Computational Physiology, Institute for Medical Engineering and Science, Cambridge, Massachusetts, United States.
- Department of Anesthesia, Critical Care and Pain Medicine, Beth Israel Deaconess Medical Center, Harvard Medical School, Boston, USA

As a PhD Candidate at Biomedical Engineering Department at Universidad Politécnica de Madrid (2016 - 2020) at the Grupo de Bioingeniería y Telemedicina (Biomedical Engineering and Telemedicine Centre) (GBT) we received advanced studies in biomedical engineering and in the main techniques that were later used during the development of this PhD dissertation.

Working as a CTO and innovation manager at Innovation Unit - Health Research Institute, Madrid General Hospital we had the thrilling task of addressing unmet medical needs through the conception, management and development of technological solutions such as eHealth web / mobile apps and biomedical devices in an innovative state-of-the-art multidisciplinary

environment working with a team composed by clinicians from different specialties of the hospital, engineers and other experts.

As an affiliate at LCP, (Dir.: Professor Dr. Roger Mark) and MIT Critical Data group, (Dir. Leo Celi) we acquired experience working with large and complex data sets related to critically ill patients (from the Intensive Care Unit and the Operating Room). We have been performing non-routine analysis problems by applying machine learning and robust statistical methods and supporting the 2018 & 2019.HST.953 Course: 'Collaborative Data Science in Medicine' at MIT as Faculty. We also have the privilege of mentoring and organizing datathons taking place in Asia, South America, Europe and North America; these are inter-institutional sprint-like events where international experts, clinicians and data scientists are brought together to compete during 48 hours to generate new knowledge employing data science and derive new clinical insights in an interdisciplinary atmosphere.

Working at the Center for Anesthesia Research Excellence (BIDMC) we helped to expand the analytic capabilities and to design new strategies and applications within this dynamic and innovative organization. We applied state-of-the-art advanced analytic, quantitative tools and modeling techniques to derive insights, solve complex problems and improve decisions about both patients and providers from the department.

III.1. Promotion of Critical Inpatients' Open Big Data Repositories

We believe in the idea that only an international and multidisciplinary team, working with the goal of extracting the know-how from big data sets can bring the revolution XXI century's healthcare is needing. Because in the end any healthcare innovation (whether technological or not) integrated in the clinical workflow of any hospital department must be centered in bringing value to the patient. In order to carry out the **promotion of critical inpatients' open big data repositories** and the promotion of an innovative culture, it is necessary to conduct datathons where hospitals, academic institutions, clinicians, researchers and engineers can be made aware of the importance of open data, collaborative research and transparent and generalizable algorithms. Below I will explain several examples of datathon and initiatives where I have participated and have successfully allowed to carry out this work, 13 of the projects described here have resulted in publications in high impact journals:

1. Serpa Neto, A., Kugener, G., Bulgarelli, L., Rabello Filho, R., **Armengol de La Hoz, M. Á.**, D. L., Johnson, A. E., ... & Ferraz, L. J. R. (2018). First Brazilian datathon in critical care. *Revista Brasileira de terapia intensiva*, 30(1), 6-8.
2. de Toledo Piza, F. M., Celi, L. A., Deliberato, R. O., Bulgarelli, L., de Carvalho, F. R. T., Rabello Filho, R., **Armengol de La Hoz, M. Á.**... & Kesselheim, J. C. (2018). Assessing team effectiveness and affective learning in a datathon. *International Journal of Medical Informatics*, 112, 40-44.

III.2. Integration and management of Critical Inpatients' Open Big Data Repositories

Before proceeding with data analysis, the data engineer must perform a review of the **integration and management tools of large open data repositories of critical inpatients** and thus ensure that the data meets specific quality standards, standardization and anonymization requirements through data governance, quality check, quality assurance, normalization, standardization and de-identification techniques performed on the repositories, detailed below in a conceptual manner and providing example code on how to carry out these tasks. The review we undertook resulted in the publication of the following manuscripts:

3. Núñez Reiz A, Martínez Sagasti F, Álvarez González M, Blesa Malpica A, Martín Benítez JC, Nieto Cabrera M, Del Pino Ramírez Á, Gil Perdomo JM, Prada Alonso J, Celi LA, **Armengol de la Hoz MÁ**, Deliberato R, Paik K, Pollard T, Raffa J, Torres F, **Mayol J**, Chafer J, González Ferrer A, Rey Á, González Luengo H, Fico G, Lombroni I, Hernandez L, López L, Merino B, Cabrera MF, Arredondo MT, Bodí M, Gómez J, Rodríguez A, Sánchez García M. Big data and machine learning in critical care: Opportunities for collaborative research. *Med Intensiva*. 2019 Jan-Feb;43(1):52-57. English, Spanish. doi: 10.1016/j.medin.2018.06.002. Epub 2018 Aug 2.
4. Reiz, A. N., **Armengol de la Hoz, MÁ.**, & García, M. S. (2019). Big data analysis and machine learning in intensive care units. *Medicina Intensiva* (English Edition).

III.3. Processing of Critical Inpatients' Open Big Data Repositories

Finally, and only when the two previous steps have been followed, we will have the certainty that we have a reliable and quality high-resolution critical patient data repository with which you can answer specific clinical questions by the **processing of critical inpatients' open big data repositories**. Although machine learning is largely used in the traditional way in diagnosis (medical imaging, deep learning), treatment optimization will benefit considerably from automatic learning. At some point, hospitals in developing countries could also create their own databases. Researchers who cannot afford clinical trials can understand their own patient populations and treat them better, democratizing the creation and validation of new knowledge. In fact, a database of patients' health records contains the results of millions of mini experiments involving their patients. In our specific case we have carried out data analysis of three high-resolution repositories of critical patients, two of which are in the Intensive Care Unit and one of which is in the operating room:

- MIMIC III
  - MIMIC-III (Medical Information Mart for Intensive Care III) is a large, freely-available database comprising de-identified health-related data associated with over forty thousand patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012 (Alistair E. W. Johnson et al., 2016).

- The database includes information such as demographics, vital sign measurements made at the bedside (~1 data point per hour), laboratory test results, procedures, medications, caregiver notes, imaging reports, and mortality (both in and out of hospital).
- MIMIC supports a diverse range of analytic studies spanning epidemiology, clinical decision-rule improvement, and electronic tool development. It is notable for three factors:
  - It is freely available to researchers worldwide
  - It encompasses a diverse and very large population of ICU patients
  - It contains high temporal resolution data including lab results, electronic documentation, and bedside monitor trends and waveforms.
- eICU
  - The eICU Collaborative Research Database is populated with data from a combination of many critical care units throughout the continental United States. The data in the collaborative database covers patients who were admitted to critical care units in 2014 and 2015 (Pollard et al., 2018).
- AIMS
  - Our subset of AIMS is a database including over five thousand patients who underwent any kind of cardiac surgery in the operating room of the Beth Israel Deaconess Medical Center between January 2008 and June 2016. Based on updates from previous years, the cohort is expected to increase to an average of 687 cases per year.
  - Source: Anesthesia Information Management System (AIMS).
  - Some of the Information we can find about every procedure: Allergens, Blood Fluids, Bolus Drugs, Infusion drugs , Comments, Events, Fluids, inhalation drugs, medications, laboratory data, procedure type, vitalsigns.
  - Hemodynamic Vital Signs Resolution: 15 seconds.

Given the diversity of data available in these repositories, the variety of clinical problems that can be solved from retrospective analysis of EHR data is enormous. In the projects detailed in the development of this dissertation, we address clinical questions in very diverse departments (Intensive Care Unit and Anesthesia) in very heterogeneous patients (those who need to undergo Coronary artery bypass surgery, valve replacement, both or other kinds of cardiac surgery, the ones who suffer from obesity, terminal patients, septic ones or those who develop acute renal insufficiency during their ICU stay). Below I will explain several examples of clinical questions that I have helped to solve, n of the projects here have resulted in publications in high impact journals:

5. **Armengol de La Hoz, M.Á.**, Hao, S., Celi, L.A., **Gómez, E.J.***, Danziger, J.*. Altered Consciousness Mediates the Effect of Hypernatremia but Not Hyponatremia, On Mortality. (Under review).
6. **Armengol de La Hoz, M.Á.**, Celi, L.A., **Gómez, E.J.**, Danziger, J. Admission Disorientation and Risk of Code Status Change (Under review).
7. Fernandes M.[†], **Armengol de la Hoz, M. Á.**[†], Rangasamy V., Subramaniam B. Machine Learning Models with Preoperative Risk Factors and Intraoperative Hypotension Parameters Predict Mortality Following Cardiac Surgery (Under review).
8. **Armengol de la Hoz, M. Á.**, Valluvan Rangasamy*, Andres Brenes Bastos, Xinling Xu, Victor Novack, Bernd Saugel MD* Balachundhar Subramaniam. Association of Intraoperative Hypotension with Postoperative Stroke, Acute Kidney Injury, and Death in Patients Having Cardiac Surgery with Cardiopulmonary Bypass (Under review).
9. Satoshi Kimura, **Armengol de la Hoz, M. Á.**, Nathan Raines, Leo Anthony Celi. Association Of Chloride Ion And Sodium-Chloride Difference With Acute Kidney Injury And Mortality In Critically Ill Patients (Under review).
10. Fehnel, C. R. [†], **Armengol de la Hoz, M. Á.**[†], Celi, L. A., Campbell, M. L., Hanafy, K., Nozari, A., ... & Mitchell, S. L. (2020). Incidence and risk model development for severe tachypnea following terminal extubation. *Chest*.
11. Danziger, J., **Armengol de la Hoz, M.Á.**, Li, W., Komorowski, M., Octávio Deliberato, R., Rush, B. N., ... & Badawi, O. (2020). Temporal Trends in Critical Care Outcomes in United States Minority Serving Hospitals. *American Journal of Respiratory and Critical Care Medicine*, (ja).
12. Cosgriff, C. V., Celi, L. A., Ko, S., Sundaresan, T., **Armengol de la Hoz, M.Á.**, Kaufman, A. R., ... & Deliberato, R. O. (2019). Developing well-calibrated illness severity scores for decision support in the critically ill. *NPJ digital medicine*, 2(1), 1-8.
13. Deliberato, R. O., Ko, S., Komorowski, M., **Armengol de La Hoz, M.Á.**, Frushicheva, M. P., Raffa, J. D., ... & Stone, D. J. (2018). Severity of illness scores may misclassify critically ill obese patients. *Critical care medicine*, 46(3), 394-400.

IV. ASSESSING TEAM EFFECTIVENESS AND AFFECTIVE LEARNING IN A DATATHON

IV.1. Introduction and problem statement

Hackathons, whose worldwide prevalence has been rising steadily, share a common purpose: connect people and form teams to propose and design technology-based solutions to problems (Aboab et al., 2016; M. Komssi et al., 2015; Olson et al., 2017). The term “datathon” is a portmanteau of the words “data” and “hackathon” and refers to the application of the hackathon model to data analytics. The objective of a health datathon is to assemble clinicians and data scientists to analyze a dataset in order to answer a research question, build and validate prediction or classification algorithms, or optimize the performance of a specific analytic task (Aboab et al., 2016; M. Komssi et al., 2015). Team building is a strategy widely used for organizational development (Liebowitz & de Meuse, 1982). Many variables are considered essential for successful teams such as effective leadership, team achievement, working methods, creativity and individual development (Meuse, 1992; Tannenbaum et al., 1992). Educators have a particular interest in teamwork because learning may be triggered or enhanced in a successful team model (Fransen et al., 2011). Learning is a complex process that intertwines affection, cognition and psychomotor domains (Bloom et al., 1956). Effective teams can enhance apprenticeship by fostering members to contribute to the best of one’s abilities. Learners that recognize team commitment and effective leadership are more likely to contribute to and benefit from a task or a project (Aubert & Kelsey, 2003; Baldwin et al., 1997). Few studies have addressed a health hackathon from learning perspective. Previous health hackathon assessments have failed to demonstrate value perhaps because they focused on the business plan development, publications produced, patents for and start-up companies around a product including software (Olson et al., 2017; Palmer, 2014; Penn & Penn, 2015). The primary purpose of this study is to explore the association between teamwork and learning in a health datathon. The secondary objective is to measure affective learning and investigate how it relates to teamwork. The research questions of this paper are:

- 1) Does a datathon promote affective learning and effective teamwork?
- 2) Is effective teamwork during a datathon associated with affective learning?

IV.2. Methods

IV.2.1. Study design and participants selection

This is a cross sectional study conducted at Hospital Israelita Albert Einstein (HIAE) during the MIT-HIAE Health Conference and Datathon. The event was held during the first weekend of May 2017 (6th and 7th) in São Paulo, Brazil and was hosted by the Critical Care Department of HIAE together with the Laboratory for Computational Physiology from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. The event provided the Medical Information Mart for Intensive Care version 3.0 (MIMIC-III) database to the participants (Alistair E. W. Johnson et al., 2016). MIMIC-III is a large, open-access de-identified dataset of patients admitted to the intensive care units (ICU) of the Beth Israel Deaconess Medical Center (BIDMC) in Boston, Massachusetts, USA. The database contains more than 60,000 ICU admissions from 2001 to 2012 and is hosted

by the Laboratory for Computational Physiology at MIT. It includes high temporal resolution data from various clinical systems – laboratory, pharmacy, bedside monitors, and others. The Institutional Review Boards of both BIDMC and MIT approved the creation and the use of the MIMIC database for research. Since the dataset was anonymized, the Institutional Review Boards of both institutions granted a waiver of consent for its use. Fifty participants with diverse backgrounds (clinicians, engineers, data scientists) were recruited early to attend the datathon. The objective was to prepare and familiarize them with the MIMIC database before the event. All of the attendees were invited to participate in this study; there were no exclusion criteria. However one participant refused to enroll in the study. Participation in the study was voluntary and without any financial incentive. The study protocol was approved by the ethics committee of HIAE and all participants filled a consent form prior to their participation.

IV.2.2. Questionnaire and data collection

For the datathon, the participants were divided into 8 groups with at least one clinician and one data scientist per group. A paper format survey with 25 items consisting of participant demographic information, participants' satisfaction, teamwork and affective learning questions was administered at the end of the event prior to the closing ceremony. The Affective Learning scale, designed by Andersen in 1979 (Andersen, 1979) demonstrates consistently high reliability across different settings. Cronbach alpha has varied from 0.86 to 0.98 (Kearney et al., 1985; Plax et al., 1986; Rubin, 2010). It is validated to measure lower-order affect, such as attitudes, and higher-order learning, such as behavioral commitment (Andersen, 1979; Kearney et al., 1985; Plax et al., 1986; Rubin, 2010). It is a 7-point Likert scale among five dimensions of affective learning (Andersen, 1979; Rubin, 2010). For some items the most positive/desirable number is "1" while in other cases it is "7". This study uses item 4 as a primary outcome measure. 1. Behavior recommended in the course; 2. Content/subject matter in the course; 3. Course instructor(s); 4. Real life situations, your likelihood of actually attempting to engage in behaviors recommended in this course; 5. Your likelihood of actually enrolling in another course of related content if your schedule so permits. To assess team effectiveness we used an adapted version of the Team-Review Questionnaire (TRQ) (Francis & Young, 1979; Rubin, 2010). Gribas has shown reliability of the 10-item scale with Cronbach alpha ranging from 0.86 to 0.92. TRQ items include a 7-point Likert scale. The primary outcome measure of the study is effective leadership (Day et al., 2004; Heifetz et al., 2010; Sarin & McDermott, n.d.; Zaccaro et al., 2001, 2009). To assess team effectiveness we used 10 of the 12 items of the TRQ. These items are:

1. Effective leadership. The team leader is committed and able to develop a team approach, and management in the team is seen as a shared function.
2. Suitable Membership. Members are capable of contributing a "mix" of skills that provides for an appropriate balance.
3. Team commitment. Members feel individual commitment to the aims of the team and are willing to devote personal energy to building the team and supporting other team members.

4. Team climate. People feel relaxed, able to be direct and open, and prepared to take risks.
5. Team achievement. Objectives are clear and felt to be worthwhile, targets for performance are felt to be stretching but achievable, energy is devoted to the achievement of the results, and team performance is reviewed frequently.
6. Corporate Role. The team has contributed to the corporate MIT/ HIAE (Datathon) planning and has an important role in the overall organization.
7. Work methods. The team has developed lively, systematic, and effective ways to solve problems together.
8. Critiquing. Errors and weaknesses are examined without personal attack to enable the group to learn from the experience.
9. Individual development. Member development is sought and the team can cope with strong individual contributions.
10. Creativity Capacity. New ideas are created through member interaction, innovative risk taking is rewarded, and the team will support new ideas from individual members or from outside.

The data obtained from the participants included demographic variables, such as age, gender, type of job, and number of years working. Lastly, information regarding participants' satisfaction and likelihood to recommend the event was also collected.

	Mean (SD)	Range	–	–	–
Age	32.5 (6.9)	22–51	–	–	–
Professional working years	8.3 (7.5)	0–30	–	–	–
Datathon learning score ^a	8.4 (1.2)	5–10	–	–	–
Datathon Teamwork learning relevancy ^b	1.2 (0.4)	1–2	–	–	–
Teamwork frequency in participant's usual life ^c	1.6 (1)	1–6	–	–	–
Datathon Recommendation ^c	1.3 (0.6)	1–4	–	–	–
Satisfaction ^d	2 (1.2)	1–5	–	–	–
Male, n (%)	37 (76)		–	–	–
Professional Distribution, n (%)	MD 16 (32)	DS 14 (29)	PT 4 (8)	Nurse 1 (3)	Other 14 (28)

Note: MD (Medical Doctor), PT (Physical Therapist), DS (Data Scientists).

^a 1: most negative and 10: most positive.

^b 1: most relevant and 5: least relevant.

^c 1: most common and 6: least common.

^d 1: most satisfied and 5: least satisfied.

^e 1: most likely and 5: least likely.

Table 1. Datathon HIAE's results

Mean, Standard Deviation, Range and Scale Interpretation, n = 49.

IV.2.3. Statistical analysis

Stata software (version 13.0, Stata Corp, College Station, Texas, USA) was used to conduct the statistical analyses. Descriptive statistics was performed as follows. Normally distributed continuous variables were summarized as mean $\pm$ standard deviation (SD). Frequencies and proportions were presented for discrete variables. Bivariate Pearson's correlations were performed between outcome and predictors to measure their association. The simple bivariate correlation identifies when to partial out and control for a predictor that is highly correlated with other predictors in a multivariate regression analysis. Multi-variable regression models were conducted to identify the association between the outcome variable and its predictors. Statistical significance was established at a p value less than 0.05.

IV.3. Results

As shown in Table 1, participants of the HIAE datathon were mostly males [76% (37/49)]; 32% were physicians and 29% were data scientists. Mean age was 32.48 years [SD = 6.9] with 8.32 [SD = 7.5] of those years working. Overall, the impression of the event was positive with the

majority of the subjects satisfied and likely to recommend the event to others. Participant-reported learning had a mean of 8.38 [$\pm$ 1.23] out of 10. Participants rated teamwork as relevant to their learning experience: mean 1.20 [$\pm$ 0.40] out of 5, with "1" being most relevant. The learning score in the event and teamwork relevance were highly correlated (absolute $r = 0.36$, $p = 0.009$). Participants who graded the learning experience high also ranked teamwork relevance high (Table 1). Teamwork experience was highly rated by the participants, with most variables with a mean of 6 points on the Likert scale (Table 2). We observed a significant association between the variables. Effective leadership variable was most predictive of team performance: it is significantly associated with 90% (9/10) of teamwork variables, and highly correlated ($r > 0.70$) with three of the variables: suitable membership, team achievement and work methods. As shown in Table 3, scores also demonstrate positive perceptions of the learning process during the datathon. Pearson's bivariate correlation analysis demonstrated significant association among the affective learning variables and among the effective leadership variables. Real life engagement behavior, the primary affective learning outcome due to its higher-order learning (behavioral commitment) was modestly associated with another affective learning variable: enrollment in a similar event in the future ($r = 0.28$, $p < 0.05$). Effective leadership, our key predictor variable for team performance, is significantly associated with three affective learning variables: event suggested behavior, course instructors and real life engagement. Table 4 presents a series of multi-variable regression models to predict real life engagement behavior as the outcome of interest. The predictors from the teamwork questionnaire and the affective learning variables were included as candidate variable for the models. Model A contained only the team performance variables. In this model the key predictor (effective leadership) is not associated with real life engagement behavior (beta coefficient = -0.07). Only team achievement is significantly correlated with the outcome (beta coefficient = 0.48 , $p < 0.01$). Model R² was 47%. Model B evaluated the key question predictor and all the affective learning variables but excluded the rest of the teamwork variables since most of them were not significant in Model A. In this model, two learning variables (content matter and course instructor) were associated with real life engagement variable. Once

again, effective leadership was not found to be associated with real life engagement behavior. Model R2 dropped to 31% (Table 4). In model C, we included all the learning and teamwork variables. In this our key question predictor coefficient is statistically significant ($\beta = -0.27$, $p < 0.01$). All the learning and four teamwork variables are associated with real life engagement behavior in this model. The model R2 is highest at 75%. Finally, model D evaluated the variables in model C, but excluded those that were not statistically significant. In this model the key question predictor coefficient remains significant ($\beta = -0.32$, $p < 0.01$) as with the other variables, but the R2 dropped to 71%. Our final model is model C because it (1) includes all teamwork and affective learning variables; and (2) has the highest R2 . The prediction model for Affective Learning in Model C = $4.57 - 0.27\text{leadership} - 0.07\text{membership} + 0.01\text{commitment} - 0.25\text{climate} + 0.50\text{achievement} + 0.14\text{corporativerole} - 0.06\text{workmethods} - 0.28\text{critiquing} + 0.35\text{individual development} - 0.25\text{creativity} - 0.29\text{eventbehavior} + 0.41\text{contentmatter} + 0.28\text{course instructor} + 0.11\text{future enrollment}$.

IV.4. Discussion

We have several findings in the study. First, affective learning and teamwork variables were highly rated by the participants. We demonstrated that it is feasible to improve affective knowledge and teamwork skills in a datathon. Datathons have been previously assessed based on outcomes such as publications produced, patents for prediction, classification, image recognition and other types of software, and start-up companies arising from the event (Aboab et al., 2016; Marko Komssi et al., 2015; Palmer, 2014; Penn & Penn, 2015). As such, evaluations failed to demonstrate value with regard to these outcomes (Palmer, 2014; Penn & Penn, 2015). Second, our final model (Model C) demonstrates a significant association between higher-level order affective learning and teamwork variables.

	Mean	1	2	3	4	5	6	7	8	9
1. Effective Leadership	5.91	–								
2. Suitable Membership	5.91	0.70***	–							
3. Team Commitment	6.26	0.33**	0.39**	–						
4. Team Climate	6.53	0.18	0.17	0.24	–					
5. Team Achievement	5.87	0.75***	0.62***	0.32*	0.37**	–				
6. Work Methods	5.81	0.71***	0.58***	0.31*	0.53***	0.79***	–			
7. Critiquing	6.30	0.28*	0.22	0.11	0.41**	0.54***	0.47***	–		
8. Individual Development	6.10	0.59***	0.43***	0.28*	0.33*	0.71***	0.59***	0.41**	–	
9. Creative capacity	6.20	0.36*	0.31*	0.27*	0.68*	0.54***	0.58***	0.60***	0.56***	–
10. Corporate role	6.24	0.52***	0.49***	0.32*	0.55***	0.76***	0.75***	0.61***	0.51***	0.62***

Note: 7-point Likert Scale ranging from “1” totally disagree to “7” totally agree.

* $p < 0.05$.

** $p < 0.01$.

*** $p < 0.001$.

Table 2. Means and Pearson’s correlation coefficient matrix for simple correlations on teamwork variables.

$n = 49$.

	n	Mean (SD)	1	2	3	4	5
1. Event suggested behavior	46	1.65 (1.21)	–				
2. Content matter	46	6.43 (0.71)	–0.37*	–			
3. Course Instructors	49	1.67 (1.28)	0.29*	–0.44**	–		
4. Real life engagement	47	6.46 (0.77)	–0.26	0.23	0.05	–	
5. Future enrollment	48	6.16 (1.70)	0.004	–0.08	0.07	0.28*	–
6. Effective Leadership	49	5.91 (1.45)	–0.57***	0.24	–0.39**	0.32*	0.10

Note: 7-point Likert Scale ranging from “1” to “7”. Some questions “1” is the most positive answer and some questions the most negative answer.

* $p < 0.05$.

** $p < 0.01$.

*** $p < 0.001$.

Table 3. Pearson's correlation coefficient matrix for simple correlations on Affective Learning and Teamwork Effectiveness Leadership, Mean and Standard Deviation.

$n=49$.

Moreover, this final model explains 75% (R^2) of the association between teamwork and affective learning and real world engagement behavior, the primary outcome of interest. Our key question predictor, effective leadership, is significantly associated with the outcome after controlling for all learning and teamwork variables. Team achievement, group critiquing capacity and individual development were significantly correlated with real world engagement behavior as well. Our analysis depicts a robust relationship between teamwork and learning in these events.

Researches have shown that collaborative learning is the process of generating knowledge through discussion, critique, debate, which develops deep learning and individual change (Fransen et al., 2011; Sarin & McDermott, n.d.). In addition, effective teams accomplish more than the sum of the efforts of each team member, even those of the most capable members (Gribas, 1990). Team suitable membership, commitment, argumentation, working methods and effective leadership are considered conditional for building an interdependency mechanism within team members, which fosters learning (Heifetz et al., 2010; Sarin & McDermott, n.d.).

Affection is a domain for learning that is crucial to team effectiveness (dependent variable) in the conceptual team model framework (Andersen, 1979; Bloom et al., 1956; Fransen et al., 2011). Although variables related to the individual team members and the task are relevant, the variables that drive successful teams are anchored by affection such as trust, shared values, members' behavior and adaptive capacity (Fransen et al., 2011). It is reasonable to understand how effective learning might be generated in a successful group: affection variables are core components that lead to team effectiveness.

Models	A	B	C	D
Effective Leadership [CI]	-0.07 [-0.31, 0.17]	0.14 [-0.03, 0.32]	-0.27** [-0.51, -0.04]	-0.32** [-0.53, -0.11]
Suitable Membership	-0.08	-	-0.07	-
Commitment	0.07	-	0.01	-
Climate	-0.09	-	-0.25	-
Achievement	0.48**	-	0.50**	0.49***
Corporative role	0.05	-	0.14	-
Work methods	-0.2	-	-0.06	-
Critiquing	0.03	-	-0.28*	-0.26*
Individual develop	0.16	-	0.35**	0.36***
Creativity	-0.06	-	-0.25**	-0.31**
<i>Learning variables</i>				
Event behavior	-	-0.01	-0.29**	-0.28**
Content matter	-	0.47*	0.41*	0.37*
Course Instructors	-	0.21*	0.28***	0.25***
Future enrollment	-	0.10	0.11*	0.13*
Cons	4.71***	1.61	4.57**	3.62**
N	47	42	42	42
R ²	47%	31%	75%	71%
RMSE	0.63	0.68	0.46	0.46

Note: Model A: Teamwork variables only. The key predictor (effective leadership) is not associated with real life engagement behavior.

Model B: Affective learning variables only. The key predictor (effective leadership) is not associated with real life engagement behavior.

Model C: Affective learning and teamwork variables. The key predictor (effective leadership) is statistically significant ($\beta = -0.27, p < 0.01$). All the learning and four teamwork variables are associated with real life engagement behavior in this model with highest $R^2 = 75\%$.

Model D: Affective learning and some of the teamwork variables, excluding those that were not statistically significant in Model C. The key predictor (effective leadership) remains significant ($\beta = -0.32, p < 0.01$) as the other variables, but the R^2 dropped to 71%.

[CI]: 95% Confidence interval.

RMSE: Root Mean Square Error.

* $p < 0.05$.

** $p < 0.01$.

*** $p < 0.001$.

Table 4. Regression Models Affective Learning on Real Life engagement behavior outcome.

Leadership is essential for team effectiveness (Day et al., 2004; Heifetz et al., 2010; Sarin & McDermott, n.d.; Zaccaro et al., 2001, 2009). Our study confirms effective leadership as the foundation of teamwork. It is significantly correlated with 9 out of 10 teamwork variables and is highly correlated ($r > 0.70$) with suitable membership, team achievement and working methods. Successful teams rely on their leaders' capacity to help participants achieve synergistic threshold where all the members share the gains and losses and are transformed in the process (Heifetz et al., 2010; Sarin & McDermott, n.d.). On the other hand, ineffective leadership results from the inability to recoup individual losses that are at times necessary for a group purpose gain (Day et al., 2004; Heifetz et al., 2010). Communication is a key component that enables effective leaders to intertwine affection and action that are critical to successful task accomplishment (Day et al., 2004; Zaccaro et al., 2001). This is achieved when leaders promote individual development (learning) and instill values towards the team vision (Day et al., 2004; Sarin & McDermott, n.d.; Zaccaro et al., 2001). Shared or adaptive leadership pertains to the capacity to mobilize members to perform well even when a clear path has yet to be established and share responsibilities as well as challenges within the team (Day et al., 2004; Pearce & Sims, 2002; Zaccaro et al., 2009). In contrast, vertical leadership is the ability to match the team's expectations by providing direction, order and psychological safety. Pearce and colleagues found that both types of

leadership – vertical and shared – are significantly associated with team effectiveness (Pearce & Sims, 2002). Interactions between effective leadership, team success and individual affective learning merit further research in both real world settings and simulations including hackathons and datathons (Liebowitz & de Meuse, 1982; Meuse, 1992; Tannenbaum et al., 1992). Health datathons are becoming popular worldwide. Future datathons will help further explorations and more reliable investigations of these associations. Our study has several limitations. As it is observational, causality cannot be established. The sample size is small and the subjects are mostly from a single center. Assessment of the learning and teamwork was based on self-reporting; it might not reflect the actual participants' behavior and beliefs, not be feasible to see tangible effects on educational and training outputs over a short period of time and may take months or even years, which would require a far longer follow-up. Nevertheless, this is the first study to measure affective learning and teamwork in a health datathon setting.

V.FIRST BRAZILIAN DATATHON IN CRITICAL CARE

V.1. Introduction and problem statement

Sistema Único de Saúde (SUS) was established in 1990 by the Brazilian Federal Constitution to ensure comprehensive, universal and free access to healthcare for the entire population.(Paiva & Teixeira, 2014)In 1988, half of Brazil's population had no health coverage. Two decades after establishing SUS, more than 75% of the country's estimated 190 million people rely exclusively on SUS for their health care coverage. (Hummel, 2016) Provision of these services presents a rich opportunity to capture digital health data, which are a resource for developing locally relevant clinical practice guidelines rather than adopting those of other countries. Despite this potential for data acquisition, electronic health records (EHR) in Brazil are currently used primarily to support administrative and billing functions and do not store clinical information in a machine-ready format as required for data analysis.(Wolters Kluwer, 2016) A recent survey showed that 70% of the facilities that have used the Internet in the last 12 months in Brazil have some sort of electronic record for medical information.(Marin et al., 2014) In 48% of the facilities, records are partially on paper and partially digital. Paperless information systems were present only in 22% of the facilities, and this rate was slightly higher at 33% for private facilities. On the other hand, 30% of the facilities keep all their patient records on paper with a much higher proportion among public facilities at 51%.(Marin et al., 2014) Two government departments could play leading roles in moving health information technology and data analytics forward in Brazil: (Paiva and Teixeira 2014) DATASUS, which provides information systems support to all divisions of SUS,(Oliveira et al., 2011) and (Hummel, 2016) Telessaúde Brasil Redes, a national program designed to improve SUS's quality of care, integrating teaching and service through information technology tools. With the integration of SUS, it is possible to create an EHR for every citizen, a repository of an individual's records of services carried out and clinical data, and when aggregated, an accurate record of the trajectory of health and disease in Brazil. Databases drawn from EHR in Brazil are limited in scope and accessible only to investigators internally within a hospital or organization. Indeed, two central factors prevent secondary analysis of health records in Brazil to improve population health: the lack of interoperability between information systems used in the health sector and the general attitude towards data as a resource. Although DATASUS provides open access to general government data to the public, these data do not currently constitute a research resource and are spread across systems. However, it is important to emphasize some important initiatives in Brazil regarding EHR. TBweb is an online database that is used for epidemiological surveillance and monitoring of cases of tuberculosis in São Paulo state. It is a real-time system, and data can be uploaded and assessed online throughout the course of the disease.(Prof. Alexandre Vranjac, 2008) Additionally, CadUnico is a great example of an instrument that merges demographic characteristics, interventions and socioeconomic markers from different databases.(Brasil. Governo Federal, 2016) In this scenario, initiatives promoting direct interaction between medical doctors, data scientists, managers and computer scientists with successful systems for Big Data Analytics could add practical value and foster future data integration in this country. We organized an event bringing together experts and a high-quality database for a hands-on experience in health data analytics. On May 6-7, 2017, a critical care datathon in Brazil was held at Hospital Israelita Albert Einstein (HIAE). The term "datathon" is a portmanteau of data + hackathon, accentuating application of the hackathon model to data analytics. In this Brazilian edition, members of the Laboratory for Computational Physiology (LCP) at the Massachusetts Institute of Technology (MIT) provided guided access to Medical Information Mart for Intensive Care III (MIMIC-III), a well-curated open access clinical database developed and maintained by

LCP and supported by a vibrant research community. (Alistair E. W. Johnson et al., 2016) As has been the model for past datathons hosted by LCP, the MIT-HIAE event was split into two components over two days: a half-day conference on health informatics followed by a day and a half of hands-on analysis of clinical data. The half-day health data conference was attended by 204 participants. Topics covered by presentations and panel discussions included the value of data sharing and multi-disciplinary collaboration around the data, privacy and security issues surrounding data sharing as it pertains to Brazil, and cultivation of an ecosystem of innovation around big health data and biotechnology. The critical care datathon immediately followed the conference. It was attended by 31 data scientists and 19 clinicians for a total of 50 attendees. Doctors pitched research questions that have been screened by the MIT and HIAE faculty to evaluate whether they could be addressed with the MIMIC database. These topics included the following: “day-night variation in sedation practices and patient outcomes”, “mechanical power in mechanically ventilated patients”, “the association between serum lactate trajectory and specific organ dysfunction”, “effect of hyperoxia in patients with sepsis”, “outcomes after surgery in very elderly patients admitted in the intensive care unit”, “behavior of brain natriuretic peptide in acute kidney injury”, and finally, “the impact of nighttime vs. daytime admissions on clinical outcomes.” Groups were formed based on expertise and interest in the particular clinical question. Early in the second and final day of the event, feedback sessions were held to gauge the progress of each group and provide suggestions regarding study design and additional analyses. At the end of the event, all teams gave a brief presentation of their research projects. Bringing together clinicians and data scientists at the MIT-HIAE datathon served to demonstrate the value of each other’s expertise. Most importantly, the datathon generated interest at HIAE to contribute to a high-resolution critical care database for the research community, supplementing existing resources, such as MIMIC. This decision might encourage other institutions in this country towards integrating and sharing medical databases in a standardized format, thus providing valuable insights to all health care ecosystems. Change is on the horizon with growing interest in secondary analysis of health records to fuel a learning health system. However, for a true health data revolution to occur in Brazil, the environment - the technology, policies, and people, both providers and patients - needs to be supportive of change. Those at the forefront of the health data revolution must earn and maintain society’s trust and demonstrate that data analysis benefits patients. The proponents of health data sharing tread a slippery path as they move into uncharted area. We hope that the datathon experience inspires similar initiatives in this country that will leverage data routinely collected in the process of care to inform clinical practice and health policies.

VI. BIG DATA AND MACHINE LEARNING IN CRITICAL CARE: OPPORTUNITIES FOR COLLABORATIVE RESEARCH

VI.1. Introduction and problem statement

One of the fundamental challenge intensivists face in the care of the critically ill is the management of a huge amount of information gathered from patients. In intensive care units equipped with electronic information systems, hundreds of individual elements of structured information are collected every day in an electronic chart, including vital signs, basic and advanced monitoring records, laboratory data, drug prescriptions and their administration, as well as nursing records, among others. A lot of textual information, like nurse's and physician's comments and progress notes, is also generated, as well as images of different formats and heterogeneous structure (Jesse et al., 1992).

It is impossible for the human brain to fully assimilate this abundance of data, so one of the fundamental qualities of the good intensivist is knowing how to prioritize and filter the most relevant information for a given patient. The exponential growth in computerized information storage and processing systems, however, enables a different approach in which all the available information is taken into account and processed.

Big data analysis (BDA) can be applied to an individual patient, but also be extended to population management of critically ill patients, or to subgroups of patients meeting certain criteria (M. Ghassemi et al., 2015). The large volume of data may partially address the lack of “formal” randomization of tests and treatments by presenting instrumental variables with which patients are “pseudo-randomized” to receive or not receive a particular intervention. Analysis of observational data can also generate hypotheses and inform the study design of prospective trials to evaluate those hypotheses. BDA is already routinely employed in other fields as varied as marketing, strategic business decision-making, insurance, banking, transport and logistics services and fraud detection in electronic commerce. Additionally, BDA requires that high quality data are entered by clinicians and nurses.

Apart from the data, the other component of BDA is what is referred as “Machine Learning” (ML), which fuels what is collectively known as artificial intelligence (AI). This technology uses different methodologies (Table 5) to identify patterns from the data which typically involve either classification or event prediction and subsequently apply the software output to a very specific task, such as automatic interpretation of medical images (Leo A. Celi et al., 2014). This technology has been used successfully in domains that, until recently, seemed unlikely use cases, such as Jeopardy and the board game Go. Traditional computer systems differ from AI in that in the former, rules are agreed upon by experts and fed into the computers as algorithms. With AI, data is fed into the computers, and the computers “discover” and then implement the rules. In addition, the computers continuously assess the rules and re-calibrate them as needed.

Methodology	Definition
Decision trees	Uses a decision tree as a predictive model with a value assigned to each branch.

Methodology	Definition
Association rules	Serves to discover interesting relationships between variables of large databases.
Neural networks	Based on the operational characteristics of biological neural networks, with different nodes and inferences stored in their interconnections. Non-linear, they serve to explore complex relationships between input and output variables of the system.
Intensive learning (Deep learning)	Neural networks of multiple layers, using hardware with great processing capacity. They try to simulate the model of the human brain and have been used for visual analysis and language recognition.
Inductive logic programming	Uses logical and functional programming to derive software models that replicate the knowledge and relationships that exist in the system.
Support vector machines	Used to create predictive models between two possible outcomes (for example, survival or death). Uses classification and regression methods.
Clustering	Used to allocate observations to subsets of similar study subjects.
Bayesian networks	Represents the interdependence between variables through directed acyclic graphs. Used, for example, to compute the probability of the presence of a disease, given a set of symptoms.
Reinforcement-based learning	Serves to create agents that perform actions in a system, maximizing long term benefits, even if conditions are suboptimal.
Representation learning	Intended to improve the way in which the entries are presented to the system, discovering their general properties.
Learning similarity and metrics	Pairs of similar examples and a similarity metrics are used to generate a model of the system.
Learning of dispersed terms	Used in situations where there is information that is only scarcely represented in the examples.

Methodology	Definition
Genetic Algorithms	A heuristic search using mechanisms of natural selection, evolution, mutation and crossing to progressively improve algorithms.
Rule-based learning	Translates the characteristics of the system into rules that can then be applied in new examples.
Boosted Decision Trees (Ensemble Model) ⁴	Supervised learning method that learns by adjusting the residual of the previous trees.
Random Forests (Ensemble Model) ⁵	Supervised classification algorithm that creates a forest with a direct relation between the number of trees and the results the model can get.
Adaboost (Ensemble Model) ⁶	It is a boosting algorithm developed for binary classification. Boosting algorithm aims to create a strong classifier from a number of weak classifiers
Natural Language Processing ⁷	Artificial intelligence technique to interpret and manipulate human language to computer understanding.

Table 5. Machine learning methodologies

(Brownlee, 2016; S. Gu & Wu, 2017; SAS, 2018; Takaki et al., 2018)

Critical care is particularly well suited for machine learning technology, because the technology is a large component of the care provided in the ICU with data as an exhaust of the technology (Leo Anthony Celi et al., 2013; Iwashyna & Liu, 2014) In fact, there have been ML applications in the ICU, such as in natural language processing, pharmacovigilance and decision support (Ross et al., 2014). Providers, including intensivists, however, are typically not trained to develop, implement and evaluate such technologies. The next section deals with how to solve this problem.

VI.2. Collaborative work

Since it would be extremely unusual that one individual would have both the clinical training and the computer science background for BDA and ML, teams with professionals across different fields are required, composed of data scientists including statisticians and epidemiologists, clinicians and other biomedical researchers. The coordination of a group with different schedules,

the establishment of clear objectives and an effective leadership to manage members with different cultures and who “speak different languages” are all individual challenges in themselves, (J. T. Wu et al., 2018) although in critical care medicine teamwork is routinely applied in the day-to-day workflow.

This article describes an initiative event organized by the Massachusetts Institute of Technology (MIT) Critical Data group, the Life Supporting Technologies (LifeSTech) research group at Universidad Politécnica de Madrid (UPM) and the Innovation Unit and Critical Care Department of Hospital Clínico de San Carlos (HCSC) in Madrid.

VI.3. Databases of critically ill patients

The use of BDA and ML techniques creates a new avenue for clinical research that includes non-traditional, non-academic investigators, especially when large anonymized databases of critical patients are shared under pre-specified data user agreement. The pioneers in this field are Beth Israel Deaconess Medical Center and MIT, whose partnership created the MIMIC database (Medical Information Mart in Intensive Care, <http://mimic.physionet.org>) for the research community. The MIMIC-III version contains data from more than 50,000 admissions to Beth Israel ICUs from the years 2001 to 2012 and is available online under a data user agreement. It includes vital signs and other monitoring data, laboratory results, data of prescription and administration of drugs, as well as progress notes, diagnostic codes, physiologic waveforms, both in the ICU and for some variables, from throughout the entire stay in the hospital.

Similar high-resolution ICU databases exist, like the Critical Care – Health Informatics Collaborative (CC-HIC) Database, (Cambridge, 2018) but MIMIC is unique because it is completely open-access. In addition, the MIT group published a textbook that is available at no cost and teaches a course to guide investigators on the analysis of the database. (Badawi et al., 2014; Leo A. Celi et al., 2014; Data, 2016; Alistair E. W. Johnson et al., 2016) Since 2016, they have also organized several datathons (Aboab et al., 2016) in various regions of the world (MIT, London, Paris, Melbourne, Beijing, Sao Paulo and Singapore), to jumpstart disseminate the field of machine learning as applied to health data.

The creators of MIMIC have proposed a system of learning and continuous improvement in Intensive Care they call “closed circuit” (Leo Anthony Celi et al., 2013). This strategy consists of using the patient database to answer questions of clinical interest and then taking it back to the bedside.

VI.4. Initiatives in Spain

Although adoption of electronic medical records and databases of critically ill patients is still in the growth phase in our country, and not all Units have implemented an integrated digital information system, there are already some initiatives under way.

The Spanish Society of Critical Care (Sociedad Española de Medicina Intensiva, Crítica y Unidades Coronarias, SEMICYUC) has published reference standards for the implementation of Clinical Information Systems (CIS), (Tello et al., 2011) and is in the process of building the national electronic record of the minimum basic data set (CMBD) registry to facilitate quality measurement, benchmarking and clinical research.

In some services and hospitals, the process of creating anonymous data bases for collaborative research has already begun. The Critical Care Department at Hospital Clínico San Carlos in Madrid, in collaboration with Philips, has already anonymized its database including its first 12,000 admissions after implementation of the IntelliSpace Critical Care and Anesthesia (ICCA™, Philips) clinical information system. Other ICUs in Spain, such as the Unit at Hospital Joan XXIII in Tarragona, are working on similar initiatives, using the data obtained automatically from their CIS. (Bodí et al., 2017; Sirgo et al., 2018)

The creation and research use of these patient-level databases have some important legal implications, which need to be addressed to comply with regulatory policies, particularly those under the Spanish Data Protection Law. Projects that employ BDA and ML of these patient databases also require prior evaluation and approval by local Ethics Committees.

VI.5. Critical Care Datathon Madrid

In order to promote the use of BDA and ML in healthcare in our country, the seventh Critical Care Datathon, organized jointly by the MIT Critical Data Group, the HCSC in Madrid and the LifeSTech from UPM, took place on December 1, 2 and 3, 2017 at Impact-Hub co-working space in Madrid. The event brought together clinicians, computer scientists and other professionals interested in health data science. HCSC and LifeSTech-UPM group collaborated in the event diffusion informing participants about the content and meaning of the Datathon.

Prior to the event, a group of clinicians from the Critical Care Department of Hospital Clínico San Carlos prepared a list of eight research questions of clinical interest, by way of brain storming, discussion and reaching internal agreement, to be explored during the event:

- (1) the choice of empirical antibiotics in the ICU on hospital mortality.
- (2) risk factors for prolonged ICU stay and mortality after cardiac surgery
- (3) prediction of poor prognosis after admission for resuscitated out-of-hospital cardiac arrest.
- (4) prediction of mortality in patients with non-elective ICU admission.

- (5) prognostic implications of hyperglycemia in critically ill patients with and without diabetes.
- (6) the epidemiology of massive transfusion of packed blood cells.
- (7) understanding the association of positive fluid balance and mortality.
- (8) seasonality of subarachnoid hemorrhage.

After brief presentations of the background and the rationale behind the research questions with consideration of the non-healthcare professionals among the Datathon attendees, eight groups were formed, one for each topic, with the stipulation that all the necessary expertise (clinical, data extraction and processing, machine learning) is represented in each group.

The MIT Critical Group with the support of LifeSTech from UPM set up for the event virtual servers with the MIMIC-III database with access to platforms and software for data analysis (Python, R, Jupyter notebook). LifeSTech evaluated using natural language processing techniques to extract information from clinician's reports written in free text, with the purpose of codifying them through SNOMED-CT terminology. 20 Mentors from the MIT team and LifeSTech were on hand throughout the weekend to help troubleshoot issues, both technical and those related to the MIMIC database or to data modeling. Finally, each group presented their methods and results to the rest of attendees and a panel of judges from MIT, UPM and HCSC.

As an illustration of the relevant conclusion rendered by the collaborative work of the groups during the datathon, we briefly describe the experience of the group who worked on the project looking at the prediction of mortality among patients with non-elective ICU admission.

First, the clinician in the team presented the existing critical care mortality prediction models (APACHE II, MPM, SAPS 2, SAPS 3), and interest in developing an updated score with better calibration from a larger database. A literature search was carried out to select the scores with the best performance for comparison. The OASIS(Kherallah et al., 2008) and APACHE II (Knaus WA et al., 1985) scores were chosen as the current “gold standard”. OASIS, which was constructed using ML techniques, has the best performance among critically ill patients in the MIMIC III database (area under the receiver operating curve or AUROC, 0.82) (personal communication from MIMIC team), while the APACHE II score, the most frequently used for benchmarking and clinical research, has an AUROC of 0.75 (Kherallah et al., 2008).

Likewise, the clinician identified variables in MIMIC-III potentially related to prognosis. The data scientists were responsible for generating the table of more than 40,000 patients with all the selected variables. The entire team went over phases of the model building. In the first phase (training), Support Vector Machines (SVM) was trained using 25,000 patients and produced a model of 30 variables with an AUROC of 0.81. This performance was validated in a different

subset of 15,000 patients, with a similar AUROC, confirming the robustness of the model. Presently, the group is working in refining the model and evaluate its performance in the anonymized ICCA™ (Philips) database with 12,000 patients of our Critical Care Department.

The same collaborative work between clinicians and data scientists performed in previous similar events and that we witnessed in the Madrid Datathon, has led to numerous publications, (M. Ghassemi et al., 2014; M. M. Ghassemi et al., 2014; Paonessa et al., 2015; Pirracchio et al., 2015; Shrive et al., 2016) and we have the same expectation from the Madrid event.

The meeting, which was organized under the auspices of SEMICYUC, also served to connect Spanish ICUs working in the field of BDA and fortified partnerships among them to move this initiative forward.

VI.6. What the future holds and conclusions

We are conceivably at the dawn of a new era in intensive care, in which clinical decision making will increasingly be assisted by computers that perform data integration and analysis. The clinician will therefore have to navigate a specialty of critical care, that harnesses the power of data to individualize care in order to improve population health, where collaborative work with other non-healthcare specialists in the area of data science is essential to leverage all the information that is routinely collected in the process of care, but without compromising patient rights to privacy.

We encourage all those interested in participating in this project to contact the authors, in order to establish a national collaborative network in the field of BDA and ML as applied to healthcare data collected in the ICU, under the auspices of SEMICYUC.

VII. BIG DATA ANALYSIS AND MACHINE LEARNING IN INTENSIVE CARE UNITS

VII.1. Introduction and problem statement

As professionals in intensive care medicine, we live immersed in a sea of data. In a digitalized Department of Intensive Care Medicine such as that of Hospital Clínico San Carlos (Madrid, Spain) (with three Intensive Care Units [ICUs] and approximately 2400 admissions a year), an average of 1400 new units of information are entered in the electronic database per admitted patient on a normal working day --- this implying about 10 million units of information a year (Núñez A., personal communication). The exponential development of computing technology and the incorporation of systems with great storage and processing capacities that can be acquired at an accessible cost imply the registry of a great volume of information that can be used in different ways. Even those Units that still lack electronic case histories or intensive care software applications can take advantage of the latest-generation computing methods to improve the way in which daily work is done. An example is the application of Natural Language Processing (NLP) to a series of reports filed in MS Word, PDF or other similar free-text (non-structured) formats.(Buurman et al., 2015) The data managed by intensivists can come from different sources. Health professionals continuously add information (whether structured or not) to the documentation of the patient. It has been estimated that physicians spend almost 2 h documenting for every hour of direct patient care. (Arndt et al., 2017) Physicians, nurses and technicians do this in the form of free-text notes or reports of different specialties, or as coded data referred to diagnoses or procedures. We also generate treatment instructions and drug administration registries, and receive a large body of data produced by medical devices and systems: laboratory analyzer results, vital signs, advanced monitoring data, ventilator parameters, operating parameters of complex equipment such as perfusion pumps, dialysis monitors, extracorporeal membrane oxygenation (ECMO) systems, and information in the form of images, audio, video and much more. Until only a few years ago, all this information was lost or, at most, filed as case histories in paper format. Nowadays, however, it is possible to store and process this information automatically in digital format and use it to extract new knowledge and obtain guidance for improved patient care. The data we use can be structured in a wide variety of standards or structures. It is much easier to conduct clinical research when the information is available in a structured format, though health professionals have not yet reached consensus regarding an unequivocal way to express each health-related concept. This does not mean that we do not have standards. The SNOMED CT, (Bhattacharyya, 2015) HL7,(Benson, 2012) UMLS,(NIH, 2018) DICOM,(DICOM, 2018) LOINC(Institute, 2018) and many others are examples of such standards, and are used in different healthcare settings --- allowing automated processing of data and the exchange of information between systems. But standards are not such in absolute terms: we can express a given diagnosis using different standards, such as MESH,(Romano, 2018) UMLS, SNOMED CT, ICD9 (Diseases ICo, 2018) or ICD-10,(Diseases ICo, 2018) and it often proves necessary to establish “translations” between standards (the technical term for this being “mapping”) in order to transfer information from one system to another. Solutions involving new approaches to the problem of multiple standards (for example OMOP(Zhao et al., 2018)) offer us tools for performing such mapping processes in a systematic way. Not only data are important. The context in which the information is set is also important. In some cases, a piece of information has no value at all unless we can associate it to more relevant information about the patient or the clinical situation. As an example, a blood pressure recording of 90/60 mmHg is interpreted very differently in a young woman undergoing plastic surgery versus a hypertensive elderly patient with bleeding. One same concept moreover can be reflected in different ways by

different professionals, in different situations and --- in the case of numerical values --- using different units. Our structured data can be categorical, consist of whole numbers or with decimals, comprise dates, hours or durations, or may be grouped (e.g., blood pressure with its two components: systolic and diastolic) in a way similar to the lists, tuples or objects we see as data structures in programming languages. Sometimes our data are matrixes of bytes organized to represent an image or video, such as for example a DICOM file that can be used to display a computed tomography (CT) image. In this concrete example there is another interesting concept, namely “metadata”: information about the information (e.g., in the case of the CT scan, the image acquisition parameters would represent metadata). Some data are subjective. As an example, a physician may reason the way in which he or she reaches a certain diagnosis through phrases in a free-text note. In contrast, other data are objective: our hemofilter has extracted exactly 52 ml of ultrafiltrate from the patient during the last hour, and this information is conveyed to the electronic plot via the network connections.

Figure 1. Dataflow in intensive care medicine.

The quality of care and of strategic decision making depends on processing of the information contained in the active data repository (information that is being generated in each moment and which is available for decision making) and on reutilization of the knowledge previously stored in local or shared databases. The automated processing of information through Machine Learning (ML), and the possibility of rapid access to large volumes of data of a heterogeneous structure by means of Big Data Analysis (BDA), allow us to improve the care processes and extract knowledge from the data. Adapted from Celi et al., (Leo Anthony Celi et al., 2013) with permission from the authors.

In performing secondary analyses of clinical data we must consider a number of aspects referred to privacy, security and sensitivity. It may be perfectly acceptable for a patient to know that his or her respiratory frequency is publicly disclosed, but possibly the same cannot be said of seropositivity status for the human immunodeficiency virus. Data are of different origins because they are obtained for different reasons (accountancy, research or clinical management of the patient, for example), and so there are also different actors involved in the process (managers, clinicians, researchers, service suppliers). If we wish to extract valid information from all this, with a view to helping health professionals to make better decisions and guide our clinical research, we must be able to work efficiently with this clinical jigsaw puzzle of thousands of small pieces generated by daily patient care.

VII.2. Methods

Dataflow in intensive care medicine Information flows in our setting through different phases which we will briefly consider (Figure 1), since in different moments of the clinical care process critical events occur that influence our capacity to make efficient use of the information.

VII.2.1. Data input

A part of the data input process is performed automatically and in a structured format allowing for efficient information use. However, there is a great opportunity for improvement in the acquisition of data in which free text is currently used. We will try to explain this with an example:

“A 58-year-old male reporting one hour ago to the emergency room due to dyspnea and central chest pain irradiating to the left arm. . .”.

In this brief paragraph of 23 words there are at least 8 concepts that can be expressed in a structured manner in the following xml document (a standard format for displaying information(Tucker, 2005)):

```
<Episode id=xxxx>
  <Patient>
    <ID>xxxx</ID>
    <Sex>Male</Sex>
    <Age unit = 'years'>58</Age>
  </Patient>
  <Place>Emergencies</Place>
  <Moment>2018-08-12 15:30:00</Moment>
  <Registry date>2018-08-12 16:30:00</Registry date >
  <Symptoms>
    <Dyspnea/>
    <Pain>
      <Location>Central chest</Location>
      <Irradiation>Left arm</Irradiation>
    </Pain>
  </Symptoms>
</Episode>
```

For a physician it is usually much easier to interpret the free text paragraph, while in the case of a computer system the second way of representing the information is much more efficient. The ideal situation would be to develop a tool allowing the human to enter the information at least as fast and conveniently as with the free text, but also allowing direct input in a structured format, facilitating reading and understanding of the data by both humans and computers.

VII.2.2. Processing of the data in vivo

The data entered in the system is in a “passive” state, and it is the health professional who must make the pertinent inferences after compiling the data that he or she feels to be relevant in a concrete moment of the patient course. However, we have the possibility of transforming the filed information into “active” data capable of triggering system responses through a so-called Clinical Decision Support System (CDSS). Continuing with the previous example, suppose we have entered the above paragraph, and the system has a concept on file:

```
<Syndrome>
  <Name>Acute coronary syndrome</Name>
  <Symptoms>
    <Pain>
      <Location>Central chest</Location>
      <Irradiation>Left arm</Irradiation>
    </Pain>
  </Symptoms>
</Syndrome>
```

By entering the patient information, the system may alert us directly to the fact that the patient has a symptom consistent with acute coronary syndrome. This is a very simple example, but what is really useful about the software tools is that this type of processing can be done automatically for thousands of concepts and in a much more systematic way than is done by the human brain. To understand this by means of an analogy, it is as if the system were running a continuous checklist with our patients. In addition, the current artificial intelligence (AI) techniques allow the use of experience gained with previous patients in the form of structured data for the assessment of the next individual patient. In future, this may have a great impact upon clinical practice. Another example is the use of real time control systems, for example to continuously adjust the insulin perfusion dose of a patient according to the insulin sensitivity he or she has shown previously; stress condition assessed from different data of the plots and laboratory tests; and the caloric and carbohydrate supply being provided in that moment. Over time, the system can learn about the patient and gradually optimize glycemic control within concrete safety parameters.

VII.2.3. Data storage

There are two ways to store the information of our patients:

- (a) in relational databases that use the SQL (Structured Query Language) to retrieve and process the stored information.

(b) in non-structured data repositories (NoSQL).

Nowadays there are tools that allow the extraction and processing of information also from data of this kind. Specific types of information can be stored in specialized databases such as for example a PACS for the filing of clinical images, using the DICOM format. A posteriori data analysis and the sharing of information Once we have stored the data, they can be processed in different ways. For example, we can obtain an automatic report on the activity of our Unit by consulting our SQL database with the demographic and clinical information of our patients, or we can find out which antibiotics we use most often, or what type of disease conditions are predominantly seen in our Unit, without having to again review our patients one by one. This information can be used for benchmarking (comparison of results between Units or within one same Unit when processes or resources are modified), or for planning strategic actions. We recommend the article published by Celi et al., (Leo Anthony Celi et al., 2013) where a review is made of all this dataflow cycle and the concept of “closing the loop” is addressed, explaining the way in which AI can contribute to the development of intensive care medicine.

VII.2.4. What is Big Data?

The classical approach of the clinician to the management of data for clinical research is based on the premise that it is necessary to obtain quality information in order to secure reliable results that are applicable to patients. The difficulty of obtaining such information leads to attempts to optimize the process, applying prospective designs, randomization and a working hypothesis before the data are analyzed. As an alternative to the exclusive use of data collected in an orthodox manner, Big Data Analysis (BDA) offers the novelty of detecting the underlying structure and knowledge in large bodies of information, even when the latter do not seem to be structured. A popular definition of the concept (Mauro et al., 2016) is that “Big Data consists of data sets with such a large volume and such a broad structural variety that specific technology and analytical methods are needed to process them and transform them into knowledge or value”. Specific BDA techniques have been successfully used in fields such as marketing, strategic decision making in the business world, banking, transport, logistics, insurance or the detection of fraud in electronic commerce. There are no reasons to believe that BDA cannot be applied to our setting, where we continuously make strategic or concrete decisions in given patients whose characteristics --- while complex --- are often repetitive. A recent example, which we will address again further below, is the follow-up of influenza epidemics based on analysis of the searches made on the internet. In this case we use databases with many millions of registries in order to draw rapid and reliable epidemiological conclusions (Ginsberg et al., 2009). The open code philosophy has acquired a strong presence in the world of Big Data, allowing its use without the need for major economical investments. Projects such as the Apache Hadoop (Foundation, 2014; Mehta R., 2018) include a whole range of resources in their ecosystem (HDFS, Spark, MapReduce, Impala, HBase, Hive) that allow the low-cost construction of a Big Data batch processing system (analysis of data already stored in large relational or non-relational databases) or Big Data stream processing system (analysis and processing of data as they are generated).

VII.2.5. What is Machine Learning?

The concept of Machine Learning (ML) dates back to the midtwentieth century, and was defined in an article published by Samuel in 1959 (Samuel, 1959) as an area of AI that uses computational algorithms and statistics to give computers the capacity to “learn”, i.e., to improve their results in a specific task after processing a sufficient volume of information and without explicit external instructions (which may be biased) from the programmer.

	A	B	C	D	E	F	G	H
1	Episode	Age	Sex	APACHE II	SOFA	Type of admission	Lactic acid	Death (Y/N)
2	3654	67	H	22	3	Elective surgery	1.2	N
3	2342	34	H	11	2	Elective surgery	0.9	N
4	1156	81	Medical	34	8	Medical	2.4	S
5	9856	59	Medical	30	7	Elective surgery	3.0	N
6	2317	75	H	21	1	Medical	4.9	N
7	9042	75	Medical	16	3	Emergency surgery	2.9	N
8	4510	82	Medical	25	4	Elective surgery	0.9	N
9	5557	60	H	24	3	Medical	1.7	N
10	7623	63	H	12	1	Elective surgery	0.5	N
11	1995	71	H	19	3	Medical	1.8	N
12	5489	47	H	30	5	Emergency surgery	5.0	S
13	1115	69	Medical	21	3	Emergency surgery	1.2	N
14	6737	78	H	12	1	Elective surgery	0.5	N
15	2499	74	Medical	12	2	Elective surgery	1.2	N
16	2009	45	Medical	10	3	Elective surgery	1.9	N

Figure 2. Typical structure of a data frame

It is composed of rows for each of the registries, with an anonymized identifier, and columns corresponding to the values of different variables for each case. The last column shows the target variable (in this case mortality). Here we can see a section of some registries of the database used as an example in the material included in the GitHub repository.

The scope of ML is closely related to other fields such as simulation and modeling, the optimization of systems and statistics. All these areas make intensive use of common mathematical techniques that require specific training. Some basic concepts need to be explained at this point. We must use a format referred to as a “data frame” to work with data in ML. A data frame is a matrix where each row corresponds to one of the patients or registries, and each column corresponds to one of the recorded variables (Figure 2). As an example, suppose we wish to develop a model for the prediction of mortality according to the following variables: type of admission (medical, elective surgery or emergency surgery), age, sex, APACHE II score upon admission, SOFA score upon admission, and blood lactic acid levels. In this case we have as

many rows as there are recorded patients, and each row has a column identifying the episode (on an anonymous basis) and a column for each of the explanatory variables that we have defined, plus another column corresponding to the target variable (in this case mortality as a dichotomic variable [YES/NO]). For those readers who wish to reproduce the examples provided in this review, we have prepared a series of anonymized data from our own database, and have entered them in an MS Excel table in a GitHub repository ([https://github.com/anunezr/revision medicina intensiva](https://github.com/anunezr/revision-medicina-intensiva)).

There are three main variants of machine learning:

- Supervised learning. Each registry is labeled with a value of the target variable, and use is made of different techniques capable of predicting the value of that variable in a new registry. In our above example, the target variable is mortality (YES/NO). Once the system has been “trained”, it must be able to predict the value of that variable corresponding to an episode that has not been previously presented to the system. Therefore, the training data and the data of the test must be different and stored separately. This variant is used in classification, prediction and similarity detection tasks.
- Unsupervised learning. In this case the aim is to detect patterns or trends in data without using a target variable. This variant is used for example to automatically classify patients into groups, and to reduce the number of variables and the complexity of the models.
- Reinforcement learning. In this case the system pursues an objective or reward, and progressively learns as the environment with which it interacts is explored, and which is not known beforehand --- avoiding actions with a negative reward and seeking to conduct actions with a positive reward. An example of this would be a system that learns and adjusts the empirical antibiotic regimens prescribed by the clinicians as they receive the results and characteristics of the septic patients they see.

Table 1 Sup, included in Appendix, summarizes the different ML techniques for each of these variants. The use of such techniques requires knowledge in programming and control of the concepts used in AI, which normally are not within the reach of the clinician. Because of this, a new collaborative working approach is being consolidated, including the introduction in the Units of a new professional with basic clinical knowledge and advanced skills in statistics and the tools and methods of BDA and ML. These professionals, working jointly with the clinicians, can help us to draw value from a large amount of clinical information which right now is filed in our databases. This “man in the middle”, now already found in leading technological development centers, will soon become an important element in the teams of our Units, and will help to plan strategic decisions and optimize the software tools in order to get the most out of them.

Filtering, reorganization and pre-processing

Once extracted, the data are checked for quality and are given an adequate format and scale, resolving problems referred to missing information and inconsistencies in order to prepare the data for processing with the corresponding model
These first two tasks in fact are the most time-consuming steps in a Big Data Analysis (BDA) and/or Machine Learning (ML) process
Selection of attributes
Selection is made of those variables that are going to be used in the learning process, keeping their number and dimensionality as low as possible. However, an advantage of ML over conventional statistical models is that it can assimilate a larger number of variables and, based on them, is able to establish more powerful predictions
Creation of the training and validation datasets
We decide the type of sampling and how we are going to divide the datasets among training, validation and testing
Selection of the model and hyper-parametric adjustment
We select the learning algorithms we are going to use (selection of model, cross-validation, result metrics, optimization of hyper-parameters). Comparison is made of the results between the different algorithms and models used
Evaluation of functioning of the selected model
Prediction
Use of the model in new cases and re-evaluation of its functioning

Table 6. Steps in constructing a Machine Learning system.

For those readers with knowledge of R or Python and who wish to acquire basic practice in the field of ML, the GitHub repository associated to this review (https://github.com/anunezr/revisión_medicina_intensiva) includes a series of scripts that use the different techniques reflected in Table 1 Sup, fundamented on the anonymized information of the database provided as an example. Table 6 provides a “road map” for putting an ML system into practice, adapted to a concrete problem (Raschka, 2017). Each of the considered aspects would merit a review as extensive as this article, if not more so. The aim of the table is to explain that the use of BDA and ML requires complex methodology and systematization that in turn demand

specific knowledge and experience in order to obtain results. Only under these premises can we extract knowledge from the data in an efficient manner.

Tools

In order to make use of BDA and ML techniques, we must master at least some of the analytical statistical computation languages (R, Python or Java); control SQL as a database consulting tool; and know how to use the code libraries employed in this field. The Supplementary Material provides a brief description of the most widely used software.

VII.2.6. Difficulties and strategies for applying these techniques in a Department of Intensive Care Medicine

In the face of this mix of acronyms and technicisms, intensivists tend to ask themselves whether BDA and ML techniques can really be applied in their working environment, and how to do so in an accessible way. In truth, the purpose of this review is to make the clinician aware that entering the world of AI applied to intensive care medicine is possible but demands structural changes and investment in human resources and technology, as well as an expanded vision of the concept of clinical research. Logically, the first requirement is a data acquisition and storage system allowing subsequent analytical processing. In Spain we are still far from achieving this, since the existing electronic clinical information systems are not universally implemented in our Units, and an inter-hospital standard is even less common. Furthermore, many of those centers that do have such applications lack efficient access to the stored data --- often because the system purchasing contract did not initially contemplate this aspect, which usually comes at an added cost. These paradoxical situations, which border on ethical neglect by keeping clinical data “hostage” of the suppliers of the commercial software, must be urgently resolved if we want to work with the new BDA and ML tools for the good of our patients. A second consideration is the need for hospitals to assign specialized staff from the Information Technology Department to the custody, processing and analysis of the clinical data of our Units, working in collaboration with physicians and nurses in order to get the most out of the stored information. A third consideration is the need to work with the local and regional Ethics Committees to guarantee data security and privacy, and to not obstruct opportunities for improvement of the processes and clinical care which exploitation of the stored data can afford.

VII.2.7. Shared databases

Secondary analysis of electronic health records (EHRs) (Data, 2016) refers to processing of the clinical data of patients generated as a sub-product of their acquisition with healthcare purposes, and can be used as an aid in strategic decision making in a Unit, or on a point basis for a concrete patient. In seeking to endorse clinical research in the intensive care setting, some institutions with very large clinical databases have placed them at the disposal of investigators throughout the world, following due anonymization to preserve data privacy. The currently most popular initiative

of this kind is the Medical Information Mart in Intensive Care (MIMIC-III),(Alistair E. W. Johnson et al., 2016; MIT Lab for Computational Physiology, 2018) the main characteristics of which are its public accessibility and the excellent quality of the filed data, comprising clinical information of all kinds, drawn from over 58,000 critical patients --- both adults and newborn infants. Another example of such databases is the eICU Collaborative Research Database, (Pollard et al., 2018) comprising information from a combination of many Intensive Care Units throughout the United States. The registries of the collaborative database compile data from almost 200,000 patients admitted to critical care in 2014 and 2015. As we can imagine, although such secondary analysis may be very useful for a Department of Intensive Care Medicine or for a clinical investigator, its potential is boosted if the data from several Units can be combined to produce a large multicenter database. The effect is similar to what we see on comparing the conclusions that can be drawn from a single center trial versus a multicenter trial. However, a number of difficulties must be overcome in order to do this. Each Unit compiles the data using software that might not be the same as that used in the rest of the Units we seek to merge. We therefore must ensure conversion to a common data structure allowing combined or pooled analysis. This is what we referred to above as “mapping”. In this regard, ontologisms have been introduced that allow us to express and map the filed concepts in a uniform manner. Some of them, such as the Observational Medical Outcomes Partnership (OMOP)(Zhao et al., 2018) or Informatics for Integrating Biology and the Bedside (i2b2),(Computing NCfB, 2018, p. 2) offer tools to facilitate local database migration to a common data standard. On only considering the studies published in recent years using the public MIMIC (Alistair E. W. Johnson et al., 2016) database and software libraries with latest-generation algorithms and free access, we can see that interesting results have been obtained in very diverse areas of intensive care medicine (Table 7). Furthermore, an advantage of this strategy is that the results are fully reproducible: we can take the data set used by the authors and reproduce the statistical and ML methods they have used, even with our own information, since they are routinely available in public repositories such as GitHub (Alistair E. W. Johnson et al., 2016; MIT Lab for Computational Physiology, 2018).

Usefulness of the use of echocardiography in sepsis (Feng et al., 2018)
Usefulness of knowing the variability of blood glucose levels in the prognosis of non-diabetic patients (Liu et al., 2016)
Improvement in the prediction of in-ICU mortality according to physiological parameters (Calvert et al., 2016)
Early detection of sepsis (Desautels et al., 2016) and the possibility of generating early alerts in patients at a high risk of developing septic shock (Henry et al., 2015)
Difference between the water balance targets to be established between the first and the second 24 h of sepsis resuscitation (Shen et al., 2018)
Predictive value of the nursing notes in relation to long term patient prognosis (Waudby-Smith et al., 2018)
Usefulness of early biliary tract drainage in acute cholangitis (Aboelsoud et al., 2018)

Capacity of natural language processing tools to detect diagnoses not coded in free text notes (Gehrmann et al., 2018)
Negative prognostic value of prolonged tachycardia (Sandfort et al., 2018)
Paradoxical relationship between obesity and mortality in the ICU among patients with chronological disease conditions (Pan et al., 2017)
Prediction of undesired early readmissions (Desautels et al., 2017)
Use of additional clinical variables to better adjust heparin perfusion doses (M. M. Ghassemi et al., 2014)
Poorer prognosis after admission to the ICU among patients administered serotonin reuptake inhibitors (M. Ghassemi et al., 2014)

Table 7. Aspects of clinical research studied using Big Data Analysis and Machine Learning techniques.

This causes the studies to be subject to additional quality control and allows anyone to check both the methods and the results obtained. There are conditioning elements of a legal and ethical nature that can complicate the data sharing process, and it may prove necessary for the local Ethics Committees to authorize access and check safekeeping of data security and privacy. The logistics of the process require coordination among the participating centers and technological and hardware support concordant with the project, involving multidisciplinary teams capable of dealing with the challenges which the different aspects of such a project require. Once these difficulties have been overcome, the opportunities for improved management, benchmarking, and of progress in clinical research are enormous, and without doubt will compensate the efforts required to incorporate a Department of Intensive Care Medicine to projects of this kind.

VII.2.8. Natural language processing

Since most of the information which professionals enter in the case history is in the form of free text (also known as natural language), a possible application of AI techniques in medicine is the automated creation of structured information from free text, as well as the classification or phenotyping of patients into different groups according to the contents of the text written by the professionals. These techniques, known as natural language processing (NLP), are advancing quickly (Pai et al., 2014) There are two possible approaches:

- The extraction of concepts from the text using tools such as cTAKES (Apache Software Foundation, 2018) is quite well developed in the Anglo-Saxon world, though there are few practical options for texts not written in English. The research group in Data Mining of the Biomedical Technology Center of the Universidad Politécnica de Madrid (Spain) is working with a tool derived from the cTAKES known as TIDA, (Costumero et al., 2014) and in future it is possible that systems of this kind will become part of our computer-based resources.

- Another alternative is the use of convolutional neural networks (CNNs), (Gehrmann et al., 2018) but this type of strategy poses an inconvenience in that the algorithms obtained are not easily interpretable by physicians --- though this could become irrelevant if the model functions correctly --- and the result obtained are variable and conditioned to the sought phenotypes.

VII.2.9. Retrospective Big Data analysis versus randomized clinical trials

Since the clinical data stored in our clinical information systems are not compiled with the same demanding criteria as in the context of clinical trials, how can the secondary analysis of patient care information allow reliable conclusions to be drawn? The key lies in the volume of the data and in the capacity of the ML tools to detect structure in the midst of chaos (Data, 2016). As an example, we can use follow-up of the incidence of the H1N1 influenza epidemic of 2008 based on the counting of Google searches made using the term “flu”, (Ginsberg et al., 2009) and which has continued to be validated posteriorly (Figure 3) --- exhibiting perfect correlation with the registry of cases obtained by means of more orthodox methods. Although to date it cannot be affirmed that BDA and ML have displaced the traditional clinical research methods, it is true that they allow the detection of trends or patterns in large bodies of data that may go unnoticed to the investigator, and which can guide the design of new studies adhered to the conventional methodology. (Mayo et al., 2017) Furthermore, they are the only alternative in those cases where it is logistically not possible to contemplate an orthodox clinical trial of sufficient statistical power to resolve a relevant question or issue.

VII.2.10. Ethical and legal issues

In a recent article appearing in Intensive Care Medicine, McLennan et al (McLennan et al., 2018) offered an excellent analysis of the problems which the sharing of clinical data included in databases may pose. Since the coming into effect of the European Directive on General Data Protection Regulation (GDPR) (EU GDPR Portal, 2018) in May 2018, the institutions must comply with legislation referred to the handling of personal data on patients in the European Union. The equivalent of this in the United States is known as Health Insurance Portability and Accountability Act (HIPAA), of 1996 (HHS.gov, 2018).

Figure 3. One of the first examples of the application of BDA to clinical research.

During the N1H1 influenza epidemic of 2008, the Google searches using the term “flu” or its symptoms proved very precise in predicting the increase in number of cases and the severity of the clinical condition. The image at top shows the number of searches, while the image at bottom shows the evolution of cases according to the Official Influenza Surveillance System in Spain (the time scales have been adjusted to ensure vertical correspondence between the two figures). This finding has subsequently been used as an epidemiological surveillance system, with great success.

The key issue is data anonymization: if the information is completely anonymous, there is no legal problem in sharing it from the right to privacy and security perspective. But when can we consider the data to be completely anonymous? Only when they do not lead to unequivocal identification of the patient upon being complemented with other data (e.g., even if we do not know the name of the patient or his or her history number, if we know the date of admission and age, we may identify the patient involved). We also must avoid unequivocal identification of the health staff appearing in the case history of the patient. In all other cases the data are considered to be pseudo-anonymized, and in this situation the GDPR directive would oblige us to request patient consent to use of the information. Database anonymization is therefore crucial, and we must standardize a process that allows compliance with the European regulations while losing as little information value as possible. Investigators must work jointly with the local Ethics Committees to harmonize the safeguarding of data privacy and security with the advances in clinical research which the BDA tools can offer us in future. Mechanisms must be developed to facilitate patient consent (or withdrawal of consent) to the use of the information generated during healthcare,

conveniently harmonized for clinical research. Another important issue is data property. In this case we are not dealing with privacy or security questions but with the intrinsic value of the information (even regarded as a commercial item). Does the patient (as the source of the information) or the clinician (as the subject entering the information in the system) have the right to request economical compensation or to prohibit the use of the data for clinical research purposes or any other use? Although the current reality is that the data are often in the hands of the companies that develop the medical software, and which seek to exploit such information for their own commercial interests, (Tanner A., 2016) this issue is the subject of great controversy, and it would be desirable to establish legal norms seeking to resolve the problems we presently face in this field.

VIII. SEVERITY OF ILLNESS SCORES MAY MISCLASSIFY CRITICALLY ILL OBESE PATIENTS

VIII.1. Introduction and problem statement

Obesity is currently a global pandemic, responsible for 3–4 million deaths per year (Ng et al., 2014), with an increasing prevalence in adults as well as children and adolescents (Flegal et al., 2013). Obesity is overrepresented in the ICU, comprising approximately one-third of patients (Oliveros & Villamor, 2008), compared with the 20% prevalence of being overweight or obese worldwide. ICUs commonly use severity of illness scores, such as Acute Physiology and Chronic Health Evaluation (APACHE), Simplified Acute Physiology score version II (SAPS-II), or Sequential Organ Failure Assessment (SOFA) to predict mortality (Gall, 1993; Knaus WA et al., 1985; J.-L. Vincent et al., 1996), but none of these scoring systems incorporates obesity into their risk adjustment variables. These scores rely on the assumptions that patients have the same normal physiologic values at baseline and that similar severity of illness scores represent the same degree of deviation from the baseline state. However, this may not always hold true for different population groups in different intensive care settings. In obese patients, prior studies have shown abnormal laboratory markers including WBC and liver enzymes (Dixon & Brien, 2006; Johansson et al., 2011), platelet counts (Samocha-Bonet et al., 2008), and respiratory physiological values (Parameswaran et al., 2006; Salome et al., 2010). Therefore, although obese and normal weight patients may present to the ICU with a similar physiological “snapshot or phenotype” as reflected by the same severity of illness scores, these identical scores may actually represent inherently different levels of deviation from the prior baseline state. This may inadvertently result in misclassification, leading to potential errors in mortality prediction and severity adjustment. The case of obese ICU patients is made more complex in that, despite their higher prevalence of chronic diseases that would be expected to result in generally higher all-cause mortality compared with normal weight individuals (Flegal et al., 2013), critically ill obese patients have been reported to paradoxically have better clinical outcomes than nonobese patients (Abhyankar et al., 2012). We postulated that prognostic severity misclassification may be an artifact of applying the same scoring system to these two diverse populations without regard to divergences in evolving anomalies. This question is critical in determining whether the use of conventional scoring systems produce reliable predictions in conditions associated with diverse physiologies, including obesity. To investigate this question, we analyzed a large ICU database (which included baseline laboratory results prior to hospital admission) to compare the deviation of laboratory tests utilized in scoring systems from baseline to ICU admission in both obese and normal weight patients.

VIII.2. Methods

VIII.2.1. Data Sources

For this study, we used the Medical Information Mart for Intensive Care (MIMIC-III) database, a large, open-access dataset of patients admitted to the Beth Israel Deaconess Medical Center (Boston, MA). This database contains data for more than 60,000 deidentified patient admissions to ICUs between 2001 and 2012 and is hosted by the Laboratory for Computational Physiology at the Massachusetts Institute of Technology. MIMIC-III contains data that were downloaded from

different sources, including archives from critical care information systems, hospital electronic health record databases, and outof-hospital records of patient mortality. Such data include high resolution vital signs, laboratory results, prescribed medications, fluid balance, procedure codes, diagnostic codes, imaging reports, provider notes, and hospital administrative data. All these data were collected during routine clinical care and data collection was not visible to caregivers, which means there was no interference with their workflow (Alistair E. W. Johnson et al., 2016). The use of the MIMIC-III database has been approved by the Institutional Review Boards of Beth Israel Deaconess Medical Center (2001- P-001699/14) and MIT (No. 0403000206).

VIII.2.2. Study Population

Study inclusion criteria were first ICU admission of patients 16 years and older and available documentation of height and weight, as well laboratory test results at baseline and at ICU admission. We excluded data from subsequent admissions if patients were admitted to ICU more than once. We defined baseline laboratory values as the mean laboratory result of all readings available between 3 days before to 1 year prior to ICU admission. ICU values were defined as the most abnormal laboratory result in the first 24 hours of ICU admission, similar to the analysis in the calculation of the SAPS-II and SOFA scores (Gall, 1993; J.-L. Vincent et al., 1996). Obesity was determined according to World Health Organization classification. The height measured during the hospital admission and the average of weights measured 24 hours before and 24 hours after the ICU admission were used. Only obese (body mass index [BMI], ≥ 30) and normal weight patients (BMI, ≥ 18.5 and < 25) were included as comparison groups to maximize the difference between study groups.

VIII.2.3. Study Variables

The following baseline patient level characteristics were collected: age, gender, ethnicity, marital status, insurance coverage, and comorbidities as defined by Elixhauser et al (Elixhauser et al., 1998) combined in a composite score by van Walraven et al (Walraven et al., 2009), here after referred to as the comorbidity index. Smoking status was identified using Natural Language Processing searches for history of active smoking in the provider notes. Hospital characteristics, procedures in the first 24 hours of the ICU admission, as well as SAPS-II and SOFA score on ICU admission were also included (Granholm et al., 2016; J.-L. Vincent et al., 1996). The exposure variable was BMI status, comparing obese to normal weight individuals, and the primary outcome was the deviation in laboratory results between that measured at baseline and during ICU admission. We selected laboratory results that were used in the SAPS-II or SOFA scores in our analyses, except for bilirubin, which was not included due to a significant fraction of missing baseline data.

VIII.2.4. Statistical Analyses

We used quantile-quantile normal plots to assess the appropriateness of assuming normality. Continuous variables were summarized using the mean and sd while those with a nonnormal distribution were summarized with the median and interquartile range. For the continuous variables, mean values were compared using two-sample t tests, and median values were assessed using the Mann-Whitney test. Tests for association between categorical variables and BMI status were assessed using a chi-square test. Absolute values at baseline as well as deviations from baseline were compared between normal weight and obese individuals. The differences in deviation from baseline between both groups were compared using multivariable linear regression adjusted for age, gender, comorbidity index (Walraven et al., 2009), SAPS-II score (Gall, 1993) or SOFA score (J.-L. Vincent et al., 1996), and type of ICU, and the relevant baseline laboratory result. A full model comprising the BMI status and all the covariates was initially fit and subjected to stepwise backward elimination retaining BMI status in the model, until a final model was obtained with only statistically significant variables. Statistical significance was assessed at the 0.05 level. For variables violating the modeling assumptions of linear regression models, the logarithm (base 10) of the baseline and ICU laboratory results were calculated and the regression analysis performed on the log transformed values. We also assessed the effect on hospital mortality of any statistically significant deviations found comparing the obese and normal weight groups using logistic regression. A null (baseline) model was fit composed of SAPS-II, SOFA, age, and the ICU values of the laboratory tests, which were found to be statistically significant when comparing the deviations from baseline between obese and normal weight subjects. A model fit using all variables in the null model, in addition to any laboratory deviation variables found to have a statistically significant difference between the normal weight and obese patients, was compared with the null model using a likelihood ratio test. Information about the number of missing laboratory values is provided in detail in Appendix, Supplemental Table 1.

Figure 4. Patient fluxogram.

BMI = body mass index, MIMIC-III = Medical Information Mart for Intensive Care III.

All analyses were performed with R version 3.3.2 (R Foundation for Statistical Computing, Vienna, Austria). We have made all our data extraction and modeling queries and codes available online: <https://github.com/deliberato/Obesity-project>.

VIII.3. Results

Of the 61,532 admissions in the MIMIC-III database, 38,367 were unique patients greater than 16 years old. Of these, 3,205 (8.35%) had laboratory results available prior to hospital admission and also had height and weight data available during admission. After excluding underweight and overweight individuals, a total of 2,027 patients were included in the final cohort (Figure 4). A total of 769 normal weight patients and 1,258 obese patients were compared. Baseline characteristics are summarized in Table 8. The obese patients were slightly younger (median age of 64.3 vs. 66.9; $p = 0.01$), less likely to be white (74% vs. 76%; $p < 0.001$) and more likely to have private insurance (41% vs. 32%; $p < 0.001$). In addition, obese patients had a lower comorbidity index

(median of 2 vs. 5; $p < 0.001$), and comprised a higher proportion of cardiac surgery recovery unit (CSRU) patients (58% vs. 39%; $p < 0.001$) and a smaller proportion of medical ICU patients (22% vs. 38%; $p < 0.001$). During the first 24 hours of ICU stay, the obese patients were more likely to require mechanical ventilation (74% vs. 59%; $p < 0.001$) and vasopressor therapy (58% vs. 45%; $p < 0.001$). A crude comparison of laboratory results at baseline showed that the obese patient group had lower platelet counts (231 vs. $245 \times 10^9/L$; $p < 0.001$) and higher sodium (140 vs. 139 ; $p < 0.001$) than normal weight individuals (Table 9). The deviation in WBC (6.4 vs. $5 \times 10^9/L$; $p < 0.001$), sodium (-3.2 vs. -2.6 ; $p = 0.003$) and potassium (1 vs. 0.8 ; $p = 0.001$) in ICU from each individual's baseline was also significantly greater in obese individuals (Table 9). The laboratory parameters included in SAPS-II scoring were adjusted for SAPS-II score as well as baseline laboratory result, ICU type, age, gender, and comorbidity index (Table 10). The WBC deviation was 0.80 (95% CI, $0.27-1.33$) $\times 10^9/L$ and the log(BUN) deviation was 0.01 (95% CI, $0.00-0.02$), both were statistically significant higher in obese patients. There were no statistically significant differences in the deviation in values in both groups with respect to sodium, potassium, and bicarbonate.

Variables	Normal Weight (BMI, 18.5–24.9), n = 769	Obese (BMI, > 30), n = 1,258	p
Age (yr), (median, IQR)	66.9 (53–78.5)	64.3 (55.5–72.7)	0.01
Male, n (%)	431 (56)	741 (59)	0.22
Ethnicity: white, n (%)	583 (76)	935 (74)	< 0.001
Marital status, n (%)			0.11
Married	415 (54)	738 (59)	
Single/divorced/separated	227 (30)	352 (28)	
Other/unknown	127 (16)	168 (13)	
Insurance, n (%)			< 0.001
Medicare/Medicaid	497 (65)	709 (56)	
Private/other	272 (35)	549 (44)	
Comorbidity index (median, IQR)	5 (0–10)	2 (0–7)	< 0.001
BMI (kg/m ²), (median, IQR)	23.1 (21.3–24.1)	34.3 (31.9–38.2)	< 0.001
Smoker, yes, n (%)	379 (49)	649 (52)	0.20
Admission type, n (%)			< 0.001
Elective	321 (42)	752 (60)	
Emergency	433 (56)	491 (39)	
Urgent	15 (2)	15 (1)	
Source of admission, n (%)			< 0.001
Emergency room	310 (40)	319 (25)	
Physician referral	367 (48)	822 (65)	
Other, n (%)	92 (12)	117 (10)	
ICU admission type, n (%)			< 0.001
Cardiac surgery recovery unit	300 (39)	726 (58)	
Medical ICU	286 (38)	272 (22)	
Surgical ICU/trauma ICU/coronary care unit	183 (23)	260 (20)	
Primary International Classification of Disease version 9 diagnosis, n (%)			0.01
Sepsis, including pneumonia	110 (14)	188 (15)	
Cardiovascular disease	357 (46)	528 (42)	
Other respiratory condition	49 (7)	50 (4)	
Neurological condition	47 (6)	88 (7)	
Other	206 (27)	404 (32)	
Procedures in the first 24 hr of ICU admission, n (%)			
Mechanical ventilation	453 (59)	937 (74)	< 0.001
Vasopressors	346 (45)	731 (58)	< 0.001
Renal replacement therapy	32 (4)	35 (3)	0.12
Severity of illness			
Simplified Acute Physiology score version II (median, IQR)	35 (27–45)	32 (25.2–40)	< 0.001
Sequential Organ Failure Assessment score (median, IQR)	4 (2–6)	4 (2–6)	0.02

BMI = body mass index, IQR = interquartile range.

Table 8. Baseline Demographic Characteristics

Baseline ^a				Δ (ICU – baseline) ^a			
Variables	Normal Weight	Obese	p	Variables	Normal Weight	Obese	p
SAPS-II laboratory parameters				SAPS-II laboratory parameters			
WBC × 10 ⁹ /L, median (IQR)	7.3 (5.6–9.5)	7.4 (6–9.1)	0.39	WBC × 10 ⁹ /L, mean ± SD	5 ± 5.9	6.4 ± 5.9	< 0.001
Sodium, mmol/L, median (IQR)	139 (136–141)	140 (137–141)	< 0.001	Sodium, mmol/L, mean ± SD	–2.6 ± 4.3	–3.2 ± 3.6	0.003
Potassium, mmol/L, median (IQR)	4.2 ± 0.5	4.2 ± 0.4	0.63	Potassium, mmol/L, mean ± SD	0.8 ± 1.1	1 ± 0.9	0.001
BUN, mg/dL, median (IQR)	18.5 (14–25)	18.4 (15–24.4)	0.49	Log (BUN, mg/dL), mean ± SD	0.02 ± 0.2	0.01 ± 0.2	0.26
BIC, mg/dL, median (IQR)	27 (25–28.7)	27 (25–28.5)	0.73	BIC, mg/dL, mean ± SD	–4 ± 4.1	–4 ± 3.6	0.8
SOFA laboratory parameters				SOFA laboratory parameters			
Creatinine, mg/dL, median (IQR)	0.9 (0.7–1.2)	1 (0.8–1.2)	0.06	Log (creatinine, mg/dL), mean ± SD	0.01 ± 0.2	0.03 ± 0.2	0.05
Platelets × 10 ⁹ /L, median (IQR)	245 (189–315)	231 (186–284)	< 0.001	Platelets × 10 ⁹ /L, mean ± SD	–75 ± 119	–70.6 ± 79	0.37

Table 9. Laboratory Results at Baseline and Change in Results Between ICU and Baseline

BIC = bicarbonate, BUN = blood urea nitrogen, IQR = interquartile range, SAPS-II = Simplified Acute Physiology Score version II, SOFA = Sequential Organ Failure Assessment score.^a Information about the number of missing laboratory values might be found in Supplemental Table 1 (Supplemental Digital Content 1, <http://links.lww.com/CCM/D72>).

The laboratory parameters included in SOFA scoring were adjusted for SOFA score as well as baseline laboratory result, ICU type, age, gender, and comorbidity index (Table 10). The deviation in log (creatinine) was 0.03 (95% CI, 0.02–0.05) higher in obese patients ($p < 0.001$); there was no statistically significant difference in the deviation of platelets. In the logistic regression analysis of hospital mortality, a comparison of the null model and the model which incorporated all variables in the null model, in addition to the deviation from patient baseline of WBC, log-BUN and log-creatinine (found to be the statistically significant variables in Table 10) was performed. In this three degree of freedom test, we found that we would reject the null model ($p = 0.009$), suggesting that one or more of log (BUN), log (creatinine), or WBC deviation variables had a statistically significant effect on mortality after adjusting for the variables in the null model.

VIII.4. Discussion

Our results suggest that the deviations in WBC, creatinine, and BUN from baseline to the most abnormal value in the first 24 hours of ICU stay are significantly higher in obese patients, adjusting for the severity of illness score. This supports the hypothesis that within the same severity of illness category, obese patients may have inherently different degrees of deviation from their usual state during critical illness.

Variables	Adjusted Difference* in Deviation Δ (ICU – Baseline) Between Obese and Normal Weight Individuals (95% CI)	p
Simplified Acute Physiology Score version II laboratory parameters		
Δ WBC, $\times 10^9/L$	0.80 (0.27–1.33)	0.003
Δ Sodium, mmol/L	–0.06 (–0.40 to 0.28)	0.712
Δ Potassium, mmol/L	0.01 (–0.07 to 0.09)	0.857
Δ log (blood urea nitrogen, mg/dL)	0.01 (0.00–0.02)	0.014
Δ Bicarbonate, mg/dL	–0.19 (–0.50 to 0.13)	0.254
Sequential Organ Failure Assessment score laboratory parameters		
Δ log (creatinine, mg/dL)	0.03 (0.02–0.05)	< 0.001
Δ Platelets, $\times 10^9/L$	4.94 (–2.48 to 12.36)	0.192

*Effect while keeping all other model covariates constant: models were fit via stepwise backward elimination with the full model composed of relevant score (Simplified Acute Physiology Score version II or Sequential Organ Failure Assessment score), baseline laboratory result, ICU type, age, gender, and comorbidity index.
 Δ : laboratory deviation = ICU laboratory result – baseline laboratory result.

Table 10. Multivariable Linear Regression of Laboratory Deviation

(Δ = ICU – Baseline)

The existing literature has reported differences in physiological and laboratory parameters between obese and nonobese patients in stable settings such as prior to and after surgery (Dixon & Brien, 2006; Johansson et al., 2011), or in outpatient settings (Parameswaran et al., 2006; Salome et al., 2010; Samocha-Bonet et al., 2008; Yadav et al., 2017) but our study is the first to describe this phenomenon in the setting of critical illness. The instinctual response of a clinician to these small differences in laboratory results is that they are clinically insignificant. However, we feel that such small changes are incredibly important. As electronic health records of larger patient populations are secondarily analyzed in a context of medicine that is increasingly precise and personal, even small differences that are consistent and statistically significant, will compound and impact data analysis. As such analysis assumes an increasing role in assisting clinicians with decision making that is less susceptible to cognitive bias, the accumulation of many small changes that may seem trivial to a clinician's eye, may affect models for classification, prediction, and/or prognostication. Small changes may shift segments of populations across classes, such as disease severity from mild to moderate, or moderate to severe, for example. Furthermore, established scoring systems, as opposed to practicing clinicians, operate on nonfuzzy, fixed thresholds from transitioning from nonpoint assignments to awarding a point (such as moving from a score of 1 to 2). Therefore, even very small changes in values, for example, in the new Sepsis-3 definitions (Seymour et al., 2016) where an increase in SOFA score of 2 or more points

(with other criteria) defines sepsis, may produce modifications that could change the classification of a patient from “not septic” to “septic.” In our attempts toward precision and personalized medicine, if an identifiable subset of patients has a consistent difference that marks that as unique, it behooves us to attempt to capture this information so that it can be applied in any predictive, real time, or retrospective analysis of clinical information. The traditional approach to severity of illness scoring assumes a normal baseline, and hence, similar degrees of deviation among those values belonging to the same category. Such assumptions may contribute to an inaccurate representation of actual severity of illness or organ dysfunction. We believe that any prior differences do carry over into the critical care time period and must be addressed in order to certify that severity stratification methods are robust, research that uses critical care databases is reliable, and that data driven clinical decision support applications are safe and useful. Although our study is not designed to elucidate why the deviation in laboratory values differ between obese and nonobese patients, others have suggested that excess adiposity is associated with proliferation of macrophages and other immune cells in response to adipocyte apoptosis, which in turn secretes proinflammatory cytokines (Dijk et al., 2015; Heymsfield & Wadden, 2017). The differences in physiological response to a state of critical illness may account for these small differences in deviations from baseline. In addition, an abnormal baseline value, in obese patients even in the absence of apparent comorbidities, may be indicative of reduced physiological reserves, which may be associated with poorer outcomes. Another related and more general issue is whether dynamic score changes represent the same severity and prognostic states as more static ones. For example, does a SOFA score of 6 in someone who came in the ICU with a score of zero represent the same severity and prognostic state as a patient who also has a score of 6, but had it before admission due to chronic obstructive pulmonary disease, liver and kidney disease, decreased platelets, and maybe a touch of encephalopathy or dementia? Ideally, an optimal scoring system can be developed that represents a true measure of biological severity and is immune to the effects of prior states and nonstandardized interventions (e.g., use of vasoactive drugs). The use of an individual patient’s baseline for scoring would be a step toward more personalized and precision medicine and may contribute to mortality prediction models with better calibration. SAPS-II was developed based on mortality outcomes of a North American and European cohort and may be vulnerable not only to unique factors of the socio demographic makeup of that particular population but also external factors such as cultural preferences, variations in practice, and differing quality of care. The impact of these contextual factors may account for the poor performance of severity of illness scores in particular populations such as the elderly (Sanchez-Hurtado et al., 2016) or in other countries—for example, a validation study of SAPS-II in a cohort of patients from 11 countries showed poorer performance than the original SAPS-II validation sample (Granholt et al., 2016). Recognizing that some of these factors may be uncontrollable, consideration of the extent to which critically ill patients deviate from their own baseline may represent one way to increase the objectivity, precision, and generalizability of mortality prediction and severity adjustment models. Our study has several limitations. First, our study was confined to those patients who had pre-ICU baseline values available, which comprised only 8.35% of the entire database. This subset may be sicker as the need for these laboratory tests may signal the presence of more comorbidities that require follow-up in the outpatient setting. Exclusion of otherwise healthy patients in the study cohort who did not have any outpatient follow-up and are more likely to have normal weight, may underestimate the difference in deviation from baseline between the two groups of patients. Second, the differences in the composition of the obese and normal weight groups may contribute to selection bias. There was a higher proportion of obese patients who were electively admitted to the ICU, comprising 60% of

admissions, or 27% excluding those admitted to the CSRU; compared with nonobese patients. Although we adjusted for these differences in the regression model, there may be residual confounders between these groups that we were unable to adjust for. Third, BMI may not be the best measure of obesity and may not be reflective of the physiological processes that we try to model. However, it is the most frequently used indicator of obesity in current clinical context and would be the most generalizable. Fourth, our data were from a single center, which may also limit generalizability. Fifth, we were unable to add deviations in vital signs into our model due to absence of baseline data, which may account for some residual confounding. Finally, as our analysis only compared normal weight individuals to obese individuals, we may not be able to extrapolate our findings to overweight individuals. Our study does not reveal whether obese patients behave, with respect to clinical outcome, like other patient types with similarly higher scores, or have better outcomes than those scores would have indicated. This is a topic for a future study—this article represents a starting point in this regard where such differences are duly noted and analyzed to form a basis for such studies. In fact, this kind of unavoidable heterogeneity in prior conditions and physiological states is not unique to the issue of obesity and poses a general problem in the formulation of valid and reliable scoring systems. APACHE deals with this issue at a high level by assigning weights to selected historical conditions. Future directions include examining for differences in deviations of physiological parameters such as heart rate, blood pressure, or partial pressure of oxygen (P_{aO_2}), which are included in SOFA and SAPS-II. As with previous studies that demonstrated differences in cardiac autonomic activity and heart rate (Yadav et al., 2017), and respiratory physiology (Parameswaran et al., 2006; Salome et al., 2010), between obese and normal weight individuals, it may well be that deviations in such parameters are also different in obese and normal weight patients with similar severity of illness scores. Further studies are required to assess if severity of illness scores based on deviations from usual state rather than assumed normal baseline values will be superior in mortality predictions. Finally, while we have done this for obese adults, other such populations may also be identified for whom the “one scoring system for all” principle may not apply.

IX. DEVELOPING WELL-CALIBRATED ILLNESS SEVERITY SCORES FOR DECISION SUPPORT IN THE CRITICALLY ILL

IX.1. Introduction and problem statement

Illness severity scores are regularly employed for quality improvement and benchmarking in the intensive care unit (ICU) and as covariates for risk adjustment in observational research of critically ill patient cohorts (Breslow & Badawi, 2012b, 2012a). Common models for estimating severity of illness scores include the Acute Physiology and Chronic Health Evaluation (APACHE), the Simplified Acute Physiology Score, and the Mortality Probability Model (Lemeshow, 1993; Moreno, 2005; J.-L. Vincent et al., 1996; J. E. Zimmerman, Kramer, McNair, & Malila, 2006). Poor generalization and inadequate model calibration have limited the use of these models for clinical decision support (Pappachan et al., 1999; Rowan, 1993, 1994). As medical data become progressively digitized and hence readily analyzable, there is increased interest in, and potential for, the use of this data to develop useful clinical decision support tools (Beam & Kohane, 2018). As accurate quantification of patient mortality risk may inform goals of care discussions and resource allocation, it is an area of clinical predictive modeling that is ripe for improvement. One of the key challenges has been developing models that are well calibrated. (Calster & Vickers, 2015; Kramer et al., 2014, 2015). Although prior models have often achieved acceptable observed-to-predicted mortality ratios (OPRs), they frequently under- or over-predict risk across the risk spectrum; in particular, the models tend to be most poorly calibrated in the sickest patients. Critically ill patients represent a heterogeneous risk population with only a minority (about 1/3) of patients constituting what may be defined as high mortality risk, i.e., $\geq 10\%$ (Beck et al., 2002). The effect of this heterogeneity on predictive models may be best understood in the context of the bias–variance tradeoff. Although logistic regression allows for the modeling of non-linearities (e.g., polynomial terms, restricted cubic splines) and the manual specification of interactions, complex interactions have been overlooked in the development of prior severity score models. Therefore, the contribution of a feature, for example, the worst heart rate over the first 24 h of hospitalization, to the risk of mortality is considered equal and constant across the spectrum of illness severity. While ensuring less variance and simplifying the models, intuitively, this also results in the creation of a biased model of the underlying data generating process. More sophisticated machine learning approaches do not require as much input from the modeler and automatically learn relationships directly from the data. This extra flexibility allows these approaches to model non-linearity and feature interactions at the cost of higher variance. Excitement over the power and promise of these methods has continued to swell, but recent work from Christodoulou et al. has challenged the usefulness of machine learning in the development of clinical prediction models (Christodoulou, 2019). Their review did not consider logistic regression, penalized or otherwise, technically to be a form of machine learning and did not consider perceptual tasks where the performance of deep learning is uncontested. However, their critique of the state of machine learning in the development of clinical prediction models remains apt. We hypothesized that, while high-risk patients in ICU cohorts represent a distinct and identifiable group of patients, prior severity scores have been built with methods that are too simple and inflexible to learn to forecast mortality accurately across such heterogeneous groups. While the aforementioned sophisticated machine learning approaches may offer the flexibility to automatically overcome this, these data-hungry methods can be bested by aptly including domain knowledge, especially when data are limited. Because we hypothesized that the model simply needed to treat high-risk patients differently, we sought to explore a sequential modeling approach wherein, after classification of patients as high risk with an initial regression model, a high-risk-specific regression model is then applied. We explored this approach, as well as a sophisticated machine learning approach, in a

large, publicly available, multicenter critical care database. As the electronic health record (EHR) has given rise to an unprecedented abundance of highly granular clinical data, we expect care centers to increasingly craft bespoke models for severity scoring as well as clinical decision support. With the known shortcomings of previous approaches to severity score development, we hope to inform future model development strategies so as to ensure accurate risk quantification across the board in critically ill patients.

IX.2. Methods

IX.2.1. Data source

The eICU Collaborative Research Database (eICU-CRD) is a large, open access, multicenter critical care database made available by Philips Healthcare in partnership with the MIT Laboratory for Computational Physiology (Pollard et al., 2018). It holds data associated with 200,859 ICU admission for 139,367 unique patients across the United States; we employed data from 2014 and 2015. Version 2.0 of the database is publicly available at <https://eicu-crd.mit.edu/>.

IX.2.2. Ethical approval

This study was exempt from institutional review board approval due to the retrospective design, lack of direct patient intervention, and the security schema for which the re-identification risk was certified as meeting safe harbor standards by Privacert (Cambridge, MA) (Health Insurance Portability and Accountability Act Certification no. 1031219-2).

IX.2.3. Cohort feature extraction

Patients were required to have the variables necessary to calculate the APACHE IVa score as well as having a valid APACHE IVa result recorded in the database. By requiring a valid APACHE IVa score, the following exclusion criteria were implicitly imposed: age <16 years, readmissions, ICU transfers, patients with ICU length of stay <4 h, patients with hospital length of stay >365 days, and patients admitted for burns or non-renal, non-hepatic transplant. In addition to these criteria, patients were also required to have at least one recording of each of the following: automated vital sign recording, laboratory result data, and treatment documentation.

The models developed in this study were meant to recapitulate and extend other well-known mortality models. They consist of a combination of the original APACHE IVa features with features engineered from routinely collected EHR vital signs, laboratory test results, diagnoses, and administered treatments.

The APACHE IVa features included were: admission diagnoses, whether thrombolytics were given, whether the patient was actively receiving any of the certain predefined treatments, the components of the Glasgow coma scale, whether or not the patient had an internal mammary graft, whether or not the patient had a myocardial infarction within the past 6 months, and ventilation and intubation indicator variables. The admission diagnoses were grouped into meaningful categories using a special text string for describing a search pattern known as a regular expression. In this approach, the presence or absence of a search string is used to classify the strings into groups.

With respect to the vital sign data, the models we develop incorporate the mean over all values in the first 24 h in the ICU; these mean values were used for the following variables: heart rate, respiratory rate, oxygen saturation, systolic, diastolic, and mean arterial pressure. Although the APACHE IVa model uses the worst values, our new models pull data that are derived from automatically captured signals, and these streaming systems tend to have noise; thus the minimum and maximum values may reflect artifact. Although the automatically captured temperature was extracted for each patient, it was missing for a substantial majority and was excluded from the analysis as its fidelity was considered suspect.

For laboratory data, the minimum and maximum values in the first 24 h were initially extracted. For serum bicarbonate, chloride, calcium, magnesium, base excess, platelets, hemoglobin, phosphate, fibrinogen, pH, and hematocrit, the minimum value was selected. For serum creatinine, blood urea nitrogen, bilirubin, lactate, troponin, amylase, lipase, B-natriuretic peptide, creatinine phosphokinase and prothrombin time, the maximum value was selected. For serum sodium and glucose, the most abnormal value was defined as the value with the greatest deviation from the normal range. For white blood cell and neutrophil counts, if any measurements were lower than the lower limit of the normal range, the minimum value was used; if the minimum was within normal range, then the maximum was used.

Treatment information from the first 24 h regarding the administration of vasopressors, antiarrhythmic agents, antibiotics, sedatives, diuretics, and blood products was also extracted and these data were included as binary indicator variables.

IX.2.4. Training and validation split

Following the extraction of the features from the database for all included patients, 25% of the dataset was randomly sampled and held out for derivation of a validation cohort. The remaining 75% of the data were then used to develop the models.

IX.2.5. Modeling procedures

Logistic regression (Logit) was chosen as the base model and for the components of the sequential models as this approach is common in clinical prediction and severity score development, is easily interpretable, and often produces results as good, if not superior to, more complex approaches (Christodoulou, 2019). The sequential models were constructed using a cutoff definition of 0.10 risk of mortality and 0.50 risk of mortality; they are titled SeqLogit-0.10 and SeqLogit-0.50, respectively. In these models, the base Logit model, which is fit on data from the entire training set, is used to initially predict the risk of mortality and if that predicted risk is above the cutoff, a second multivariate logistic regression model trained only in patients above the cutoff is used to predict the mortality. When deployed in the validation cohort, the base Logit model first predicts the risk and if the risk is above the cutoff, the second model is used to predict the risk of mortality.

In order to compare the Logit model and sequential models to the current state-of-the-art approach for prediction with structured data, a gradient boosting machine model was implemented. This approach has recently gained traction in clinical predictive modeling with state-of-the-art performance at predicting risk of readmission and risk of acute kidney injury (Koyner et al., 2018; Rojas, 2018). We utilized the XGB implementation developed by Tianqi Chen (Chen & Guestrin, 2016). The gradient boosting approach allows for the development of a strong learner from many weak learners. For the classification task, classification trees of a specified complexity are trained iteratively, each tree being trained on the residual from the linear combination of all prior trees (with the first tree trained on a pseudo-residual). The complexity of the gradient boosting machine is controlled by various factors including the number of trees to be fit, the size of the random sample of the data to be used for each tree, the number of features to be randomly selected for the construction of each tree, the tree depth, and the shrinkage factor (learning rate). These hyperparameters allow for fine control of the model flexibility and thus, if chosen correctly, allow for the development of a model that is hypothetically close to the optima of the bias–variance tradeoff and reduces overfitting.

All hyperparameters were selected via ten-fold CV using only the training data. Because there are many hyperparameters to select with this approach, and because fitting these models is computationally expensive, the CV hyperparameter search was carried using a random search strategy: a grid of possible hyperparameter combinations was randomly sampled from 100 times for a total of 1000 model fits. This approach has been shown to be highly successful and spares substantial wasted CPU time in areas of the hyperparameter space that are far from optimal.

Because of the differences between Logit and XGB, different preprocessing steps were applied depending on which method was being applied. Regression based methods are not capable of dealing with missing data, and samples with any missing values are excluded, that is, missing data are assumed to be MCAR unless adjustment for missingness is performed (e.g., inverse probability weighting to model the mechanism of missingness). To overcome this, the fitting of regression models was carried out using a pipeline that performed multiple imputation using Bayesian Ridge Regression as implemented in the current development version of scikit-learn (Pedregosa, 2011). In addition, this pipeline also normalized the data using the mean and standard deviation from the training data. In contrast, tree-based methods are implemented such that they may automatically learn to handle missing data directly from the data, and there is no need for normalization. As such, no preprocessing was performed on the data used to train the XGB models.

No model recalibration procedures were applied to either approach. Logistic regression-based approaches maximize the likelihood estimator and in turn, minimize the negative log loss given by:

$$-\frac{1}{N} \sum y_i \log(p_i) + (1 - y_i) \log(1 - p_i)$$

Minimization of this loss function directly optimizes the probabilities, p_i , produced by the model and therefore logistic regression produces inherently well-calibrated probabilities. Although Niculescu-Mizil et al. have demonstrated that gradient boosting classification tree-based approaches do not produce well-calibrated probabilities, their work specifically examined AdaBoost, a different implementation, which minimizes the squared exponential loss function (Niculescu-Mizil & Caruana, 2005). In contrast, XGB can be trained with its objective to minimize the log loss, and therefore, like regression, can produce well-calibrated probabilities without recalibration procedures. Eschewing this step allowed us to use all of the training data to fit the models as re-calibration requires a second held out dataset.

IX.2.6. Model performance evaluation

Calibration was examined graphically with the use of reliability curves; this is recommended as the optimal way to assess calibration, and the Hosmer-Lemeshow test was not performed as it offers minimal insight and can be misleading in large samples (Alba, 2017). Although not an optimal measure of calibration across the risk spectrum, the overall OPR was also calculated for each model. Discrimination was examined graphically by plotting receiver operating characteristic curves and quantitatively by calculating the AUC. Precision-recall curves were also generated, and AP was also assessed. Uncertainty around AUC, AP, and observed-to-predicted mortality estimates was quantified by the construction of 95% confidence intervals obtained via bootstrapping: 2000 bootstrap samples were used for each interval.

IX.2.7. Statistical analysis

Normality was assessed by examination of variable distribution. Continuous variables are presented as means and standard deviations if normally distributed or medians with 25th and 75th percentiles otherwise. Categorical variables are presented as the percentage of the whole and sum total. As mentioned above, confidence intervals were derived via the construction of 2000 bootstrap samples for estimated metrics and a null-hypothesis testing framework was not applied.

IX.2.8. Software

The data were extracted from the database with the use of structured query language. This work was developed with Python 3.6. The scikit-learn and xgBoost packages were used for model development, and graphics were produced with the use of the matplotlib package (Chen & Guestrin, 2016; Pedregosa, 2011) .

IX.2.9. Reporting summary

Further information on research design is available in the Nature Research Reporting Summary linked to this article.

IX.2.10. Data availability

The dataset analyzed during the current study is publicly available at <https://eicu-crd.mit.edu/>.

IX.2.11. Code availability

All of the code for this work is publicly available at <https://github.com/cosgriffc/seq-severityscore>.

IX.3. Results

IX.3.1. Cohort overview

Of the 200,859 ICU stays in the Phillips eICU Collaborative Research Database (eICU-CRD), 148,532 had the required APACHE IVa variables recorded, and 136,231 had the required APACHE IVa variables and also met criteria for the calculation of an APACHE IVa mortality prediction. After excluding patients who lacked at least one recording of automatically captured vital sign data, 134,946 ICU stays remained. The cohort selection is summarized in Figure 5. Demographic variables and a subset of the features are summarized in Table 11. After randomly splitting off 25% of the initial cohort to form a validation cohort, 101,167 patients remained for model training. All model evaluation metrics are reported on the validation cohort, composed of the held out 33,779 patients who were never used in model training.

Figure 5. Fluxogram. Cohort selection process

Variable	Summary measure
Age	63.12 (17.32)
Gender Prop.	54.03% (72,875)
Caucasian ethnicity	77.08% (103,971)
Sepsis	13.32% (17,970)
Cardiac arrest	6.72% (9066)
GI bleed	5.36% (7230)
CVA	7.13% (9616)
Trauma	4.28% (5780)
Unit type: MICU	8.56% (11,545)
Unit type: SICU	6.42% (8664)
Unit type: mixed	55.18% (74,434)
Unit type: other	29.84% (40,247)
Mean HR	84.29 (16.42)
Mean MAP	77.83 (10.14)
Mean RR	19.34 (4.74)
Mean SpO ₂	97.02 (95.5, 98.42)
GCS motor	6 (6.0, 6.0)
GCS eyes	4 (3.0, 4.0)
GCS verbal	5 (4.0, 5.0)

Antibiotics	26.34% (29,785)
Vasopressors	12.06% (13,635)
Ventilatory support	32.96% (44,456)
APACHE IVa mortality probability	0.05 (0.02, 0.13)
Hospital mortality	8.84% (11,925)

Table 11. General overview of the cohort features

APACHE Acute Physiology and Chronic Health Evaluation, CVA cerebrovascular accident, GCS Glasgow coma scale, GI gastrointestinal, HR heart rate, MAP mean arterial pressure, MICU medical intensive care unit, RR respiratory rate, SICU surgical intensive care unit.

IX.3.2. Linear models

Reliability curves for the APACHE IVa model and our logistic regression model (Logit) applied to our validation cohort are shown in Figure 6. The older APACHE IVa model systematically over-predicts mortality across the risk spectrum. The degradation of model calibration in models like APACHE has been previously attributed to changes in treatment efficacy over time, cultural changes in propensity to forgo care, ICU utilization, and frequency of early discharge (J. E. Zimmerman, Kramer, McNair, Malila, et al., 2006). However, other work examining model degradation attributed the phenomenon to changes in case mix rather than changes in predictor outcome relationships or outcome frequency (Davis et al., 2018). The Logit model appears better calibrated in patients on the lower end of the predicted risk spectrum but becomes less well calibrated as the predicted risk of mortality increases. Despite this evident miscalibration, the OPR for the Logit model is 0.99 [0.96, 1.02] as compared to 0.74 [0.72, 0.76] for the APACHE IVa model. The Logit model also achieved superior discrimination with an area under the receiver operating characteristic curve (AUC) of 0.887 [0.882, 0.893] as compared to an AUC of 0.864 [0.858, 0.871] for the APACHE IVa model. Similarly, the Logit model had an average precision (AP) of 0.529 [0.512, 0.549] as compared to an AP of 0.460 [0.442, 0.480] for the APACHE IVa model. Figure 7 displays the receiver operating characteristic curves and precision-recall curves in the held-out validation cohort for these models, as well as all models trained in this study.

Figure 6. Reliability curves for the APACHE IVa and Logit models.

Figure 7. Receiver operating characteristic and precision-recall curves.

Receiver operating characteristic and precision-recall curves for all the models. Area under the receiver operating characteristic curve (AUC) and average precision (AP) are provided for each model along with 95% confidence intervals obtained from bootstrapping.

IX.3.3. Sequential models

The OPR for SeqLogit-0.10 and SeqLogit-0.50 in the validation cohort are 0.98 [0.95, 1.01] and 1.00 [0.97, 1.03], respectively, indicating similar overall calibration performance, but the reliability curves for these models shown in Figure 8 suggest that both sequential models are better calibrated than the baseline Logit model alone at the higher end of the risk spectrum. As is evident in Figure 7, the AUC and AP of the sequential models is equivalent to the Logit model. Inspection of the feature importance of the Logit model and the high-risk components of the sequential models provides support for the general intuition of sequential approach. For example, in the Logit model, age was the most important feature with a coefficient of 0.52 on the log odds scale; it drops to fifth most important feature in the high-risk component of the SeqLogit-0.10 model with a coefficient of 0.38 and eighth most important in the high-risk component of the SeqLogit-0.50 model with a coefficient of 0.32. The different contribution of these features can be understood in terms of the change in distribution of the feature in data on which the model is being trained as case mix necessarily shifts. The mean age in the training cohort was 63 years with a standard deviation of 17.30, but when we examine the high-risk cohort identified by the first step of the SeqLogit-0.10 model, we see the mean age increase to 71 years with a decreased standard deviation of 14.78. In the higher-risk subset, age is higher, but with lower variability, and thus less informative for prediction. Similarly, the maximum lactate level on the first day was a top 10 feature in the high-risk component of the SeqLogit-0.50 model with a coefficient of 0.37 but only a

coefficient of 0.16 in the Logit model. The intuition is similar to that of age but in reverse. In the whole training cohort, we expect a wide distribution of lactate and a normal to slightly elevated average lactate, but in the higher-risk patients we expect a higher average lactate and a broader spread. In fact, the mean of the maximum lactate on day 1 for patients in the whole training cohort was 3.11 with a standard deviation of 3.17, and this same feature had a mean of 8.44 with a standard deviation of 5.90 in patients stratified as high risk in the first step of the SeqLogit-0.50 model. The coefficients for each feature of the Logit model and the high-risk components of the sequential models are provided in full in Supplemental Materials.

Figure 8. Reliability curves for sequential models.

IX.3.4. Gradient boosting machine

A subset of the hyperparameters for the xgBoost (extreme gradient boosting (XGB)) model as selected by ten-fold cross-validation (CV) are summarized in Table 12. The best model was chosen based on the optimal negative log-loss and is composed of 1000 trees with a permitted maximum tree depth of 12. The reliability curve for the XGB model is shown in Figure 9: the model appears well calibrated but varies between slight under- and over-prediction as the predicted risk increases resulting in an overall OPR of 1.04 [1.01, 1.07]. The AUC and AP were slightly superior to all other models in this study at 0.907 [0.902, 0.913] and 0.596 [0.579, 0.615], respectively.

Hyperparameter	Value
Learning rate	0.01
Number of trees	1000
Max. tree depth	12
Row sampling	0.6
Column sampling	0.75

Table 12. The XGB model hyperparameters as selected by ten-fold cross-validation XGB extreme gradient boosting.

Figure 9. Reliability curves for the extreme gradient boosting model.

IX.4. Discussion

This work explores strategies for improving the calibration of severity of illness models. We hypothesized that existing models are too inflexible to adequately model the risk of mortality across the full spectrum of illness severity. Furthermore, we postulated that, at minimum, patients with a high risk of mortality signify a distinct group of patients requiring a separate disease severity model mapping their physiological data to the probability of mortality. To examine this further, we developed an updated and augmented severity score based off the APACHE IVa model using

multivariate logistic regression as a baseline and then developed models composed of a combination of two logistic regression models, termed sequential models. The sequential models first used the full cohort regression model to assign risk of mortality, and when that risk exceeds a predefined cutoff, a second model trained in only patients above that cutoff in the training set is then employed to estimate risk. We examined this approach with a cutoff of 0.10 and 0.50 risk of mortality. We compared this simple approach to a sophisticated, state-of-the-art machine learning method: the gradient boosting machine.

Inspection of the reliability curves suggests that the sequential modeling approach results in an improvement in calibration in higher-risk patients over the full cohort model alone, but with no effect on AUC or AP, and minimal effect on the overall OPR. In comparison, the XGB model slightly bests the linear models in terms of AUC, with a larger advantage in terms of AP; the XGB model appears well calibrated on inspection of the reliability curve but varies between under- and over-prediction across the risk spectrum resulting in an over-prediction of overall mortality by 4%. As would be expected, all models improve on APACHE IVa given that they are trained and deployed in data from the same generating source and thus are closer to the validation data both temporally and spatially.

The fundamental premise of this work, that accurate risk quantification with severity of illness models requires that the model be able to quantify risk differently in patients with higher disease severity, is supported by the improved calibration seen on the reliability curves for the sequential models. In agreement with our intuition, features that are important in discerning risk of death across the whole cohort are less important in the highest-risk patients, and the sequential approach allows a re-weighting that supports improved risk quantification at the higher end of the risk spectrum. While the XGB model allows for up to 12 levels of interaction, the gain in discrimination is marginal at the cost of increased variance, with slight overall over-prediction of mortality in the validation cohort; furthermore, this complex model is harder to interpret and developers will need to weigh this difficulty against the small discriminative gains provided.

However, there are two other important conclusions to draw from this work. Foremost, the worst performing model, APACHE IVa, still demonstrated an AUC of 0.864 [0.858, 0.871], not substantially lower than the XGB model that had the highest AUC at 0.907 [0.902, 0.913], and yet it was very poorly calibrated. Shah et al. highlighted this problem in the broader context of clinical machine learning, referring to calibration as the “Achilles’ heel of prediction.” Proper calibration is crucial, especially for informing care decisions, and even strong discrimination performance alone is insufficient for this purpose. That is, while thresholding risk at 50% can assign a label of survival or expiration, for clinical decision support a 10% chance of mortality is not the same as a 40% chance, just as a 95% chance differs from a 55% chance. In addition, with respect to discrimination, the fact that a sophisticated method capable of modeling complex interactions did not substantially outperform less complex methods, or even the out-of-date APACHE IVa model, suggests to us that substantial gains are unlikely to be related to algorithmic advancement alone, at least while we are reliant on a small subset of structured clinical data. While complex methods like xgBoost have led to performance gains in various task domains, allowing models to capture non-linearity and interaction directly from engineered features is less important than the engineering of such features (A. E. W. Johnson & Mark, 2018; Koyner et al., 2018; Rojas, 2018). Better quantification of mortality risk with improved discriminative performance will likely require the identification of other variables that explain more of the variance observed. As a significant amount of patient data reside in unstructured formats like images and clinical notes, modeling

approaches (such as deep learning) that can automatically extract novel features from these sources represent an emerging strategy for advancing the current state-of-the-art (Hinton, 2018; Naylor, 2018; Wainberg et al., 2018). However, we emphasize that it is these rich data sources that demonstrate the usefulness of deep learning, and the application of deep learning to structured inputs traditionally used in clinical models has not yielded performance gains. Extending the intuition behind the sequential approach, we not only anticipate that certain features will contribute more strongly in certain subpopulations but also that the incorporation of new features selectively within relevant subgroups will improve predictive performance.

This work has several important limitations. Foremost, because no other publicly available database (including MIMIC-III) provides the variables used to derive the APACHE IVa score and we chose to use this expansive feature set, we were unable to test the sequential models on a truly external validation set as is expected by the Transparent Reporting of a multivariable prediction model for Individual Prognosis Or Diagnosis (TRIPOD) statement (Collins et al., 2015). We have attempted to allay some of this shortcoming by selecting hyperparameters with ten-fold CV, evaluating our final model on a validation set that the model never encountered during training, using data that is publicly available online, and making our code publicly available online (Cosgriff, 2019). However, the lack of another publicly available database in which to test our models in full compliance with the TRIPOD statement represents this work's greatest shortcoming.

Another important limitation of this work lies in the feature engineering. Unlike the APACHE IVa system that uses the worst value for vital sign parameters such as heart rate, the models we developed automatically processed vital sign inputs from streaming 5-min medians and averaged these data over the first 24 h. Values curated to be the worst are not only used in the standard APACHE score calculation and might be more predictive of mortality than an average over the first day but are also more subject to technical (artifacts, monitoring inaccuracies), user input, and data storage and retrieval issues. In this current digital age, perhaps precise definitions need to be modified so "worse" values are better defined in terms of the data collected, which is often stored as a median value of some kind such as ≥ 5 min time interval medians. In addition, average values may be more representative of the physiological state as sustained over the period of the APACHE day. However, the goal of this work was to compare the sequential approach to the commonly used simple linear modeling approach and a more complex machine learning approach, and this type of feature selection was used in an internally consistent manner across all of these methods. In fact, our model performed as well or better than the standard APACHE worse value method. As such, it is unlikely that it diminishes the conclusion of this paper that sequential modeling improves severity score calibration.

A further limitation is that arbitrary cutoffs for high risk of 0.10 and 0.50 were used, and in future work the cutoff point or points should be considered a hyperparameter chosen via a CV procedure in a data-driven manner. Selection bias also represents an important shortcoming: a significant limitation of this study is the number of patients for whom APACHE data were not present and/or an APACHE IVa score was not calculated as they were excluded. It is unlikely that these data were missing completely at random (MCAR), and thus the missingness is likely informative. Therefore, exclusion of these patients may have biased our models and comparisons. While this does not specifically detract from the goal of this study, it is important to note that models developed in this manner would not be applicable to patients who would have been inadvertently

excluded by this selection bias, and prospective analysis is crucial to safely developing and deploying these tools.

X. TEMPORAL TRENDS IN CRITICAL CARE OUTCOMES IN UNITED STATES MINORITY SERVING HOSPITALS

X.1. Introduction and problem statement

Health disparities continue to plague the United States medical system (Schneider, 2002). Despite higher rates of comorbidities (HHS.gov, 2019), minorities have less access to preventative medicine (Ayanian et al., 1999; Bach et al., 2004; Kendrick et al., 2015), seeking care in lower-performance hospitals with higher complication (Creanga et al., 2014; Ly et al., 2010), readmission (Joynt, 2011), and mortality rates (Haider, 2012; Jha et al., 2007; López & Jha, 2012; Skinner et al., 2005). While the higher acuity and resource utilization of critical illness might seem immune to such disparities, racial differences in the intensive care unit (ICU) have similarly been described (Bime et al., 2016; Lauerma et al., 2015; Rapoport, 2000; Soto et al., 2013; Tyler et al., 2018).

Accordingly, we examined whether improvements in critical care outcomes over the last decade (Erickson et al., 2009; Lilly et al., 2017; Oud & Watkins, 2015; Stevenson et al., 2014)^{19–22} extend to minority serving hospitals. Using a large repository of almost 1.1 million ICU admissions from hospitals across the United States (Alistair E. W. Johnson et al., 2017; Komorowski et al., 2018; Pollard et al., 2018), we describe the temporal trends of critical illness outcomes according to hospital minority composition and whether these trends differed by ethnicity.

X.2. Methods

X.2.1. Data Source

Philips Healthcare, a major vendor of ICU equipment and services, provides a telehealth ICU platform to over 300 hospitals across the United States. Data from participating hospitals is anonymously curated in the electronic Intensive Care Unit Research Institute Database (eICU-RI), a collaborative partnership between Philips Healthcare and the Laboratory of Computational Physiology at Massachusetts Institute of Technology (Alistair E. W. Johnson et al., 2017; Komorowski et al., 2018; McShea et al., 2010; Pollard et al., 2018). It contains high-resolution patient data including demographics, vital signs, laboratory tests, illness severity scores, fluid intake and outputs, and diagnostic coding from patients admitted between 2003 and 2016. Participating hospitals trained clinicians to use the Philips platform, using primary data entry and drop-down boxes to adjudicate patient information and diagnoses, with direct synchronization with laboratory and clinical data.

The most up-to-date formulation of the eICU-RI contains 1.7 million unique first critical illness hospitalizations from 301 hospitals. Missing data included hospital regional location ($n=430,137$) and Acute Physiology and Chronic Health Evaluation (APACHE) IV severity of illness scoring ($n=104,041$). We excluded those admitted prior to 2006 ($n=30,207$) due to low participation and unreliability of data entry, leaving 1,088,109 patients. Of these, 48,514 lacked documentation of length of critical stay, leaving a cohort of 1,039,595 for primary analysis.

X.2.2. Exposure

We used 2010 United States Census data to determine the African American and Hispanic regional means and defined minority-serving hospitals as those with a greater than two-fold African American or Hispanic ICU census than the corresponding regional mean. The cutpoints for African American and Hispanics were 11.30% and 7.0% in the Midwest, 13% and 12.6% in the Northeast, 20.1% and 15.9% in the South and 5.7% and 28.6% in the West, respectively. 28 As an alternate definition, we defined minority-serving hospitals as those with a greater than 25% African American or Hispanic ICU patient census (Haider, 2012).

X.2.3. Outcomes

Our primary outcome was death during critical illness hospitalization. The secondary outcomes were ICU and hospital lengths of stay.

Categorization of trends

We examined year of admission as a categorical (i.e., as individual two-year groups) and continuous variable.

X.2.4. Variables

Basic demographics included age, gender, and ethnicity. Ethnicity was self-reported as white, African American, Hispanic, Asian, Native American, other, or unknown. Admission diagnoses were adjudicated by trained clinicians within the first 24 hours of ICU admission as part of the APACHE IV score system (Harris et al., 2017), and were categorized into the fifteen most common clinical categories, including sepsis, myocardial infarction/angina, trauma, gastrointestinal bleed, arrhythmia, drug/alcohol complications, cerebrovascular accident, coronary artery bypass grafting, pneumonia, malignancy related, congestive heart failure, cardiac arrest, angina, diabetes related, intracranial bleed, other and unknown. The admission APACHE IV score, obtained within 24 hours of ICU admission, was used to quantify severity of illness. The Charlson comorbidity scoring system was used to describe preexisting illness burden (Charlson et al., 1987). ICU unit type (medical, medical surgical, surgical, cardiac, cardiothoracic, and neurological) was included as a series of indicator variables.

X.2.5. Analysis

Baseline characteristics were presented as percentages for categorical variables and mean and standard deviation for continuous variables by hospital minority composition. We used Cox proportional hazards model to estimate hazard ratios (HRs) and 95% confidence intervals (CIs) for the associations between categorical year of admission and mortality. Time to event was defined as the length of stay between ICU admission and date of death or censoring. Patients who were discharged were censored at that time. The models were adjusted for age, sex, ethnicity (white, African American, Hispanic, or other), unit type (categorical), admission diagnosis

(categorical), APACHE IV severity of illness and Charlson comorbidity scores (continuous), and year of admission (categorical; 2006-2008 as the reference group). We used multiplicative interactions to determine whether the effect of admission year on mortality differed according to hospital minority composition and explored whether these findings were consistent across ethnicity.

In secondary analyses, we defined minority-serving hospitals as those with a >25% African American or Hispanic ICU census and conducted survival analyses as in our primary analysis. Second, we applied generalized estimating equations (GEE) with Poisson error distribution, log link function, and exchangeable covariance structure to examine the associations between categorical admission year and mortality in those hospitals with greater than 500 admissions. This approach allowed us to account for within-hospital correlation. Third, to account for hospital participation, we examined our primary analysis in those hospitals that had consistent participation in four consecutive time periods.

As secondary endpoints, we describe ICU and critical illness hospitalization lengths of stay according to minority hospital composition. Using standard least squares regression, including all variables from the primary analysis and an indicator for hospital mortality, we describe the adjusted differences in lengths of stay in minority-serving and non-minority hospitals and how these trends have changed over time.

To determine whether there were differences in critical illness resource utilization across hospitals, we examined the delay to ICU transfer in those patients admitted directly from the emergency department (Chalfin et al., 2007). Using all variables above, and an indicator variable for hospital mortality, we describe whether the delay differed by hospital composition and how these trends have changed over time.

All analyses were performed using JMP Pro 12 and PROC PHREG and PROC GENMOD in SAS 9.4 (both produced by SAS Institute, Cary, NC).

X.3. Results

X.3.1. Usage and characteristics of minority serving hospitals

Of almost 1.1 million critically ill patients, 10% (n=109,022) were cared for in one of 14(7% of sampled hospitals) minority serving hospitals. There was significant ethnic variation in usage of such hospitals, with 25% (n=27,242) of African American and 48% (n=26,743) of Hispanic patients receiving critical care in a minority serving hospital, compared to 5.2% (n=42,941) of white patients. Patients in minority serving hospitals tended to be younger, with a lower comorbidity burden (Table 13), yet a higher level of illness severity on ICU presentation. Minority serving hospitals had a higher relative percentage of ICU admissions for trauma, myocardial infarction, and heart failure, and a lower percentage for sepsis and drug and alcohol complications, than non-minority hospitals. Hospital mortality proportions were higher in minority than non-minority hospitals (10.5 vs 9.5%, $p<0.001$), consistently across ethnic groupings.

Patient and Hospital Characteristics	Minority-Serving Hospitals	Non-Minority-Serving Hospitals
No. of patients	109,022	979,087
No. of hospitals	14	194
Patient characteristics		
Age, yr	61.8 (18.2)	62.8 (17.6)
Sex, F	46.5	45.8
Ethnicity/race		
White	39.4	80.6
African American	25.0	8.6
Hispanic	24.5	3.0
Other/unknown	11.1	7.8
ICU type		
Cardiac/cardiothoracic	27.9	22.5
Medical	14.7	15.7
Medical/surgical	48.7	55.4
Surgical	8.7	6.4
Charlson comorbidity index	3.4 (2.7)	3.5 (2.7)
APACHE IV	55.7 (27.0)	53.4 (25.5)
Admission diagnosis		
Sepsis	7.7	10.2
MI/angina	9.3	8.0
CABG	3.8	3.9
CHF	3.7	3.0
Trauma	5.5	5.1
Drug and alcohol related	2.6	4.4
CVA	3.6	4.0
Pneumonia	3.6	3.6
Intracranial bleed	2.6	1.6
Hospital characteristics		
Academic	28.6	6.7
Hospital beds		
>500	14.2	11.3
250 to <500	36.0	15.4
Hospital mortality	10.5	9.5

Table 13. Baseline Patient and Hospital Characteristics

Definition of abbreviations: APACHE IV = Acute Physiology and Chronic Health Evaluation IV; CABG = coronary artery bypass grafting; CHF = congestive heart failure; CVA = cerebrovascular accident; MI = myocardial infarction. Percentages for categorical variables and mean (SD) for continuous variables are provided.

X.3.2. Temporal trends in critical care mortality

While the incidence of critical illness mortality (Figure 10) and adjusted mortality (Table 14) steadily decreased from 2006 to 2016, the trends differed between minority and non-minority serving hospitals (multiplicative interaction between calendar year and minority-serving hospital p value <0.001). A steady decline in critical illness mortality (HR 0.98, 95%CI 0.97-0.98 per additional calendar year) was observed in non-minority hospitals, but not in minority serving hospitals (Table 14), supplemental table 1).

Figure 10. Incidence of Critical illness mortality

Critical illness mortality (95% confidence interval) in minority and non-minority-serving hospitals from 2006 to 2016 in the United States ($n = 1,088,109$). Trend P values for minority-serving and non-minority-serving hospitals were 0.002 and ,0.001, respectively.

	Admission Year					Multiplicative Interaction Term P Value
	2006–2008	2009–2010	2011–2012	2013–2014	2015–2016	
All hospitals	Ref.	0.86 (0.84–0.89)	0.82 (0.80–0.84)	0.81 (0.79–0.82)	0.80 (0.78–0.82)	—
Minority-serving hospitals	Ref.	0.83 (0.77–0.89)	0.88 (0.82–0.95)	0.79 (0.74–0.85)	0.88 (0.81–0.94)	<0.001
Non-minority-serving hospitals	Ref.	0.87 (0.85–0.89)	0.80 (0.79–0.82)	0.81 (0.79–0.82)	0.79 (0.77–0.81)	
Minority hospital defined as having greater than 25% African American or Hispanic census						
Minority-serving hospitals	Ref.	0.85 (0.80–0.90)	0.83 (0.78–0.87)	0.79 (0.75–0.83)	0.85 (0.80–0.90)	0.05
Non-minority-serving hospitals	Ref.	0.87 (0.84–0.89)	0.81 (0.79–0.83)	0.81 (0.79–0.83)	0.79 (0.77–0.81)	
GEE analysis in hospitals with >500 admissions to account for hospital correlation						
Minority-serving hospitals	Ref.	0.94 (0.92–0.95)	0.93 (0.91–0.95)	0.90 (0.88–0.93)	0.94 (0.91–0.97)	0.05
Non-minority-serving hospitals	Ref.	0.94 (0.92–0.95)	0.90 (0.88–0.92)	0.89 (0.87–0.91)	0.88 (0.86–0.90)	
Cox regression in hospitals with participation in four consecutive time periods						
Minority-serving hospitals	Ref.	0.82 (0.76–0.88)	0.87 (0.81–0.94)	0.75 (0.70–0.81)	0.87 (0.81–0.94)	0.003
Non-minority-serving hospitals	Ref.	0.87 (0.85–0.89)	0.81 (0.79–0.83)	0.82 (0.80–0.84)	0.80 (0.77–0.82)	

Table 14. Hazard Ratio (95% Confidence Interval) of Critical Illness Mortality per Admission Year Category Provided, with 2006–2008 Considered as Reference for All Analyses.

Definition of abbreviations: GEE = generalized estimating equation; Ref. = reference. Adjusted hazard ratios for critical illness mortality according to admission year and hospital minority composition. Adjusted for age, sex, ethnicity, unit type, admission diagnosis, Charlson comorbidity score, and illness severity. In addition, alternative definition of hospital minority composition, and analytic approaches to account for within hospital correlation and hospital participation, are provided. Multiplicative interaction P value between indicator for minority-serving hospitals and admission year (defined continuously) provided.

This temporal inequality was most apparent amongst African American patients (multiplicative interaction between calendar year and minority-serving hospital p values 0.02, 0.07, and 0.04 amongst African American, Hispanic, and white patients, respectively), where each additional calendar year was associated with 3% lower adjusted mortality (HR 0.97, 95%CI 0.96–0.97) in non-minority hospitals, compared to no change in minority-serving hospitals (HR 0.99, 95%CI 0.97–1.01) (Figure 11).

Figure 11. Adjusted hazard ratio (95% CI) of hospital mortality per additional calendar year of admission between 2006 and 2016, stratified by ethnicity.

The hazard ratios were adjusted for age, sex, unit type, admission diagnosis, Charlson comorbidity score, illness severity, and year of admission (defined continuously). The *P* values for the multiplicative interaction between calendar year and minority-serving hospital were 0.02, 0.07, and 0.04 among African American, Hispanic, and white patients, respectively. *N* = 1,039,595 patients in 208 hospitals. CI = confidence interval.

X.3.3. Sensitivity analyses of mortality

Using a threshold of more than a 25% African American or Hispanic ICU census to define a minority hospital resulted in 26 minority-serving hospitals serving 177,186 patients. Patients within these hospitals had higher mortality rates and similarly less temporal improvement (multiplicative interaction between each additional calendar year and minority-serving hospital *p* value 0.05) than those in non-minority hospitals (Table 15). Analyses that accounted for within-hospital correlation and participation resulted in similar findings.

		Admission Year					Multiplicative Interaction Term <i>P</i> Value
		2006–2008	2009–2010	2011–2012	2013–2014	2015–2016	
Adjusted change in ICU length of stay							
Minority-serving hospitals	Ref.	−0.35 (−0.43 to −0.27)	−0.48 (−0.56 to −0.40)	−0.41 (−0.49 to −0.34)	−0.44 (−0.52 to −0.36)		<0.001
Non-minority-serving hospitals	Ref.	−0.42 (−0.45 to −0.39)	−0.56 (−0.59 to −0.53)	−0.64 (−0.66 to −0.61)	−0.67 (−0.70 to −0.65)		
Adjusted change in hospital length of stay							
Minority-serving hospitals	Ref.	−0.74 (−0.91 to −0.59)	−1.12 (−1.27 to −0.96)	−1.01 (−1.17 to −0.87)	−0.98 (−1.13 to −0.81)		<0.001
Non-minority-serving hospitals	Ref.	−0.56 (−0.63 to −0.50)	−0.90 (−0.95 to −0.85)	−1.14 (−1.18 to −1.09)	−1.37 (−1.41 to −1.32)		

Table 15. Change (95% CI) in Length of ICU and Critical Illness Hospital Stay (in Days),

Relative to 2006–2008, Adjusted for Age, Sex, Ethnicity, Unit Type, Admission Diagnosis, Charlson Comorbidity Score, Illness Severity, Admission Year Category, and Hospital Mortality

Definition of abbreviation: Ref. = reference. CI = Confidence Interval.

Adjusted change of ICU and critical illness hospitalization lengths of stay from 2006 to 2016. Multiplicative interaction between indicator for minority-serving hospitals and admission year (defined continuously) are provided.

X.3.4. Temporal trends in ICU and hospital lengths of stay

The lengths of ICU stay and critical illness hospitalization were higher among minority than non-minority serving hospitals (3.1±3.9 and 7.3±6.9 days compared to 2.9±3.6 and 6.4±6.2 days, respectively), a difference that remained in an adjusted analysis that included hospital mortality [0.03 (95%CI 0.02-0.04, p<0.001) and 0.21(95%CI 0.20-0.23, p<0.001) days longer ICU and hospital stays in minority serving hospitals, respectively]. ICU and hospital lengths of stay steadily decreased in non-minority serving hospitals [–0.08 (95%CI –0.08,–0.07, p<0.001) and –0.16 (95% CI –0.16,–0.15, p<0.001) days per additional calendar year, respectively], but significantly less so amongst minority serving hospitals (multiplicative interaction between minority hospital and admission year p values <0.001 for both ICU and hospital lengths of stay), remaining essentially constant from 2011 to 2016 (Figure 12, Table 15). This temporal disparity was most apparent in African American patients, for whom length of stay decreased in non-minority hospitals, but not in minority serving hospitals (Table 16).

Figure 12. Temporal trends in hospital and ICU length of stay in minority- and non-minority-serving hospitals.

Mean (95% confidence interval) lengths of stay according to year of admission are provided. Trend *P* values were all less than 0.001, except for ICU length of stay in minority-serving hospitals (*P*= 0.06).

	ICU Length of Stay		Hospital Length of Stay	
	Minority-Serving Hospital	Non-Minority-Serving Hospital	Minority-Serving Hospital	Non-Minority-Serving Hospital
African American				
Adjusted change (95% CI), d	-0.01 (-0.02 to 0.01)	-0.09 (-0.10 to -0.08)	-0.03 (-0.07 to 0.01)	-0.21 (-0.23 to -0.20)
<i>P</i> value	0.46	<0.001	0.19	<0.001
Hispanic				
Adjusted change (95% CI), d	-0.09 (-0.10 to -0.07)	-0.06 (-0.07 to -0.04)	-0.20 (-0.23 to -0.16)	-0.12 (-0.15 to -0.09)
<i>P</i> value	<0.001	<0.001	<0.001	<0.001
White				
Adjusted change (95% CI), d	-0.04 (-0.05 to -0.02)	-0.08 (-0.08 to -0.07)	-0.08 (-0.11 to -0.06)	-0.16 (-0.17 to -0.16)
<i>P</i> value	<0.001	<0.001	<0.001	<0.001

Table 16. Adjusted Change (95% CI) per Additional Calendar Year in the Length (in Days) of ICU and Critical Illness Hospital Stay according to Ethnicity.

Definition of abbreviation: CI = confidence interval. Adjusted change in length of stay per additional calendar year according to ethnicity. The change in length of stay was adjusted for age, sex, ethnicity, unit type, admission diagnosis, Charlson comorbidity score, illness severity, admission year (defined continuously), and hospital mortality. Multiplicative interaction P values between indicator for minority-serving hospitals and admission year (defined continuously) were less than 0.001 within each ethnic stratum.

X.3.5. Temporal trends in delay to ICU transfer in patients admitted from the emergency department

Amongst 567,325 ICU admissions from the emergency department, the mean $\pm$ SD delay until ICU admission was 3.9 $\pm$ 16.3 hours among non-minority hospitals and 5.9 $\pm$ 19.4 hours among minority-serving hospitals ($p<0.001$). Over the last ten years, the adjusted delay decreased by 7.6 minutes (95%CI -8.66 to -6.51, $p<0.001$) per additional calendar year in non-minority hospitals yet had little change (1.0 minutes; 95%CI-2.62 to 4.67, $p=0.58$) in minority-serving hospitals.

X.4. Discussion

In our sampling of approximately 200 hospitals across the United States, almost a third of critically ill African American and half of critically ill Hispanic patients received critical care in just 7% of surveyed hospitals. These minority-serving hospitals showed significantly less decline in critical illness mortality and length of stay over the last decade, compared to non-minority hospitals. While this inequality was consistent across ethnicities, it was most pronounced amongst African Americans, in whom we observed no temporal improvement in critical care mortality or reduction in length of stay during this period.

Minority serving hospitals tended to care for younger patients, with a lower overall burden of disease comorbidity, yet with a paradoxically higher severity of illness severity and mortality. Accordingly, it is difficult to determine whether our findings reflect caring for an increasingly disadvantaged population or differences in hospital resource utilization. As a proxy for hospital practice patterns, we examined the delay to ICU admission for those patients admitted through the emergency department, a clinically important indicator (Chalfin et al., 2007; Churpek et al., 2013; Intas et al., 2011; Rincon et al., 2010). We found that minority-serving hospitals had significantly longer delays with little temporal improvement, while the adjusted delay to ICU admission decreased by almost eight minutes per year in non-minority hospitals.

Regardless of how much of the increased mortality risk is due to patient or hospital-specific issues, the high minority usage of these hospitals highlights the obstacles facing African Americans in the United States. The “neighborhood effect”, whereby location of residence has a profound effect on outcomes (Fowler et al., 2010; Frei et al., 2010; Haider, 2008; Kind et al., 2014; Yergan et al., 1987), highlights the socioeconomic barriers to achieving equitable health care access, compounded by differences in practice patterns and resource utilization that extend into the ICU (Rapoport, 2000; Shippee et al., 2011; Williams et al., 1995). From the perspective of health care delivery, recognizing the challenges facing minority-serving hospitals is particularly important in the current “pay for performance” reimbursement paradigm (Ryan, 2013) so as not to unfairly penalize the most vulnerable hospitals (Joynt & Jha, 2013). Our data provides clinical context for

this concern and underscores the need for additional support for minority-serving hospitals to ensure they have the appropriate resources to meet their strenuous clinical demand.

Our analysis has several notable limitations. Confounding due to either admission or discharge bias is possible, particularly since minority patients tend to receive more intensive therapy and testing towards end of life. Whether the ethnic distributions in the ICU were similar to those of the hospital were not known. In addition, how representative hospitals that choose to use the Phillips platform are is not known, and important patient characteristics, such as income, insurance type, and lifestyle choices, as well as hospital information, were not available. Finally, using more granular population census definitions, such as county codes, could improve the precision of minority-hospital definitions. However, we examined two different definitions of minority hospitals, with similar results, and our primary findings were consistent through a range of sensitivity analyses and statistical approaches. Despite these limitations, the sheer size and granularity of this dataset are an important strength, providing a unique snapshot of modern American critical care over the last ten years.

XI. INCIDENCE AND RISK MODEL DEVELOPMENT FOR SEVERE TACHYPNEA FOLLOWING TERMINAL EXTUBATION

XI.1. Introduction and problem statement

Nearly 25% of deaths in the United States occur in intensive care units (ICUs) (Angus et al., 2004; Khandelwal et al., 2016). The majority of deaths in the ICU reflect a transition from curative to comfort focused care through a process of palliative ventilator withdrawal (PVW) (Cook & Rocker, 2014). Numerous professional societies and patient groups have advocated for improved management of ventilated ICU patients transitioning to palliative care (CARLET et al., 2004; Holloway et al., 2014; Lanken et al., 2008; Salinas et al., 2016; Steinhauser, 2000; Teno, 2004; Truog et al., 2008). Research conducted over the past 15 years has informed several aspects of palliative care in the ICU setting, particularly how to improve communication and family-centered care (Aslakson et al., 2017; Carson et al., 2016; Connors et al., 1995; Creutzfeldt et al., 2017; Curtis, Downey, et al., 2016; Curtis et al., 2002; Curtis, Treece, et al., 2016; Davidson et al., 2017; Engelberg et al., 2010; Lautrette et al., 2007; Nelson et al., 2013; Seaman et al., 2017; Sullivan et al., 2016; Truog et al., 2008; D. B. White et al., 2012; Douglas B. White et al., 2008, 2016; Zier et al., 2009). In contrast, relatively little research has focused on the process of PVW, thus the practice varies widely across ICUs (Abarshi et al., 2014; Brinkkemper et al., 2011; Campbell, 2007; Mayer & Kossoff, 1999; Papavasiliou et al., 2014; Robert et al., 2017; van Beinum et al., 2015), and little is known about factors associated with better symptom control among terminally ill patients undergoing this procedure.

Patients undergoing PVW who are able to report symptoms describe the sensation of breathlessness or dyspnea, as the most distressing symptom they experienced (Campbell, 2012; Campbell et al., 1999; Puntillo et al., 2013; Truog et al., 2008). Unfortunately, there is little data to guide the clinical approach to minimize respiratory distress during PVW. While evidence suggests lower rates of respiratory distress with gradual reduction of ventilator support (terminal weaning) versus immediate extubation (Robert et al., 2017), it is not known whether administering analgesia/sedation pre-extubation (anticipatory dosing) (Billings, 2012; Delaney & Downar, 2016) relieves distress more effectively than giving these drugs only in response to observed symptoms (Truog et al., 2001, 2012).

Assessing physical discomfort among patients undergoing PVW is challenging. A study of terminally ill ICU patients revealed that 50% were unable to communicate, and 20-30% were comatose (Smedira et al., 1990). Empirical evidence suggests that many patients deemed comatose retain the ability to experience the sensation of respiratory distress (Campbell, 2008). Moreover, a study of 104 ICU patients, most with neurological injuries, found 40% of patients deemed comatose by highly trained physicians actually displayed clinical signs of consciousness (Schnakers et al., 2009).

Taken together, this report leverages the Medical Information Mart for Intensive Care III (MIMIC-III) database (Alistair E. W. Johnson et al., 2016) to better examine severe tachypnea in a cohort of ICU patients undergoing PVW. MIMIC is a rich dataset that collects detailed physiological and clinical data from patients cared for in the 7 ICUs at Beth Israel Deaconess Medical Center in Boston.

The main objectives of the study were to:

- 1) Describe the prevalence.

2) Identify modifiable factors associated with the occurrence of severe tachypnea among patients who underwent PVW through development of multivariable prediction models.

XI.2. Methods

XI.2.1. Data Source

Data were ascertained from the MIMIC-III (Alistair E. W. Johnson et al., 2016); a publicly available dataset managed by the Laboratory for Computational Physiology at the Massachusetts Institute of Technology (MIT-LCP, 2018), which includes data from 7 medical, surgical, and subspecialty ICUs at the Beth Israel Deaconess Medical Center (BIDMC) in Boston. Data from January 1, 2008- December 31, 2012 were used for this study. MIMIC-III contains de-identified high resolution physiological and treatment data, including chart notes, continuous telemetry data and vital signs, nursing assessments, and the complete electronic health record (EHR) (Alistair E. W. Johnson et al., 2016). The study received exempt status by the Institutional Review Board at BIDMC.

XI.2.2. Study population

Mechanically ventilated subjects undergoing PVW were identified from MIMIC-III based on the following eligibility criteria:

1. Age ≥ 18 .
2. Mechanical ventilation followed by extubation.
3. Comfort measures only (CMO) documented in an order or physician note prior to ventilator withdrawal (CMO order indicates all therapies are directed towards comfort but does not specify particular treatment approaches).
4. Death within 24-hours of extubation. Sample size was determined by convenience sample of all data available for analysis.

XI.2.3. Data Elements

Patient-level variables were extracted from the dataset as potentially associated with tachypnea or treatment of dyspnea based upon prior literature (Abarshi et al., 2014; Binks et al., 2017; Campbell, 2007; Campbell et al., 1999, 2015; Cook et al., 2003; Cottreau et al., 2016; Curtis, 2005; Deschepper et al., 2013; Engelberg et al., 2010; Hawryluck et al., 2002; O'Mahony et al., 2003; Parshall et al., 2012; Robert et al., 2017; Sanders et al., 2012; Soffler et al., 2017; Truog et al., 2001; van Beinum et al., 2015; Willms & Brewer, 2005) and clinical experience. These variable included: demographics, insurance status, Glasgow Coma Score (GCS) (score > 8 indicates lack of severe coma) (G. Teasdale & Jennett, 1974), the number of ventilator days (dichotomized at the median of 48 hours), and Sequential Organ Failure Assessment (SOFA)(Jean-Louis Vincent et al., 1998) score (dichotomized at the median > 8), and its components (e.g., Partial pressure of arterial oxygen content to fraction of inspired oxygen content (PaO₂:FiO₂) ratio most proximal to extubation.(Knaus WA et al., 1985) A binary variable was created indicating the presence of

reduced diffusion capacity of lung $\text{PaO}_2:\text{FiO}_2 < 200$, a marker of the acute respiratory distress syndrome. Anticipatory dosing with opioids was defined as the administration (by any route, bolus or increase in continuous infusion rate) of morphine, fentanyl, or hydromorphone, within one hour pre-extubation (Billings, 2012; Mazer et al., 2011; Truog et al., 2012). Fentanyl and hydromorphone were converted to morphine equivalent doses and a binary variable was created indicating whether opiates were administered within 60 minutes prior to extubation. Post extubation opiate administration was indicated by variables created in a similar fashion (morphine, and morphine equivalent doses of fentanyl and hydromorphone) at 1-hour, and 6-hours post-extubation respectively. Variables indicating benzodiazepine (midazolam, lorazepam, diazepam) and propofol administration were created at the time points of within 60-minutes before, and 60-minutes post extubation. Primary and secondary ICU diagnoses were extracted from the electronic health record. Primary neurological diagnosis was defined as by the presence of any International Classification of Disease (10th revision) code for ischemic stroke, intracerebral hemorrhage, subarachnoid hemorrhage, status epilepticus, anoxic brain injury, cerebral edema, brain herniation/compression, hydrocephalus, or brain mass (van Drimmelen-Krabbe et al., 1998). Physiologic measures such as heart rate, respiratory rate and other vital signs were obtained from telemetry data for up to 6 hours post extubation or death. All telemetry data are validated by the patient's ICU nurse, with inaccurate values corrected or removed as part of routine clinical documentation. ICU type (e.g. medical, surgical, subspecialty) was also extracted.

XI.2.4. Outcomes

The primary outcome of interest was severe tachypnea following PVW. The American Thoracic Society defines dyspnea as the “subjective experience of breathing discomfort that consists of qualitatively distinct sensations that vary in intensity” (Parshall et al., 2012). Therefore dyspnea is not uniformly assessed, and while tachypnea is not equivalent to dyspnea, it is a measurable objective sign that correlates with breathlessness in controlled settings (Banzett et al., 2008; Simon et al., 1989). We used a respiratory rate of greater than 30 breaths per minute (validated from telemetry) to define severe tachypnea (McGee, 2012), which was measured up to 1-hour, and until 6-hours from the time of extubation or death.

XI.2.5. Statistical Analysis

Frequencies were used to describe all variables. Bivariate analyses using logistic regression measured the association of individual independent variables and the occurrence of a severe tachypnea episode measured at two time points; within 1-hour and 6-hours post-extubation (outcomes). Patient characteristics (described in Data Elements section) included as independent variables were: demographic information, Glasgow Coma Score > 8 , mechanical ventilation > 48 hours prior to PVW, SOFA Score > 8 , $\text{PaO}_2:\text{FiO}_2$ ratio < 200 , anticipatory dosing with opiates, pre and post-extubation benzodiazepines, pre and post-extubation propofol, opiate dosing within 1-hour and 6-hours post extubation, non-neurological primary diagnosis, and ICU type. The process for variables selection was as follows: independent variables associated with the outcomes at $P < 0.20$ in these unadjusted analyses were included in a multivariable logistic regression model for the outcome at 1-hour and 6-hours controlling for time at risk (patient survival time). Repeated episodes of severe tachypnea within the same patient, and patients who died,

were censored from the analysis of population at risk. Complete case analysis was used for missing data. Odds ratios with 95% confidence intervals were generated from these analyses. Cox-proportional hazard model described time to tachypnea event and Kaplan Meier time to event curves were generated comparing patients receiving and not receiving anticipatory dosing of opiates. Analyses were performed using R and Stata software packages.

XI.3. Results

Among 17,692 patients in the MIMIC-III database during the years 2008-2012 there were 1,251 patients undergoing PVW. The analytical sample consisted of 822 patients with complete CMO and post-extubation respiratory rate data available for analysis, among whom 248 (30%) patients experienced at least one episode of severe tachypnea following PVW. Median age was 73 years (Inter-quartile Range (IQR) 60-82) with 68% of patients over age 65 (Table 17). Of note, 50% of patients had a GCS ≤ 8 , indicating severe coma, and only 67% had a non-neurological diagnosis. As shown in Figure 13, the highest proportion of patients experienced an episode of severe tachypnea within first hour of extubation (19%). Only 20% received anticipatory dosing with opiates. The median morphine equivalent bolus dose administered within 60-minutes before extubation was 4mg (IQR 2-5), median change in continuous infusion rate was 5mg/hour (IQR 3-10). Post extubation, the majority of patients received opiates (58%), followed by benzodiazepines (30%) and propofol (8%).

Figure 13. Number of patients with post-extubation tachypnea by time.

(n=822)

Characteristic	No. (%) Patients With Characteristic	No. (%)* with Severe Tachypnea at 1-hour (n=153)	Odds Ratio	No. (%)* with Severe Tachypnea at 6-hours (n=248)	Odds Ratio
Age, y					
18-40	31 (4)	8 (26)	1.54 (0.68-3.51)	10 (32)	1.00
41-60	156 (19)	32 (21)	1.15 (0.75-1.78)	49 (31)	0.93 (0.40-2.15)
61-80	381 (47)	66 (17)	0.84 (0.59-1.20)	114 (30)	0.86 (0.39-1.93)
81-100	250 (31)	47 (18)	1.00 (0.69-1.48)	73 (29)	0.83 (0.37-1.87)
Male	449 (55)	92 (60)	1.30 (0.92-1.84)	142 (57)	1.20 (0.89-1.62)
Ethnicity					
White	593 (85)	106 (17)	0.67 (0.41-1.10)	168 (28)	1.00
Black	41 (6)	16 (39)	3.00 (1.55-5.80)	18 (44)	1.83 (0.96-3.48)
Asian	23 (3)	3 (13)	0.90 (0.30-2.70)	9 (39)	1.50 (0.64-3.54)
Hispanic	21 (3)	4 (19)	0.67 (0.41-1.10)	5 (23)	0.78 (0.28-2.18)
Other	22 (3)	3 (14)	0.64 (0.20-2.30)	7 (32)	1.09 (0.44-2.73)
Insurance					
Medicare	571 (70)	108 (19)	1.00	178 (31)	1.00
Medicaid	51 (6)	3 (18)	0.93 (0.44-1.95)	15 (29)	0.85 (0.45-1.59)
Private	180 (22)	34 (19)	1.01 (0.66-1.55)	49 (27)	0.81 (0.55-1.17)
Uninsured	16 (2)	2 (13)	0.62 (0.14-2.74)	4 (25)	0.68 (0.22-2.14)
GCS > 8	406 (50)	131 (85)	1.65 (1.02-2.66)	104 (26)	1.47 (0.99-2.17)
Ventilation >48 hours	402 (51)	75 (52)	1.05 (0.73-1.50)	126 (31)	1.16 (0.85-1.58)
SOFA score >8	402 (49)	89 (58)	1.56 (1.09-2.23)	127 (52)	1.20 (0.88-1.62)
PaO ₂ :FiO ₂ < 200	268 (34)	76 (49)	2.13(1.50-3.05)	100 (40)	1.48 (1.09-2.03)
No Anticipatory Dosing	652 (79)	128 (83)	1.35 (0.85-2.14)	128 (20)	1.34 (0.86-2.08)
Non-neurological primary diagnosis	548 (67)	124 (81)	2.39 (1.55-3.65)	52 (21)	1.76 (1.23-2.51)
ICU type					
Surgical	199 (24)	24 (16)	1.00	50 (26)	1.00
Medical	397 (48)	95 (62)	2.29 (1.41-3.71)	134 (36)	1.61 (1.09-2.36)
Trauma	104 (13)	14 (9)	1.13 (0.56-2.30)	25 (25)	0.97 (0.55-1.68)
Cardiac	81 (10)	14 (9)	1.52 (0.74-3.12)	27 (34)	1.51 (0.86-2.65)
Cardiac Surgical	36 (4)	6 (4)	1.46 (0.55-3.87)	10 (32)	1.38 (0.61-3.13)

Table 17. Patient characteristics and their unadjusted association with severe tachypnea episodes within 1-hour, and 6-hours of extubation.

(N=822). * Reported proportions are by row and will not sum to 100. Bolded items indicate statistical significance at $P < 0.05$. Glasgow Coma Scale (GCS) < 9 indicates a state of severe coma. SOFA, sequential organ failure assessment score. ICU, Intensive Care Unit. PaO₂, Partial pressure of oxygen. FiO₂, Fraction of inspired oxygen. Anticipatory dosing indicates administration of opiates within 60 minutes before extubation. PaO₂:FiO₂ < 200 indicates substantial impairment in lung oxygen diffusion capacity and is a marker of the Acute Respiratory Distress Syndrome (ARDS).

In the unadjusted analysis, factors associated with having an episode of tachypnea within 1 hour post extubation at the $P < 0.20$ level were: male gender, black race, GCS > 8 , SOFA score > 8 , PaO₂:FiO₂ ratio < 200 , no anticipatory dosing, no post extubation opiates, non-neurological primary diagnosis, and ICU-type. When these variables were entered into a multivariable model (Table 18), factors that remained independently associated with having a severe tachypnea episode were: no anticipatory dosing (aOR 2.08 1.03-4.19), no post extubation opiates 1.90 (1.19-3.00), GCS > 8 (aOR 2.21, 1.30-3.00), PaO₂:FiO₂ < 200 (aOR 3.33, 2.19-5.04). The SOFA score was excluded from the final model to avoid collinearity with PaO₂:FiO₂ ratio. To estimate the potential for synergistic effects for anticipatory dosing and post-extubation opiates, an interaction term (e.g. the product of these effects) was included. We also explored the potential for interaction between gender and race, but observed little evidence in favor of this, and so the model assumes a gender effect consistent across categories of racial identification, and vice-versa. The concordance statistic for the final model was 0.65 (95% CI 0.60-0.70).

Characteristic	1-hour post PVW Adjusted Odds Ratio (95% CI)	6-hours post PVW Adjusted Odds Ratio (95% CI)
Male gender	1.54 (1.02-2.34)	1.48 (1.03-2.12)
No Anticipatory Dosing	1.77 (1.03-3.03)	NS
GCS > 9	1.88 (1.09-3.25)	1.99 (1.26-3.15)
PaO ₂ :FiO ₂ < 200	2.70 (1.75-4.14)	1.98 (1.35-2.94)
Non-neurological primary diagnosis	3.02 (1.84-4.95)	NS

Table 18. Independent predictors* of severe tachypnea at 1-hour and 6-hours of PVW.

PVW, Palliative Ventilator Withdrawal. CI, Confidence Interval. NS, Not significant. PaO₂, Partial pressure of oxygen. FiO₂, Fraction of inspired oxygen. PaO₂:FiO₂ <200 indicates substantial impairment in lung oxygen diffusion capacity and is a marker of the Acute Respiratory Distress Syndrome (ARDS). Anticipatory dosing indicates administration of opiates within 60 minutes before extubation. *Covariates included in the model were identified gender, race, no anticipatory dosing, GCS > 8, PaO₂:FiO₂ ratio <200, non-neurological primary diagnosis, and ICU type.

In the unadjusted analysis of factors associated with having an episode of tachypnea within 6 hours post PVW at the $P < 0.20$ level were: male gender, black race, GCS > 8, PaO₂:FiO₂ ratio <200, no post-extubation opiates up to 6-hours, non-neurological primary diagnosis and ICU-type. When these variables were entered into a multivariable model, factors that remained independently associated with having a severe tachypnea episode were: male gender (aOR 1.48, 1.03-2.09), no post extubation opiates up to 6-hours (aOR 2.06 (1.42-3.00), GCS > 8 (aOR 2.25, 1.45-3.53), and PaO₂:FiO₂ ratio <200 (aOR 2.34, 1.60-3.41).

Cox proportional hazard modeling including the covariates of: male gender, black race, GCS > 8, PaO₂:FiO₂ ratio <200, no anticipatory dosing, no post-extubation opiates up to 6-hours, non-neurological primary diagnosis and ICU-type revealed only patients with PaO₂:FiO₂ ratio <200 were significantly associated with time to tachypnea episode (Hazard Ratio 1.48, 1.11-1.97). Kaplan Meier time to event curves comparing patients receiving and not receiving anticipatory dosing revealed no difference in time to death over the full 24-hour post extubation period between groups (Figure 14). Log-rank test for equality of survivor functions ($P=0.28$).

Figure 14. Kaplan-Meier survival curves by anticipatory dosing of opiates pre-extubation.

XI.4. Discussion

We identified a 30% rate of severe tachypnea during palliative ventilator withdrawal among a mixed population of critically ill patients. The highest incidence of severe tachypnea occurred in the first hour following extubation. A key finding of this analysis is the odds of severe tachypnea following PVW are more than doubled without anticipatory dosing of opiates. While a similar strength and direction of association was seen with post-extubation opiate dosing, these findings lend empirical support to the theoretical rationale for anticipatory dosing of opiates pre-extubation with the goal of preventing uncontrolled symptoms during PVW. There was notably no difference in time to death between groups of patients receiving anticipatory dosing and those that did not. Males, a higher level of consciousness, no opiates given within 1-hour of extubation, and lung injury ($\text{PaO}_2\text{:FiO}_2$ ratio < 200) were also associated with greater risk for severe tachypnea following PVW.

Despite the immense resources devoted to the monitoring and care of ICU patients, the relatively high prevalence of this marker of distress during the transition to comfort measures is surprising, but consistent with prior estimates (Mayer & Kossoff, 1999; Morris & Galicia-Castillo, 2016). A small study of 32 terminally extubated patients in a neurological ICU reported a 59% rate of labored breathing (Mayer & Kossoff, 1999). Apart from the small sample size of the study, the assessment of labored breathing is not fully standardized and may have led to biased estimates.

Others have estimated rates of dyspnea among terminally extubated patients of 34% (Mayer & Kossoff, 1999; Morris & Galicia-Castillo, 2016).

Previous evaluations of the use of anticipatory dosing of opiates in the setting of PVW have yielded unclear results. Two studies found anticipatory dosing was not associated with earlier time to death, but did not examine symptom control or compare it with reactive dosing (Ankrom et al., 2001; Chan et al., 2004). A small study of 29 patients described anticipatory dosing as a treatment option for patients who were alert and at risk for poor post-extubation symptom control, however characteristics of patients at risk were not delineated (O'Mahony et al., 2003).

Lastly, a study of 32 patients undergoing different PVW practices advised caution when using anticipatory dosing, particularly among comatose patients (Campbell et al., 1999). While reviews and guidelines treat the practice of dosing medications in anticipation of end-of-life symptoms as ethically and clinically acceptable, they also cite a lack of detailed empirical evidence (Downar et al., 2016; Rubenfeld, 2004).

There were differences in risk factors identified by logistic regression and Cox proportional hazard models. These differences are in part attributable to logistic regression models estimating risk at the time points of 1 and 6-hours post extubation, while the Cox model estimated risk at time of the tachypnea event. The effect of time varying covariates may be less evident in the Cox model, such as waning pharmacologic effect of anticipatory dosing (pre-extubation opiates), but more apparent when comparing 1-hour and 6-hours post extubation using logistic regression. Taken together, interpretation of both models suggests that patients with $\text{PaO}_2:\text{FiO}_2 < 200$ are at highest risk for a tachypnea episode. This characteristic is readily available in ICU patients prior to extubation and may be used to tailor approaches for reducing the risk of poorly controlled symptoms in the process of PVW. The mechanisms by which males and higher levels of consciousness appear to be at risk for higher rates of severe tachypnea is unclear and warrants further exploration in carefully constructed prospective analysis. In the interim, particular attention should be paid to these risk groups during the process of PVW with carefully titrated symptom directed treatments.

A broader context at odds with our findings, are studies reporting that the majority of family members of patients dying in the ICU perceive deaths to have occurred with comfort (Rocker et al., 2004). However the family members surveyed in this prospective study of 155 ICU patients undergoing terminal extubation in Canada identified 30% of families reporting patients were “mostly comfortable,” and 9% of families reporting “slightly uncomfortable” or “very uncomfortable” (Rocker et al., 2004). While family perception of comfort and support during dying process in the ICU is a core component of high-quality ICU care, the individual patient's experience remains paramount. Studies utilizing repeated measurement of objective markers of symptom control during PVW are rare in the medical literature.

Our study offers the advantage of precise calculation of the population at risk to provide robust estimates of episodes of severe tachypnea during PVW. It is the largest study of symptom control among PVW patients to date. This study is limited by the fact that tachypnea is not equivalent to the subjective experience of dyspnea among patients able to self-report. The respiratory distress observation scale (RDOS) (Campbell et al., 2010) is the only validated measure for dyspnea in this population, but it has yet to be universally adopted. The reporting rate of the RDOS was low in our population and limited its use for this analysis. Other markers of respiratory distress, such as labored or obstructed patterns of breathing are used as markers of comfort during the PVW

process, but the integration of a validated, systematic approach to identification and management of patients during PVW remains to be standardized in critical care practice (Campbell, 2007; Puntillo et al., 2013). The use of observational data introduces the potential for uncontrolled confounding. However, analysis of the complete EHR, including free text documentation of each patient's entire hospital course renders the possibility of overlooking clinically relevant patient or ICU characteristics unlikely. While ICU nurses validate all vital signs, some validation could occur erroneously. There were no major institutional changes in the process of palliative ventilator withdrawal during the study period. Nevertheless, the process of de-identification within the dataset limits formal statistical tests for time period effects, which could alter the reported associations.

XII. ADMISSION DISORIENTATION AND RISK OF CODE STATUS CHANGE

XII.1. Introduction and problem statement

Disorientation occurs commonly in hospitalized and critically ill patients, particularly when associated with delirium, and has been linked to a range of poor patient outcomes (Gleason et al., 2015). Patient disorientation also affects qualitative aspects of the intensive care unit experience, such as perception of increased patient suffering and nursing stress. How disorientation might affect medical decision making has been less researched.

Changing goals of care during a critical illness stay is one of the most important medical decisions to make. While factors predictably associated with poor outcomes, such as advanced age, illness severity, and low likelihood of meaningful recovery, lead to the de-escalation of care, whether patient disorientation might influence care de-escalation has not specifically explored. Using a large, national sample of critically ill patients, we examined the effect of disorientation on Intensive Care Unit (ICU) admission and risk of de-escalation of code status. Using the verbal component of the GCS score (G. M. Teasdale et al., 1998), which characterizes level of disorientation independently of eye and motor function, in those patients admitted as full code, we describe the adjusted risk of a code status change to DNR, DNI or similar reduction of care.

XII.2. Methods

XII.2.1. Data Source

Phillips Healthcare, a major vendor of ICU equipment and services, provides a telehealth ICU platform to over 300 hospitals across the United States. Data from participating hospitals is anonymously curated in the electronic Intensive Care Unit Research Institute Database (eICU-RI), a collaborative partnership between Philips Healthcare and the Laboratory of Computational Physiology at Massachusetts Institute of Technology (Alistair E. W. Johnson et al., 2017; Komorowski et al., 2018; McShea et al., 2010; Pollard et al., 2018). It contains high-resolution patient data including demographics, vital signs, laboratory tests, illness severity scores, fluid intake and outputs, and diagnostic coding from patients admitted between 2003 and 2016. Participating hospitals trained clinicians to use the Philips platform, using primary data entry and drop-down boxes to adjudicate patient information and diagnoses, with direct synchronization with laboratory and clinical data. The public accessible version of the eICU-RI contains 139,367 unique first critical illness hospitalizations during 2015-16 from 208 hospitals. Of 122,835 full code ICU admissions, 80,178 had documented assessment of patient orientation on admission to comprise the primary cohort. Version 2.0 of the database is publicly available at <https://eicu-crd.mit.edu/>.

XII.2.2. Exposure

Patient orientation was assessed by clinical staff and documented in the “Best verbal response” of the admission Glasgow Coma Score. We defined disorientation as “confused”, having “inappropriate words” or “incomprehensible sounds”, but with a normal assessment of the GCS motor “obey commands” and eye opening “spontaneously” responses.

XII.2.3. Outcome

De-escalation of code status was defined by an order to change code status to one of the following: Do not resuscitate (n=5983), No Cardioversion (n=225), No Cardiopulmonary Resuscitation (n=576), No Cardioversion (n=225), No Intubation (n=468), No Vasopressors/inotropes (n=60), Comfort Measures Only (n=104), and No Augmentation of Care (n=104), and No Blood Draws (n=4).

XII.2.4. Analysis

Baseline characteristics were presented as percentages for categorical variables and mean and standard deviation for continuous variables according to admission orientation. Time to event was from ICU admission until code status change or censoring. Patients who died or were discharged prior to code status change were censored at that time. We used Cox proportional hazards model to estimate hazard ratios (HRs) and 95% confidence intervals (CIs) for the associations between disorientation and code status change. The models were adjusted for age, sex, ethnicity (white, African American, Hispanic, or other), unit type (categorical), admission diagnoses (categorical), and APACHE IV severity of illness. We used multiplicative interactions to determine whether the effect of disorientation on risk of code status change differed according to age, illness severity, or ethnicity.

All analyses were performed using JMP Pro 12 (SAS Institute, Cary, NC). R version 3.5.2, RStudio version 1.2.1335. Data was queried from R using the package bigrquery version 1.2.0.

XII.3. Results

Of the 10% (n=6,088) of patients were disoriented upon ICU admission, 44% were described as “disoriented”, 17% as having “inappropriate words” and 39% with “incomprehensible sounds”. Disoriented patients tended to be older with a greater overall level of illness severity. Code status was de-escalated in 15.8% (n=963) of disoriented patients, compared to 9.2% (n=6,782) of fully oriented patients ($p<0.001$). The median time to code status change was 44.1(51.9) hours, occurring earlier amongst disoriented patients [35.4 (44.2) versus 45.3 (52.8) hours, $p<0.001$] (Table 19).

Patient and hospital characteristics				
	Overall	Non-Disoriented Patients	Disoriented Patients	p
Number of Patients	80,178	74,090	6,088	
Age (mean (SD))	62.11 (17.41)	61.60 (17.36)	68.20 (16.89)	<0.001
Gender = Female (%)	36481 (45.5)	33588 (45.4)	2893 (47.5)	0.001
Ethnicity (%)				0.008
African American	9331 (11.7)	8672 (11.8)	659 (10.9)	
Caucasian	61486 (77.1)	56753 (77.0)	4733 (78.2)	
Hispanic	3148 (3.9)	2885 (3.9)	263 (4.3)	
Other/Unknown	5777 (7.2)	5381 (7.3)	396 (6.5)	
Admission diagnosis Group (%)				<0.001
Acute coronary syndrome	5311 (6.6)	5172 (7.0)	139 (2.3)	
Acute renal failure	1029 (1.3)	875 (1.2)	154 (2.5)	
Asthma-emphysema	1954 (2.4)	1844 (2.5)	110 (1.8)	
Coronary artery bypass grafting	2596 (3.2)	2537 (3.4)	59 (1.0)	
Cardiac arrest	5354 (6.7)	5129 (6.9)	225 (3.7)	
Chest pain	509 (0.6)	495 (0.7)	14 (0.2)	
Congestive heart failure	2837 (3.5)	2641 (3.6)	196 (3.2)	
Coma	1256 (1.6)	1067 (1.4)	189 (3.1)	
Cerebrovascular accident	6444 (8.0)	5635 (7.6)	809 (13.3)	
Other cardiovascular disease	2126 (2.7)	2033 (2.7)	93 (1.5)	
Diabetic ketoacidosis	2184 (2.7)	2067 (2.8)	117 (1.9)	
Gastrointestinal bleed	4179 (5.2)	3925 (5.3)	254 (4.2)	
Gastrointestinal obstruction	685 (0.9)	643 (0.9)	42 (0.7)	
Neurological	2871 (3.6)	2564 (3.5)	307 (5.0)	
Other	15276 (19.1)	14115 (19.1)	1161 (19.1)	
Overdose	2881 (3.6)	2665 (3.6)	216 (3.5)	
Pneumonia	2419 (3.0)	2211 (3.0)	208 (3.4)	
Other respiratory diseases	4611 (5.8)	4323 (5.8)	288 (4.7)	
Sepsis	10047 (12.5)	8929 (12.1)	1118 (18.4)	
Trauma	4204 (5.2)	3857 (5.2)	347 (5.7)	
Valve disease	1405 (1.8)	1363 (1.8)	42 (0.7)	
Apache illness severity score (mean (SD))	55.50 (26.72)	55.22 (27.04)	58.90 (22.22)	<0.001

Table 19. Patient and hospital characteristics.

Percentages for categorical variables and mean (standard deviation) for continuous variables provided. Abbreviations: SD-Standard Deviation.

In models that included measurement of overall illness severity, disorientation was associated with a 1.34 (95%CI 1.25-1.44) higher risk of de-escalation of care. This observation was true across strata of admission illness severity. We provide open access to all our data-extraction, filtering, data-wrangling, modelling, figures and tables code and queries on: https://github.com/theonesp/adm_disorientation_risk_code_stat/.

XII.4. Discussion

Patients who are disoriented upon admission to the ICU are more likely to have a de-escalation in goals of care than oriented patients. This was observed in models that adjusted for overall illness severity, as well as across strata of illness severity. Our findings raise questions about whether disorientation might affect the stakeholders involved in shared decision making⁸, including the patient him/herself, the patient's family, and the health care team. Furthermore, whether disoriented patients can meaningfully comprehend and participate in goals of care discussion is controversial, particularly with the increasing realization that ICU confusion is not transient, but instead associated with chronic and long term sequele (Girard et al., 2010; Pandharipande et al., 2013; Rengel et al., 2019). Additionally, caring for confused patients is stressful for both caregivers and nursing staff (Breitbart et al., 2002; Partridge et al., 2012), and accordingly might influence decisions to limit care in order to mitigate perceived suffering (Martins et al., 2017, p. 201). Furthermore, given the poor prognostication associated with episodes of confusion, its presence plausibly influences decisions about further withdrawal of care.

Our analysis has important limitations. We could not characterize the underlying cause of disorientation, and while our adjusted models accounted for admission diagnosis and illness severity, underlying metabolic, hepatic, neurologic, and psychiatric disease states might explain the de-escalation of care. In addition, we lacked important information about psychoactive medications, such as pain medications and sedatives, which are sources of confounding, but would similarly be expected to affect other components of the GCS score. Furthermore, while we did not have access to more comprehensive assessment of delirium, the use of simple bedside assessment of patient orientation has widespread clinical benefits in ease of use and applicability.

XIII. ALTERED CONSCIOUSNESS MEDIATES THE EFFECT OF HYPERNATREMIA BUT NOT HYPONATREMIA, ON MORTALITY

XIII.1. Introduction and problem statement

Water conservation is tightly regulated by a cascade of neurohormonal and renal mechanisms. Disorders of water imbalance, characterized clinically as dysnatremias, result in alterations of cell size and function (Danziger & Zeidel, 2014). Both hyponatremia and hypernatremia, presumably through cerebral swelling and shrinkage, respectively, have been linked to a range of neurologic sequelae, including cognitive dysfunction (Nowak et al., 2018; Verbalis et al., 2016), physical impairment (Gunathilake et al., 2013) and dementia (Chung et al., 2017), and are generally thought the pathologic explanation behind dysnatremias and increased mortality (Arampatzis et al., 2012; Funk et al., 2009; Rosner & Ronco, 2010; Wald, 2010, p. 20). Accordingly, restoration of water balance is one of the cardinal therapies in treating dysnatremia (Corona et al., 2015).

Increasingly however, particularly with the failure of aquaretic drugs to provide meaningful benefit in hyponatremia, whether the water imbalance itself, or rather the underlying disease that leads to the water imbalance, more plausibly accounts for the association of dysnatremia with increased mortality is unclear (Chawla et al., 2011; Hoorn & Zietse, 2011, 2013). Given that water retaining pathophysiologic mechanisms, such as sensed volume mediated stimulation of vasopressin, are complex and often unmeasurable, unaccounted confounding due to illness severity might better explain the association with poor outcomes.

To better understand the relationship between dysnatremia and mortality, we tested how much of the association between dysnatremia and increased mortality was mediated via decreased consciousness. Hypothesizing that changes in neuronal size and function in the setting of dysnatremia would manifest as an abnormal Glasgow Coma Scores (GCS)(15), a widely used assessment of critical illness consciousness, we examined whether decreased consciousness was on the pathogenic pathway between dysnatremia and mortality.

XIII.2. Methods

XIII.2.1. Data Source

Phillips Healthcare, a major vendor of ICU equipment and services, provides a telehealth ICU platform to over 300 hospitals across the United States. Data from participating hospitals is anonymously curated in the electronic Intensive Care Unit Research Institute Database (eICU-RI), a collaborative partnership between Philips Healthcare and the Laboratory of Computational Physiology at Massachusetts Institute of Technology (Alistair E. W. Johnson et al., 2017; Komorowski et al., 2018; McShea et al., 2010; Pollard et al., 2018). It contains high-resolution patient data including demographics, vital signs, laboratory tests, illness severity scores, fluid intake and outputs, and diagnostic coding from patients admitted between 2003 and 2016. Participating hospitals trained clinicians to use the Philips platform, using primary data entry and drop-down boxes to adjudicate patient information and diagnoses, with direct synchronization with laboratory and clinical data. Currently, the publicly accessible version of the eICU-RI contains data from 131,495 unique first critical illness hospitalizations. Of 102,489 with documented GCS

scores within 24 hours of admission, 5,899 were missing basic demographic or laboratory values, leaving a cohort of 96,590 for primary analysis.

XIII.2.2. Exposures

The primary exposures was the ICU admission sodium concentration, examined continuously and in 5 mEq/L increments. The lowest Glasgow Coma Severity score obtained within 24 hours of admission was used to describe level of consciousness.

XIII.2.3. Outcome

Critical illness hospital mortality as it was recorded on the eicu database uploaded to BigQuery on the Physionet project on 4/2020.

XIII.2.4. Variables

Basic demographics included age, gender, and ethnicity. Ethnicity was self-reported as white, African American, Hispanic, Asian, Native American, Other, or Unknown. Admission diagnoses were adjudicated by trained clinicians within the first 24 hours of ICU admission as part of the Apache scoring system (Harris et al., 2017), and were categorized into the ten most common clinical categories, including sepsis, myocardial infarction/angina, trauma, gastrointestinal bleed, arrhythmia, drug/alcohol complications, cerebrovascular accident, coronary artery bypass grafting, pneumonia, malignancy related, congestive heart failure, unknown and other. A history of metastatic disease, AIDS, liver disease, stroke, renal disease, diabetes, cancer, leukemia, lymphoma, myocardial infarction, congestive heart failure, peripheral vascular disease, transient ischemic attack, dementia, chronic obstructive pulmonary disease, connective tissue disease, and peptic ulcer disease, as adjudicated in the Charlson comorbidity scoring system were used to describe preexisting illness burden, and included as separate variables (Charlson et al., 1987). The source of admission (emergency department, floor, other hospital, direct admission, recovery unit/operating room and other), and unit type (medical, medical surgical, surgical, cardiac, cardiothoracic, and neurological) were included as categorical variables. Mechanical ventilation during the first 24 hours of care was included as binary variable.

XIII.2.5. Statistics

Baseline characteristics stratified by the presence of either hyponatremia (<135 mEq/L) or hypernatremia (>145mEq/L) are presented. Multivariable analysis and logistic regression were used to describe the adjusted association between serum sodium category and admission GCS score and risk of decreased consciousness (GCS score <15), respectively, using all variables described above. Sodium was categorized into 5 mEq/L increments, with ≥ 135 to 140 mEq/L as reference, as well as analyzed continuously by stratifying at 140 mEq/L. We describe the adjusted association between sodium concentration and mortality, and how the addition of the admission GCS score changed the association. We graphically represent the mediating effect of the GCS score by calculating the percentage change in mortality risk due to GCS inclusion in adjusted analysis.

XIII.3. Results

XIII.3.1. Baseline characteristics

Amongst 96,950 critically ill patients in ICUS across the United States, 21% (n=20,277) and 4% (3,813) were hyponatremic and hypernatremic respectively. There were differences in baseline characteristics accordingly, with more heart failure and admissions amongst hyponatremic patients, while hypernatremic patients tended. Both hypo and hypernatremia were associated with altered consciousness (Table 20), where a one mEq/L higher sodium was associated with a 2% (HR 0.98, 95%CI 0.98-1.12, p<0.001) lower and 15% (HR 1.15, 95%CI 1.14-1.16, p<0.001) higher risk of decreased consciousness, respectively. The mortality rates were significantly higher in hypernatremia than hyponatremia (12% (n=2,391) vs 20% (n=762), p<0.001, respectively), particularly when accompanied with decreased consciousness (Figure 15). In adjusted analysis, both hypo and hypernatremia severity were closely associated with increased mortality (Table 20), where a one mEq/L higher sodium was associated with a 4% (HR 0.96, 95% 0.96-0.97, p<0.001) lower and 6% (HR 1.06, 95% CI1.05-1.07, p<0.001) higher risk of mortality, respectively. However, while decreased consciousness seemed to mediate the affect of hypernatremia on mortality, it had no effect on mortality in hyponatremic patients (Table 20, Figure 15).

Adjusted association between sodium concentration, consciousness, and mortality amongst the critically ill								
	<120	120 - <125	125 - <130	130 - <135	135 - <140	140 - <145	145 - <150	≥150
GCS ¹	-0.15 -0.24,- 0.07 p<0.001	-0.10 -0.10,- 0.02 p=0.01	-0.10 -0.14,- 0.05 p<0.001	-0.04 -0.06,- 0.01 p=0.009	Ref.	-0.10 -0.12,- 0.07 p<0.001	-0.53 -0.57,- 0.48 p<0.001	-1.27 -1.35,- 1.18 p<0.001
Risk of decreased GCS ¹	1.49 1.29,1.71 p<0.001	1.24 1.09-1.42 p<0.001	1.13 1.05-1.22 p=0.002	1.06 1.02-1.11 p=0.006	Ref.	1.21 1.17-1.25 p<0.001	2.43 2.25-2.64 p<0.001	11.09 8.65-14.2 p<0.001
Mortality risk ¹	1.30 1.01-1.67 P=0.04	1.95 1.61-2.35 p<0.001	1.81 1.62-2.01 p<0.001	1.36 1.27-1.46 p<0.001	Ref.	0.99 0.93-1.05 p=0.81	1.70 1.54-1.88 p<0.001	2.14 1.83-2.51 p<0.001
Mortality risk adjusting for GCS ²	1.23 0.94-1.59 P=0.12	1.98 1.63-2.40 P<0.001	1.82 1.63-2.04 P<0.001	1.37 1.28-1.47 P<0.001	Ref.	0.94 0.88-0.99 p=0.04	1.36 1.23-1.50 P<0.001	1.30 1/10-1.52 P<0.001

Table 20. Adjusted association between sodium concentration, consciousness, and mortality in the critically ill.

Adjusted changed in GCS score(95%CI), and adjusted risk(95%CI), provided. Adjusted for age, gender, ethnicity, ICU type, comorbidities, admission diagnosis, and mechanical ventilation¹ and GCS score². Sodium category of 135 to 130 mEq/L reference for all analyses. Abbreviations: GCS- Glasgow Coma Score.

Figure 15. Potential mechanistic linking hyponatremia with increased mortality.

(A) The water excess of hyponatremia, and deficiency of hypernatremia, cause death by changes in neuronal size and function. (B) Underlying illness simultaneously causes water retention and mortality. There is no direct causal link between dysnatremia and mortality.

Figure 16. Effect of consciousness on mortality rates in dysnatremia.

Proportions of hospital mortality according to admission serum sodium and decreased consciousness, as defined by a decreased Glasgow Coma Score.

Figure 17. Mediating effect of decreased consciousness on the association between sodium concentrations and mortality.

Adjusted for age, gender, ethnicity, ICU type, comorbidities, admission diagnosis, and mechanical ventilation, with and without GCS score (continuous). Shaded figure represents the percent of the mortality risk mediated by decreased consciousness. <120 mEq/L data not visualized.

XIII.4. Discussion

Our analysis further highlights the complex relationship between dysnatremia and outcomes. While hyponatremia associates with mortality in part through decreased consciousness, presumably due to neuronal dysfunction in the setting of cell shrinkage, hyponatremia's association with mortality occurs independently of decreased consciousness.

The absence of a mediating effect of decreased consciousness on the association between hyponatremia and increased mortality raises important questions about causality and for the potential for confounding. The myriad mechanisms that orchestrate water retention, including osmolar independent activation of vasopressin, as might occur in heart failure, cirrhosis, or in the syndrome of inappropriate ADH release, are not readily measurable, and instead rely on the clinical assessment of a patient's sensed volume status. Given that determining the "sensed" fluid volume remains one of the most challenging aspects in medicine (Danziger et al., 2012; Danziger & Hoenig, 2016), confounding due to underlying illness severity is likely. In other words, since heart failure is a strong determinant of hyponatremia and mortality, and since heart failure's

effect on vasopressin release is not easily quantified, hyponatremia associated mortality might simply reflect heart failure severity rather than an effect of water excess per se.

In addition, the lower overall mortality and risk of decreased consciousness observed in hyponatremia, when compared to hypernatremia, might reflect more robust cell volume regulatory mechanism in states of water excess, whereby cells active extrude osmotically active particles to drive compensatory water egress, thereby restoring cell size. Per one mEq/L sodium change was associated with a 2% versus 15% higher risk of decreased consciousness in hypo versus hypernatremia, respectively, potentially reflecting greater ability to protect cell size in states of water excess than deficiency. In addition to potential differences in cell volume regulation, the brain structure might be particularly susceptible to hypernatremia, as the shrinking neurons pull away from the surrounding vascular network, inducing cerebral hemorrhage (Arora, 2011).

While symptomatic hyponatremia remains a medical emergency, requiring careful and judicious restoration of water homeostasis, whether asymptomatic hyponatremia can be more conservatively managed is unclear (Sterns, 2015, 2018). In our cohort, more than % of the hyponatremic patients had a normal GCS score. While we could not delineate the other factors warranting ICU admission, whether asymptomatic hyponatremia necessarily warrants high acuity of care warrants discussion.

Our analysis has several important limitations. While the accuracy of GCS scoring as a measure of neuronal size and function is unknown, its mediating effect in hypernatremia supports its biologic plausibility. In addition, the underlying pathophysiology accounting for hyponatremia was not known, and more granular datasets enriched with measures of urine concentration and urinary sodium handling would be helpful. In addition, while we lacked follow up sodium concentrations, since water replacement is generally simpler and faster than aquaresis, our findings further highlight the risk of hypernatremia, particularly when associated with decreased consciousness.

XIV. ASSOCIATION OF INTRAOPERATIVE HYPOTENSION WITH POSTOPERATIVE STROKE, ACUTE KIDNEY INJURY, AND DEATH IN PATIENTS HAVING CARDIAC SURGERY WITH CARDIOPULMONARY BYPASS

XIV.1. Introduction and problem statement

Postoperative major adverse events (MAE) frequently occurs after cardiac surgery, especially with an increasing number of older and complex patients presenting for cardiac surgery (W.-J. Gu et al., 2018; Ngaage et al., 2008). In patients having noncardiac surgery, intraoperative hypotension (IOH) is associated with postoperative acute kidney injury (AKI), (Salmasi et al., 2017; van Waes et al., 2016; Walsh et al., 2013) myocardial injury (Salmasi et al., 2017; Sun et al., 2015; Walsh et al., 2013), stroke, (Bijker et al., 2012) delirium, (Hirsch et al., 2015) and death (Bijker et al., 2009; Mascha et al., 2015; Monk et al., 2015). IOH may also be a modifiable risk factor for MAE in patients having cardiac surgery (Fischer et al., 2002; Kanji et al., 2010; Monk et al., 2015; Ono et al., 2014).

While a mean arterial pressure (MAP) of 65 mmHg has been suggested as a population harm threshold in noncardiac surgery patients (Sessler et al., 2019), there is no clear consensus regarding an optimal arterial pressure (AP) intervention threshold during cardiac surgery with cardiopulmonary bypass (CPB) (Murphy et al., 2009). Existing fixed absolute AP values used as lower intervention thresholds during cardiac surgery were chosen based on the principle that cerebral blood flow (CBF) autoregulation remains functional during CPB (Hogue et al., 2006; Murkin et al., 1987). However, current evidence suggests that lower limits of autoregulation can vary from 40-160 mmHg (Joshi et al., 2012). Therefore, IOH may be an important modifiable risk factor for MAE in patients having cardiac surgery. Recently, IOH (MAP <65 mmHg) even for only >10 minutes during CPB has been shown to increase the risk of stroke (Sun et al., 2018). Vasopressors and inotropes are used to treat IOH during cardiac surgery. The association of IOH and intraoperative vasopressor/inotrope use with postoperative outcomes remains elusive. We thus studied the relationship between a) IOH (defined as a MAP <65 mmHg) and b) the fraction of total IOH occurring during CPB (fraction IOH-CPB) with a composite of three MAEs (stroke, acute kidney injury, and death) accounting for intraoperative vasopressor and inotrope dose in patients having cardiac surgery with cardiopulmonary bypass (CPB).

XIV.2. Methods

XIV.2.1. Study design and participants

In this retrospective observational study, we analyzed data that were prospectively collected in patients having cardiac surgery from institutional electronic medical records, Society of Thoracic Surgery (STS), and Anesthesia Information Systems (AIMS) after institutional review board (IRB, Beth Israel Deaconess Medical Center, Boston, MA, US. Protocol #2020P000074) approval. The IRB waived informed consent, and the manuscript adheres to the applicable Strengthening the Reporting of Observational Studies in Epidemiology (STROBE) standards for observational studies (von Elm et al., 2007). Adult patients (>18 years age) who had cardiac surgery with CPB between January 2008 and June 2016 were included. Patients who underwent emergency cardiac surgery or those with invalid, incomplete, or unavailable data were excluded (Figure 18).

Figure 18. Flow chart presenting patient selection and analysis.

XIV.2.2. Anesthesia management

The perioperative details of anesthesia management have been described previously (Jinadasa et al., 2018). In brief, preoperative medications were continued until the time of the surgery unless contraindicated by the patient's condition. All patients received preoperative carotid scan. Anyone with 80% or more carotid stenosis on either side or symptoms suggestive of carotid stenosis were consulted with vascular surgery for appropriate management before surgery. All patients underwent general anesthesia with endotracheal intubation. Standard intravenous (IV) induction was performed using IV propofol and fentanyl. Rocuronium was used for muscle relaxation and isoflurane (0.5%–1%) for maintenance. Valvular surgeries were performed under cardiac arrest using a standard institutional cardioplegia solution (K 60 mEq, Mg 8 mEq, dextrose 2.5 g, Tham 10 mEq, and normal saline 500 mL). Patients were maintained under mild hypothermia, and an alpha-stat pH strategy was used for blood gas management during cardiopulmonary bypass.

XIV.2.3. Hemodynamic monitoring

All patients received transoesophageal echocardiography (TEE) monitoring throughout the surgery unless contraindicated. The rate, rhythm, preload, afterload, and contractility were maintained with information obtained from arterial catheters, pulmonary artery or central venous catheters, and TEE. All cardiac anesthesiologists involved in patient care were TEE board certified. Preoperatively, intravenous phenylephrine or norepinephrine infusions were started to maintain a systolic AP (SAP) of 90 to 120 mmHg.

XIV.2.4. Vasopressor-inotrope treatment

Following CPB, in patients with low to moderately dysfunctional ventricular function (preoperative LVEF <30% or with significant mitral or aortic regurgitation), IV epinephrine was started to maintain a cardiac index >2L/min/m². Intravenous milrinone was added to this regimen to achieve the same goals at the clinician's discretion. Other medications, such as dopamine and dobutamine, were rarely added to this strategy. Afterload was maintained with phenylephrine, norepinephrine, or vasopressin in combination or by themselves along with the inotropes, as mentioned earlier. The management was guided by direct visualization of the ventricles, echocardiographic assessment of cardiac function with or without pulmonary artery catheter measurements of mixed venous oxygen saturation, and cardiac index.

Vasopressor and inotrope dose was calculated by adding norepinephrine equivalents of total norepinephrine, epinephrine, phenylephrine, milrinone, and vasopressin dose used during surgery; using the formula, total vasopressor-inotrope dose = [norepinephrine] + [epinephrine] + [phenylephrine ÷ 10] + [milrinone ÷ 5] + [vasopressin × 8.33] (Mancl & Muzevich, 2012, p. 201; Russell et al., 2008).

XIV.2.5. BP threshold for IOH

Invasive AP was recorded every 15 seconds, and artifacts were removed using the rules described elsewhere (Maheshwari et al., 2018; Salmasi et al., 2017). IOH was defined as MAP <65 mmHg in accordance with the noncardiac surgical literature (Sessler et al., 2019). IOH was characterized by a) IOH duration as cumulative minutes and b) area under a MAP of 65 mmHg (AUC-MAP <65 mmHg) measured based on the trapezoidal rule (Epstein et al., 2017; Monk et al., 2015). Using this method, values below the threshold were subtracted from the threshold, multiplied by 0.25 (which corresponds to 15 seconds), and summed together. For example, each MAP value 'x' below 65 mmHg will be subtracted from 65, multiplied by 0.25 minutes, and all such values were summed together. $AUC = \sum [(65-x) * 0.25] \text{ mmHg} * \text{min}$.

XIV.2.6. IOH during CPB

The duration of IOH was divided into three categories based on how much of the total IOH occurred during CPB (fraction IOH-CPB): fraction IOH-CPB >80%, fraction IOH-CPB 80-60%, fraction IOH-CPB <60%.

XIV.2.7. Study Outcomes

The primary outcome was a composite of three MAE, namely stroke, acute kidney injury, and death based on STS definition, version 2.81 (Appendix VI).

XIV.2.8. Statistical Analysis

Descriptive statistics were computed as mean values with standard deviations (SD) for continuous variables and frequencies with percentages for categorical variables. Multivariate logistic regression was used to assess the association between IOH variables and the composite primary endpoint and between vasopressor-inotrope dose and the primary outcome. Confounders were predefined based on clinical plausibility, which included age in years, gender, surgery category, STS risk scores in tertiles, left ventricular ejection fraction, delta hematocrit (HCT) percentage, and aortic cross-clamp time (XCT). The same confounders were used for both IOH variables (total duration and AUC-MAP <65mmHg).

The linearity assumption of the effects of both IOH and vasopressor-inotrope dose on the log odds of the outcome variable was assessed using restricted cubic splines with three knots and generalized additive models. In the case where significant non-linear terms were detected, the corresponding variables were categorized using quartiles. The effect modification of vasopressor-inotrope dose on the association between IOH and MAE was evaluated by including an interaction term of vasopressor-inotrope dose and IOH in the logistic regression models. In the case of nonsignificant interaction, the final model only with main effects was reported. Hosmer-Lemeshow goodness-of-fit test and area under the receiver-operating characteristic (AUROC) curve were used to assess the model fit. Statistical analyses were performed in R version 3.5.1 and RStudio version 1.1.463. All tests were two-sided, with 0.05 as the level of significance.

No specific power calculation was performed for this retrospective observational cohort study. In this cohort with more than 300 MAE, it was assumed that a minimum of 30 variables could be inserted into the logistic regression analysis.

XIV.3. Results

XIV.3.1. Baseline characteristics

Figure 18 & Table 21 display an overview of patient selection and analysis. Data describing 6024 patients who had cardiac surgeries from 2008 to 2016 were collected. We excluded 269 patients who had emergent surgeries. From the remaining 5755 patients, 597 patients were excluded due to inadequate data, and the final analysis included 5158 patients.

Table 21 summarizes the baseline characteristics of patients stratified by postoperative MAEs. The mean (SD) age of our study population was 66.8 (11.8) years. There were 3594 (70%) male patients, 1183 (23%) patients had diabetes, 4026 (78%) patients had chronic arterial hypertension, and 3880 (75%) patients had dyslipidemia. Previous myocardial infarction was seen in 1578 (31%) patients, congestive heart failure in 1766 (34%) patients, and 85 (1.6%) patients were on dialysis before surgery.

Demographics	All patients N = 5158	MAE		P value
		Absent N = 4884 (94.7)	Present N = 274 (5.3)	
Age (years), mean (SD)	66.87 (11.80)	66.69 (11.76)	70.08 (12.13)	<0.001
Male	3594 (69.7)	3421 (70.0)	173 (63.1)	0.019
Diabetes	1183 (22.9)	1105 (22.6)	78 (28.5)	0.030
Dyslipidaemia	3880 (75.2)	3673 (75.2)	207 (75.5)	0.955
Hypertension	4026 (78.1)	3793 (77.7)	233 (85.0)	0.005
Smoking	1489 (28.9)	1405 (28.8)	84 (30.7)	0.546
Congestive Heart Failure	1766 (34.2)	1610 (33.0)	156 (56.9)	<0.001
Previous MI	1578 (30.6)	1469 (30.1)	109 (39.8)	0.001
Chronic Lung Disease	623 (12.1)	571 (11.7)	52 (19.0)	<0.001
Dialysis	85 (1.6)	79 (1.6)	6 (2.2)	0.631
Preoperative medications				
Beta blockers	3840 (74.4)	3640 (74.5)	200 (73.0)	0.620
ACEI or ARB	2249 (43.6)	2130 (43.6)	119 (43.4)	1.000

Inotropes	47 (0.9)	34 (0.7)	13 (4.7)	<0.001
Steroids	175 (3.4)	155 (3.2)	20 (7.3)	<0.001
Aspirin	4237 (82.1)	4023 (82.4)	214 (78.1)	0.086
Statins	3959 (76.8)	3754 (76.9)	205 (74.8)	0.480
STS risk score, mean (SD)	0.02 (0.03)	0.02 (0.03)	0.04 (0.06)	<0.001
Category	<0.001			
CABG	2550 (49.4)	2466 (50.5)	84 (30.7)	
Valve	1488 (28.8)	1408 (28.8)	80 (29.2)	
CABG + Valve	946 (18.3)	854 (17.5)	92 (33.6)	
Aortic Surgery	17 (0.3)	13 (0.3)	4 (1.5)	
Other	157 (3.0)	143 (2.9)	14 (5.1)	
Duration of phase (minutes), mean (SD)				
Cardiopulmonary Bypass	94.63 (38.70)	93.00 (36.29)	123.76 (62.01)	<0.001
Aortic cross clamp	73.16 (29.99)	72.30 (28.93)	88.93 (42.41)	<0.001

Table 21. Demographics and Clinical Characteristics stratified by Major Adverse Events (MAE)*

Data are presented as numbers (percentages) unless otherwise indicated.

IQR, interquartile range Major Adverse Events (MAE)*, composite of renal failure, stroke and death MI, myocardial infarction; ACE-I, angiotensin converting enzyme inhibitors; ARB, angiotensin receptor blockers; STS, Society of Thoracic Surgeons; CABG, Coronary Artery Bypass Grafting; AUC, Area Under the Curve

XIV.3.2. Postoperative major adverse events (MAE)

Postoperative MAE are displayed in Supplemental Table 1. The composite primary outcome occurred in 274 (5.3%). One hundred thirty-four (2.6%) patients had acute kidney injury, 69 (1.3%) patients had stroke, and 117 (2.3%) patients died after surgery. Baseline characteristics of patients stratified by composite primary outcome are presented in Table 21. Patients having one of the MAEs of the composite primary endpoint were significantly older [70 (12) vs 67 (12) years, $P < .001$], more likely to be male (65% vs 70%, $P = .03$), and have diabetes (28% vs 23%, $P = .03$) (Table 21. Demographics and Clinical Characteristics stratified by Major Adverse Events (MAE)*Table 21). They also had higher incidence of congestive heart failure (57% vs 33%, $P < .001$), previous myocardial injury (MI) (40% vs 30%, $P < .001$), chronic lung disease (19% vs 12%,

P < .001), need for inotropic support (4.7% vs 0.7%, P < .001) and steroid medications (7.3% vs 3.2%, P < .001) as compared to those without MAE. STS risk scores were significantly higher in patients with MAE compared to those without MAE [mean (SD); 0.04 (0.06) vs 0.02 (0.03); P < .001].

In addition, patients with MAE had a significantly increased mean (SD) CPB time [124 (62) vs 93 (36) minutes; P <0.001], cross clamp time [89 (42) vs 72 (29) minutes; P <0.001], duration MAP <65mmHg [142 (75) vs 104 (48) minutes; P <0.001], and AUC-MAP <65mmHg [1530 (1144) vs 1073 (663) mmHg.min; P <0.001] compared to those without MAE. Similarly, median (IQR) vasopressor-inotrope dose were significantly higher in patients with MAE compared to those without MAE [1.08 (0.54, 2.87) vs 0.66 (0.37, 1.13) mcg; P <0.001] (Table 21).

XIV.3.3. IOH duration, AUC-MAP < 65mmHg and vasopressor-inotrope dose

Supplemental Table 2 presents the values of IOH duration, AUC-MAP < 65mmHg, and vasopressor-inotrope dose when categorized as quartiles. In quartile 4 (Q4), the duration of IOH was > 128 minutes, AUC-MAP > 1339 mmHg•min, and vasopressor-inotrope dose > 1.17 mg.

Figure 19. Association between AUC-MAP < 65 mmHg, vasopressor-inotrope dose, and Composite Outcome for Renal Failure, Death and Stroke.

XIV.3.4. IOH and vasopressor-inotrope dose association with MAEs

For both IOH and vasopressor-inotrope dose, significant nonlinearity was detected. Total IOH duration, AUC-MAP <65mmHg, and vasopressor-inotrope dose were categorized into quartiles. The results of multivariable logistic regression analysis showing the association between exposure measures and the primary composite endpoint are displayed in Table 22 & Table 23. Only the 4th quartile for total IOH duration and AUC-MAP <65mmHg were significantly associated with MAE [Odds ratio (OR) 1.65; 95% confidence interval (CI) 1.35, 3.28) and OR 1.65; 95% CI

(1.11, 2.49)] (Table 22). Similarly, only the 4th quartile of vasopressor-inotrope dose showed significant association with MAE [OR 1.71; 95% CI (1.12, 2.66)]

Variables	OR	95% CI^a	
Age	1.0018	0.9896	1.0146
Female	1.1288	0.8529	1.4884
Category			
Aortic Surgery	1.8734	0.4408	6.569
CABG + Valve	1.0024	0.6885	1.4605
Other	1.051	0.5043	2.0563
Valve	1.0634	0.7488	1.5078
Delta HCT	0.9824	0.9646	0.9994
Ejection fraction	1.0031	0.9938	1.0124
Duration of surgery	1.0048	1.0028	1.0069
Aortic cross clamp time	0.9967	0.9915	1.0019
STS Risk ^b			
T2	1.5958	0.9899	2.6163
T3	4.7434	2.9302	7.8482
Duration MAP<65mmHg			
Q2	1.1623	0.7227	1.8838
Q3	1.4978	0.9627	2.3694
Q4	2.0853	1.3521	3.2847
Vasopressor dose ^d			
Q2	1.4114	0.9114	2.2122
Q3	1.2387	0.803	1.9369
Q4	1.6506	1.0822	2.5647

Table 22. Multivariable logistic regression model evaluating the association between intraoperative hypotension duration MAP <65mmHg, vasopressor-inotrope dose and Major Adverse Events (MAE)*

Abbreviations: OR, Odds Ratio; CI, Confidence Intervals; CABG- Coronary Artery Bypass Grafting; HCT, Haematocrit; STS, Society of Thoracic Surgeons; T, Tertile; AUC, Area Under the Curve; MAP, Mean arterial Pressure; Q, Quartile

^aAdjusted for age, gender, type of surgery, STS risk score, left ventricular ejection fraction, delta Haematocrit (HCT) percentage.

^bSTS risk algorithm score categorized into tertiles: < 0.01, 0.01-0.02 and 0.02-0.52.

^dvasopressor dose categorized into quartiles (mg): < 0.38, 0.38 - 0.68, 0.68-1.17 and 1.17-115875.89

Variables	OR	95% CI ^a	
Age	1.0023	0.9901	1.0152
Female	1.1147	0.8422	1.4698
Category			
Aortic Surgery	1.5508	0.3696	5.3997
CABG + Valve	1.0206	0.701	1.487
Other	0.9739	0.4659	1.9093
Valve	1.0485	0.7384	1.4867
Delta HCT	0.9829	0.9652	0.9998
Ejection fraction	1.0033	0.9941	1.0126
Duration of surgery	1.0051	1.003	1.0071
Aortic cross clamp time	0.9976	0.9925	1.0027
STS Risk ^b			
T2	1.6369	1.0153	2.6838
T3	4.9585	3.0626	8.2046
AUC-MAP<65mmHg			
Q2	1.2352	0.8016	1.9159
Q3	1.1585	0.7565	1.7892
Q4	1.6531	1.1108	2.4996
Vasopressor dose ^d			
Q2	1.4251	0.9212	2.2313
Q3	1.2744	0.8268	1.9911
Q4	1.7157	1.1242	2.6672

Table 23. Multivariable logistic regression model evaluating the association between intraoperative hypotension AUC-MAP <65mmHg, vasopressor-inotrope dose and Major Adverse Events (MAE)*

Major Adverse Events (MAE)*, composite of renal failure, stroke and death

Abbreviations: OR, Odds Ratio; CI, Confidence Intervals; CABG- Coronary Artery Bypass Grafting; HCT, Haematocrit; STS, Society of Thoracic Surgeons; T, Tertile; AUC, Area Under the Curve; MAP, Mean arterial Pressure; Q, Quartile

^aAdjusted for age, gender, type of surgery, STS risk score, left ventricular ejection fraction, delta Haematocrit (HCT) percentage.

^bSTS risk algorithm score categorized into tertiles: < 0.01, 0.01-0.02 and 0.02-0.52.

^dvasopressor dose categorized into quartiles (mg): < 0.38, 0.38 - 0.68, 0.68-1.17 and 1.17-115875.89

XIV.3.5. Fraction of IOH during CPB and its association with MAEs

The association between varying fraction IOH-CPB and the composite primary outcome, as well as individual MAEs, were presented in Table 24 and Figure 20. When compared to fraction IOH-CPB > 80%, fraction IOH-CPB 80-60% and < 60% were associated with an increased risk for composite outcome [OR 1.4; 95% CI (0.94, 2.12) and OR 1.6; 95% CI (1.07, 2.45)].

IOH during CPB	OR	95% CI ^a	
<i>MAE (Composite outcome)^b</i>			
80% to 60%	1.4003	0.9458	2.1220
< 60%	1.6092	1.0771	2.4587
<i>Renal failure</i>			
80% to 60%	3.2191	1.5955	7.4285
< 60%	3.9801	1.9616	9.2233
<i>Death</i>			
80% to 60%	2.1646	1.0883	4.8106
< 60%	2.8168	1.4136	6.2748
<i>Stroke</i>			
80% to 60%	0.5493	0.3014	1.0156
< 60%	0.4969	0.2601	0.9559

Table 24. Model evaluating the association between intraoperative hypotension during CPB and outcomes (composite, renal failure, stroke and death).

Abbreviations: OR, Odds Ratio; CI, Confidence Intervals

^a Adjusted for age, gender, type of surgery, STS risk score, left ventricular ejection fraction, delta Haematocrit (HCT) percentage, duration of surgery, aortic cross clamp time, vasopressor-inotrope dose.

^b composite outcome comprises of renal failure , death and stroke.

Figure 20. Association between intraoperative hypotension during CPB and outcome (composite outcome, renal failure, stroke & death).

Similarly, when exploring the associations with individual components of composite outcome, fraction IOH-CPB 80-60% and < 60% were associated with an increased risk for renal failure [OR 3.2; 95% CI (1.60, 7.43) and OR 4; 95% CI (1.96, 9.23)] and death [OR 2.2; 95% CI (1.09, 4.81) and OR 2.8; 95% CI (1.41, 6.27)] when compared with fraction IOH-CPB > 80%. However, on comparing with fraction IOH-CPB > 80%, fraction IOH-CPB 80-60% and < 60% were observed to be protective for stroke [OR 0.55; 95% CI (0.30, 1.01) and OR 0.50; 95% CI (0.26, 0.95)]

We provide open access to all our data-extraction, filtering, data-wrangling, and modeling, figures, and tables' code and queries on https://github.com/theonesp/vasopressor_dose_mae . An interactive dashboard, including the most important figures of the study, can be found on <https://plot.ly/dashboard/ma.armengol:9/present#/> . A de-identified version of vital-signs and drugs AIMS data used in this project will eventually be available on <https://physionet.org/>.

XIV.4. Discussion

We found that the 4th quartiles of total IOH duration, AUC-MAP <65mmHg and vasopressor-inotrope dose were significantly associated with the composite primary outcome in patients having cardiac surgery with CPB. Fraction IOH-CPB <60% and 60-80% when compared to fraction IOH-CPB >80% were associated with a higher risk for renal injury and death. However, in comparison with fraction IOH-CPB >80%, fraction IOH-CPB <60% and 60-80% were protective for stroke [0.55 (0.30, 1.01) and 0.50 (0.26, 0.95)].

Advanced hemodynamic monitoring and protocolized treatment strategies in the cardiac surgical setting could identify and treat hemodynamic changes earlier. This also accords with our observations, which showed that only the 4th quartiles of total IOH duration and AUC-MAP <65 mmHg were associated with the composite primary outcome. Moreover, in every quartile of the IOH duration, increasing vasopressor-inotrope doses increased the risk of the composite primary outcome. Patients in the 4th quartile of IOH duration despite receiving maximum treatment with vasopressor-inotrope dose were increasingly associated with outcome. These patients might potentially represent those with high preoperative comorbid conditions, risk, and compromised physiology. Accordingly, it is possible that in a highly monitored setting, the higher preoperative risk could prevent a positive change in outcome despite treatment based on a specific MAP threshold. We could hypothesize that, in a highly monitored environment that was not specifically targeting a MAP threshold, higher preoperative risk precludes a positive change in the outcome.

Increasing treatment (vasopressor-inotrope dose) for hypotension should decrease MAE. However, increasing treatment doses were significantly associated with the composite primary outcome. These findings are similar to several other studies. In a study, when the vasopressor dose was increased to achieve higher MAP thresholds, a decrease in the incidence of adverse outcomes were not observed. achieved with vasopressors, a decrease in the incidence of adverse outcomes was not observed as compared to moderate MAP thresholds (X. Wu et al., 2017). Moreover, an individualized AP management strategy was found beneficial compared to the interventions aimed to maintain a fixed AP threshold (Futier et al., 2017). Other studies targeting a higher MAP during CPB using vasopressors also showed increased adverse events (D. Hori et al., 2014; Jinadasa et al., 2018). In patients with a high preoperative risk with maximal monitoring and interventions, MAE might be driven by the preoperative patient and cardiac surgical risks.

In addition to duration, we characterized the severity of IOH as AUC-MAP <65mmHg. Both total IOH duration and AUC-MAP <65mmHg showed similar results. Even though these results were though consistent with previous studies (Maheshwari et al., 2018; Salmasi et al., 2017), only the 4th quartiles (or the extreme case) of total IOH duration and AUC-MAP <65mmHg were associated with the composite primary outcome. Furthermore, in absolute duration terms, while MAP <65mmHg for ≥ 13 minutes was significantly associated with MAE5 our findings were

significant only when the duration was ≥ 128 minutes. This discrepancy could be attributed to the CPB time, where patients often tend to have MAPs below 65 mmHg. Furthermore, the primary composite outcome in our study included death, renal failure, and stroke compared to the studies in noncardiac surgical settings where the myocardial and renal injury was predominant outcomes.

Weis et al (Weis et al., 2006) observed that cardiac surgical patients needing more vasopressors had a greater degree of a systemic inflammatory response that has been associated with poor outcomes (Kilger et al., 2003). Moreover, significant associations between increased vasopressor support and renal replacement therapy, length of stay, periods of ventilation after cardiac surgery were reported (Weis et al., 2006). Our results further confirm the association between vasopressor dose and MAEs. Accordingly, we can infer that patients needing a high vasopressor dose might have developed end-organ damage resulting in adverse events.

In our study, with a decrease in IOH duration during CPB, a protective effect for stroke was observed. These results mirror the findings from a previous study that demonstrated a strong association between increased duration of IOH (MAP < 65 mmHg) during CPB and postoperative stroke (Sun et al., 2018). On the other hand, we didn't observe a similar protective effect for renal failure or death. However, irrespective of varying durations, IOH during CPB significantly increased the risk of renal failure and death. These findings are contrary to a recent study that found no association with IOH (similar AP threshold) during CPB and renal failure (Rettig et al., 2017). The hypotension predominance during or outside CPB needs to be studied in larger databases to identify the phase that needs appropriate interventions.

Evidence regarding optimal AP during cardiac surgery and CPB remains controversial. The current practice of maintaining MAP as low as 50 mmHg during CPB is based on the assumption that cerebral blood flow and autoregulation remains functional during CPB (Cartwright & Mora Mangano, 1998; Joshi et al., 2012). However, cerebral autoregulation could be impaired in cardiac patients (Schoof et al., 2007) and studies using transcranial Doppler and near-infrared spectroscopy observed varying limits of autoregulation (Daijiro Hori et al., 2016, 2017; Joshi et al., 2012). Moreover, an empirically increasing MAP target resulted in adverse outcomes (D. Hori et al., 2014). These data support the role of MAP targets tailored to individuals^{35,36} rather than an absolute fixed BP threshold in cardiac surgery.

This is a retrospective study and finding a causative link between IOH or vasopressor-inotrope dose and postoperative outcomes is thus not possible. Additionally, the vasopressor-inotrope dose was calculated by adding NE equivalents of vasopressors and inotropes used. Since the molecular target for each drug differs, the possibility of an individual medication effect on outcome should not be missed. The sample size may be relatively small to evaluate the true interaction between IOH, vasopressor-inotrope dose, and outcome. However, we had more than 6000 patients in this study. Our analyses didn't include postoperative hypotension, which might have an impact on the incidence of adverse outcomes. Despite the limitations of this observational study, our results do seem to provide important insights regarding the association between IOH and vasopressor-inotrope dose and postoperative MAEs.

To conclude, in patients having cardiac surgery with CPB, the 4th quartiles of total IOH duration, AUC-MAP < 65 mmHg, and vasopressor-inotrope dose were significantly associated with a composite primary outcome of stroke, acute kidney injury, and death. Larger databases from multiple centers can offer further confirmation and specific insights into the relative contributions

of perioperative factors for postoperative MAE. Differential effects of IOH during or outside the CPB phase merits further exploration.

XV. MACHINE LEARNING MODELS WITH PREOPERATIVE RISK FACTORS AND INTRAOPERATIVE HYPOTENSION PARAMETERS PREDICT MORTALITY FOLLOWING CARDIAC SURGERY

XV.1. Introduction and problem statement

Despite all the advancements in modern medicine, perioperative mortality still remains as a leading cause of death in the United states (Bartels et al., 2013). With an increase in elderly population, more patients are presenting for cardiac surgery with complex comorbidities. The Society of Thoracic Surgeons (STS) risk score and the European System for Cardiac Operative Risk Evaluation (EuroSCORE) are the commonly used risk prediction models. However, they have certain limitations in estimating perioperative risk for major surgeries and elderly patients (Barili et al., 2012; Puskas et al., 2012; Wendt et al., 2009). Most of the prediction models, were constructed assuming that the factors interact in a linear and additive manner (Merath et al., 2019), though the interactions are often multifactorial and non-linear (Bertsimas et al., 2018). Moreover, these models are generally based on conventional statistical approaches with inherent limitations (R Shouval et al., 2013).

Machine learning, a scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, is an emerging tool for predicting outcomes due to their comprehensive nature and nonlinear properties (Carlos et al., 2018). It analyzes large datasets and facilitates model building using algorithms that iteratively learn from the data by detecting underlying patterns avoiding pre-assumptions as in conventional statistical approach (R Shouval et al., 2013). Machine learning models accurately predicted mortality after gastrointestinal surgeries (Merath et al., 2019), aortic aneurysmal repair (Wise et al., 2015), liver cancer surgery (Shi et al., 2012) and ST elevation myocardial infarction (Roni Shouval et al., 2017). However, studies exploring the utility of these models in cardiac surgical setting remain sparse.

Significant changes during intraoperative period could result in complications. Cardiopulmonary bypass (CPB) (Wang et al., 2011), intraoperative hypotension (IOH) (Wesselink et al., 2018), increasing use of vasopressors (D. Hori et al., 2014) and cross clamp time (Al-Sarraf et al., 2011) have been significantly associated with adverse outcomes. Incorporating intraoperative factors in conventional risk prediction models accurately predicted mortality (Lamarche et al., 2017) and improved preoperative risk prediction (Stoica, 2002) in cardiac surgical patients. However, these factors are rarely considered, and only preoperative factors were used to design machine learning in predicting mortality after cardiac surgery (Allyn et al., 2017). Furthermore, CPB that is associated with unique physiological perturbations could influence mortality (Jinadasa et al., 2018). This warrant studying IOH during and outside CPB to delineate CPB specific effects. We hypothesized that incorporating adverse intraoperative factors that include IOH during and outside CPB along with the preoperative risk factors into the machine learning models will improve discrimination, specificity, sensitivity and positive predictive value for mortality prediction following cardiac surgery.

In cardiac surgical patients, we aimed to explore, 1) the performance of commonly used machine learning models incorporating IOH and other intraoperative adverse factors for 30-day mortality prediction and 2) the performance of these models with IOH during and outside CPB phase for mortality prediction.

XV.2. Methodology

XV.2.1. Study design and participants

In this retrospective study from prospectively collected data, adult patients (>18 years age) undergoing cardiac surgery with CPB from January 2008 to June 2016 were included after institutional review board (IRB, Beth Israel Deaconess Medical Center, Boston, MA, US. Protocol #2008P000478) approval. Informed consent was waived by our IRB. The manuscript adheres to the applicable Strengthening the Reporting of Observational Studies in Epidemiology (STROBE) standards for observational studies (von Elm et al., 2007). Data were collected from institutional electronic medical records, Society of Thoracic Surgery (STS) database and anesthesia information management system (AIMS). Patients who underwent emergency surgeries or those with inadequate data were excluded.

XV.2.2. Perioperative management

All patients received standard perioperative care and the details have been described previously (Jinadasa et al., 2018). In brief, standard institutional practice was followed for anesthesia, monitoring, mild hypothermia and an α stat pH strategy for blood gas management during CPB. Invasive BP was recorded every 15 seconds and artefacts were removed using rules published elsewhere (Salmasi et al., 2017).

XV.2.3. Factors used as predictors in machine learning models

Table 25. Phase-specific factors used as predictors for machine learning models presents the list of factors used as predictors in the models. Machine learning models were constructed using both pre and phase-specific intraoperative factors. Preoperative factors consisted of baseline patient characteristics such as age, gender, surgery type, ejection fraction, renal failure, STS risk and hematocrit. Intraoperative factors were IOH, vasopressor-inotrope dose, phase-specific duration of surgery and cross clamp time (absent in outside-CPB phase). Intraoperative period was divided into three phases based on CPB:

- 1) Outside CPB – before and after CPB (excluding CPB).
- 2) CPB - only during CPB.
- 3) Total surgery - outside CPB + CPB for the purpose of quantifying IOH.

Outside CPB†	CPB	‡Total surgery
Gender	Gender	Gender
Age	Age	Age
Renal Failure	Renal Failure	Renal Failure
Category of surgery	Category of surgery	Category of surgery
Ejection fraction	Ejection fraction	Ejection fraction
Delta HCT	Delta HCT	Delta HCT
STS risk	STS risk	STS risk
Duration	Duration	Duration
--	Cross clamp time	Cross clamp time
AUC-MAP<65mmHg	AUC-MAP<65mmHg	AUC-MAP<65mmHg
Vasopressor-inotrope dose	Vasopressor-inotrope dose	Vasopressor-inotrope dose

Table 25. Phase-specific factors used as predictors for machine learning models

CPB, cardiopulmonary bypass; † Outside CPB: duration before and after CPB (Total surgery duration – CPB duration); ‡ Total surgery: CPB + outside CPB

HCT, hematocrit; STS, Society of thoracic surgeons; AUC, area under the curve; MAP, mean arterial pressure.

XV.2.4. IOH definition and characterization

IOH was defined as MAP < 65mmHg (Wang et al., 2011) and its severity was characterized by area under the curve (AUC-MAP < 65 mmHg) measured using trapezoidal rule (Epstein et al., 2017; Monk et al., 2015). In brief, values below the threshold were subtracted from the threshold, multiplied by 0.25 (which corresponds to 15 seconds data resolution) and summed together. For example, each MAP value ‘x’ below 65mmHg will be subtracted from 65, multiplied by 0.25 minutes and all such values were summed together as $AUC = \sum [(65-x) * 0.25] \text{ mmHg}\cdot\text{min}$.

XV.2.5. Vasopressor-Inotrope dose

Vasopressor-Inotrope dose was calculated by adding norepinephrine equivalents of total norepinephrine, epinephrine, phenylephrine, milrinone and vasopressin dose used during

surgery; using the formula, total vasopressor-inotrope dose = [norepinephrine] + [epinephrine] + [phenylephrine ÷ 10] + [milrinone ÷ 5] + [vasopressin × 8.33] (Mancl & Muzevich, 2012, p. 201; Russell et al., 2008).

XV.2.6. Primary outcome

Mortality was the primary outcome of interest and was defined based on Society of Thoracic Surgeons (STS) version 2.81 (Appendix VII). It includes:

- (1) all deaths, regardless of cause, occurring during the hospitalization in which the operation was performed, even if after 30 days (including patients transferred to other acute care facilities).
- (2) all deaths, regardless of cause, occurring after discharge from the hospital, but before the end of the thirtieth postoperative day. If a patient was discharged, they were given a 30-day appointment. Those who missed the 30-day appointment were contacted through phone by the STS database coordinator to note the morbidity and mortality. State STS coordinators also run the Social Security Death Index to capture those who died within 30 days after cardiac surgery, and this information was sent to the individual hospital.

XV.2.7. Model training and validation

We trained five different models:

- 1) Logistic Regression, a statistical model that in its basic form uses a logistic function to model a binary dependent variable (Tolles & Meurer, 2016).
- 2) Random forests (RF), an ensemble learning method that uses a set of learning algorithms for classification that operates by constructing a multitude of decision trees models in train and outputting a classification, which corresponds to the mode of the classifications given by the individual decision trees (Tin Kam Ho, 1998).
- 3) Neural Network (NN), a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates.
- 4) Support Vector Machine (SVM), a representation of the examples in our case each patient's data is an example, as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible.
- 5) Extreme Gradient Boosting (XGB), a machine learning technique for classification problems, which produces a prediction model in the form of an ensemble (uses a set of learning algorithms) of weak prediction models (Breiman, 1997).

The abovementioned five models were trained using the three intraoperative phases (outside CPB, CPB and total surgery) separately. For each intraoperative phase, we applied stratified

random sampling in order to split them into two: 75% for model training and 25% for testing, stratified by mortality. The class distributions were balanced, i.e. the same proportion of positive and negative classes existed in each split. We performed 10-fold cross validation while training our models and performed hyper-parameter tuning until best performance was achieved. We selected the best performing model as the one with the highest value of the area under the receiver operating characteristic curve (AUROC). The model with the best performance during model training was selected for testing in a hold-out dataset.

XV.2.8. Model performance

Predictive ability of models was assessed by AUROC. In addition, we measured the positive predictive value (PPV), sensitivity and specificity. The AUROC is a function of the true positive rate (sensitivity) versus the false positive rate (1-specificity), integrated over all thresholds. An AUROC of 0.50 is achieved through random predictions, 1 represents a perfect discrimination. The precision or PPV (Positive Predictive Value) is intuitively the ability of the classifier not to label as positive a sample that is negative. The sensitivity represents the cases where the patient is correctly classified as being positive and the specificity represents the cases where the patient is correctly classified as being negative.

We used a simple bootstrapping method with replacement, where the original dataset was sampled to form a new sample (a “resample” or bootstrap sample) with the same size. The bootstrap sample was taken from the original one by using sampling with replacement and this process was repeated 100 times, which was considered adequate given the dimension of the dataset, and for each of these bootstrap samples the mean was computed in 95% Confidence Intervals (CI). Performance was compared using a pair-wise corrected resampled t-test. Statistical significance was considered with $P < 0.05$.

A non-parametric isotonic calibration was performed for the model exhibiting the best performance. Calibration curves, also referred to as reliability diagrams, present the fraction of patients in the positive class against the predicted probabilities. The mean of the predicted probabilities was computed for each decile. A well calibrated binary classifier by norm has an increasing number of true cases as one goes from the decile with the lowest mean predicted probability to the decile with the highest mean predicted probability.

Appendix VII presents an extended and detailed description of machine learning models, their performance and methodology.

XV.2.9. Data imputation

When there was a gap smaller or equal to 5 minutes, we expanded it by using median imputation. Since this gap can be located in the beginning of the time series using another approach such as the LPV (Last Present Value) would have resulted in forward filling the missing values with more missing values. Each observation (row) corresponds to one case, there are 3650 data points factors (columns) in the full dataset, 1 every 15 seconds. This is because the longest case contains 3650 data points using 15 seconds as Sampling Time (ST). Cases shorter than this one will contain missing data from the last time-stamp of the surgery until data-point number 3650.

Instead of removing the rows with missing data values when surgery finishes, we replaced these missing data values with a specific value that did not appear naturally in the input, we chose -1. This way, the problem can be modeled as-is and we can encourage the model to learn that a specific value means “missing.” We used median imputation for hematocrit change (Delta), ejection fraction and cross clamp time. Artifacts were removed using the rules described elsewhere (Salmasi et al., 2017).

XV.2.10. Statistical Analysis

Results were expressed as mean and SD for continuous factors and as frequencies/percentages for categorical factors. In our analysis, 3 datasets containing different surgery phases data comprising both patients with record and no record of STS Risk were used to train the Machine Learning Models previously listed, for the patients with no record, the score was imputed using median imputation. Normality was assessed in continuous factors making use of a histogram and those who did not have a normal distribution were log-transformed. Statistical analyses were performed in R version 3.5.2, RStudio version 1.2.1335, Python version 3.7.3 and JupyterLab version 1.1.1. All tests were two-sided with a 0.05 level of significance.

XV.3. Results

XV.3.1. Baseline characteristics

Figure 21. Flow chart presenting patient selection and analysis. presents an overview of patient selection and analysis. Data describing 5979 patients who underwent cardiac surgery between January 2008 to June 2016 were collected. We excluded 157 patients who underwent emergent procedures and 807 patients with inadequate data. The final analysis had 5015 patients. Table 26 summarizes the baseline characteristics of patients stratified by mortality. The mean (standard deviation, SD) duration of surgery was 294 (67) minutes, CPB duration was 94 (38) minutes and cross clamp time was 73 (30) minutes.

Figure 21. Flow chart presenting patient selection and analysis.

Demographics	All patients N = 5015	Mortality		P value
		Absent N = 4903 (97.76)	Present N = 112 (2.23)	
Gender, Male [†]	3502 (69.8)	3433 (70.0)	69 (61.6)	0.070
Age, years [‡]	66.88 (11.77)	66.78 (11.75)	71.60 (11.80)	<0.001
Hematocrit [§]	2.79 (8.18)	2.84 (8.20)	0.76 (6.77)	0.008
Ejection fraction [‡]	54.17 (13.67)	54.22 (13.64)	52.30 (14.65)	0.142
Surgery category [†]				<0.001
CABG	2482 (49.5)	2440 (49.8)	42 (37.5)	
Valve	1457 (29.1)	1437 (29.3)	20 (17.9)	
CABG + Valve	914 (18.2)	871 (17.8)	43 (38.4)	
Other	162 (3.2)	155 (3.2)	7 (6.3)	
Duration of CPB [‡]	94.29 (38.09)	93.40 (36.37)	133.52 (74.45)	<0.001
Duration Outside CPB [‡]	199.81 (44.31)	199.11 (43.30)	230.45 (70.15)	<0.001
Duration of Total surgery [‡]	294.11 (66.73)	292.51 (64.38)	363.96 (114.36)	<0.001
Cross clamp time [‡]	72.91 (29.46)	72.46 (28.68)	92.74 (49.75)	<0.001
AUC-MAP<65mmHg, mmHg.min [‡]				
CPB	802.96 (550.27)	793.79 (521.31)	1204.24 (1228.78)	<0.001
Outside CPB	293.23 (293.29)	286.74 (281.69)	577.18 (546.19)	<0.001
Total Surgery	1096.29 (692.39)	1080.64 (655.71)	1781.42 (1477.38)	<0.001
Vasopressor-inotrope, mg [*]	0.67 (0.38, 1.16)	0.66 (0.37, 1.14)	1.22 (0.68, 5.40)	<0.001
STS risk [‡]	0.02 (0.03)	0.02 (0.03)	0.05 (0.06)	<0.001

Table 26. Baseline patient and surgical characteristics stratified by mortality

[†] number (percentage), [‡] mean $\pm$ standard deviation, ^{*}median (quartile 1, quartile 3) CABG, Coronary Artery Bypass Grafting; CPB, Cardiopulmonary bypass; [§] Delta – difference in hematocrit from baseline Duration Outside CPB = Duration of total surgery – Duration of CPB; Duration of Total surgery = Duration of CPB + Duration Outside CPB; AUC, Area under the curve; STS, Society of Thoracic Surgeons.

XV.3.2. Postoperative mortality

In total, 112 (2.2%) patients had mortality (Table 25). These patients were significantly older [71.6 (11.8) vs 66.8 (11.8) years, $P < 0.001$], had longer duration of surgery [364 (114.4) vs 293 (64.4) minutes, $P < 0.001$], CPB time [133.5 (74.6) vs 93.4 (36.4) minutes, $P < 0.001$] and cross clamp time [92.7 (49.8) vs 72.5 (28.7) minutes, $P < 0.001$] as compared to those patients who survived.

XV.3.3. IOH and vasopressor-inotrope dose

Patients with mortality had significantly higher incidence of intraoperative hypotension and received higher vasopressor-inotrope dose. IOH characterized as AUC -MAP < 65 mmHg were significantly higher in patients with mortality [1781.4 (1477.4) vs 1080.6 (655.7) mmHg.min, $P < 0.001$]. Additionally, they required a higher median (Q1, Q3) dose of vasopressor-inotropes [1.22 (0.68, 5.40) vs 0.66 (0.37, 1.14) mg, $P < 0.001$] as compared to those patients who survived (Table 20).

XV.3.4. Predictive ability of models

Ability of phase-specific models in predicting mortality were presented in Table 27. Models incorporating intraoperative factors significantly predicted mortality. In outside-CPB phase, all five models demonstrated good predictive ability (AUROC values ≥ 0.7). However, support vector machine from CPB and total surgery phases, didn't show good performance (AUROC values ≤ 0.7). Logistic regression, Random forests and Extreme Gradient Boosting from all three phases were strongly predictive of mortality (AUROC values ≥ 0.8).

Models	AUROC	PPV	Specificity	Sensitivity
<i>Outside CPB phase [†]</i>				
Logistic Regression	0.80 [0.69-0.88]	0.08 [0.04-0.11]	0.8 [0.77-0.82]	0.71 [0.57-0.88]
Random Forests	0.82 [0.73-0.90]	0.09 [0.05-0.14]	0.86 [0.83-0.88]	0.68 [0.48-0.84]
Neural Networks	0.78 [0.68-0.85]	0.11 [0.06-0.19]	0.92 [0.90-0.93]	0.36 [0.20-0.55]
Support Vector Machine	0.70 [0.59-0.79]	0.06 [0.02-0.10]	0.89 [0.87-0.91]	0.36 [0.20-0.55]
Extreme Gradient Boosting	0.87 [0.81-0.93]	0.10 [0.06-0.15]	0.85 [0.83-0.87]	0.71 [0.52-0.85]
<i>CPB phase</i>				
Logistic Regression	0.83 [0.72-0.90]	0.07 [0.04-0.10]	0.77 [0.76-0.8]	0.75 [0.57-0.89]
Random Forests	0.86 [0.76-0.92]	0.09 [0.05-0.13]	0.81 [0.79-0.83]	0.79 [0.61-0.92]
Neural Networks	0.72 [0.59-0.85]	0.09 [0.04-0.13]	0.89 [0.88-0.91]	0.43 [0.20-0.61]
Support Vector Machine	0.69 [0.57-0.80]	0.08 [0.04-0.11]	0.84 [0.82-0.86]	0.57 [0.44-0.71]
Extreme Gradient Boosting	0.80 [0.67-0.90]	0.09 [0.06-0.14]	0.83 [0.81-0.85]	0.79 [0.61-0.91]
<i>Total surgery phase [‡]</i>				
Logistic Regression	0.80 [0.70-0.88]	0.08 [0.04-0.12]	0.81 [0.79-0.83]	0.68 [0.51-0.84]
Random Forests	0.83 [0.74-0.92]	0.10 [0.06-0.14]	0.85 [0.83-0.88]	0.68 [0.52-0.89]
Neural Networks	0.70 [0.59-0.80]	0.10 [0.03-0.18]	0.96 [0.94-0.97]	0.21 [0.07-0.36]
Support Vector Machine	0.66 [0.52-0.79]	0.08 [0.04-0.12]	0.87 [0.86-0.89]	0.46 [0.30-0.65]
Extreme Gradient Boosting	0.85 [0.76-0.91]	0.10 [0.06-0.15]	0.86 [0.84-0.88]	0.68 [0.50-0.81]

Table 27. Ability of models in predicting mortality based on specific phases of surgery

AUROC, area under the receptor operating curve; CPB, cardiopulmonary bypass; [†] Outside CPB: duration before and after CPB (Total surgery duration – CPB duration); [‡] Total surgery: CPB + outside CPB.

Of all the phase-specific models tested, Extreme Gradient Boosting model from outside-CPB phase had strong predictive ability [AUROC: 0.87 (95% CI: 0.81-0.93)] together with better PPV [0.10 (95% CI: 0.06-0.15)] , specificity [0.85 (95% CI: 0.83-0.87)] and sensitivity [0.71 (95% CI: 0.52-0.85)] compared to other models. The AUROC values for all machine learning methods trained with outside-CPB phase were presented in Figure 22.

Figure 22. Area Under the Receiver Operating Characteristic curves for Calibrated versus non-calibrated eXtreme Gradient Boosting with outside-CPB phase data.

The calibration of the best model was assessed to understand the reliability of the predictions. The AUROC curves for the model before and after calibration is presented in Figure 23 and the calibration plot in Figure 24.

Figure 23. AUROC comparison.

Figure 24. Calibration plot.

We applied isotonic calibration and observed an improvement in AUROC [0.88 (95% CI: 0.83-0.94)], specificity [0.86 (95% CI: 0.84-0.88)] and sensitivity [0.75 (95% CI: 0.57-0.90)] (Figure 23), with the other performance measures maintaining the same values. From the calibration curve, we concluded that the model over-estimated the risk of mortality. However, an over estimation of the probabilities of mortality in this case can be preferable to an under-estimation of the risk.

We provide open access to all our data-extraction, filtering, data-wrangling, modelling, figures and tables code and queries on: https://github.com/theonesp/bp_machine_learning/ . A deidentified version of vital-signs and drugs AIMS data used in this project will soon be available on <https://physionet.org/> .

XV.4. Discussion

The present study was designed to evaluate the mortality prediction of the commonly used machine learning models when CPB specific intraoperative hypotension and other intraoperative adverse factors were added to the preoperative risk factors. The main findings of this study that evaluated mortality prediction following cardiac surgery are:

- 1) machine learning models demonstrated excellent predictive ability with pre- and intraoperative risk factors that included intraoperative hypotension.
- 2) Extreme Gradient Boosting model from outside-CPB phase was a better predictor of mortality with higher discrimination (AUROC), positive predictive value, sensitivity and specificity compare to the other phase-specific machine learning models.

The importance of perioperative risk stratification is well known. Several conventional risk prediction scores and machine learning models for cardiac surgery are mostly based on static patient and surgical factors. Dynamic physiological changes during intraoperative period significantly affects the outcome. Several studies have demonstrated the association between IOH and other changes during CPB with adverse outcomes after cardiac surgery (Jinadasa et al., 2018; Sun et al., 2018; Wang et al., 2011). In a study among cardiac surgical patients, logistic regression models with adverse intraoperative factors enhanced prediction of postoperative morbidity-mortality risk (Stoica, 2002). Similarly, our results demonstrate that including intraoperative factors in the machine learning models offered better discrimination and prediction of postoperative mortality.

In this study machine learning models accurately predicted postoperative mortality. These findings are consistent with other machine learning model based risk stratification (Allyn et al., 2017; Merath et al., 2019; Shi et al., 2012; Wise et al., 2015). Machine learning models have been shown to better predict mortality than conventional logistic regression following liver cancer and aortic aneurysm surgery (Shi et al., 2012; Wise et al., 2015). Thus, our results broadly support other work with machine learning models and mortality prediction.

The most interesting finding in our study was that the machine learning model (XGB) based on outside-CPB phase (before and after CPB) had a better predictive ability of mortality. Induction of anesthesia could lead to clinically significant hypotension (Reich et al., 2005, p. 200) and a significant fall in blood pressure complexity measures (Rangasamy et al., 2019). Postinduction hypotension and loss of complexity have been associated with significant postoperative adverse events (Suán et al., 1997, p. 199). Furthermore, prolonged bypass times and cross clamp times

have been associated with postoperative adverse events. It is uncertain whether there are modifiable targets such as the blood pressure during CPB. Post bypass-phase has been reported as the most vulnerable period (Reves et al., 1984). Thus, the association of outside CPB with postoperative mortality is significant enough to warrant large database studies from another center.

Random Forest modeling has been shown to predict mortality better than other models in cardiac surgical patients (Allyn et al., 2017) and patients with myocardial infarction (Roni Shouval et al., 2017). However, our study found extreme gradient boosting to be more accurate compared to other models. These findings may partly be explained by the incorporation of intraoperative factors such as IOH in our study. Incorporating intraoperative adverse events might be of significant value when refining risk prediction models in future.

Machine learning comprises of algorithms that could learn and improve on its own. This allows prediction easier and accurate with more information. Recently, these models are increasingly used for predicting perioperative outcomes. While previous machine learning models were exclusively based on preoperative factors, we incorporated phase-specific intraoperative factors such as IOH, vasopressor-inotrope dose, duration of surgery and cross clamp time. These factors are independent risk factors and could be readily obtained. Accordingly, combining pre- and intraoperative factors would be a logical approach that could offer several advantages over other machine learning driven models. Although they are late and/or closer to the event, still valuable inputs can be obtained for triaging and postoperative care of these patients.

Overall, the relevance of our results could be interpreted on several levels. We provide an additional proof of concept for the existing literature on machine learning models especially for cardiac surgery. Moreover, in an era of digitalized medical records and accessibility to big data, using machine learning might become imperative. In addition, from a clinical perspective, the importance of intraoperative adverse events like IOH and vasopressor-inotrope dose becomes evident and may serve as a target for improving outcomes.

The present study has several limitations. First, as with any retrospective study it is susceptible to selection bias and finding a causative link is not possible. Moreover, the performance of machine learning models depends on the quality of data analyzed. It is possible that other factors that were not included might have changed our results. However, we analyzed a prospectively collected data, captured BP every 15 seconds that could have reliably captured IOH. Moreover, we characterized IOH as AUC that represents the depth in addition to the duration of hypotension. Second, our outcome mortality from STS database were limited to 30-day follow up period and events after 30 postoperative days could have been missed. Third, our models considered time points as independent features, thus as a future work we propose to use other techniques that are able to incorporate time series such as long short-term memory neural networks. Finally, we couldn't do external validation for our models.

In conclusion, XGB machine learning model trained with IOH outside CPB-phase seems to offer a better discrimination, sensitivity, specificity, and PPV compared to other models. Machine learning models incorporating intraoperative adverse factors have the potential to offer better discriminative ability for risk stratification and triaging of patients following cardiac surgery. In an era of precision medicine, developing accurate machine learning models with all available information could improve prediction of mortality following cardiac surgery.

XVI. ASSOCIATION OF CHLORIDE ION AND SODIUM-CHLORIDE DIFFERENCE WITH ACUTE KIDNEY INJURY AND MORTALITY IN CRITICALLY ILL PATIENTS

XVI.1. Introduction and problem statement

Chloride is the most abundant anion in the extracellular fluid and plays an important role in numerous physiologic functions (Berend et al., 2012). Recently, researchers have described an association between hyperchloremia and acute kidney injury (AKI) after abdominal surgery (Toyonaga & Kikura, 2017); after non-cardiac surgery (McCluskey et al., 2013); in septic shock (Suetrong et al., 2016); with subarachnoid hemorrhage (Sadan, 28504980); and in unselected intensive care unit (ICU) patients (Marttinen et al., 2016; Zhang et al., 2013). Hyperchloremia has also been associated with increased mortality in postsurgical and intensive care unit (ICU) populations (Boniatti et al., 2011; Lee et al., 2016; McCluskey et al., 2013; Shaw et al., 2014; Silva et al., 2009).

Stewart in 1978 proposed an alternative approach to acid-base physiology which considered serum bicarbonate a dependent variable, with “strong ions” (ions which completely dissociate from hydrogen in physiologic conditions) as well as “weak nonvolatile acids” (ions which can exist either associated with or dissociated from their hydrogen ion) as the independent variables along with plasma carbon dioxide (Stewart, 1978, p. 78). “Apparent strong ion difference” (aSID) could be calculated by subtracting the sum of serum chloride and lactate from the sum of serum sodium, potassium, calcium, and magnesium, and could be used to determine the presence of an acidosis or alkalosis. Subsequently, studies have shown that “main strong ion difference” (mSID), the difference between sodium ion concentration ($[Na^+]$) and chloride ion concentration ($[Cl^-]$), can be used as surrogate for aSID (Lombardi et al., 2019, p. 20; Mallat et al., 2013, p. 13).

Prior studies have shown strong ion difference calculations using the Stewart approach can be useful as a prognostic marker for mortality in ICU populations, although with no clear advantage over the more common bicarbonate-centric acid base approach (Cusack et al., 2002; Ho et al., 2016; Kaplan & Kellum, 2007, p. 200). However, many of the previous studies correlating $[Cl^-]$ with AKI did not evaluate whether $[Cl^-]$ is merely acting as a proxy for strong ion difference, which may be better evaluated using calculations such as mSID. (Marttinen et al., 2016; H. J. Oh et al., 2017; T. K. Oh et al., 2018; Suetrong et al., 2016; Toyonaga & Kikura, 2017; Yessayan et al., 2017; Zhang et al., 2013); In this multi-center, retrospective study, we evaluate the association of mSID and $[Cl^-]$ on admission to the ICU with subsequent development of AKI and ICU mortality.

XVI.2. Materials and methods

XVI.2.1. Study population and data source

Data were analyzed from patients admitted into ICU in 208 hospitals across the United States between 2014 to 2015 as captured in the eICU Collaborative Research Database v2.0 (Pollard et al., 2018). The study is exempt from institutional review board approval due to the retrospective design, lack of direct patient intervention, and the security schema, for which the re-identification risk was certified as meeting safe harbor standards by an independent privacy expert (Health Insurance Portability and Accountability Act Certification no. 1031219-2).

Inclusion criteria were:

- 1) first admission into the ICU during the study period,
- 2) 18 years of age or older.
- 3) medical ICU patients.

Exclusion criteria were:

- 1) patients who received renal replacement therapy before ICU admission.
- 2) patients who had creatinine concentration more than 3.0 mg/dl at admission.
- 3) patients who underwent surgery before ICU admission.
- 4) patients without data for mSID calculation for the first 24 hours after ICU admission.

XVI.2.2. Study variables

The primary outcome was defined as the development of AKI within seven days of ICU admission. AKI was diagnosed and classified by Kidney Disease Improving Global Outcomes (KDIGO) consensus criteria (Kellum & Lameire, 2013). According to this classification and staging system of AKI, serum creatinine concentration ≥ 1.5 times the baseline or increase in serum creatinine ≥ 0.3 mg/dl from baseline creatinine concentration constitutes stage 1, serum creatinine concentration ≥ 2.0 times the baseline constitutes stage 2, and serum creatinine concentration ≥ 3.0 times the baseline or initiation of renal replacement therapy constitutes stage 3. Baseline creatinine, as in previous studies, was considered to be creatinine on admission or the previous creatinine value before ICU admission when available (dos Santos et al., 2019; Pinheiro et al., 2019).

The secondary outcomes were mortality in the ICU and in the hospital. ICU mortality was defined as death prior to discharge from the ICU, with hospital mortality defined as death prior to discharge from hospital. Baseline characteristics including age, gender, weight, and ethnicity, and past medical history were extracted. Information on vital signs such as heart rate, blood pressure, and temperature were also extracted from the database.

[Cl⁻] at admission was defined as the first serum chloride ion measurement within 24 hours of ICU admission. [Na⁺] measured at the same time as the [Cl⁻] measurement was also extracted and used to calculate mSID on admission: [Na⁺] – [Cl⁻]. Low SID and high SID were defined as SID < 31 mEq/L and > 37 mEq/L, respectively. Hypochloremia and hyperchloremia were defined as [Cl⁻] < 98 mEq/L and > 110 mEq/L, respectively. Both mSID and [Cl⁻] were stratified into bins of 2 mEq/L in width and represented as a categorical variable.

XVI.2.3. Statistical Analyses

All results for continuous variables are expressed as mean $\pm$ standard deviation, or median (IQR, interquartile range; 25% quartile, 75% quartile) as appropriate. Shapiro-Wilk test was used to test for normal distribution. For groupwise comparisons of continuous variables, Student t test (two groups) or one-way ANOVA (n groups) was used when variables were normally distributed. Wilcoxon rank sum test (two groups) or Kruskal-Wallis test (n groups) was used when variables were not normally distributed. For categorical variables, Fisher's exact test or the chi-square test was used.

An adjusted odds ratio for an outcome was calculated by a logistic regression model, adjusting for factors that could be related to both mSID and [Cl⁻] and the outcomes. mSID and [Cl⁻] were treated as categorical variables in case of non-linear relationship with the outcomes. Missing values in continuous covariates in the model were replaced with mean values. Inverse probability weighting was used to estimate causal effects in a sensitivity analysis. Since treating mSID and [Cl⁻] as categorical variables ignores the order and could decrease power, four subgroups were created and analyzed separately: a) low mSID (≤ 34 mEq/L) and low [Cl⁻] (≤ 106 mEq/L) b) low mSID (≤ 34 mEq/L) and high [Cl⁻] (> 106 mEq/L), c) high mSID (> 34 mEq/L) and low [Cl⁻] (≤ 106 mEq/L), and d) high mSID (> 34 mEq/L) and high [Cl⁻] (> 106 mEq/L). In another sensitivity analysis, mSID and [Cl⁻] were treated as continuous variables.

All statistical comparisons were two-sided and a significant level was defined as a P value of less than 0.05. All statistical analyses were performed using R 3.6.0 (R Foundation for Statistical Computing, Vienna, Austria). Data was queried from R using the package bigrquery version 1.2.0.

XVI.3. Results

XVI.3.1. Participant characteristics

Of an initial pool of 65,855 patients, there were 19,844 postoperative patients, 5752 patients whose creatinine concentration was more than 3.0 mg/dl on admission, 106 patients who received renal replacement therapy before admission, and 5,352 patients without data for mSID calculation who were excluded, resulting in a final cohort of 34,801 participants. Mean age was 63 years, 53% were male, and 77% were Caucasian. On admission, 7,030 patients (20.2%) had low mSID and 6,198 patients (17.8%) had high mSID. 4,167 patients (12.0%) had hyponatremia and 5,336 patients (15.3%) had hypernatremia. Table 28 shows the demographics and the hemodynamic and laboratory parameters for patients based on mSID classification.

A

	31≤mSID<37 mEq/L (n=21,573, 62%)	mSID<31 mEq/L (n=7,030, 20%)	mSID>37 mEq/L (n=6,198, 18%)	p-value
Age, year (SD)	63.3 (17.7)	61.6 (18.8)	64.8 (16.2)	<0.001
Male, n (%)	11752 (54.5)	3391 (48.2)	3185 (51.4)	<0.001
Height, cm (SD)	169.5 (10.9)	167.9 (10.8)	168.7 (11.1)	<0.001
Weight, kg (SD)	83.1 (24.6)	78.3 (22.2)	88.1 (31.3)	<0.001
Ethnicity, n (%)				<0.001
African American	1975 (9.2)	503 (7.2)	724 (11.7)	
Asian	205 (1.0)	62 (0.9)	33 (0.5)	
Caucasian	16574 (76.8)	5419 (77.1)	4888 (78.9)	
Hispanic	1386 (6.4)	419 (6.0)	307 (5.0)	
Other/Unknown	1433 (6.6)	627 (8.9)	246 (4.0)	
CHF, n (%)	2419 (13.4)	549 (9.4)	1454 (26.7)	<0.001
PVD, n (%)	822 (4.6)	260 (4.5)	316 (5.8)	<0.001
Hypertension, n (%)	8447 (47.0)	2424 (41.5)	2893 (53.2)	<0.001
COPD, n (%)	4075 (22.7)	1133 (19.4)	2045 (37.6)	<0.001
Diabetes, n (%)	2325 (12.9)	830 (14.2)	931 (17.1)	<0.001
Malignancy, n (%)	0.00 (0.06)	0.00 (0.05)	0.00 (0.04)	0.13
Creatinine, mg/dl (SD)	1.08 (0.54)	1.15 (0.58)	1.13 (0.56)	<0.001

Abbreviations: mSID, main strong ion difference; SD, standard deviation; CHF, congestive heart failure; PVD, peripheral vascular disease; COPD, chronic obstructive pulmonary disease

B

	31≤mSID<37 mEq/L (n=21,573)	mSID<31 mEq/L (n=7,030)	mSID>37 mEq/L (n=6,198)	p-value
mSID, mEq/L (SD)	33.7 (1.9)	28.3 (2.0)	40.8 (3.2)	<0.001
[Cl ⁻], mEq/L (SD)	104.8 (5.3)	109.8 (5.7)	98.1 (6.5)	<0.001
[Na ⁺], mEq/L (SD)	138.5 (5.0)	138.1 (5.5)	138.9 (5.8)	<0.001
Albumin, g/dl (SD)	3.1 (0.8)	2.8 (0.8)	3.3 (0.8)	<0.001
Bilirubin, mg/dl (SD)	1.04 (2.15)	1.21 (2.41)	1.02 (2.23)	<0.001
BUN, mg/dl (SD)	22.9 (16.3)	24.7 (17.9)	26.5 (18.5)	<0.001
WBC, *10 ³ /L (SD)	11.8 (7.0)	12.6 (8.2)	12.2 (7.0)	<0.001
Platelets, *10 ³ /L (SD)	220.9 (107.4)	210.8 (113.6)	229.9 (106.7)	<0.001
Hematocrit, % (SD)	35.3 (7.4)	33.0 (7.8)	36.9 (7.5)	<0.001
Hemoglobin, g/dl (SD)	11.7 (2.6)	11.0 (2.7)	12.0 (2.6)	<0.001
Glucose, mg/dk (SD)	161 (117)	164 (124)	180 (148)	<0.001
Potassium, mEq/L (SD)	4.06 (0.64)	4.17 (0.76)	4.07 (0.76)	<0.001
Uric acid, mg/dl (SD)	6.16 (2.97)	6.23 (3.36)	7.14 (3.21)	0.054
Heart rate, bpm (SD)	89.6 (22.1)	92.2 (22.4)	92.7 (22.5)	<0.001
Systolic blood pressure, mmHg (SD)	127.9 (27.7)	119.6 (26.4)	129.6 (28.1)	<0.001
Temperature, Celsius	36.8 (0.8)	36.8 (0.9)	36.7 (0.8)	<0.001

Abbreviations: mSID, main strong ion difference; SD, standard deviation; [Cl⁻], serum chloride ion concentration; [Na⁺], serum sodium concentration; BUN, blood urea nitrogen; WBC, white cell count.

Table 28. Comparison of patients' (A) demographics and (B) hemodynamic and laboratory values based on main strong ion difference classification

Three-thousand sixty-one patients (8.8%) developed AKI within 7 days after admission. Of those patients with AKI, 2,443 patients were classified as AKI stage 1, 453 patients were classified as AKI stage 2, and 165 patients were classified as AKI stage 3. There were 5,907 patients (17.0%) who could not be staged due to missing values. There were 1,700 deaths (4.8%) in the ICU and 2,618 deaths (7.5%) in the hospital.

XVI.3.2. Univariate analyses of primary and secondary outcomes

In univariate analysis, there were significant differences in AKI development, AKI stage, ICU mortality, and hospital mortality among groups classified based on mSID. AKI incidence and mortality were both highest in the mSID >37mEq/L, followed by the mSID <31mEq/L group and lowest in the group with mSID between 31 and 37mEq/L (Table 29).

	31<=mSID<37 mEq/L (n=21,573)	mSID<31 mEq/L (n=7,030)	mSID>37 mEq/L (n=6,198)	p-value
AKI, n (%)	1695 (7.9)	633 (9.0)	733 (11.8)	<0.001
AKI stage (SD)	0.12 (0.41)	0.14 (0.44)	0.17 (0.47)	<0.001
Death in ICU, n (%)	936 (4.3)	375 (5.3)	389 (6.3)	<0.001
Death in Hospital, n (%)	1447 (6.7)	547 (7.8)	624 (10.1)	<0.001

Table 29. Comparison of outcomes based on main strong ion difference classification

Abbreviations: mSID, main strong ion difference; SD, standard deviation; AKI, acute kidney injury; ICU, intensive care unit.

Figure 25 illustrates the incidence of AKI among groups stratified into 2-mEq/L-wide mSID bins. The incidence of AKI was lowest (7.2%) in patients with mSID 32-34 mEq/L and increased as SID increased and decreased. Figure 26 illustrates the incidence of AKI among groups stratified into 2-mEq/L-wide [Cl⁻] bins, showing the similar U-shaped relationship. Figure 27 and Figure 28 show the ICU mortality among groups stratified by mSID and [Cl⁻], again showing a U-shaped curve.

Figure 25. Incidence of acute kidney injury among stratified groups based on main strong ion difference.

Abbreviations: mSID, main strong ion difference; AKI, acute kidney injury.

Figure 26. Incidence of acute kidney injury among stratified groups based on serum chloride ion concentration.

Abbreviations: AKI, acute kidney injury; [Cl⁻], serum chloride ion concentration.

Figure 27. ICU mortality among stratified groups based on mSID.

Abbreviations: mSID, main strong ion difference; ICU, intensive care unit.

Figure 28. ICU mortality among stratified groups based on [Cl⁻].

Abbreviations: mSID, main strong ion difference; ICU, intensive care unit; [Cl⁻], serum chloride ion concentration

XVI.3.3. Multivariable analyses

A multivariable logistic regression analysis with development of AKI as the outcome is shown in Supplemental Table 28. Comparison of patients' (A) demographics and (B) hemodynamic and laboratory values based on main strong ion difference classification Table 28. Compared with mSID 32-34 mEq/L as a reference, the odds for AKI increased for nearly all mSID range groups above 34 mEq/L. There was no significant difference in the odds for AKI for mSID range groupings below 32 mEq/L compared to the reference range of 32-34 mEq/L. With [Cl⁻] 104-106 mEq/L as a reference, the odds for AKI were significantly higher only in [Cl⁻] ≤94 mEq/L and [Cl⁻] 98-100

mEq/L. Age, weight, black race, a diagnosis of congestive heart failure (CHF), and a number of hematologic and laboratory parameters were also associated with increased odds for AKI. Analyses using inverse probability weighting demonstrated similar findings compared to unweighted analyses for both mSID and [Cl-].

A multivariable logistic regression analysis for ICU mortality is shown in Supplemental Table 29. Compared with mSID 32-34 mEq/L as a reference group, odds for ICU mortality were significantly higher in the mSID 36-38 mEq/L and mSID 40-42 mEq/L groups, and significantly lower in the mSID 28-30 mEq/L group. Analyses conducted using inverse probability weighting showed significantly greater odds for ICU mortality in all groups with mSID greater than 34mEq/L other than the 36-38mEq/L group, as well as the <26 mEq/L group. Using [Cl-] 104-106 mEq/L as a reference group, the odds for ICU mortality were greater in the 112-114 mEq/L and [Cl-] >116 mEq/L groups. However, these associations were not significant with inverse probability weighting. Similar trends were seen using multivariable logistic regression analysis for overall hospital mortality; mSID groups >36 mEq/L and [Cl-] groups <98 mEq/L and >110 mEq/L were associated with increased hospital mortality. A multivariable logistic regression analysis for hospital mortality is shown in Supplemental Table 3.

Supplemental Table 4 shows logistic regression models with mSID and [Cl-] as continuous variables within 4 subgroups of mSID and [Cl-] ranges. In the group with high mSID and low [Cl-], both increased mSID and decreased [Cl-] were associated with AKI. Decreased mSID but not [Cl-] had significant association with AKI in the low mSID/high [Cl-] group. Increased mSID but not [Cl-] had significant association with AKI in the high mSID/high [Cl-] group. The only significant associations seen in sensitivity analyses for mortality were increasing [Cl-] and mSID in the high mSID/high [Cl-] group and increasing [Cl-] in the low mSID / high [Cl-] group.

Supplemental Figure 1 shows mSID plotted against serum bicarbonate in the same individual, demonstrating an absence of a linear relationship between the two variables. Supplemental Figure 2 shows [Cl-] plotted against mSID in the same individual.

XVI.4. Discussion

The hospital mortality from this large, predominantly Caucasian US-based population was 7.5%, which is somewhat lower than previous studies of ICU populations using the APACHE database (Jack E Zimmerman et al., 2013). Studies of AKI rates in ICU populations of this size have not been published, but smaller studies have demonstrated substantially higher rates of AKI than the 8.8% seen in this study (Martinen et al., 2016; Zhang et al., 2013). Discrepancies in mortality and AKI rates could be explained by differences in exclusion criteria, specifically those with significant elevation in creatinine or the need for renal replacement therapy prior to admission. While some studies computed serum creatinine concentration based on a formula by assuming a normal glomerular filtration rate for patients without baseline creatinine concentration before admission (de Vasconcellos & Skinner, 2018; Suetrong et al., 2016; Zhang et al., 2013), this assumption could overestimate AKI incidence after admission.

Both low and high mSID values are associated with an increased risk of AKI, severe AKI, ICU mortality, and hospital mortality overall compared to normal mSID values. Subdividing mSID values into smaller groups both above and below normal range reveals a U-shaped relationship, with greater risk as mSID deviates further from the normal range. Risk for AKI was greater for

mSID values above normal range compared with values below normal range, which was corroborated in multivariate analysis of the subdivided dataset showing nearly all groups with elevated mSID values with increased risk for AKI, whereas only the lowest mSID group among depressed mSID groups was associated with increased risk for AKI.

Prior studies have shown strong ion difference calculations using the Stewart approach can be useful as a prognostic marker for mortality in ICU populations (Cusack et al., 2002; Ho et al., 2016; Kaplan & Kellum, 2007). Here, we demonstrate a similar roughly U-shaped relationship between mSID and both ICU and in-hospital mortality. Low mSID values in the case of mortality were even more tenuously associated with increased mortality in either the hospital or ICU, and significant only in inverse probability weighted analysis.

Previous studies have demonstrated an association between abnormal serum chloride concentration and development of AKI in critically ill patients (Baalaaji et al., 2018; Commereuc et al., 2019; Kimura et al., 2019; Marttinen et al., 2016; T. K. Oh, Kim, et al., 2019; Patel et al., 2016; Sadan et al., 2017; Suetrong et al., 2016; Yessayan et al., 2017; Zhang et al., 2013). However, these studies considered [Cl⁻] as a continuous variable, assuming a linear relationship between [Cl⁻] and log odds of AKI development. This study shows a U-shaped association between [Cl⁻] and AKI, suggesting that representing [Cl⁻] as a continuous variable in a regression model may be a simplification of the relationship. Prior studies which treated [Cl⁻] as a categorical variable to evaluate a nonlinear relationship with AKI subdivided [Cl⁻] only into two or three bins due to sample size considerations (de Vasconcellos & Skinner, 2018; Mao et al., 2019; H. J. Oh et al., 2017; T. K. Oh, Do, et al., 2019; T. K. Oh et al., 2018; Shao et al., 2016; Toyonaga & Kikura, 2017) 2,17,18,25,32-34, a limitation we were able to overcome with the large size of our dataset.

Many of the studies correlating [Cl⁻] and AKI have not evaluated whether abnormal [Cl⁻] is merely a proxy for strong ion difference (Marttinen et al., 2016; H. J. Oh et al., 2017; T. K. Oh et al., 2018; Suetrong et al., 2016; Toyonaga & Kikura, 2017; Yessayan et al., 2017; Zhang et al., 2013). In this study, [Cl⁻] was not significantly associated with AKI when included in multivariate models that also contained mSID other than at the ≤ 94 level; it was also not significantly associated with ICU mortality except at the ≤ 94 level in inverse probability weighted models. These findings suggest mSID may be a better predictor for both AKI and ICU mortality than [Cl⁻] alone.

Sensitivity analyses were done to evaluate each combination of mSID and [Cl⁻] deviation from normal range: Low mSID and high [Cl⁻], most consistent with metabolic acidosis; high mSID and low [Cl⁻], most consistent with metabolic alkalosis; and low mSID/high [Cl⁻] and high mSID/low [Cl⁻] pairings, which are less consistent with pure metabolic processes. These analyses corroborated the advantage of mSID over [Cl⁻] as a predictor of AKI. The sensitivity analyses also supported but less conclusively the association between increased mSID and mortality.

mSID was originally conceived as a simplified way of evaluating acid-base physiology using the Stewart approach (Lombardi et al., 2019; Mallat et al., 2013). In this study, however, there were nearly equal numbers of individuals with mSID above and below a normal range, yet the majority of patients had a metabolic acidosis based on serum bicarbonate levels. While the majority of individuals with metabolic alkalosis based on serum bicarbonate levels had high mSID values, individuals with metabolic acidosis based on serum bicarbonate levels had a broad range of mSID values. This deviation does not appear to be driven by variation in [Na⁺], which were largely similar across mSID groups.

There are several limitations in this study. First, mSID incorporates [Cl⁻] in its calculations, so there is some degree of collinearity between the two features. However, the data does demonstrate a significant degree of discordance with the variables and their relationship to the outcomes. Second, baseline creatinine was unavailable and so the initial serum creatinine was considered baseline; while this allows for assessment of subsequent worsening of kidney function, it does not capture processes that may have started before arrival to the ICU, including the lag time for creatinine to reach steady state in cases where the cause of AKI was addressed before arrival to the ICU. More broadly, this analysis is unable to take into account any medical interventions done prior to ICU arrival which could affect the risk of AKI and mortality. Finally, the use of mSID, while clinically appealing due to its simplicity, is limited in its ability to allow inference about the underlying physiology. This study raises additional questions about the relationship between mSID, acid-base status, and clinical outcomes, which will require further investigation to better answer. Code is available at:

XVI.5. Conclusions

mSID measured on presentation to the ICU could be a useful consideration in predicting AKI and mortality risk on initial patient evaluation, with low and, in particular, high values representing increased risk. The association between [Cl⁻] and AKI or mortality largely became insignificant in models incorporating mSID, suggesting mSID is a better predictive marker than chloride on ICU presentation. However, mSID deviations from normal did not appear to correlate with changes in serum bicarbonate alone, a finding which merits further exploration.

XVII. CONCLUSIONS AND FUTURE WORKS

XVII.1. Publications

As a result of the work carried out during the PhD dissertation, we were able to publish the following manuscripts developed between 2017-2020:

1. Armengol de La Hoz, M.Á. †, Hao, S., Celi, L.A., Gómez, E.J.*, Danziger, J.*. Altered Consciousness Mediates the Effect of Hyponatremia but Not Hyponatremia, On Mortality.
2. Armengol de La Hoz, M.Á., Celi, L.A., Gómez, E.J., Danziger, J.. Admission Disorientation and Risk of Code Status Change.
3. Organizing Committee of the Madrid 2017 Critical Care Datathon, Núñez Reiz A, Martínez Sagasti F, Álvarez González M, Blesa Malpica A, Martín Benítez JC, Nieto Cabrera M, Del Pino Ramírez Á, Gil Perdomo JM, Prada Alonso J, Celi LA, Armengol de la Hoz MÁ, Deliberato R, Paik K, Pollard T, Raffa J, Torres F, Mayol J, Chafer J, González Ferrer A, Rey Á, González Luengo H, Fico G, Lombroni I, Hernandez L, López L, Merino B, Cabrera MF, Arredondo MT, Bodí M, Gómez J, Rodríguez A, Sánchez García M. Big data and machine learning in critical care: Opportunities for collaborative research. *Med Intensiva*. 2019 Jan-Feb;43(1):52-57. English, Spanish. doi: 10.1016/j.medin.2018.06.002. Epub 2018 Aug 2. PMID: 30077427.
4. Armengol de la Hoz, M. Á.† , Fernandes M.†, Rangasamy V., Subramaniam B. Machine Learning Models with Preoperative Risk Factors and Intraoperative Hypotension Parameters Predict Mortality Following Cardiac Surgery
5. Armengol de La Hoz, M. Á. †, Rangasamy, V.†, Novack V., Brenes Bastos A., Nabel S., Ramachandran S.K & Subramaniam B. Increasing Vasopressor Dose used to treat Intraoperative Hypotension and not the Duration of Hypotension is Strongly Associated with Adverse Outcomes following Cardiac Surgeries
6. Kimera S. †, Armengol de La Hoz, M. Á. †, Raines, N.H. , Celi, L.A., Association of Chloride Ion and Sodium-Chloride Difference with Acute Kidney Injury and Mortality in Critically Ill Patients
7. Cosgriff C. V. , Celi L. A., Ko S., Sundaresan T., Johnson A. , Pollard T., Armengol de la Hoz M. Á., Kaufman A. R., Stonen D. J., Badawi O., Deliberato R. O. .Developing Well Calibrated Illness Severity Scores for Decision Support in the Critically Ill. *Nature Digital Medicine*
8. Reiz, A., Sanchez Garcia, M., Armengol de La Hoz, M. Á. Big Data Analysis and Machine Learning in Intensive Care Medicine. *Medicina Intensiva* (2018).
9. Deliberato, R. O., Ko, S., Komorowski, M., Armengol de La Hoz, M. Á., Frushicheva, M. P., Raffa, J. D., ... & Stone, D. J. (2018). Severity of illness scores may misclassify critically ill obese patients. *Critical care medicine*, 46(3), 394-400.
10. Serpa Neto, A., Kugener, G., Bulgarelli, L., Rabello Filho, R., Armengol de La Hoz, M. Á.. D. L., Johnson, A. E., ... & Ferraz, L. J. R. (2018). First Brazilian datathon in critical care. *Revista Brasileira de terapia intensiva*, 30(1), 6-8.
11. de Toledo Piza, F. M., Celi, L. A., Deliberato, R. O., Bulgarelli, L., de Carvalho, F. R. T., Rabello Filho, R., Armengol de La Hoz, M. Á.... & Kesselheim, J. C. (2018). Assessing team effectiveness and affective learning in a datathon. *International Journal of Medical Informatics*, 112, 40-44.

12. Fehnel, C. R. †, de la Hoz, M. A. †, Celi, L. A., Campbell, M. L., Hanafy, K., Nozari, A., ... & Mitchell, S. L. (2020). Incidence and risk model development for severe tachypnea following terminal extubation. *Chest*.
13. Danziger, J., Armengol de La Hoz, M. Á., Li, W., Komorowski, M., Deliberato, R. O., Rush, B. N. M., Mukamal, K.J., Leo Celi, Badawi, O. Temporal trends in critical illness outcomes in United States minority serving hospitals.

† Indicates co-first authors.

* Indicates co-senior authors.

XVII.2. Research hypotheses verification

XVII.2.1. General Hypothesis

Our main hypothesis is that an effective strategy of promotion, integration, management and processing of critical inpatients' Open Big Data Repositories can overcome decisive social, technological and clinical barriers boosting the enhancement of research that eventually could improve the quality of life of critical inpatients.

Hypothesis confirmed

During the development of this dissertation we were able to overcome different organizational, bureaucratic, cultural and professional barriers in the field of study of the promotion of Critical Inpatients' Open Big Data Repositories organizing and participating in different datathons around the world in numerous roles:

- Sao Paulo Datathon (May 2017). *Data mentor, judging committee member.*
- Taipei Datathon (Oct. 2017). *Data mentor, judging committee member.*
- Madrid Datathon (Dec 2017). *Speaker, Data mentor, Judging committee member, Logistic Manager, Marketing/Design Manager, Webpage Developer.*
- Buenos Aires Datathon (May 2018). *Workshop mentor.*
- Tarragona Datathon, Spain (Nov 2018): *Data mentor and speaker.*
- New York University Langone Health Tech Datathon (May 2019): *Data mentor.*
- Bogotá Datathon (Dec 2019), *workshop mentor*: 'Introduction to modelling vital signs and lab results to solve complex problems and improve decision making of patients and providers'.
- Beth Israel Deaconess Medical Center Datathon, Boston 2020: *Data mentor.*

This work allowed the publication of two manuscripts so far, one of them being a white paper on the first Brazilian critical care datathon and the other one addressing the importance of team effectiveness and affective learning. Most importantly it helped to raise awareness around the world about the importance of open data, transparent algorithms and collaborative research.

Furthermore, we were able to carry out a review on how the Integration and management of Critical Inpatients' Open Big Data Repositories can help to boost the research around this kind of patients if we work side by side with members of the intensive care units addressing their unmet needs and focusing on bringing better value to their patients.

This work allowed the publication of two manuscripts so far, while we were able to learn about the numerous techniques that can boost new knowledge discovery at the bedside both in the Intensive Care Unit and in the Operating Room.

Last but not most important we addressed specific clinical questions in the two departments of the hospitals where we focused our efforts:

- **Intensive Care Unit:** we helped to gather more knowledge about the obesity paradox, we study and improved the way severity scores are developed while providing open access to our exceptionally performing score for decision support, we studied how critical care outcomes differ throughout the United States geography, we developed a model for severe tachypnea risk in terminal patients, we address admission disorientation and risk of code status change, we studied how GCS mediates the baseline sodium effect when predicting mortality, finally we studied the association of chloride ion and sodium-chloride difference with acute kidney injury and mortality in critically ill patients.
- **Operating room:** we addressed the association of intraoperative hypotension with postoperative stroke, acute kidney injury and death in patients having cardiac surgery with cardiopulmonary bypass. In a more analytically advanced version of this project, we trained Machine Learning models with preoperative risk factors and intraoperative hypotension parameters, so we were able to predict mortality following cardiac surgery.

This work allowed the publication of eight manuscripts so far, but most importantly allowed us and our collaborating physicians to broaden our knowledge about critically ill patients and their treatment optimization. It also helped us to acquire excellent data analysis skills that aid us to tackle a wide variety of clinically relevant questions.

XVII.2.2. Specific Hypothesis

XVII.2.2.1. Social aspects: Promotion of Critical Inpatients' Open Big Data Repositories

H#1: We hypothesized that a datathon promotes effective learning and effective teamwork. This hypothesis was addressed in chapter V: *Assessing team effectiveness and affective learning in a datathon*.

H#2: We hypothesized that effective teamwork during a datathon is associated with affective learning. This hypothesis was addressed in chapter V: *Assessing team effectiveness and affective learning in a datathon*.

Hypotheses confirmed

Team effectiveness was highly rated by participants during a datathon event in São Paulo, Brazil. Datathon may provide an environment that promotes affective learning and effective teamwork. Successful teams are associated with higher affective learning gains. Effective leadership is a key component to integrate teamwork with affective learning.

XVII.2.2.2. Technological aspects: Integration and management of Critical Inpatients' Open Big Data Repositories

H#3: We hypothesized that using data governance, quality check, quality assurance, normalization, standardization and de-identification techniques performed on the repositories the data we are going to analyze will meet the desired quality standards, standardization and anonymization requirements. This hypothesis was addressed in chapter VII: Big Data Analysis and Machine Learning in Intensive Care Units.

Hypothesis confirmed

Big Data Analysis and Machine Learning tools offer a great opportunity for improving the strategic management of our Units, the handling of concrete clinical cases, and clinical research. However, in order to take advantage of this new methodology, we need to evolve and incorporate new human resources (specialized staff with clinical knowledge and training in AI) and technological assets. Furthermore, we must be able to harmonize the requirements referred to patient data privacy and security with the possibility of using large clinical databases efficiently. We are convinced that an international and multidisciplinary team working collectively to draw knowledge from large bodies of clinical data, as explained in the course of this article, may bring a revolution to the modern practice of critical patient care.

H#4: Intensive care is an ideal environment for the use of Big Data Analysis (BDA) and Machine Learning (ML), due to the huge amount of information processed and stored in electronic format in relation to such care. We hypothesized these tools can improve our clinical research capabilities and clinical decision making in the future. This hypothesis was addressed in chapter VI: *Big Data and Machine Learning in Critical Care: Opportunities for Collaborative Research*.

Hypothesis confirmed

We are conceivably at the dawn of a new era in intensive care, in which clinical decision making will increasingly be assisted by computers that perform data integration and analysis. The clinician will therefore have to navigate a specialty of critical care, that harnesses the power of data to individualize care in order to improve population health, where collaborative work with other non-healthcare specialists in the area of data science is essential to leverage all the information that is routinely collected in the process of care, but without compromising patient rights to privacy.

XVII.2.2.3. Clinical aspects: Processing of Critical Inpatients' Open Big Data Repositories

H#5: We hypothesized that within the same severity of illness category, obese patients may have inherently different degrees of deviation from their usual state during critical illness. This hypothesis was addressed in chapter VIII: *Severity of Illness Scores May Misclassify Critically Ill Obese Patients*.

Hypothesis confirmed

Our findings suggest that for the same severity of illness scores, obese and normal weight patients have different deviations of laboratory markers from baseline. This could potentially lead to severity of illness score misclassification. We encourage others to develop severity of illness scores based on deviation from usual physiologic state in order to maximize the personalization of the scores and minimize the impact of uncontrollable local factors external to the patient.

H#6: We hypothesized that, while high-risk patients in ICU cohorts represent a distinct and identifiable group of patients, prior severity scores have been built with methods that are too simple and inflexible to learn to forecast mortality accurately across such heterogeneous groups; therefore a more efficient severity score can be built. This hypothesis was addressed in chapter IX: *Developing well-calibrated illness severity scores for decision support in the critically ill*.

Hypothesis confirmed

As healthcare continues to digitize and EHR data continues to proliferate, we expect that hospitals will increasingly craft bespoke models in order to quantify the probability of outcomes relevant to clinical decision-making. In the ICU setting, the quantification of disease severity and the risk of hospital mortality can optimize resource allocation; motivate goals of care conversations; and allow clinicians, patients, and families prognostic insight. Because of the wide spectrum of disease severity present in ICU cohorts, we believe strategies for adequately calibrating models across the risk spectrum are necessary. We demonstrate that a sequential approach combining two linear models is a valid strategy but provides only a modest improvement, and further work is needed to ultimately deliver models that accurately quantify risk in every ICU patient.

H#7: We hypothesized that there is a difference between hospitals which predominantly care for minority patients and non-minority hospitals regarding improvements in critical care outcomes. This hypothesis was addressed in chapter X: *Temporal trends in critical illness outcomes in United States minority serving hospitals*.

Hypothesis confirmed

A large proportion of minority patients receive critical illness care in a small number of minority-serving hospitals, which over the last ten years, have not enjoyed the steady decrease in mortality and length of stay that non-minority hospitals have. Whether this reflects a more systemic disparity, whereby African Americans are more medically disadvantaged upon presentation, or differences in hospital care and resources, is not known, but regardless, this observation highlights the profound obstacles facing minorities and minority-serving hospitals.

H#8: We hypothesized that administering analgesia/sedation pre-extubation (anticipatory dosing) relieves distress more effectively than giving these drugs only in response to observed symptoms. This hypothesis was addressed in chapter XI: *Incidence and Risk Model Development for Severe Tachypnea Following Terminal Extubation*

Hypothesis confirmed

The prevalence of poorly controlled symptoms as measured by severe tachypnea during PVW is common. The risk of severe tachypnea may be reduced with anticipatory dosing of opiates. Patients who are male, with higher levels of consciousness, or have more severe lung injury appear to be at greater risk for severe tachypnea. Further prospective study using standardized measurements of dyspnea, and interventions to improve symptom control during PVW should be directed at these risk groups.

H#9: While there are many factors that enter into the decision to de-escalate intensity of critical illness care, the influence of mental status has not been previously directly measured. We hypothesized that disoriented patients were more likely to have code status change to Do not resuscitate/Intubate (DNR/I) or other type of care-de-escalation. This hypothesis was addressed in chapter XII. Admission Disorientation and Risk of Code Status Change.

Hypothesis confirmed

In conclusion, in models that adjusted for severity of illness, disoriented patients are more likely to have a code status changed to DNR, DNI, or other form of care de-escalation, than oriented patients. How patient disorientation affects the myriad pathways of medical decision making, and what are the safeguards protecting disoriented patients from engaging in goals of care discussions during critical illness, requires further study.

H#10: Although disorders of water balance have been associated with increased mortality, the mechanisms explaining the association are not clear. We hypothesized that decreased consciousness is a mediator on the association between dysnatremia and mortality. This hypothesis was addressed in chapter XIII: Altered Consciousness Mediates the Effect of Hyponatremia, but not Hyponatremia, on Mortality.

Hypothesis confirmed

The absence of a mediating effect of altered consciousness on the association of hyponatremia and mortality suggest that pathogenic mechanisms that lead to water excess, rather than water excess per se, are in part responsible for poor outcomes seen across hyponatremic disorders.

H#11: Intra Operative Hypotension (IOH) (MAP <65 mmHg) even for only >10 minutes during CPB has been shown to increase the risk of stroke. Vasopressors and inotropes are used to treat IOH during cardiac surgery. We believe there is an association between IOH and intraoperative vasopressor/inotrope use with postoperative outcomes.

Hypothesis confirmed

The 4th quartiles of total IOH duration, AUC-MAP <65mmHg, and vasopressor-inotrope dose were significantly associated with the composite primary outcome. Reducing IOH duration during CPB may reduce the incidence of stroke.

H#12: We hypothesized that incorporating adverse intraoperative factors that include Intraoperative hypotension (IOH) during and outside Cardiopulmonary Bypass (CPB) along with the preoperative risk factors into the machine learning models such as eXtreme gradient boosting or Random Forests will improve discrimination, specificity, sensitivity and positive predictive value for mortality prediction following cardiac surgery.

Hypothesis confirmed

XGB machine learning model from IOH outside-CPB phase seems to offer a better discrimination, sensitivity, specificity, and PPV compared to other models. Machine learning models incorporating intraoperative adverse factors might offer better predictive ability for risk stratification and triaging of patients following cardiac surgery.

H#13: Derangements of chloride ion concentration ($[Cl^-]$) have been shown to be associated with acute kidney injury (AKI) and other adverse outcomes. For a physicochemical approach, however, $[Cl^-]$ should be considered with sodium ion concentration ($[Na^+]$). We hypothesized there is an association of $[Cl^-]$ and the main strong ion difference ($mSID = [Na^+] - [Cl^-]$) during the first 24 hours after admission into intensive care unit (ICU) with the development of AKI and mortality.

Hypothesis confirmed

$mSID$ measured on ICU presentation to the ICU predicts AKI within 7 days, with low and in particular high values representing increased risk. The association between chloride levels and AKI are statistically insignificant in models incorporating $mSID$, suggesting $mSID$ is a better predictive marker than chloride on ICU admission.

XVII.3. Future Works

XVII.3.1. Social aspects: Promotion of Critical Inpatients' Open Big Data Repositories

XVII.3.1.1. Assessing team effectiveness and affective learning in a datathon

Since real life engagement behavior, the primary affective learning outcome due to its higher-order learning (behavioral commitment) was modestly associated with another affective learning variable: enrollment in a similar event in the future ($r = 0.28$, $p < 0.05$) this confirms our suspect that datathons can

Health datathons keep becoming popular worldwide. We plan to keep celebrating datathons since future events of this kind will help further explorations and more reliable investigations of associations such as those between effective leadership, team success and individual affective learning which merit further research in both real world settings and simulations including hackathons and datathons (Liebowitz & de Meuse, 1982; Meuse, 1992; Tannenbaum et al., 1992).

XVII.3.2. Technological aspects: Integration and management of Critical Inpatients' Open Big Data Repositories

XVII.3.2.1. Big data and machine learning in critical care: Opportunities for collaborative research

For future works, we encourage all those interested in participating in this project to contact the authors in order to establish a national collaborative network in the field of BDA and ML as applied to healthcare data collected in the ICU, under the auspices of SEMICYUC, we will be providing both research and engineering support for those interested in joining the consortium.

XVII.3.2.2. Big data analysis and machine learning in intensive care units

To understand this by means of an analogy, it is as if the system were running a continuous checklist with our patients. In addition, the current artificial intelligence (AI) techniques allow the use of experience gained with previous patients in the form of structured data for the assessment of the next individual patient. In future, this may have a great impact upon clinical practice.

Another example is the use of real time control systems, for example to continuously adjust the insulin perfusion dose of a patient according to the insulin sensitivity he or she has shown previously; stress condition assessed from different data of the plots and laboratory tests; and the caloric and carbohydrate supply being provided in that moment. Over time, the system can learn about the patient and gradually optimize glycemic control within concrete safety parameters.

The extraction of concepts from the text using tools such as cTAKES (Apache Software Foundation, 2018) is quite well developed in the Anglo-Saxon world, though there are few practical options for texts not written in English. The research group in Data Mining of the Biomedical Technology Center of the Universidad Politécnica de Madrid (Spain) is working with a tool derived from the cTAKES known as TIDA, (Costumero et al., 2014) and in future it is possible that systems of this kind will become part of our computer-based resources.

Investigators must work jointly with the local Ethics Committees to harmonize the safeguarding of data privacy and security with the advances in clinical research which the BDA tools can offer us in future. Mechanisms must be developed to facilitate patient consent (or withdrawal of consent) to the use of the information generated during healthcare, conveniently harmonized for clinical research. Another important issue is data property. In this case we are not dealing with privacy or security questions but with the intrinsic value of the information (even regarded as a commercial item).

XVII.3.3. Clinical aspects: Processing of Critical Inpatients' Open Big Data Repositories

XVII.3.3.1. Severity of illness scores may misclassify critically ill obese patients

Since our analysis only compared normal weight individuals to obese individuals, we may not be able to extrapolate our findings to overweight individuals, therefore in future work we plan to address overweight individuals as well.

Another topic for future study is whether obese patients behave, with respect to clinical outcome, like other patient types with similarly higher scores, or have better outcomes than those scores would have indicated, this article represents a starting point in this regard where such differences are duly noted and analyzed to form a basis for such studies.

In fact, this kind of unavoidable heterogeneity in prior conditions and physiological states is not unique to the issue of obesity and poses a general problem in the formulation of valid and reliable scoring systems. APACHE deals with this issue at a high level by assigning weights to selected historical conditions. Future directions include examining for differences in deviations of physiological parameters such as heart rate, blood pressure, or partial pressure of oxygen (Pao₂), which are included in SOFA and SAPS-II. As with previous studies that demonstrated differences in cardiac autonomic activity and heart rate (Yadav et al., 2017), and respiratory physiology (Parameswaran et al., 2006; Salome et al., 2010), between obese and normal weight individuals, it may well be that deviations in such parameters are also different in obese and normal weight patients with similar severity of illness scores.

XVII.3.3.2.   Developing well-calibrated illness severity scores for decision support in the critically ill

With the known shortcomings of previous approaches to severity score development, we hope to inform future model development strategies so as to ensure accurate risk quantification across the board in critically ill patients as the electronic health record (EHR) continues to rise to the unprecedented abundance of highly granular clinical data it has made available so far.

In future works the cutoff point or points should be chosen by means of hyperparameter tuning via a Cross Validation procedure in a data-driven manner since we used arbitrary cutoffs for high risk of 0.10 and 0.50. Selection bias also represents an important shortcoming: a significant limitation of this study was the number of patients for whom APACHE data were not present and/or an APACHE IVa score was not calculated, this patients were finally excluded so if in its new version eICU database provides more patients with apache score, in future works we plan to repeat the analysis using these.

XVII.3.3.3.   Association of Intraoperative Hypotension with Postoperative Stroke, Acute Kidney Injury, and Death in Patients Having Cardiac Surgery with Cardiopulmonary Bypass

Larger databases from multiple centers can offer further confirmation and specific insights into the relative contributions of perioperative factors for postoperative MAE. Differential effects of IOH during or outside the CPB phase merits further exploration.

XVII.3.3.4.   Machine learning models with preoperative risk factors and intraoperative hypotension parameters predict mortality following cardiac surgery

Variability among anesthesiologists for the incidence of hypotension during cardiac surgery was not explored in this work where we found that the threshold below 65 mm of Hg was predictive of postoperative outcomes. One of the future projects we plan to conduct is to study the variability among and within anesthesia providers with respect to incidence of hypotension using mixed effect models, furthermore, since our models considered time points as independent features, as a future work we propose to use other techniques that are able to incorporate time series such as long short-term memory neural networks. Finally, we couldn't do external validation for our models, so we plan to validate our results using data coming from a different hospital in the future.

XVII.3.3.5.   Association of Chloride Ion and Sodium-Chloride Difference with Acute Kidney Injury and Mortality in Critically Ill Patients

The use of mSID, while clinically appealing due to its simplicity, is limited in its ability to allow inference about the underlying physiology. Our study raises additional questions about the relationship between mSID, acid-base status, and clinical outcomes, which will require further investigation to better answer. Finally, mSID deviations from normal did not appear to correlate with changes in serum bicarbonate alone, a finding which merits further exploration.

REFERENCES

- Abarshi, E. A., Papavasiliou, E. S., Preston, N., Brown, J., & Payne, S. (2014). The Complexity of Nurses' Attitudes and Practice of Sedation at the End of Life: A Systematic Literature Review. *Journal of Pain and Symptom Management*, 47(5), 915-925.e11. <https://doi.org/10.1016/j.jpainsymman.2013.06.011>
- Abhyankar, S., Leishear, K., Callaghan, F. M., Demner-Fushman, D., & McDonald, C. J. (2012). Lower short- and long-term mortality associated with overweight and obesity in a large cohort study of adult intensive care unit patients. *Critical Care*, 16(6), R235. <https://doi.org/10.1186/cc11903>
- Aboab, J., Celi, L. A., Charlton, P., Feng, M., Ghassemi, M., Marshall, D. C., Mayaud, L., Naumann, T., McCague, N., Paik, K. E., Pollard, T. J., Resche-Rigon, M., Saliccioli, J. D., & Stone, D. J. (2016). A "datathon" model to support cross-disciplinary collaboration. *Science Translational Medicine*, 8(333), 333ps8. <https://doi.org/10.1126/scitranslmed.aad9072>
- Aboelsoud, M., Siddique, O., Morales, A., Seol, Y., & Al-Qadi, M. (2018). Early biliary drainage is associated with favourable outcomes in critically-ill patients with acute cholangitis. *Gastroenterology Review*, 13(1), 16–21. <https://doi.org/10.5114/pg.2018.74557>
- Alba, A. C. (2017). Discrimination and calibration of clinical prediction models: Users' guides to the medical literature. *JAMA*, 318. <https://doi.org/10.1001/jama.2017.12126>
- Allyn, J., Allou, N., Augustin, P., Philip, I., Martinet, O., Belghiti, M., Provenchere, S., Montravers, P., & Ferdynus, C. (2017). A Comparison of a Machine Learning Model with EuroSCORE II in Predicting Mortality after Elective Cardiac Surgery: A Decision Curve Analysis. *PLOS ONE*, 12(1), e0169772. <https://doi.org/10.1371/journal.pone.0169772>
- Al-Sarraf, N., Thalib, L., Hughes, A., Houlihan, M., Tolan, M., Young, V., & McGovern, E. (2011). Cross-clamp time is an independent predictor of mortality and morbidity in low- and high-risk cardiac patients. *International Journal of Surgery*, 9(1), 104–109. <https://doi.org/10.1016/j.ijsu.2010.10.007>
- Andersen, J. F. (1979). Teacher Immediacy as a Predictor of Teaching Effectiveness. *Annals of the International Communication Association*, 3(1), 543–559. <https://doi.org/10.1080/23808985.1979.11923782>
- Angus, D. C., Barnato, A. E., Linde-Zwirble, W. T., Weissfeld, L. A., Watson, R. S., Rickert, T., & Rubenfeld, G. D. (2004). Use of intensive care at the end of life in the United States: An epidemiologic study*. *Critical Care Medicine*, 32(3), 638–643. <https://doi.org/10.1097/01.ccm.0000114816.62331.08>
- Ankrom, M., Zelesnick, L., Barofsky, I., Georas, S., Finucane, T. E., & Greenough, W. B. (2001). Elective Discontinuation of Life-Sustaining Mechanical Ventilation on a Chronic Ventilator Unit. *Journal of the American Geriatrics Society*, 49(11), 1549–1554. <https://doi.org/10.1046/j.1532-5415.2001.4911252.x>
- Apache Software Foundation. (2018). *Apache cTAKES™*. <http://ctakes.apache.org/>.
- Arampatzis, S., Frauchiger, B., Fiedler, G.-M., Leichtle, A. B., Buhl, D., Schwarz, C., Funk, G.-C., Zimmermann, H., Exadaktylos, A. K., & Lindner, G. (2012). Characteristics, Symptoms, and Outcome of Severe Dysnatremias Present on Hospital Admission. *The American Journal of Medicine*, 125(11), 1125.e1-1125.e7. <https://doi.org/10.1016/j.amjmed.2012.04.041>
- Arndt, B. G., Beasley, J. W., Watkinson, M. D., Temte, J. L., Tuan, W.-J., Sinsky, C. A., & Gilchrist, V. J. (2017). Tethered to the EHR: Primary Care Physician Workload Assessment Using EHR Event Log Data and Time-Motion Observations. *The Annals of Family Medicine*, 15(5), 419–426. <https://doi.org/10.1370/afm.2121>
- Arora, S. K. (2011). Hypernatremic Disorders in the Intensive Care Unit. *Journal of Intensive Care Medicine*, 28(1), 37–45. <https://doi.org/10.1177/0885066611403994>
- Aslakson, R. A., Reinke, L. F., Cox, C., Kross, E. K., Benzo, R. P., & Curtis, J. R. (2017). Developing a Research Agenda for Integrating Palliative Care into Critical Care and Pulmonary Practice To Improve Patient and Family Outcomes. *Journal of Palliative Medicine*, 20(4), 329–343. <https://doi.org/10.1089/jpm.2016.0567>
- Aubert, B. A., & Kelsey, B. L. (2003). Further Understanding of Trust and Performance in Virtual Teams. *Small Group Research*, 34(5), 575–618. <https://doi.org/10.1177/1046496403256011>

- Ayanian, J. Z., Cleary, P. D., Weissman, J. S., & Epstein, A. M. (1999). The Effect of Patients' Preferences on Racial Differences in Access to Renal Transplantation. *New England Journal of Medicine*, 341(22), 1661–1669. <https://doi.org/10.1056/nejm199911253412206>
- Baalaaji, M., Jayashree, M., Nallasamy, K., Singhi, S., & Bansal, A. (2018). Predictors and Outcome of Acute Kidney Injury in Children with Diabetic Ketoacidosis. *Indian Pediatrics*, 55(4), 311–314. <https://doi.org/10.1007/s13312-018-1274-8>
- Bach, P. B., Pham, H. H., Schrag, D., Tate, R. C., & Hargraves, J. L. (2004). Primary Care Physicians Who Treat Blacks and Whites. *New England Journal of Medicine*, 351(6), 575–584. <https://doi.org/10.1056/nejmsa040609>
- Badawi, O., Brennan, T., Celi, L. A., Feng, M., Ghassemi, M., Ippolito, A., Johnson, A., Mark, R. G., Mayaud, L., Moody, G., Moses, C., Naumann, T., Nikore, V., Pimentel, M., Pollard, T. J., Santos, M., Stone, D. J., & and, A. Z. (2014). Making Big Data Useful for Health Care: A Summary of the Inaugural MIT Critical Data Conference. *JMIR Medical Informatics*, 2(2), e22. <https://doi.org/10.2196/medinform.3447>
- Baldwin, T. T., BEDELL, M. D., & JOHNSON, J. L. (1997). The Social Fabric Of A Team-based M.b.a. Program: Network Effects On Student Satisfaction And Performance. *Academy of Management Journal*, 40(6), 1369–1397. <https://doi.org/10.2307/257037>
- Banzett, R. B., Pedersen, S. H., Schwartzstein, R. M., & Lansing, R. W. (2008). The Affective Dimension of Laboratory Dyspnea. *American Journal of Respiratory and Critical Care Medicine*, 177(12), 1384–1390. <https://doi.org/10.1164/rccm.200711-1675oc>
- Barili, F., Pacini, D., Capo, A., Rasovic, O., Grossi, C., Alamanni, F., Di Bartolomeo, R., & Parolari, A. (2012). Does EuroSCORE II perform better than its original versions? A multicentre validation study. *European Heart Journal*, 34(1), 22–29. <https://doi.org/10.1093/eurheartj/ehs342>
- Bartels, K., Karhausen, J., Clambey, E. T., Grenz, A., & Eltzschig, H. K. (2013). Perioperative Organ Injury. *Anesthesiology*, 119(6), 1474–1489. <https://doi.org/10.1097/aln.0000000000000022>
- Beam, A. L., & Kohane, I. S. (2018). Big data and machine learning in health care. *JAMA*, 319. <https://doi.org/10.1001/jama.2017.18391>
- Beck, D. H., Smith, G. B., & Taylor, B. L. (2002). The impact of low-risk intensive care unit admissions on mortality probabilities by SAPS II, APACHE II and APACHE III. *Anaesthesia*, 57. <https://doi.org/10.1046/j.1365-2044.2002.02362.x>
- Benson, T. (2012). HL7 Version 2. In *Principles of Health Interoperability HL7 and SNOMED* (pp. 101–119). Springer London. https://doi.org/10.1007/978-1-4471-2801-4_7
- Berend, K., van Hulsteijn, L. H., & Gans, R. O. B. (2012). Chloride: The queen of electrolytes? *European Journal of Internal Medicine*, 23(3), 203–211. <https://doi.org/10.1016/j.ejim.2011.11.013>
- Bertsimas, D., Dunn, J., Velmahos, G. C., & Kaafarani, H. M. A. (2018). Surgical Risk Is Not Linear. *Annals of Surgery*, 268(4), 574–583. <https://doi.org/10.1097/sla.0000000000002956>
- Bhattacharyya, S. B. (2015). SNOMED CT Basics. In *Introduction to SNOMED CT* (pp. 25–60). Springer Singapore. https://doi.org/10.1007/978-981-287-895-3_4
- Bijker, J. B., Persoon, S., Peelen, L. M., Moons, K. G. M., Kalkman, C. J., Kappelle, L. J., & van Klei, W. A. (2012). Intraoperative Hypotension and Perioperative Ischemic Stroke after General Surgery. *Anesthesiology*, 116(3), 658–664. <https://doi.org/10.1097/aln.0b013e3182472320>
- Bijker, J. B., van Klei, W. A., Vergouwe, Y., Eleveld, D. J., van Wolfswinkel, L., Moons, K. G. M., & Kalkman, C. J. (2009). Intraoperative Hypotension and 1-Year Mortality after Noncardiac Surgery. *Anesthesiology*, 111(6), 1217–1226. <https://doi.org/10.1097/aln.0b013e3181c14930>
- Billings, J. A. (2012). Humane terminal extubation reconsidered. *Critical Care Medicine*, 40(2), 625–630. <https://doi.org/10.1097/ccm.0b013e318228235d>
- Bime, C., Poongkunran, C., Borgstrom, M., Natt, B., Desai, H., Parthasarathy, S., & Garcia, J. G. N. (2016). Racial Differences in Mortality from Severe Acute Respiratory Failure in the United States, 2008–2012. *Annals of the American Thoracic Society*, 13(12), 2184–2189. <https://doi.org/10.1513/annalsats.201605-359oc>

- Binks, A. P., Desjardin, S., & Riker, R. (2017). ICU Clinicians Underestimate Breathing Discomfort in Ventilated Subjects. *Respiratory Care*, 62(2), 150–155. <https://doi.org/10.4187/respcare.04927>
- Bloom, Krathwohl, & Masia. (1956). Appendix B: Taxonomy of educational objectives. *AAHE-ERIC/Higher Education Research Report*, 3(1), 59–60. <https://doi.org/10.1002/aehe.3640030109>
- Bodí, M., Blanch, L., & Maspons, R. (2017). Clinical information systems: An opportunity to measure value, investigate and innovate from the real world. *Medicina Intensiva (English Edition)*, 41(5), 316–318. <https://doi.org/10.1016/j.medine.2017.04.002>
- Boniatti, M. M., Cardoso, P. R. C., Castilho, R. K., & Vieira, S. R. R. (2011). Is hyperchloremia associated with mortality in critically ill patients? A prospective cohort study. *Journal of Critical Care*, 26(2), 175–179. <https://doi.org/10.1016/j.jcrc.2010.04.013>
- Brasil. Governo Federal. (2016). *Economia e emprego*. <http://bit.ly/2gJIB0v>
- Breiman, L. (1997). *Arcing the Edge*. <https://statistics.berkeley.edu/tech-reports/486>
- Breitbart, W., Gibson, C., & Tremblay, A. (2002). The Delirium Experience: Delirium Recall and Delirium-Related Distress in Hospitalized Patients With Cancer, Their Spouses/Caregivers, and Their Nurses. *Psychosomatics*, 43(3), 183–194. <https://doi.org/10.1176/appi.psy.43.3.183>
- Breslow, M. J., & Badawi, O. (2012a). Severity scoring in the critically ill: Part 1—interpretation and accuracy of outcome prediction scoring systems. *Chest*, 141. <https://doi.org/10.1378/chest.11-0330>
- Breslow, M. J., & Badawi, O. (2012b). Severity scoring in the critically ill: Part 2: Maximizing value from outcome prediction scoring systems. *Chest*, 141. <https://doi.org/10.1378/chest.11-0331>
- Brinkkemper, T., van Norel, A. M., Szadek, K. M., Loer, S. A., Zuurmond, W. W., & Perez, R. S. (2011). The use of observational scales to monitor symptom control and depth of sedation in patients requiring palliative sedation: A systematic review. *Palliative Medicine*, 27(1), 54–67. <https://doi.org/10.1177/0269216311425421>
- Brownlee, J. (2016). *Boosting and AdaBoost for Machine Learning*. <https://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning>.
- Buurman, J., Liu, P., Peters, J. F., Chang, P. J., & Sevenster, M. (2015). Natural Language Processing Techniques for Extracting and Categorizing Finding Measurements in Narrative Radiology Reports. *Applied Clinical Informatics*, 06(03), 600–610. <https://doi.org/10.4338/aci-2014-11-ra-0110>
- Calster, B., & Vickers, A. J. (2015). Calibration of risk prediction models: Impact on decision-analytic performance. *Med. Decis. Making*, 35. <https://doi.org/10.1177/0272989X14547233>
- Calvert, J., Mao, Q., Hoffman, J. L., Jay, M., Desautels, T., Mohamadlou, H., Chettipally, U., & Das, R. (2016). Using electronic health record collected clinical variables to predict medical intensive care unit mortality. *Annals of Medicine and Surgery*, 11, 52–57. <https://doi.org/10.1016/j.amsu.2016.09.002>
- Cambridge, U. (2018). *NIHR Health Informatics Collaborative*. Available from: <http://www.clinicalinformatics.group.cam.ac.uk/hic/intensive-care/>.
- Campbell, M. L. (2007). How to Withdraw Mechanical Ventilation. *AACN Advanced Critical Care*, 18(4), 397–403. <https://doi.org/10.1097/01.aacn.0000298631.53276.f7>
- Campbell, M. L. (2008). Respiratory distress: A model of responses and behaviors to an asphyxial threat for patients who are unable to self-report. *Heart & Lung*, 37(1), 54–60. <https://doi.org/10.1016/j.hrtlng.2007.05.007>
- Campbell, M. L. (2012). Dyspnea prevalence, trajectories, and measurement in critical care and at life's end. *Current Opinion in Supportive and Palliative Care*, 6(2), 168–171. <https://doi.org/10.1097/spc.0b013e328352b67f>
- Campbell, M. L., Bizek, K. S., & Thill, M. (1999). Patient responses during rapid terminal weaning from mechanical ventilation. *Critical Care Medicine*, 27(1), 73–77. <https://doi.org/10.1097/00003246-199901000-00029>
- Campbell, M. L., Templin, T., & Walch, J. (2010). A Respiratory Distress Observation Scale for Patients Unable To Self-Report Dyspnea. *Journal of Palliative Medicine*, 13(3), 285–290. <https://doi.org/10.1089/jpm.2009.0229>
- Campbell, M. L., Yarandi, H. N., & Mendez, M. (2015). A Two-Group Trial of a Terminal Ventilator Withdrawal Algorithm: Pilot Testing. *Journal of Palliative Medicine*, 18(9), 781–785. <https://doi.org/10.1089/jpm.2015.0111>

- CARLET, J., THIJS, L., ANTONELLI, M., CASSELL, J., COX, P., HILL, N., HINDS, C., PIMENTEL, J., REINHART, K., & THOMPSON, B. (2004). Statement of the 5th International Consensus Conference in Critical Care Challenges in end-of-life care in the ICU☆Brussels, Belgium, April 2003. *Réanimation*, 13(6–7), 440–452. [https://doi.org/10.1016/s1624-0693\(04\)00133-1](https://doi.org/10.1016/s1624-0693(04)00133-1)
- Carlos, R. C., Kahn, C. E., & Halabi, S. (2018). Data Science: Big Data, Machine Learning, and Artificial Intelligence. *Journal of the American College of Radiology*, 15(3), 497–498. <https://doi.org/10.1016/j.jacr.2018.01.029>
- Carson, S. S., Cox, C. E., Wallenstein, S., Hanson, L. C., Danis, M., Tulskey, J. A., Chai, E., & Nelson, J. E. (2016). Effect of Palliative Care–Led Meetings for Families of Patients With Chronic Critical Illness. *JAMA*, 316(1), 51. <https://doi.org/10.1001/jama.2016.8474>
- Cartwright, C. R., & Mora Mangano, C. (1998). Con: During cardiopulmonary bypass for elective coronary artery bypass grafting, perfusion pressure should not routinely be greater than 70 mmHg. *Journal of Cardiothoracic and Vascular Anesthesia*, 12(3), 361–364. [https://doi.org/10.1016/s1053-0770\(98\)90023-4](https://doi.org/10.1016/s1053-0770(98)90023-4)
- Celi, Leo A., Csete, M., & Stone, D. (2014). Optimal data systems. *Current Opinion in Critical Care*, 20(5), 573–580. <https://doi.org/10.1097/mcc.0000000000000137>
- Celi, Leo Anthony, Mark, R. G., Stone, D. J., & Montgomery, R. A. (2013). “Big Data” in the Intensive Care Unit. Closing the Data Loop. *American Journal of Respiratory and Critical Care Medicine*, 187(11), 1157–1160. <https://doi.org/10.1164/rccm.201212-2311ed>
- Chalfin, D. B., Trzeciak, S., Likourezos, A., Baumann, B. M., & Dellinger, R. P. (2007). Impact of delayed transfer of critically ill patients from the emergency department to the intensive care unit†ast. *Critical Care Medicine*, 35(6), 1477–1483. <https://doi.org/10.1097/01.ccm.0000266585.74905.5a>
- Chan, J. D., Treece, P. D., Engelberg, R. A., Crowley, L., Rubenfeld, G. D., Steinberg, K. P., & Curtis, J. R. (2004). Narcotic and Benzodiazepine Use After Withdrawal of Life Support. *Chest*, 126(1), 286–293. [https://doi.org/10.1016/s0012-3692\(15\)32925-1](https://doi.org/10.1016/s0012-3692(15)32925-1)
- Charlson, M. E., Pompei, P., Ales, K. L., & MacKenzie, C. R. (1987). A new method of classifying prognostic comorbidity in longitudinal studies: Development and validation. *Journal of Chronic Diseases*, 40(5), 373–383. [https://doi.org/10.1016/0021-9681\(87\)90171-8](https://doi.org/10.1016/0021-9681(87)90171-8)
- Chawla, A., Sterns, R. H., Nigwekar, S. U., & Cappuccio, J. D. (2011). Mortality and Serum Sodium: Do Patients Die from or with Hyponatremia? *Clinical Journal of the American Society of Nephrology*, 6(5), 960–965. <https://doi.org/10.2215/cjn.10101110>
- Chen, T., & Guestrin, C. (2016). *XGBoost: A scalable tree boosting system*. In *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ACM, San Francisco, CA)*. 785–794.
- Christodoulou, E. (2019). A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. *J. Clin. Epidemiol.*, 110. <https://doi.org/10.1016/j.jclinepi.2019.02.004>
- Chung, M.-C., Yu, T.-M., Shu, K.-H., Wu, M.-J., Chang, C.-H., Muo, C.-H., & Chung, C.-J. (2017). Hyponatremia and increased risk of dementia: A population-based retrospective cohort study. *PLOS ONE*, 12(6), e0178977. <https://doi.org/10.1371/journal.pone.0178977>
- Churpek, M. M., Yuen, T. C., & Edelson, D. P. (2013). Delayed Intensive Care Unit Transfer Is Associated With Increased Mortality In Ward Patients. In *C24. QUALITY IMPROVEMENT INITIATIVES IN PULMONARY AND CRITICAL CARE* (Vol. 1–276, pp. A6074–A6074). American Thoracic Society. https://doi.org/10.1164/ajrccm-conference.2013.187.1_MeetingAbstracts.A6074
- Collins, G. S., Reitsma, J. B., Altman, D. G., & Moons, K. G. M. (2015). Transparent Reporting of a multivariable prediction model for Individual Prognosis Or Diagnosis (TRIPOD): The TRIPOD Statement. *Ann. Intern. Med.*, 162. <https://doi.org/10.7326/M14-0697>
- Commereuc, M., Nevoret, C., Radermacher, P., Katsahian, S., Asfar, P., & Schortgen, F. (2019). Hyperchloremia is not associated with AKI or death in septic shock patients: Results of a post hoc analysis of the “HYPER2S” trial. *Annals of Intensive Care*, 9(1). <https://doi.org/10.1186/s13613-019-0570-3>
- Computing NCfB. (2018). *I2b2 (Informatics for Integrating Biology and the Bedside)*. <https://www.i2b2.org>.
- Connors, A. F., Dawson, N. V., Desbiens, N. A., Fulkerson, W. J., Goldman, L., Knaus, W. A., Lynn, J., Oye, R. K., Bergner, M., Damiano, A., Hakim, R., Murphy, D. J., Teno, J., Virnig, B., Wagner, D. P., Wu, A. W., Yasui, Y.,

- Robinson, D. K., Kreling, B., ... Ransohoff, D. (1995). A Controlled Trial to Improve Care for Seriously Ill Hospitalized Patients: The Study to Understand Prognoses and Preferences for Outcomes and Risks of Treatments (SUPPORT). *JAMA*, 274(20), 1591–1598. <https://doi.org/10.1001/jama.1995.03530200027032>
- Cook, D., & Rocker, G. (2014). Dying with Dignity in the Intensive Care Unit. *New England Journal of Medicine*, 370(26), 2506–2514. <https://doi.org/10.1056/nejmra1208795>
- Cook, D., Rocker, G., Marshall, J., Sjøkvist, P., Dodek, P., Griffith, L., Freitag, A., Varon, J., Bradley, C., Levy, M., Finfer, S., Hamielec, C., McMullin, J., Weaver, B., Walter, S., & Guyatt, G. (2003). Withdrawal of Mechanical Ventilation in Anticipation of Death in the Intensive Care Unit. *New England Journal of Medicine*, 349(12), 1123–1132. <https://doi.org/10.1056/nejmoa030083>
- Corona, G., Giuliani, C., Verbalis, J. G., Forti, G., Maggi, M., & Peri, A. (2015). Hyponatremia Improvement Is Associated with a Reduced Risk of Mortality: Evidence from a Meta-Analysis. *PLOS ONE*, 10(4), e0124105. <https://doi.org/10.1371/journal.pone.0124105>
- Cosgriff, C. V. (2019). *et al. Developing well calibrated illness severity scores for decision support in the critically ill*. <https://github.com/cosgriffc/seq-severityscore>
- Costumero, R., Gonzalo, C., & Menasalvas, E. (2014). TIDA: A Spanish EHR Semantic Search Engine. In *Advances in Intelligent Systems and Computing* (pp. 235–242). Springer International Publishing. https://doi.org/10.1007/978-3-319-07581-5_28
- Cottereau, A., Robert, R., le Gouge, A., Adda, M., Audibert, J., Barbier, F., Bardou, P., Bourcier, S., Boyer, A., Brenas, F., Canet, E., Da Silva, D., Das, V., Desachy, A., Devaquet, J., Embriaco, N., Eon, B., Feissel, M., Friedman, D., ... Reigner, J. (2016). ICU physicians' and nurses' perceptions of terminal extubation and terminal weaning: A self-questionnaire study. *Intensive Care Medicine*, 42(8), 1248–1257. <https://doi.org/10.1007/s00134-016-4373-9>
- Creanga, A. A., Bateman, B. T., Mhyre, J. M., Kuklina, E., Shilkut, A., & Callaghan, W. M. (2014). Performance of racial and ethnic minority-serving hospitals on delivery-related indicators. *American Journal of Obstetrics and Gynecology*, 211(6), 647.e1–647.e16. <https://doi.org/10.1016/j.ajog.2014.06.006>
- Creutzfeldt, C. J., Hanna, M. G., Cheever, C. S., Lele, A. V., Spiekerman, C., Engelberg, R. A., & Curtis, J. R. (2017). Palliative Care Needs Assessment in the Neuro-ICU: Effect on Family. *Neurocritical Care*, 27(2), 163–172. <https://doi.org/10.1007/s12028-017-0426-3>
- Curtis, J. R. (2005). Interventions to Improve Care during Withdrawal of Life-Sustaining Treatments. *Journal of Palliative Medicine*, 8(supplement 1), s-116-s-131. <https://doi.org/10.1089/jpm.2005.8.s-116>
- Curtis, J. R., Downey, L., & Engelberg, R. A. (2016). The importance and challenge of measuring family experience with end-of-life care in the ICU. *Intensive Care Medicine*, 42(7), 1179–1181. <https://doi.org/10.1007/s00134-016-4346-z>
- Curtis, J. R., Engelberg, R. A., Wenrich, M. D., Nielsen, E. L., Shannon, S. E., Treece, P. D., Tonelli, M. R., Patrick, D. L., Robins, L. S., McGrath, B. B., & Rubenfeld, G. D. (2002). Studying communication about end-of-life care during the ICU family conference: Development of a framework. *Journal of Critical Care*, 17(3), 147–160. <https://doi.org/10.1053/jcrc.2002.35929>
- Curtis, J. R., Treece, P. D., Nielsen, E. L., Gold, J., Ciechanowski, P. S., Shannon, S. E., Khandelwal, N., Young, J. P., & Engelberg, R. A. (2016). Randomized Trial of Communication Facilitators to Reduce Family Distress and Intensity of End-of-Life Care. *American Journal of Respiratory and Critical Care Medicine*, 193(2), 154–162. <https://doi.org/10.1164/rccm.201505-0900oc>
- Cusack, R., Rhodes, A., Lochhead, P., Jordan, B., Perry, S., Ball, J., Grounds, R., & Bennett, E. (2002). The strong ion gap does not have prognostic value in critically ill patients in a mixed medical/surgical adult ICU. *Intensive Care Medicine*, 28(7), 864–869. <https://doi.org/10.1007/s00134-002-1318-2>
- DAB Saúde. (2018, February 9). *Departamento de Atenção Básica. Telessaúde Brasil Redes na Atenção Básica à Saúde*. <http://bit.ly/29iMWCu>
- Danziger, J., & Hoenig, M. P. (2016). The Role of the Kidney in Disorders of Volume: Core Curriculum 2016. *American Journal of Kidney Diseases*, 68(5), 808–816. <https://doi.org/10.1053/j.ajkd.2016.05.028>
- Danziger, J., & Zeidel, M. L. (2014). Osmotic Homeostasis. *Clinical Journal of the American Society of Nephrology*, 10(5), 852–862. <https://doi.org/10.2215/cjn.10741013>

- Danziger, Zeidel, & Parker. (2012). *Renal Physiology: A Clinical Approach* (Edición: 1). LWW.
- Darst, J. R., Newburger, J. W., Resch, S., Rathod, R. H., & Lock, J. E. (2010). Deciding without data. *Congenital Heart Disease*, 5(4), 339–342. <https://doi.org/10.1111/j.1747-0803.2010.00433.x>
- Data, M. C. (2016). *Secondary Analysis of Electronic Health Records*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-43742-2>
- Davidson, J. E., Aslakson, R. A., Long, A. C., Puntillo, K. A., Kross, E. K., Hart, J., Cox, C. E., Wunsch, H., Wickline, M. A., Nunnally, M. E., Netzer, G., Kentish-Barnes, N., Sprung, C. L., Hartog, C. S., Coombs, M., Gerritsen, R. T., Hopkins, R. O., Franck, L. S., Skrobik, Y., ... Curtis, J. R. (2017). Guidelines for Family-Centered Care in the Neonatal, Pediatric, and Adult ICU. *Critical Care Medicine*, 45(1), 103–128. <https://doi.org/10.1097/ccm.0000000000002169>
- Davis, S. E., Lasko, T. A., Chen, G., & Matheny, M. E. (2018). Calibration drift among regression and machine learning models for hospital mortality. *AMIA. Annu. Symp. Proc.*, 2017.
- Day, D. V., Gronn, P., & Salas, E. (2004). Leadership capacity in teams. *The Leadership Quarterly*, 15(6), 857–880. <https://doi.org/10.1016/j.leaqua.2004.09.001>
- de Vasconcellos, K., & Skinner, D. L. (2018). Hyperchloraemia is associated with acute kidney injury and mortality in the critically ill: A retrospective observational study in a multidisciplinary intensive care unit. *Journal of Critical Care*, 45, 45–51. <https://doi.org/10.1016/j.jcrc.2018.01.019>
- Delaney, J. W., & Downar, J. (2016). How is life support withdrawn in intensive care units: A narrative review. *Journal of Critical Care*, 35, 12–18. <https://doi.org/10.1016/j.jcrc.2016.04.006>
- Desautels, T., Calvert, J., Hoffman, J., Jay, M., Kerem, Y., Shieh, L., Shimabukuro, D., Chettipally, U., Feldman, M. D., Barton, C., Wales, D. J., & Das, R. (2016). Prediction of Sepsis in the Intensive Care Unit With Minimal Electronic Health Record Data: A Machine Learning Approach. *JMIR Medical Informatics*, 4(3), e28. <https://doi.org/10.2196/medinform.5909>
- Desautels, T., Das, R., Calvert, J., Trivedi, M., Summers, C., Wales, D. J., & Ercole, A. (2017). Prediction of early unplanned intensive care unit readmission in a UK tertiary care hospital: A cross-sectional machine learning approach. *BMJ Open*, 7(9), e017199. <https://doi.org/10.1136/bmjopen-2017-017199>
- Deschepper, R., Laureys, S., Hachimi-Idrissi, S., Poelaert, J., Distelmans, W., & Bilsen, J. (2013). Palliative sedation: Why we should be more concerned about the risks that patients experience an uncomfortable death. *Pain*, 154(9), 1505–1508. <https://doi.org/10.1016/j.pain.2013.04.038>
- DICOM. (2018). *DICOM 2018*. <https://www.dicomstandard.org/>.
- Dijk, S. J. van, Tellam, R. L., Morrison, J. L., Muhlhausler, B. S., & Molloy, P. L. (2015). Recent developments on the role of epigenetics in obesity and metabolic disease. *Clinical Epigenetics*, 7(1). <https://doi.org/10.1186/s13148-015-0101-5>
- Diseases ICo. (2018). *CIE 10*. <https://eciemaps.msssi.gob.es/>.
- Dixon, J., & Brien, P. O. (2006). Obesity and the White Blood Cell Count: Changes with Sustained Weight Loss. *Obesity Surgery*, 16(3), 251–257. <https://doi.org/10.1381/096089206776116453>
- dos Santos, R. P., Carvalho, A. R. da S., & Peres, L. A. B. (2019). Incidence and risk factors of acute kidney injury in critically ill patients from a single centre in Brazil: A retrospective cohort analysis. *Scientific Reports*, 9(1). <https://doi.org/10.1038/s41598-019-54674-1>
- Downar, J., Delaney, J. W., Hawryluck, L., & Kenny, L. (2016). Guidelines for the withdrawal of life-sustaining measures. *Intensive Care Medicine*, 42(6), 1003–1017. <https://doi.org/10.1007/s00134-016-4330-7>
- Elixhauser, A., Steiner, C., Harris, D. R., & Coffey, R. M. (1998). Comorbidity Measures for Use with Administrative Data. *Medical Care*, 36(1), 8–27. <https://doi.org/10.1097/00005650-199801000-00004>
- Engelberg, R. A., Downey, L., Wenrich, M. D., Carline, J. D., Silvestri, G. A., Dotolo, D., Nielsen, E. L., & Curtis, J. R. (2010). Measuring the Quality of End-of-Life Care. *Journal of Pain and Symptom Management*, 39(6), 951–971. <https://doi.org/10.1016/j.jpainsymman.2009.11.313>

- Epstein, R. H., Dexter, F., & Schwenk, E. S. (2017). Hypotension during induction of anaesthesia is neither a reliable nor a useful quality measure for comparison of anaesthetists' performance. *British Journal of Anaesthesia*, 119(1), 106–114. <https://doi.org/10.1093/bja/aex153>
- Erickson, S. E., Martin, G. S., Davis, J. L., Matthay, M. A., & Eisner, M. D. (2009). Recent trends in acute lung injury mortality: 1996–2005. *Critical Care Medicine*, 37(5), 1574–1579. <https://doi.org/10.1097/ccm.0b013e31819fefdf>
- EU GDPR Portal. (2018). *EU General Data Protection Regulation (GDPR)*. <https://www.eugdpr.org/>.
- Feng, M., McSparron, J. I., Kien, D. T., Stone, D. J., Roberts, D. H., Schwartzstein, R. M., Vieillard-Baron, A., & Celi, L. A. (2018). Transthoracic echocardiography and mortality in sepsis: Analysis of the MIMIC-III database. *Intensive Care Medicine*, 44(6), 884–892. <https://doi.org/10.1007/s00134-018-5208-7>
- Fischer, U. M., Weissenberger, W. K., Warters, R. D., Geissler, H. J., Allen, S. J., & Mehlhorn, U. (2002). Impact of cardiopulmonary bypass management on postcardiac surgery renal function. *Perfusion*, 17(6), 401–406. <https://doi.org/10.1191/0267659102pf6100a>
- Flegal, K. M., Kit, B. K., Orpana, H., & Graubard, B. I. (2013). Association of All-Cause Mortality With Overweight and Obesity Using Standard Body Mass Index Categories. *JAMA*, 309(1), 71. <https://doi.org/10.1001/jama.2012.113905>
- Foundation, A. (2014). *Apache Hadoop*. <http://hadoop.apache.org/>
- Fowler, R. A., Noyahr, L.-A., Thornton, J. D., Pinto, R., Kahn, J. M., Adhikari, N. K. J., Dodek, P. M., Khan, N. A., Kalb, T., Hill, A., O'neill, J. M., Evans, D., & Curtis, J. R. (2010). An Official American Thoracic Society Systematic Review: The Association between Health Insurance Status and Access, Care Delivery, and Outcomes for Patients Who Are Critically Ill. *American Journal of Respiratory and Critical Care Medicine*, 181(9), 1003–1011. <https://doi.org/10.1164/rccm.200902-0281st>
- Francis, D., & Young, D. (1979). *Improving work groups, a practical manual for team building*. University Associates. <https://agris.fao.org/agris-search/search.do?recordID=US8051647>
- Fransen, J., Kirschner, P. A., & Erkens, G. (2011). Mediating team effectiveness in the context of collaborative learning: The importance of team and task awareness. *Computers in Human Behavior*, 27(3), 1103–1113. <https://doi.org/10.1016/j.chb.2010.05.017>
- Frei, C. R., Mortensen, E. M., Copeland, L. A., Attridge, R. T., Pugh, M. J. V., Restrepo, M. I., Anzueto, A., Nakashima, B., & Fine, M. J. (2010). Disparities of Care for African-Americans and Caucasians with Community-Acquired Pneumonia: A Retrospective Cohort Study. *BMC Health Services Research*, 10(1). <https://doi.org/10.1186/1472-6963-10-143>
- Funk, G.-C., Lindner, G., Druml, W., Metnitz, B., Schwarz, C., Bauer, P., & Metnitz, P. G. H. (2009). Incidence and prognosis of dysnatremias present on ICU admission. *Intensive Care Medicine*, 36(2), 304–311. <https://doi.org/10.1007/s00134-009-1692-0>
- Futier, E., Lefrant, J.-Y., Guinot, P.-G., Godet, T., Lorne, E., Cuvillon, P., Bertran, S., Leone, M., Pastene, B., Piriou, V., Molliex, S., Albanese, J., Julia, J.-M., Tavernier, B., Imhoff, E., Bazin, J.-E., Constantin, J.-M., Pereira, B., & Jaber, S. (2017). Effect of Individualized vs Standard Blood Pressure Management Strategies on Postoperative Organ Dysfunction Among High-Risk Patients Undergoing Major Surgery. *JAMA*, 318(14), 1346. <https://doi.org/10.1001/jama.2017.14172>
- Gall, J. R. L. (1993). A new Simplified Acute Physiology Score (SAPS II) based on a European/North American multicenter study. *JAMA: The Journal of the American Medical Association*, 270(24), 2957–2963. <https://doi.org/10.1001/jama.270.24.2957>
- Gehrmann, S., Deroncourt, F., Li, Y., Carlson, E. T., Wu, J. T., Welt, J., Foote, J., Moseley, E. T., Grant, D. W., Tyler, P. D., & Celi, L. A. (2018). Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives. *PLOS ONE*, 13(2), e0192360. <https://doi.org/10.1371/journal.pone.0192360>
- Ghassemi, M., Celi, L., & Stone, D. J. (2015). State of the art review: The data revolution in critical care. *Critical Care*, 19(1), 118. <https://doi.org/10.1186/s13054-015-0801-4>

- Ghassemi, M. M., Richter, S. E., Eche, I. M., Chen, T. W., Danziger, J., & Celi, L. A. (2014). A data-driven approach to optimized medication dosing: A focus on heparin. *Intensive Care Medicine*, 40(9), 1332–1339. <https://doi.org/10.1007/s00134-014-3406-5>
- Ghassemi, M., Marshall, J., Singh, N., Stone, D. J., & Celi, L. A. (2014). Leveraging a Critical Care Database. *Chest*, 145(4), 745–752. <https://doi.org/10.1378/chest.13-1722>
- Ginsberg, J., Mohebbi, M. H., Patel, R. S., Brammer, L., Smolinski, M. S., & Brilliant, L. (2009). Detecting influenza epidemics using search engine query data. *Nature*, 457(7232), 1012–1014. <https://doi.org/10.1038/nature07634>
- Girard, T. D., Jackson, J. C., Pandharipande, P. P., Pun, B. T., Thompson, J. L., Shintani, A. K., Gordon, S. M., Canonico, A. E., Dittus, R. S., Bernard, G. R., & Wesley Ely, E. (2010). Delirium as a predictor of long-term cognitive impairment in survivors of critical illness. *Critical Care Medicine*, 38(7), 1513–1520. <https://doi.org/10.1097/ccm.0b013e3181e47be1>
- Gleason, L. J., Schmitt, E. M., Kosar, C. M., Tabloski, P., Saczynski, J. S., Robinson, T., Cooper, Z., Rogers, S. O., Jones, R. N., Marcantonio, E. R., & Inouye, S. K. (2015). Effect of Delirium and Other Major Complications on Outcomes After Elective Surgery in Older Adults. *JAMA Surgery*, 150(12), 1134. <https://doi.org/10.1001/jamasurg.2015.2606>
- Granhölm, A., Möller, M. H., Krag, M., Perner, A., & Hjortrup, P. B. (2016). Predictive Performance of the Simplified Acute Physiology Score (SAPS) II and the Initial Sequential Organ Failure Assessment (SOFA) Score in Acutely Ill Intensive Care Patients: Post-Hoc Analyses of the SUP-ICU Inception Cohort Study. *PLOS ONE*, 11(12), e0168948. <https://doi.org/10.1371/journal.pone.0168948>
- Gribas, J. (1990). *Characteristics of Temporary Teams: Perceptions Among Community Theatre Casts and Directors*.
- Gu, S., & Wu, Q. (2017). *How Random Forest Algorithm Works in Machine Learning*. <https://medium.com/@Synced/how-random-forest-algorithm-works-in-machine-learning-3c0fe15b6674> [cited 2018].
- Gu, W.-J., Hou, B.-L., Kwong, J. S. W., Tian, X., Qian, Y., Cui, Y., Hao, J., Li, J.-C., Ma, Z.-L., & Gu, X.-P. (2018). Association between intraoperative hypotension and 30-day mortality, major adverse cardiac events, and acute kidney injury after non-cardiac surgery: A meta-analysis of cohort studies. *International Journal of Cardiology*, 258, 68–73. <https://doi.org/10.1016/j.ijcard.2018.01.137>
- Gunathilake, R., Oldmeadow, C., McEvoy, M., Kelly, B., Inder, K., Schofield, P., & Attia, J. (2013). Mild Hyponatremia is Associated with Impaired Cognition and Falls in Community-Dwelling Older Persons. *Journal of the American Geriatrics Society*, 61(10), 1838–1839. <https://doi.org/10.1111/jgs.12468>
- Haider, A. H. (2008). Race and Insurance Status as Risk Factors for Trauma Mortality. *Archives of Surgery*, 143(10), 945. <https://doi.org/10.1001/archsurg.143.10.945>
- Haider, A. H. (2012). Association Between Hospitals Caring for a Disproportionately High Percentage of Minority Trauma Patients and Increased Mortality. *Archives of Surgery*, 147(1), 63. <https://doi.org/10.1001/archsurg.2011.254>
- Harris, K., Zhou, J., Liu, X., Hassan, E., & Badawi, O. (2017). The Obesity Paradox Is Not Observed in Critically Ill Patients on Early Enteral Nutrition†. *Critical Care Medicine*, 45(5), 828–834. <https://doi.org/10.1097/ccm.0000000000002326>
- Hawryluck, L. A., Harvey, W. R., Lemieux-Charles, L., & Singer, P. A. (2002). Consensus guidelines on analgesia and sedation in dying intensive care unit patients. *BMC Medical Ethics*, 3(1). <https://doi.org/10.1186/1472-6939-3-3>
- Heifetz, R., Grashow, A., & Linsky, M. (2010). The Practice of Adaptive Leadership: Tools and Tactics for Changing Your Organization and the World. *Personnel Psychology*, 63(1), 255–258. https://doi.org/10.1111/j.1744-6570.2009.01168_4.x
- Henry, K. E., Hager, D. N., Pronovost, P. J., & Saria, S. (2015). A targeted real-time early warning score (TREWScore) for septic shock. *Science Translational Medicine*, 7(299), 299ra122–299ra122. <https://doi.org/10.1126/scitranslmed.aab3719>
- Heymsfield, S. B., & Wadden, T. A. (2017). Mechanisms, Pathophysiology, and Management of Obesity. *New England Journal of Medicine*, 376(3), 254–266. <https://doi.org/10.1056/nejmra1514009>

- HHS.gov. (2018). *Health information privacy*.
- HHS.gov. (2019). *Office of Minority Health (OMH)*. <https://www.minorityhealth.hhs.gov/>
- Hinton, G. (2018). Deep learning-a technology with the potential to transform health care. *JAMA*, 320. <https://doi.org/10.1001/jama.2018.11100>
- Hirsch, J., DePalma, G., Tsai, T. T., Sands, L. P., & Leung, J. M. (2015). Impact of intraoperative hypotension and blood pressure fluctuations on early postoperative delirium after non-cardiac surgery † ‡. *British Journal of Anaesthesia*, 115(3), 418–426. <https://doi.org/10.1093/bja/aeu458>
- Ho, K. M., Lan, N. S. H., Williams, T. A., Harahsheh, Y., Chapman, A. R., Dobb, G. J., & Magder, S. (2016). A comparison of prognostic significance of strong ion gap (SIG) with other acid-base markers in the critically ill: A cohort study. *Journal of Intensive Care*, 4(1). <https://doi.org/10.1186/s40560-016-0166-z>
- Hogue, C. W., Palin, C. A., & Arrowsmith, J. E. (2006). Cardiopulmonary Bypass Management and Neurologic Outcomes: An Evidence-Based Appraisal of Current Practices. *Anesthesia & Analgesia*, 103(1), 21–37. <https://doi.org/10.1213/01.ane.0000220035.82989.79>
- Holloway, R. G., Arnold, R. M., Creutzfeldt, C. J., Lewis, E. F., Lutz, B. J., McCann, R. M., Rabinstein, A. A., Saposnik, G., Sheth, K. N., Zahuranec, D. B., Zipfel, G. J., & Zorowitz, R. D. (2014). Palliative and End-of-Life Care in Stroke. *Stroke*, 45(6), 1887–1916. <https://doi.org/10.1161/str.0000000000000015>
- Hoorn, E. J., & Zietse, R. (2011). Hyponatremia and Mortality: How Innocent is the Bystander?: Figure 1. *Clinical Journal of the American Society of Nephrology*, 6(5), 951–953. <https://doi.org/10.2215/cjn.01210211>
- Hoorn, E. J., & Zietse, R. (2013). Hyponatremia and Mortality: Moving Beyond Associations. *American Journal of Kidney Diseases*, 62(1), 139–149. <https://doi.org/10.1053/j.ajkd.2012.09.019>
- Hori, D., Brown, C., Ono, M., Rappold, T., Sieber, F., Gottschalk, A., Neufeld, K. J., Gottesman, R., Adachi, H., & Hogue, C. W. (2014). Arterial pressure above the upper cerebral autoregulation limit during cardiopulmonary bypass is associated with postoperative delirium. *British Journal of Anaesthesia*, 113(6), 1009–1017. <https://doi.org/10.1093/bja/aeu319>
- Hori, Daijiro, Hogue, C., Adachi, H., Max, L., Price, J., Sciortino, C., Zehr, K., Conte, J., Cameron, D., & Mandal, K. (2016). Perioperative optimal blood pressure as determined by ultrasound tagged near infrared spectroscopy and its association with postoperative acute kidney injury in cardiac surgery patients. *Interactive CardioVascular and Thoracic Surgery*, 22(4), 445–451. <https://doi.org/10.1093/icvts/ivv371>
- Hori, Daijiro, Nomura, Y., Ono, M., Joshi, B., Mandal, K., Cameron, D., Kocherginsky, M., & Hogue, C. W. (2017). Optimal blood pressure during cardiopulmonary bypass defined by cerebral autoregulation monitoring. *The Journal of Thoracic and Cardiovascular Surgery*, 154(5), 1590-1598.e2. <https://doi.org/10.1016/j.jtcvs.2017.04.091>
- Hummel, GS. (2016, November). *Brazil eHealth—Overview, Trends & Opportunities*. [http:// bit.ly/2vCR5IY](http://bit.ly/2vCR5IY)
- Institute, R. (2018). *LOINC*. <https://loinc.org/>.
- Intas, G., Pantelis, S., & Georgios, V. (2011). *Impact of critically ill patients in regards to the boarding time from the emergency department to the intensive care unit*. https://www.researchgate.net/publication/269333212_Impact_of_critically_ill_patients_in_regards_to_the_boarding_time_from_the_emergency_department_to_the_intensive_care_unit
- Iwashyna, T. J., & Liu, V. (2014). What's so different about big data? A primer for clinicians trained to think epidemiologically. *Ann Am Thorac Soc*.
- Jesse, H., Gregory, S., & John, K. (1992). *Principles of critical care* (Vol. 1–500–514). New York: McGraw-Hill, Health Professions Division.
- Jha, A. K., Orav, E. J., Li, Z., & Epstein, A. M. (2007). Concentration and Quality of Hospitals That Care for Elderly Black Patients. *Archives of Internal Medicine*, 167(11), 1177. <https://doi.org/10.1001/archinte.167.11.1177>
- Jinadasa, S. P., Mueller, A., Prasad, V., Subramaniam, K., Heldt, T., Novack, V., & Subramaniam, B. (2018). Blood Pressure Coefficient of Variation and Its Association With Cardiac Surgical Outcomes. *Anesthesia & Analgesia*, 127(4), 832–839. <https://doi.org/10.1213/ane.0000000000003362>

- Johansson, H.-E., Haenni, A., & Zethelius, B. (2011). Changes in Erythrocyte Sedimentation Rate, White Blood Cell Count, Liver Enzymes, and Magnesium after Gastric Bypass Surgery. *Journal of Obesity*, 2011, 1–6. <https://doi.org/10.1155/2011/273105>
- Johnson, A. E. W., & Mark, R. G. (2018). Real-time mortality prediction in the Intensive Care Unit. *AMIA. Annu. Symp. Proc.*, 2017.
- Johnson, Alistair E. W., Pollard, T. J., Celi, L. A., & Mark, R. G. (2017, August). Analyzing the eICU Collaborative Research Database. *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. <https://doi.org/10.1145/3107411.3107494>
- Johnson, Alistair E. W., Pollard, T. J., Shen, L., Lehman, L. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., & Mark, R. G. (2016). MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1). <https://doi.org/10.1038/sdata.2016.35>
- Joshi, B., Ono, M., Brown, C., Brady, K., Easley, R. B., Yenokyan, G., Gottesman, R. F., & Hogue, C. W. (2012). Predicting the Limits of Cerebral Autoregulation During Cardiopulmonary Bypass. *Anesthesia & Analgesia*, 114(3), 503–510. <https://doi.org/10.1213/ane.0b013e31823d292a>
- Joynt, K. E. (2011). Thirty-Day Readmission Rates for Medicare Beneficiaries by Race and Site of Care. *JAMA*, 305(7), 675. <https://doi.org/10.1001/jama.2011.123>
- Joynt, K. E., & Jha, A. K. (2013). Characteristics of Hospitals Receiving Penalties Under the Hospital Readmissions Reduction Program. *JAMA*, 309(4), 342. <https://doi.org/10.1001/jama.2012.94856>
- Kanji, H. D., Schulze, C. J., Hervas-Malo, M., Wang, P., Ross, D. B., Zibdawi, M., & Bagshaw, S. M. (2010). Difference between pre-operative and cardiopulmonary bypass mean arterial pressure is independently associated with early cardiac surgery-associated acute kidney injury. *Journal of Cardiothoracic Surgery*, 5(1). <https://doi.org/10.1186/1749-8090-5-71>
- Kaplan, L. J., & Kellum, J. A. (2007). COMPARISON OF ACID BASE MODELS FOR PREDICTION OF HOSPITAL MORTALITY FOLLOWING TRAUMA. *Shock*, 1. <https://doi.org/10.1097/shk.0b013e3181618946>
- Kearney, P., Plax, T. G., & Wendt-Wasco, N. J. (1985). Teacher immediacy for affective learning in divergent college classes. *Communication Quarterly*, 33(1), 61–74. <https://doi.org/10.1080/01463378509369579>
- Kellum, J. A., & Lameire, N. (2013). Diagnosis, evaluation, and management of acute kidney injury: A KDIGO summary (Part 1). *Critical Care*, 17(1), 204. <https://doi.org/10.1186/cc11454>
- Kendrick, J., Nuccio, E., Leiferman, J. A., & Sauaia, A. (2015). Primary Care Providers Perceptions of Racial/Ethnic and Socioeconomic Disparities in Hypertension Control. *American Journal of Hypertension*, 28(9), 1091–1097. <https://doi.org/10.1093/ajh/hpu294>
- Khandelwal, N., Benkeser, D., Coe, N. B., Engelberg, R. A., Teno, J. M., & Curtis, J. R. (2016). Patterns of Cost for Patients Dying in the Intensive Care Unit and Implications for Cost Savings of Palliative Care Interventions. *Journal of Palliative Medicine*, 19(11), 1171–1178. <https://doi.org/10.1089/jpm.2016.0133>
- Kherallah, M., Dahhan, T., Jamil, M. G., Abouchala, N., & Al-tarifi, A. (2008). COMPARISON OF STANDARDIZED MORTALITY RATIOS OF ACUTE PHYSIOLOGY AND CHRONIC HEALTH EVALUATION (APACHE) II AND IV SCORING SYSTEMS. *Chest*, 134(4), 112P. https://doi.org/10.1378/chest.134.4_meetingabstracts.p112001
- Kilger, E., Weis, F., Briegel, J., Frey, L., Goetz, A. E., Reuter, D., Nagy, A., Schuetz, A., Lamm, P., Knoll, A., & Peter, K. (2003). Stress doses of hydrocortisone reduce severe systemic inflammatory response syndrome and improve early outcome in a risk group of patients after cardiac surgery. *Critical Care Medicine*, 31(4), 1068–1074. <https://doi.org/10.1097/01.ccm.0000059646.89546.98>
- Kimura, S., Iwasaki, T., Shimizu, K., Kanazawa, T., Kawase, H., Shioji, N., Kuroe, Y., Matsuoka, Y., Isoyama, S., & Morimatsu, H. (2019). Hyperchloremia Is Not an Independent Risk Factor for Postoperative Acute Kidney Injury in Pediatric Cardiac Patients. *Journal of Cardiothoracic and Vascular Anesthesia*, 33(7), 1939–1945. <https://doi.org/10.1053/j.jvca.2018.12.009>
- Kind, A. J. H., Jencks, S., Brock, J., Yu, M., Bartels, C., Ehlenbach, W., Greenberg, C., & Smith, M. (2014). Neighborhood Socioeconomic Disadvantage and 30-Day Rehospitalization. *Annals of Internal Medicine*, 161(11), 765. <https://doi.org/10.7326/m13-2946>

- Knaus WA, Draper EA, Wagner DP, & Zimmerman JE. (1985). *APACHE II: a severity of disease classification system. Crit Care Med*; 13: Pp. 818-829.
- Komorowski, M., Celi, L. A., Badawi, O., Gordon, A. C., & Faisal, A. A. (2018). The Artificial Intelligence Clinician learns optimal treatment strategies for sepsis in intensive care. *Nature Medicine*, 24(11), 1716–1720. <https://doi.org/10.1038/s41591-018-0213-5>
- Komssi, M., Pichlis, D., Raatikainen, M., Kindström, K., & Järvinen, J. (2015). What are Hackathons for? *IEEE Software*, 32(5), 60–67. <https://doi.org/10.1109/MS.2014.78>
- Komssi, Marko, Pichlis, D., Raatikainen, M., Kindström, K., & Järvinen, J. (2015). What are Hackathons for? *IEEE Software*, 32(5), 60–67. <https://doi.org/10.1109/MS.2014.78>
- Koyner, J. L., Carey, K. A., Edelson, D. P., & Churpek, M. M. (2018). The development of a machine learning inpatient acute kidney injury prediction model. *Crit. Care Med.*, 46. <https://doi.org/10.1097/CCM.00000000000003123>
- Kramer, A. A., Higgins, T. L., & Zimmerman, J. E. (2014). Comparison of the Mortality Probability Admission Model III, National Quality Forum, and Acute Physiology and Chronic Health Evaluation IV hospital mortality models: Implications for national benchmarking. *Crit. Care Med.*, 42. <https://doi.org/10.1097/CCM.0b013e3182a66a49>
- Kramer, A. A., Higgins, T. L., & Zimmerman, J. E. (2015). Comparing observed and predicted mortality among ICUs using different prognostic systems: Why do performance assessments differ? *Crit. Care Med.*, 43. <https://doi.org/10.1097/CCM.0000000000000694>
- Krumholz, H. M. (2014). Big Data And New Knowledge In Medicine: The Thinking, Training, And Tools Needed For A Learning Health System. *Health Affairs (Project Hope)*, 33(7), 1163–1170. <https://doi.org/10.1377/hlthaff.2014.0053>
- Lamarche, Y., Elmi-Sarabi, M., Ding, L., Abel, J. G., Sirounis, D., & Denault, A. Y. (2017). A score to estimate 30-day mortality after intensive care admission after cardiac surgery. *The Journal of Thoracic and Cardiovascular Surgery*, 153(5), 1118-1125.e4. <https://doi.org/10.1016/j.jtcvs.2016.11.039>
- Lanken, P. N., Terry, P. B., DeLisser, H. M., Fahy, B. F., Hansen-Flaschen, J., Heffner, J. E., Levy, M., Mularski, R. A., Osborne, M. L., Prendergast, T. J., Rocker, G., Sibbald, W. J., Wilfond, B., & Yankaskas, J. R. (2008). An Official American Thoracic Society Clinical Policy Statement: Palliative Care for Patients with Respiratory Diseases and Critical Illnesses. *American Journal of Respiratory and Critical Care Medicine*, 177(8), 912–927. <https://doi.org/10.1164/rccm.200605-587st>
- Lauerman, M., Herrera, A., Klyushnenkova, E., Narayan, M., Bruns, B., Tesoriero, R., Buchanan, L., & Diaz, J. (2015). Racial disparities in icu resource utilization in emergency general surgery. *Critical Care Medicine*, 43(12), 281–282. <https://doi.org/10.1097/01.ccm.0000474951.39931.c3>
- Lautrette, A., Darmon, M., Megarbane, B., Joly, L. M., Chevret, S., Adrie, C., Barnoud, D., Bleichner, G., Bruel, C., Choukroun, G., Curtis, J. R., Fieux, F., Galliot, R., Garrouste-Orgeas, M., Georges, H., Goldgran-Toledano, D., Jourdain, M., Loubert, G., Reignier, J., ... Azoulay, E. (2007). A Communication Strategy and Brochure for Relatives of Patients Dying in the ICU. *New England Journal of Medicine*, 356(5), 469–478. <https://doi.org/10.1056/nejmoa063446>
- Lee, J. Y., Hong, T. H., Lee, K. W., Jung, M. J., Lee, J. G., & Lee, S. H. (2016). Hyperchloremia is associated with 30-day mortality in major trauma patients: A retrospective observational study. *Scandinavian Journal of Trauma, Resuscitation and Emergency Medicine*, 24(1). <https://doi.org/10.1186/s13049-016-0311-7>
- Lemeshow, S. (1993). Mortality Probability Models (MPM II) based on an international cohort of intensive care unit patients. *JAMA*, 270. <https://doi.org/10.1001/jama.1993.03510200084037>
- Liebowitz, S. J., & de Meuse, K. P. (1982). The Application of Team Building. *Human Relations*, 35(1), 1–18. <https://doi.org/10.1177/001872678203500102>
- Lilly, C. M., Swami, S., Liu, X., Riker, R. R., & Badawi, O. (2017). Five-Year Trends of Critical Care Practice and Outcomes. *Chest*, 152(4), 723–735. <https://doi.org/10.1016/j.chest.2017.06.050>
- Liu, W.-Y., Lin, S.-G., Zhu, G.-Q., Poucke, S. V., Braddock, M., Zhang, Z., Mao, Z., Shen, F.-X., & Zheng, M.-H. (2016). Establishment and Validation of GV-SAPS II Scoring System for Non-Diabetic Critically Ill Patients. *PLOS ONE*, 11(11), e0166085. <https://doi.org/10.1371/journal.pone.0166085>

- Lombardi, G., Ferraro, P. M., Bargagli, M., Naticchia, A., D'Alonzo, S., & Gambaro, G. (2019). Hyperchloremia and acute kidney injury: A retrospective observational cohort study on a general mixed medical-surgical not ICU-hospitalized population. *Internal and Emergency Medicine*, 15(2), 273–280. <https://doi.org/10.1007/s11739-019-02165-6>
- López, L., & Jha, A. K. (2012). Outcomes for Whites and Blacks at Hospitals That Disproportionately Care for Black Medicare Beneficiaries. *Health Services Research*, 48(1), 114–128. <https://doi.org/10.1111/j.1475-6773.2012.01445.x>
- Ly, D. P., Lopez, L., Isaac, T., & Jha, A. K. (2010). How Do Black-Serving Hospitals Perform on Patient Safety Indicators? *Medical Care*, 48(12), 1133–1137. <https://doi.org/10.1097/mlr.0b013e3181f81c7e>
- Maheshwari, K., Nathanson, B. H., Munson, S. H., Khangulov, V., Stevens, M., Badani, H., Khanna, A. K., & Sessler, D. I. (2018). The relationship between ICU hypotension and in-hospital mortality and morbidity in septic patients. *Intensive Care Medicine*, 44(6), 857–867. <https://doi.org/10.1007/s00134-018-5218-5>
- Mallat, J., Barrailler, S., Lemyze, M., Pepy, F., Gasan, G., Tronchon, L., & Thevenin, D. (2013). Use of Sodium-Chloride Difference and Corrected Anion Gap as Surrogates of Stewart Variables in Critically Ill Patients. *PLoS ONE*, 8(2), e56635. <https://doi.org/10.1371/journal.pone.0056635>
- Mancl, E. E., & Muzevich, K. M. (2012). Tolerability and Safety of Enteral Nutrition in Critically Ill Patients Receiving Intravenous Vasopressor Therapy. *Journal of Parenteral and Enteral Nutrition*, 37(5), 641–651. <https://doi.org/10.1177/0148607112470460>
- Mao, W., Wu, J., Zhang, H., Zhou, J., Ye, B., Li, G., Gao, L., Li, X., Ke, L., Tong, Z., Li, W., & Li, J. (2019). Increase in serum chloride and chloride exposure are associated with acute kidney injury in moderately severe and severe acute pancreatitis patients. *Pancreatology*, 19(1), 136–142. <https://doi.org/10.1016/j.pan.2018.11.006>
- Marin, H. F., Senne, F., & Barbosa, A. (2014). ICT health 2013: Infrastructure and adoption by healthcare providers in Brazil. *Studies in Health Technology and Informatics*, 205, 496–500.
- Martins, S., Pinho, E., Correia, R., Moreira, E., Lopes, L., Paiva, J. A., Azevedo, L., & Fernandes, L. (2017). What effect does delirium have on family and nurses of older adult patients? *Aging & Mental Health*, 22(7), 903–911. <https://doi.org/10.1080/13607863.2017.1393794>
- Martinen, M., Wilkman, E., Petäjä, L., Suojaranta-Ylinen, R., Pettilä, V., & Vaara, S. T. (2016). Association of plasma chloride values with acute kidney injury in the critically ill—A prospective observational study. *Acta Anaesthesiologica Scandinavica*, 60(6), 790–799. <https://doi.org/10.1111/aas.12694>
- Mascha, E. J., Yang, D., Weiss, S., & Sessler, D. I. (2015). Intraoperative Mean Arterial Pressure Variability and 30-day Mortality in Patients Having Noncardiac Surgery. *Anesthesiology*, 123(1), 79–91. <https://doi.org/10.1097/aln.0000000000000686>
- Mauro, A. D., Greco, M., & Grimaldi, M. (2016). A formal definition of Big Data based on its essential features. *Library Review*, 65(3), 122–135. <https://doi.org/10.1108/lr-06-2015-0061>
- Mayer, S. A., & Kossoff, S. B. (1999). Withdrawal of life support in the neurological intensive care unit. *Neurology*, 52(8), 1602–1602. <https://doi.org/10.1212/wnl.52.8.1602>
- Mayo, C. S., Matuszak, M. M., Schipper, M. J., Jolly, S., Hayman, J. A., & Haken, R. K. T. (2017). Big Data in Designing Clinical Trials: Opportunities and Challenges. *Frontiers in Oncology*, 7. <https://doi.org/10.3389/fonc.2017.00187>
- Mazer, M. A., Alligood, C. M., & Wu, Q. (2011). The Infusion of Opioids During Terminal Withdrawal of Mechanical Ventilation in the Medical Intensive Care Unit. *Journal of Pain and Symptom Management*, 42(1), 44–51. <https://doi.org/10.1016/j.jpainsymman.2010.10.256>
- McCluskey, S. A., Karkouti, K., Wijesundera, D., Minkovich, L., Tait, G., & Beattie, W. S. (2013). Hyperchloremia After Noncardiac Surgery Is Independently Associated with Increased Morbidity and Mortality. *Anesthesia & Analgesia*, 117(2), 412–421. <https://doi.org/10.1213/ane.0b013e318293d81e>
- McGee, S. (2012). *Evidence-Based Physical Diagnosis—3rd Edition*. <https://www.elsevier.com/books/evidence-based-physical-diagnosis/mcgee/978-1-4377-2207-9>
- McLennan, S., Shaw, D., & Celi, L. A. (2018). The challenge of local consent requirements for global critical care databases. *Intensive Care Medicine*, 45(2), 246–248. <https://doi.org/10.1007/s00134-018-5257-y>

- McShea, M., Holl, R., Badawi, O., Riker, R., & Silfen, E. (2010). The eICU Research Institute—A Collaboration Between Industry, Health-Care Providers, and Academia. *IEEE Engineering in Medicine and Biology Magazine*, 29(2), 18–25. <https://doi.org/10.1109/memb.2009.935720>
- Mehta R. (2018). *Big Data Analytics with Java. Big Data Analytics with Java: Data analysis, visualization & machine learning techniques. Birmingham.*
- Merath, K., Hyer, J. M., Mehta, R., Farooq, A., Bagante, F., Sahara, K., Tsilimigras, D. I., Beal, E., Paredes, A. Z., Wu, L., Ejaz, A., & Pawlik, T. M. (2019). Use of Machine Learning for Prediction of Patient Risk of Postoperative Complications After Liver, Pancreatic, and Colorectal Surgery. *Journal of Gastrointestinal Surgery*. <https://doi.org/10.1007/s11605-019-04338-2>
- Meuse, K. P. D. (1992). Teamwork: What Must Go Right/What Can Go Wrong Carl E. Larson and Frank M. J. LaFasto, Sage Publications, Newbury Park, California, 1989, No. Of Pages: 150. *Journal of Organizational Behavior*, 13(6), 637–638. <https://doi.org/10.1002/job.4030130612>
- MIT Lab for Computational Physiology. (2018). *MIMIC*. <http://mimic.physionet.org>.
- MIT-LCP. (2018). *MIMIC III webpage*. <https://github.com/MIT-LCP/mimic-code>.
- Monk, T. G., Bronsert, M. R., Henderson, W. G., Mangione, M. P., Sum-Ping, S. T. J., Bentt, D. R., Nguyen, J. D., Richman, J. S., Meguid, R. A., & Hammermeister, K. E. (2015). Association between Intraoperative Hypotension and Hypertension and 30-day Postoperative Mortality in Noncardiac Surgery. *Anesthesiology*, 123(2), 307–319. <https://doi.org/10.1097/aln.0000000000000756>
- Moreno, R. P. (2005). SAPS 3—From evaluation of the patient to evaluation of the intensive care unit. Part 2: Development of a prognostic model for hospital mortality at ICU admission. *Intensive Care Med.*, 31. <https://doi.org/10.1007/s00134-005-2763-5>
- Morris, D., & Galicia-Castillo, M. (2016). Dying With Dyspnea in the Hospital. *American Journal of Hospice and Palliative Medicine*, 34(2), 132–134. <https://doi.org/10.1177/1049909115604140>
- Murkin, J. M., Farrar, J. K., Tweed, W. A., McKenzie, F. N., & Guiraudon, G. (1987). Cerebral Autoregulation and Flow/Metabolism Coupling during Cardiopulmonary Bypass. *Anesthesia & Analgesia*, 66(9), 825–832. <https://doi.org/10.1213/00000539-198709000-00003>
- Murphy, G. S., Hessel, E. A., & Groom, R. C. (2009). Optimal Perfusion During Cardiopulmonary Bypass: An Evidence-Based Approach. *Anesthesia & Analgesia*, 108(5), 1394–1417. <https://doi.org/10.1213/ane.0b013e3181875e2e>
- Naylor, C. D. (2018). On the prospects for a (deep) learning health care system. *JAMA*, 320. <https://doi.org/10.1001/jama.2018.11103>
- Nelson, J. E., Curtis, J. R., Mulkerin, C., Campbell, M., Lustbader, D. R., Mosenthal, A. C., Puntillo, K., Ray, D. E., Bassett, R., Boss, R. D., Brasel, K. J., Frontera, J. A., Hays, R. M., & Weissman, D. E. (2013). Choosing and Using Screening Criteria for Palliative Care Consultation in the ICU. *Critical Care Medicine*, 41(10), 2318–2327. <https://doi.org/10.1097/ccm.0b013e31828cf12c>
- Ng, M., Fleming, T., Robinson, M., Thomson, B., Graetz, N., Margono, C., Mullany, E. C., Biryukov, S., Abbafati, C., Abera, S. F., Abraham, J. P., Abu-Rmeileh, N. M. E., Achoki, T., AlBuhairan, F. S., Alemu, Z. A., Alfonso, R., Ali, M. K., Ali, R., Guzman, N. A., ... Gakidou, E. (2014). Global, regional, and national prevalence of overweight and obesity in children and adults during 1980–2013: A systematic analysis for the Global Burden of Disease Study 2013. *Lancet (London, England)*, 384(9945), 766–781. [https://doi.org/10.1016/S0140-6736\(14\)60460-8](https://doi.org/10.1016/S0140-6736(14)60460-8)
- Ngaage, D. L., Cowen, M. E., Griffin, S., Guvendik, L., & Cale, A. R. (2008). Early neurological complications after coronary artery bypass grafting and valve surgery in octogenarians☆. *European Journal of Cardio-Thoracic Surgery*, 33(4), 653–659. <https://doi.org/10.1016/j.ejcts.2008.01.017>
- Niculescu-Mizil, A., & Caruana, R. (2005). Obtaining calibrated probabilities from boosting. In Proc. Twenty-First Conference on Uncertainty in Artificial Intelligence. *AUAI Press, Edinburgh*, 413–420.
- NIH. (2018). *UMLS*. <https://www.nlm-nih-gov.treadwell.idm.oclc.org/research/umls/>.

- Nowak, K. L., Yaffe, K., Orwoll, E. S., Ix, J. H., You, Z., Barrett-Connor, E., Hoffman, A. R., & Chonchol, M. (2018). Serum Sodium and Cognition in Older Community-Dwelling Men. *Clinical Journal of the American Society of Nephrology*, 13(3), 366–374. <https://doi.org/10.2215/cjn.07400717>
- Oh, H. J., Kim, S., Park, J. T., Kim, S.-J., Han, S. H., Yoo, T.-H., Ryu, D.-R., Kang, S.-W., & Chung, Y. E. (2017). Baseline Chloride Levels are Associated with the Incidence of Contrast-Associated Acute Kidney Injury. *Scientific Reports*, 7(1). <https://doi.org/10.1038/s41598-017-17763-7>
- Oh, T. K., Do, S.-H., Jeon, Y.-T., Kim, J., Na, H.-S., & Hwang, J.-W. (2019). Association of Preoperative Serum Chloride Levels With Mortality and Morbidity After Noncardiac Surgery. *Anesthesia & Analgesia*, 129(6), 1494–1501. <https://doi.org/10.1213/ane.0000000000003958>
- Oh, T. K., Kim, C.-Y., Jeon, Y.-T., Hwang, J.-W., & Do, S.-H. (2019). Perioperative Hyperchloremia and its Association With Postoperative Acute Kidney Injury After Craniotomy for Primary Brain Tumor Resection. *Journal of Neurosurgical Anesthesiology*, 31(3), 311–317. <https://doi.org/10.1097/ana.0000000000000512>
- Oh, T. K., Song, I.-A., Kim, S. J., Lim, S. Y., Do, S.-H., Hwang, J.-W., Kim, J., & Jeon, Y.-T. (2018). Hyperchloremia and postoperative acute kidney injury: A retrospective analysis of data from the surgical intensive care unit. *Critical Care*, 22(1). <https://doi.org/10.1186/s13054-018-2216-5>
- Oliveira, H. F. de, Sampaio, A. L. L., & Oliveira, C. A. C. P. de. (2011). DATASUS as a instrument for developing otologic public health policies. *Brazilian Journal of Otorhinolaryngology*, 77(3), 369–372. <https://doi.org/10.1590/s1808-86942011000300016>
- Oliveros, H., & Villamor, E. (2008). Obesity and Mortality in Critically Ill Adults: A Systematic Review and Meta-analysis. *Obesity*, 16(3), 515–521. <https://doi.org/10.1038/oby.2007.102>
- Olson, K. R., Walsh, M., Garg, P., Steel, A., Mehta, S., Data, S., Petersen, R., Guarino, A. J., Bailey, E., & Bangsberg, D. R. (2017). Health hackathons: Theatre or substance? A survey assessment of outcomes from healthcare-focused hackathons in three countries. *BMJ Innovations*, 3(1), 37–44. <https://doi.org/10.1136/bmjinnov-2016-000147>
- O'Mahony, S., McHugh, M., Zallman, L., & Selwyn, P. (2003). Ventilator withdrawal: Procedures and outcomes. Report of a collaboration between a critical care division and a palliative care service. *Journal of Pain and Symptom Management*, 26(4), 954–961. [https://doi.org/10.1016/s0885-3924\(03\)00333-6](https://doi.org/10.1016/s0885-3924(03)00333-6)
- Ono, M., Brady, K., Easley, R. B., Brown, C., Kraut, M., Gottesman, R. F., & Hogue, C. W. (2014). Duration and magnitude of blood pressure below cerebral autoregulation threshold during cardiopulmonary bypass is associated with major morbidity and operative mortality. *The Journal of Thoracic and Cardiovascular Surgery*, 147(1), 483–489. <https://doi.org/10.1016/j.jtcvs.2013.07.069>
- Oud, L., & Watkins, P. (2015). Contemporary Trends of the Clinical Characteristics, Resource Utilization and Outcomes Among Patients with Dementia Admitted to ICU: A Population-Based Study. In *C24. SEDATION, DELIRIUM, AND NEUROCOGNITIVE DYSFUNCTION IN CRITICAL CARE* (Vol. 1–257, pp. A4014–A4014). American Thoracic Society. https://doi.org/10.1164/ajrcm-conference.2015.191.1_MeetingAbstracts.A4014
- Pai, V. M., Rodgers, M., Conroy, R., Luo, J., Zhou, R., & Seto, B. (2014). Workshop on using natural language processing applications for enhancing clinical decision making: An executive summary. *Journal of the American Medical Informatics Association*, 21(e1), e2–e5. <https://doi.org/10.1136/amiainl-2013-001896>
- Paiva, C. H. A., & Teixeira, L. A. (2014). Reforma sanitária e a criação do Sistema Único de Saúde: Notas sobre contextos e autores. *História, Ciências, Saúde-Manguinhos*, 21(1), 15–36. <https://doi.org/10.1590/s0104-59702014000100002>
- Palmer, B. (2014, April 10). *Are Hackathons the Future of Medical Innovation?* Slate Magazine. <https://slate.com/business/2014/04/medical-hackathons-is-this-the-future-of-health-care-innovation.html>
- Pan, J., Shaffer, R., Sinno, Z., Tyler, M., & Ghosh, J. (2017, July). The obesity paradox in ICU patients. *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. <https://doi.org/10.1109/embc.2017.8037576>
- Pandharipande, P. P., Girard, T. D., Jackson, J. C., Morandi, A., Thompson, J. L., Pun, B. T., Brummel, N. E., Hughes, C. G., Vasilevskis, E. E., Shintani, A. K., Moons, K. G., Geevarghese, S. K., Canonico, A., Hopkins, R. O., Bernard, G. R., Dittus, R. S., & Ely, E. W. (2013). Long-Term Cognitive Impairment after Critical Illness. *New England Journal of Medicine*, 369(14), 1306–1316. <https://doi.org/10.1056/nejmoa1301372>

- Paonessa, J. R., Brennan, T., Pimentel, M., Steinhaus, D., Feng, M., & Celi, L. A. (2015). Hyperdynamic left ventricular ejection fraction in the intensive care unit. *Critical Care*, 19(1). <https://doi.org/10.1186/s13054-015-1012-8>
- Papavasiliou, E. S., Brearley, S. G., Seymour, J. E., Brown, J., & Payne, S. A. (2014). Papavasiliou ES, Brearley SG, Seymour JE, Brown J, Payne SA, on behalf of EUROIMPACT. 'From Sedation to Continuous Sedation Until Death: How Has The Conceptual Basis of Sedation in End-of-Life Care Changed Over Time? J Pain Symptom Manage 2013; 5:706–723. *Journal of Pain and Symptom Management*, 47(2), 370. <https://doi.org/10.1016/j.jpainsymman.2013.12.227>
- Pappachan, J. V., Millar, B., Bennett, E. D., & Smith, G. B. (1999). Comparison of outcome from intensive care admission after adjustment for case mix by the APACHE III prognostic system. *Chest*, 115. <https://doi.org/10.1378/chest.115.3.802>
- Parameswaran, K., Todd, D. C., & Soth, M. (2006). Altered Respiratory Physiology in Obesity. *Canadian Respiratory Journal*, 13(4), 203–210. <https://doi.org/10.1155/2006/834786>
- Parshall, M. B., Schwartzstein, R. M., Adams, L., Banzett, R. B., Manning, H. L., Bourbeau, J., Calverley, P. M., Gift, A. G., Harver, A., Lareau, S. C., Mahler, D. A., Meek, P. M., & O'Donnell, D. E. (2012). An Official American Thoracic Society Statement: Update on the Mechanisms, Assessment, and Management of Dyspnea. *American Journal of Respiratory and Critical Care Medicine*, 185(4), 435–452. <https://doi.org/10.1164/rccm.201111-2042st>
- Partridge, J. S., Martin, F. C., Harari, D., & Dhesi, J. K. (2012). The delirium experience: What is the effect on patients, relatives and staff and what can be done to modify this? *International Journal of Geriatric Psychiatry*, 28(8), 804–812. <https://doi.org/10.1002/gps.3900>
- Patel, N., Baker, S. M., Walters, R. W., Kaja, A., Kandasamy, V., Abuzaid, A., & Modrykamien, A. M. (2016). Serum Hyperchloremia as a Risk Factor for Acute Kidney Injury in Patients with ST-Segment Elevation Myocardial Infarction Undergoing Percutaneous Coronary Intervention. *Baylor University Medical Center Proceedings*, 29(1), 7–11. <https://doi.org/10.1080/08998280.2016.11929341>
- Pearce, C. L., & Sims, H. P. (2002). Vertical versus shared leadership as predictors of the effectiveness of change management teams: An examination of aversive, directive, transactional, transformational, and empowering leader behaviors. *Group Dynamics: Theory, Research, and Practice*, 6(2), 172–197. <https://doi.org/10.1037/1089-2699.6.2.172>
- Pedregosa, F. (2011). Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, 12.
- Penn, A. S. and K., & Penn, A. S. and K. (2015, December 1). *Why Hackathons Are Bad For Innovation*. Fast Company. <https://www.fastcompany.com/3054023/why-hackathons-are-bad-for-innovation>
- Pinheiro, K. H. E., Azêdo, F. A., Areco, K. C. N., & Laranja, S. M. R. (2019). Risk factors and mortality in patients with sepsis, septic and non septic acute kidney injury in ICU. *Brazilian Journal of Nephrology*, 41(4), 462–471. <https://doi.org/10.1590/2175-8239-jbn-2018-0240>
- Pirracchio, R., Petersen, M. L., Carone, M., Rigon, M. R., Chevret, S., & Laan, M. J. van der. (2015). Mortality prediction in intensive care units with the Super ICU Learner Algorithm (SICULA): A population-based study. *The Lancet Respiratory Medicine*, 3(1), 42–52. [https://doi.org/10.1016/s2213-2600\(14\)70239-5](https://doi.org/10.1016/s2213-2600(14)70239-5)
- Plax, T. G., Kearney, P., McCroskey, J. C., & Richmond, V. P. (1986). Power in the classroom VI: Verbal control strategies, nonverbal immediacy and affective learning. *Communication Education*, 35(1), 43–55. <https://doi.org/10.1080/03634528609388318>
- Pollard, T. J., Johnson, A. E. W., Raffa, J. D., Celi, L. A., Mark, R. G., & Badawi, O. (2018). The eICU Collaborative Research Database, a freely available multi-center database for critical care research. *Scientific Data*, 5(1). <https://doi.org/10.1038/sdata.2018.178>
- Prof. Alexandre Vranjac. (2008). *Manual TBWEB versão 1.6*. São Paulo (Estado). Secretaria Estadual de Saúde. Coordenadoria de Controle de Doenças. Centro de Vigilância Epidemiológica . Divisão de Tuberculose. <http://bit.ly/2vIL7ul>
- Puntillo, K., Nelson, J. E., Weissman, D., Curtis, R., Weiss, S., Frontera, J., Gabriel, M., Hays, R., Lustbader, D., Mosenthal, A., Mulkerin, C., Ray, D., Bassett, R., Boss, R., Brasel, K., & Campbell, M. (2013). Palliative care in the ICU: relief of pain, dyspnea, and thirst—A report from the IPAL-ICU Advisory Board. *Intensive Care Medicine*, 40(2), 235–248. <https://doi.org/10.1007/s00134-013-3153-z>

- Puskas, J. D., Kilgo, P. D., Thourani, V. H., Lattouf, O. M., Chen, E., Vega, J. D., Cooper, W., Guyton, R. A., & Halkos, M. (2012). The Society of Thoracic Surgeons 30-Day Predicted Risk of Mortality Score Also Predicts Long-Term Survival. *The Annals of Thoracic Surgery*, 93(1), 26–35. <https://doi.org/10.1016/j.athoracsur.2011.07.086>
- Rangasamy, V., Henriques, T. S., Mathur, P. A., Davis, R. B., Mittleman, M. A., & Subramaniam, B. (2019). Changes in nonlinear dynamic complexity measures of blood pressure during anesthesia for cardiac surgeries using cardio pulmonary bypass. *Journal of Clinical Monitoring and Computing*. <https://doi.org/10.1007/s10877-019-00370-4>
- Rapoport, J. (2000). Patient Characteristics and ICU Organizational Factors That Influence Frequency of Pulmonary Artery Catheterization. *JAMA*, 283(19), 2559. <https://doi.org/10.1001/jama.283.19.2559>
- Raschka, S. MV. (2017). *Chapter 1. Giving computers the ability to learn from data. Python Machine Learning Machine learning and deep learning with Python, scikit-learn and Tensorflow*. (Birmingham, UK).
- Reich, D. L., Hossain, S., Krol, M., Baez, B., Patel, P., Bernstein, A., & Bodian, C. A. (2005). Predictors of Hypotension After Induction of General Anesthesia. *Anesthesia & Analgesia*, 101(3), 622–628. <https://doi.org/10.1213/01.ane.0000175214.38450.91>
- Rengel, K. F., Hayhurst, C. J., Pandharipande, P. P., & Hughes, C. G. (2019). Long-term Cognitive and Functional Impairments After Critical Illness. *Anesthesia & Analgesia*, 128(4), 772–780. <https://doi.org/10.1213/ane.0000000000004066>
- Rettig, T. C. D., Peelen, L. M., Geuzebroek, G. S. C., van Klei, W. A., Boer, C., van der Veer, J. W., Hofland, J., van de Garde, E. M. W., & Noordzij, P. G. (2017). Impact of Intraoperative Hypotension During Cardiopulmonary Bypass on Acute Kidney Injury After Coronary Artery Bypass Grafting. *Journal of Cardiothoracic and Vascular Anesthesia*, 31(2), 522–528. <https://doi.org/10.1053/j.jvca.2016.07.040>
- Reves, J. G., Buttner, E., Karp, R. B., Oparil, S., McDaniel, H. G., & Smith, L. R. (1984). Elevated catecholamines during cardiac surgery: Consequences of reperfusion of the postarrested heart. *The American Journal of Cardiology*, 53(6), 722–728. [https://doi.org/10.1016/0002-9149\(84\)90393-x](https://doi.org/10.1016/0002-9149(84)90393-x)
- Rincon, F., Mayer, S. A., Rivolta, J., Stillman, J., Boden-Albala, B., Elkind, M. S. V., Marshall, R., & Chong, J. Y. (2010). Impact of Delayed Transfer of Critically Ill Stroke Patients from the Emergency Department to the Neuro-ICU. *Neurocritical Care*, 13(1), 75–81. <https://doi.org/10.1007/s12028-010-9347-0>
- Robert, R., Le Gouge, A., & Kentish-Barnes, N. (2017). Terminal weaning or immediate extubation for withdrawing mechanical ventilation in critically ill patients (the ARREVE observational study). *Intensive Care Medicine*, 44(2), 275–276. <https://doi.org/10.1007/s00134-017-5037-0>
- Rocker, G. M., Heyland, D. K., Cook, D. J., Dodek, P. M., Kutsogiannis, D. J., & O'Callaghan, C. J. (2004). Most critically ill patients are perceived to die in comfort during withdrawal of life support: A Canadian multicentre study. *Canadian Journal of Anesthesia/Journal Canadien d'anesthésie*, 51(6), 623–630. <https://doi.org/10.1007/bf03018407>
- Rojas, J. C. (2018). Predicting intensive care unit readmission with machine learning using electronic health record data. *Ann. Am. Thorac. Soc.*, 15. <https://doi.org/10.1513/AnnalsATS.201710-787OC>
- Romano, L. (2018). Using Medical Subject Headings (MeSH) in Cataloging. *Technical Services Quarterly*, 35(2), 217–219. <https://doi.org/10.1080/07317131.2018.1425351>
- Rosner, M. H., & Ronco, C. (2010). Dysnatremias in the Intensive Care Unit. *Contributions to Nephrology*, 292–298. <https://doi.org/10.1159/000313769>
- Ross, M. K., Wei, W., & Ohno-Machado, L. (2014). “Big Data” and the Electronic Health Record. *Yearbook of Medical Informatics*, 23(01), 97–104. <https://doi.org/10.15265/iy-2014-0003>
- Rowan, K. M. (1993). Intensive Care Society's APACHE II study in Britain and Ireland–II: Outcome comparisons of intensive care units after adjustment for case mix by the American APACHE II method. *BMJ (Clin. Res. Ed.)*, 307. <https://doi.org/10.1136/bmj.307.6910.977>
- Rowan, K. M. (1994). Intensive Care Society's Acute Physiology and Chronic Health Evaluation (APACHE II) study in Britain and Ireland: A prospective, multicenter, cohort study comparing two methods for predicting outcome for adult intensive care patients. *Crit. Care Med.*, 22. <https://doi.org/10.1097/00003246-199409000-00007>

- Rubinfeld, G. D. (2004). Principles and practice of withdrawing life-sustaining treatments. *Critical Care Clinics*, 20(3), 435–451. <https://doi.org/10.1016/j.ccc.2004.03.005>
- Rubin, R. B. (2010). *Communication Research Measures II*. Routledge. <https://doi.org/10.4324/9780203871539>
- Russell, J. A., Walley, K. R., Singer, J., Gordon, A. C., Hébert, P. C., Cooper, D. J., Holmes, C. L., Mehta, S., Granton, J. T., Storms, M. M., Cook, D. J., Presneill, J. J., & Ayers, D. (2008). Vasopressin versus Norepinephrine Infusion in Patients with Septic Shock. *New England Journal of Medicine*, 358(9), 877–887. <https://doi.org/10.1056/nejmoa067373>
- Ryan, A. M. (2013). Will Value-Based Purchasing Increase Disparities in Care? *New England Journal of Medicine*, 369(26), 2472–2474. <https://doi.org/10.1056/nejmp1312654>
- Sadan, O., Singbartl, K., Kandiah, P. A., Martin, K. S., & Samuels, O. B. (2017). Hyperchloremia Is Associated With Acute Kidney Injury in Patients With Subarachnoid Hemorrhage. *Critical Care Medicine*, 45(8), 1382–1388. <https://doi.org/10.1097/ccm.0000000000002497>
- Salinas, J., Sprinkhuizen, S. M., Ackerson, T., Bernhardt, J., Davie, C., George, M. G., Gething, S., Kelly, A. G., Lindsay, P., Liu, L., Martins, S. C. O., Morgan, L., Norrving, B., Ribbers, G. M., Silver, F. L., Smith, E. E., Williams, L. S., & Schwamm, L. H. (2016). An International Standard Set of Patient-Centered Outcome Measures After Stroke. *Stroke*, 47(1), 180–186. <https://doi.org/10.1161/strokeaha.115.010898>
- Salmasi, V., Maheshwari, K., Yang, D., Mascha, E. J., Singh, A., Sessler, D. I., & Kurz, A. (2017). Relationship between Intraoperative Hypotension, Defined by Either Reduction from Baseline or Absolute Thresholds, and Acute Kidney and Myocardial Injury after Noncardiac Surgery. *Anesthesiology*, 126(1), 47–65. <https://doi.org/10.1097/aln.0000000000001432>
- Salome, C. M., King, G. G., & Berend, N. (2010). Physiology of obesity and effects on lung function. *Journal of Applied Physiology*, 108(1), 206–211. <https://doi.org/10.1152/jappphysiol.00694.2009>
- Samocha-Bonet, D., Justo, D., Rogowski, O., Saar, N., Abu-Abeid, S., Shenkerman, G., Shapira, I., Berliner, S., & Tomer, A. (2008). Platelet counts and platelet activation markers in obese subjects. *Mediators of Inflammation*, 2008, 834153. <https://doi.org/10.1155/2008/834153>
- Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3), 210–229. <https://doi.org/10.1147/rd.33.0210>
- Sanchez-Hurtado, L., Ángeles-Veléz, A., Tejeda-Huezo, B., García-Cruz, J., & Juárez-Cedillo, T. (2016). Validation of a prognostic score for mortality in elderly patients admitted to Intensive Care Unit. *Indian Journal of Critical Care Medicine*, 20(12), 695–700. <https://doi.org/10.4103/0972-5229.195702>
- Sanders, R. D., Tononi, G., Laureys, S., & Sleigh, J. W. (2012). Unresponsiveness ≠ Unconsciousness. *Anesthesiology*, 116(4), 946–959. <https://doi.org/10.1097/aln.0b013e318249d0a7>
- Sandfort, V., Johnson, A. E. W., Kunz, L. M., Vargas, J. D., & Rosing, D. R. (2018). Prolonged Elevated Heart Rate and 90-Day Survival in Acutely Ill Patients: Data From the MIMIC-III Database. *Journal of Intensive Care Medicine*, 34(8), 622–629. <https://doi.org/10.1177/0885066618756828>
- Sarin, S., & McDermott, C. (n.d.). *The effect of team leader characteristics on learning, knowledge application, and performance of cross-functional new product development teams*, *Decis. Sci.* 34 (4) (2003) 707–739.
- SAS. (2018). *Natural Language Processing*. https://www.sas.com/en_us/insights/analytics/what-is-natural-language-processing-nlp.html.
- Schnakers, C., Vanhaudenhuyse, A., Giacino, J., Ventura, M., Boly, M., Majerus, S., Moonen, G., & Laureys, S. (2009). Diagnostic accuracy of the vegetative and minimally conscious state: Clinical consensus versus standardized neurobehavioral assessment. *BMC Neurology*, 9(1). <https://doi.org/10.1186/1471-2377-9-35>
- Schneider, E. C. (2002). Racial Disparities in the Quality of Care for Enrollees in Medicare Managed Care. *JAMA*, 287(10), 1288. <https://doi.org/10.1001/jama.287.10.1288>
- Schoof, J., Lubahn, W., Baeumer, M., Kross, R., Wallesch, C.-W., Kozian, A., Huth, C., & Goertler, M. (2007). Impaired cerebral autoregulation distal to carotid stenosis/occlusion is associated with increased risk of stroke at cardiac surgery with cardiopulmonary bypass. *The Journal of Thoracic and Cardiovascular Surgery*, 134(3), 690–696. <https://doi.org/10.1016/j.jtcvs.2007.03.018>

- Seaman, J. B., Arnold, R. M., Scheunemann, L. P., & White, D. B. (2017). An Integrated Framework for Effective and Efficient Communication with Families in the Adult Intensive Care Unit. *Annals of the American Thoracic Society*, 14(6), 1015–1020. <https://doi.org/10.1513/annalsats.201612-965oi>
- Sessler, D. I., Bloomstone, J. A., Aronson, S., Berry, C., Gan, T. J., Kellum, J. A., Plumb, J., Mythen, M. G., Grocott, M. P. W., Edwards, M. R., Miller, T. E., Miller, T. E., Mythen, M. G., Grocott, M. P. W., Edwards, M. R., Ackland, G. L., Brudney, C. S., Cecconi, M., Ince, C., ... Hamilton, M. (2019). Perioperative Quality Initiative consensus statement on intraoperative blood pressure, risk and outcomes for elective surgery. *British Journal of Anaesthesia*, 122(5), 563–574. <https://doi.org/10.1016/j.bja.2019.01.013>
- Seymour, C. W., Liu, V. X., Iwashyna, T. J., Brunkhorst, F. M., Rea, T. D., Scherag, A., Rubenfeld, G., Kahn, J. M., Shankar-Hari, M., Singer, M., Deutschman, C. S., Escobar, G. J., & Angus, D. C. (2016). Assessment of Clinical Criteria for Sepsis: For the Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3). *JAMA*, 315(8), 762–774. <https://doi.org/10.1001/jama.2016.0288>
- Shao, M., Li, G., Sarvottam, K., Wang, S., Thongprayoon, C., Dong, Y., Gajic, O., & Kashani, K. (2016). Dyschloremia Is a Risk Factor for the Development of Acute Kidney Injury in Critically Ill Patients. *PLOS ONE*, 11(8), e0160322. <https://doi.org/10.1371/journal.pone.0160322>
- Shaw, A. D., Raghunathan, K., Peyerl, F. W., Munson, S. H., Paluszkiwicz, S. M., & Schermer, C. R. (2014). Association between intravenous chloride load during resuscitation and in-hospital mortality among patients with SIRS. *Intensive Care Medicine*, 40(12), 1897–1905. <https://doi.org/10.1007/s00134-014-3505-3>
- Shen, Y., Ru, W., Huang, X., & Zhang, W. (2018). Time-related association between fluid balance and mortality in sepsis patients: Interaction between fluid balance and haemodynamics. *Scientific Reports*, 8(1). <https://doi.org/10.1038/s41598-018-28781-4>
- Shi, H.-Y., Lee, K.-T., Lee, H.-H., Ho, W.-H., Sun, D.-P., Wang, J.-J., & Chiu, C.-C. (2012). Comparison of Artificial Neural Network and Logistic Regression Models for Predicting In-Hospital Mortality after Primary Liver Cancer Surgery. *PLoS ONE*, 7(4), e35781. <https://doi.org/10.1371/journal.pone.0035781>
- Shippee, T. P., Ferraro, K. F., & Thorpe, R. J. (2011). Racial disparity in access to cardiac intensive care over 20 years. *Ethnicity & Health*, 16(2), 145–165. <https://doi.org/10.1080/13557858.2010.544292>
- Shouval, R., Bondi, O., Mishan, H., Shimoni, A., Unger, R., & Nagler, A. (2013). Application of machine learning algorithms for clinical predictive modeling: A data-mining approach in SCT. *Bone Marrow Transplantation*, 49(3), 332–337. <https://doi.org/10.1038/bmt.2013.146>
- Shouval, Roni, Hadanny, A., Shlomo, N., Iakobishvili, Z., Unger, R., Zahger, D., Alcalai, R., Atar, S., Gottlieb, S., Matetzky, S., Goldenberg, I., & Beigel, R. (2017). Machine learning for prediction of 30-day mortality after ST elevation myocardial infarction: An Acute Coronary Syndrome Israeli Survey data mining study. *International Journal of Cardiology*, 246, 7–13. <https://doi.org/10.1016/j.ijcard.2017.05.067>
- Shrime, M. G., Ferket, B. S., Scott, D. J., Lee, J., Barragan-Bradford, D., Pollard, T., Arabi, Y. M., Al-Dorzi, H. M., Baron, R. M., Hunink, M. G. M., Celi, L. A., & Lai, P. S. (2016). Time-Limited Trials of Intensive Care for Critically Ill Patients With Cancer. *JAMA Oncology*, 2(1), 76. <https://doi.org/10.1001/jamaoncol.2015.3336>
- Silva, J. M., Neves, E. F., Santana, T. C., Ferreira, U. P., Marti, Y. N., & Silva, J. M. C. (2009). The Importance of Intraoperative Hyperchloremia. *Brazilian Journal of Anesthesiology*, 59(3), 304–313. <https://doi.org/10.1590/s0034-70942009000300005>
- Simon, P. M., Schwartzstein, R. M., Weiss, J. W., Lahive, K., Fencl, V., Teghtsoonian, M., & Weinberger, S. E. (1989). Distinguishable Sensations of Breathlessness Induced in Normal Volunteers. *American Review of Respiratory Disease*, 140(4), 1021–1027. <https://doi.org/10.1164/ajrccm/140.4.1021>
- Sirgo, G., Esteban, F., Gómez, J., Moreno, G., Rodríguez, A., Blanch, L., Guardiola, J. J., Gracia, R., Haro, L. D., & Bodí, M. (2018). Validation of the ICU-DaMa tool for automatically extracting variables for minimum dataset and quality indicators: The importance of data quality assessment. *International Journal of Medical Informatics*, 112, 166–172. <https://doi.org/10.1016/j.ijmedinf.2018.02.007>
- Skinner, J., Chandra, A., Staiger, D., Lee, J., & McClellan, M. (2005). Mortality After Acute Myocardial Infarction in Hospitals That Disproportionately Treat Black Patients. *Circulation*, 112(17), 2634–2641. <https://doi.org/10.1161/circulationaha.105.543231>

- Smedira, N. G., Evans, B. H., Grais, L. S., Cohen, N. H., Lo, B., Cooke, M., Schechter, W. P., Fink, C., Epstein-Jaffe, E., May, C., & Luce, J. M. (1990). Withholding and Withdrawal of Life Support from the Critically Ill. *New England Journal of Medicine*, 322(5), 309–315. <https://doi.org/10.1056/nejm199002013220506>
- Soffler, M. I., Rose, A., Hayes, M. M., Banzett, R., & Schwartzstein, R. M. (2017). Treatment of Acute Dyspnea with Morphine to Avert Respiratory Failure. *Annals of the American Thoracic Society*, 14(4), 584–588. <https://doi.org/10.1513/annalsats.201611-922cc>
- Soto, G. J., Martin, G. S., & Gong, M. N. (2013). Healthcare Disparities in Critical Illness. *Critical Care Medicine*, 41(12), 2784–2793. <https://doi.org/10.1097/ccm.0b013e3182a84a43>
- Steinhauser, K. E. (2000). Factors Considered Important at the End of Life by Patients, Family, Physicians, and Other Care Providers. *JAMA*, 284(19), 2476. <https://doi.org/10.1001/jama.284.19.2476>
- Sterns, R. H. (2015). Disorders of Plasma Sodium—Causes, Consequences, and Correction. *New England Journal of Medicine*, 372(1), 55–65. <https://doi.org/10.1056/nejmra1404489>
- Sterns, R. H. (2018). Treatment of Severe Hyponatremia. *Clinical Journal of the American Society of Nephrology*, 13(4), 641–649. <https://doi.org/10.2215/cjn.10440917>
- Stevenson, E. K., Rubenstein, A. R., Radin, G. T., Wiener, R. S., & Walkey, A. J. (2014). Two Decades of Mortality Trends Among Patients With Severe Sepsis. *Critical Care Medicine*, 42(3), 625–631. <https://doi.org/10.1097/ccm.0000000000000026>
- Stewart, P. A. (1978). Independent and dependent variables of acid-base control. *Respiration Physiology*, 33(1), 9–26. [https://doi.org/10.1016/0034-5687\(78\)90079-8](https://doi.org/10.1016/0034-5687(78)90079-8)
- Stoica, S. (2002). Preoperative risk prediction and intraoperative events in cardiac surgery. *European Journal of Cardio-Thoracic Surgery*, 21(1), 41–46. [https://doi.org/10.1016/s1010-7940\(01\)01077-6](https://doi.org/10.1016/s1010-7940(01)01077-6)
- Suán, C., Pérez-Torres, C., & Herrera, A. (1997). Postoperative mortality in a general hospital. *Revista Espanola De Anestesiologia Y Reanimacion*, 44(7), 267–272.
- Suetrong, B., Pisitsak, C., Boyd, J. H., Russell, J. A., & Walley, K. R. (2016). Hyperchloremia and moderate increase in serum chloride are associated with acute kidney injury in severe sepsis and septic shock patients. *Critical Care*, 20(1). <https://doi.org/10.1186/s13054-016-1499-7>
- Sullivan, A. M., Rock, L. K., Gadmer, N. M., Norwich, D. E., & Schwartzstein, R. M. (2016). The Impact of Resident Training on Communication with Families in the ICU: Resident and Family Outcomes. *Annals of the American Thoracic Society*. <https://doi.org/10.1513/annalsats.201508-495oc>
- Sun, L. Y., Chung, A. M., Farkouh, M. E., van Diepen, S., Weinberger, J., Bourke, M., & Ruel, M. (2018). Defining an Intraoperative Hypotension Threshold in Association with Stroke in Cardiac Surgery. *Anesthesiology*, 129(3), 440–447. <https://doi.org/10.1097/aln.0000000000002298>
- Sun, L. Y., Wijesundera, D. N., Tait, G. A., & Beattie, W. S. (2015). Association of Intraoperative Hypotension with Acute Kidney Injury after Elective Noncardiac Surgery. *Anesthesiology*, 123(3), 515–523. <https://doi.org/10.1097/aln.0000000000000765>
- Takaki, J., Petersen, T., & Ericson, G. (2018, 2/2018). *Boosted Decision Tree Regression*. <https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/boosted-decision-tree-regression>.
- Tannenbaum, S. I., Beard, R. L., & Salas, E. (1992). Chapter 5 Team Building and its Influence on Team Effectiveness: An Examination of Conceptual and Empirical Developments. In *Issues, Theory, and Research in Industrial/Organizational Psychology* (pp. 117–153). Elsevier. [https://doi.org/10.1016/s0166-4115\(08\)62601-1](https://doi.org/10.1016/s0166-4115(08)62601-1)
- Tanner A. (2016). *Our bodies, our data: How companies make billions selling our medical records*. Boston: Beacon Press.
- Teasdale, G., & Jennett, B. (1974). Assessment Of Coma And Impaired Consciousness. *The Lancet*, 304(7872), 81–84. [https://doi.org/10.1016/s0140-6736\(74\)91639-0](https://doi.org/10.1016/s0140-6736(74)91639-0)

- Teasdale, G. M., Pettigrew, L. E. I., Wilson, J. t. L., Murray, G., & Jennett, B. (1998). Analyzing Outcome of Treatment of Severe Head Injury: A Review and Update on Advancing the Use of the Glasgow Outcome Scale. *Journal of Neurotrauma*, 15(8), 587–597. <https://doi.org/10.1089/neu.1998.15.587>
- Tello, V. G., Rodríguez, J. Á., Reiz, A. N., Sánchez, J. A. G., Barbará, A. H. A. de, Fresneda, M. M., Valdeolmillos, P. M., Arfelis, J. M. N., Varela, I. P., & Chicharro, M. C. (2011). Technical and functional standards and implementation of a clinical information system in intensive care units. *Medicina Intensiva (English Edition)*, 35(8), 484–496. <https://doi.org/10.1016/j.medicine.2011.12.001>
- Teno, J. M. (2004). Family Perspectives on End-of-Life Care at the Last Place of Care. *JAMA*, 291(1), 88. <https://doi.org/10.1001/jama.291.1.88>
- Tin Kam Ho. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8), 832–844. <https://doi.org/10.1109/34.709601>
- Tolles, J., & Meurer, W. J. (2016). Logistic Regression. *JAMA*, 316(5), 533. <https://doi.org/10.1001/jama.2016.7653>
- Toyonaga, Y., & Kikura, M. (2017). Hyperchloremic acidosis is associated with acute kidney injury after abdominal surgery. *Nephrology*, 22(9), 720–727. <https://doi.org/10.1111/nep.12840>
- Truog, R. D., Brock, D. W., & White, D. B. (2012). Should patients receive general anesthesia prior to extubation at the end of life?*. *Critical Care Medicine*, 40(2), 631–633. <https://doi.org/10.1097/ccm.0b013e3182413b8a>
- Truog, R. D., Campbell, M. L., Curtis, J. R., Haas, C. E., Luce, J. M., Rubenfeld, G. D., Rushton, C. H., & Kaufman, D. C. (2008). Recommendations for end-of-life care in the intensive care unit: A consensus statement by the American College of Critical Care Medicine. *Critical Care Medicine*, 36(3), 953–963. <https://doi.org/10.1097/ccm.0b013e3181659096>
- Truog, R. D., Cist, A. F. M., Brackett, S. E., Burns, J. P., Curley, M. A. Q., Danis, M., DeVita, M. A., Rosenbaum, S. H., Rothenberg, D. M., Sprung, C. L., Webb, S. A., Wlody, G. S., & Hurford, W. E. (2001). Recommendations for end-of-life care in the intensive care unit: The Ethics Committee of the Society of Critical Care Medicine. *Critical Care Medicine*, 29(12), 2332–2348. <https://doi.org/10.1097/00003246-200112000-00017>
- Tucker, R. (2005). *Extensible Markup Language (XML) databases*. McGraw-Hill Professional. <https://doi.org/10.1036/1097-8542.yb021025>
- Tyler, P. D., Stone, D. J., Geisler, B. P., McLennan, S., Celi, L. A., & Rush, B. (2018). Racial and Geographic Disparities in Interhospital ICU Transfers†. *Critical Care Medicine*, 46(1), e76–e80. <https://doi.org/10.1097/ccm.0000000000002776>
- van Beinum, A., Hornby, L., Ward, R., Ramsay, T., & Dhanani, S. (2015). Variations in the Operational Process of Withdrawal of Life-Sustaining Therapy. *Critical Care Medicine*, 43(10), e450–e457. <https://doi.org/10.1097/ccm.0000000000001163>
- van Drimmelen-Krabbe, J. J., Bradley, W. G., Orgogozo, J. M., & Sartorius, N. (1998). The application of the International Statistical Classification of Diseases to neurology: ICD-10 NA. *Journal of the Neurological Sciences*, 161(1), 2–9. [https://doi.org/10.1016/s0022-510x\(98\)00217-2](https://doi.org/10.1016/s0022-510x(98)00217-2)
- van Waes, J. A. R., van Klei, W. A., Wijeysondera, D. N., van Wolfswinkel, L., Lindsay, T. F., & Beattie, W. S. (2016). Association between Intraoperative Hypotension and Myocardial Injury after Vascular Surgery. *Anesthesiology*, 124(1), 35–44. <https://doi.org/10.1097/aln.0000000000000922>
- Verbalis, J. G., Ellison, H., Hobart, M., Krasa, H., Ouyang, J., Czerwiec, F. S., Barnum, O., Devireddy, K., Ellison, H., Franco, M., Goldstein, F., Hack, T. C., Kaveh, K., Khojasteh, A., LaFata, J., Lillestol, M. J., Olelewe, S., Rosenbaum, D., Rosner, M. H., ... Wells, W. (2016). Tolvaptan and Neurocognitive Function in Mild to Moderate Chronic Hyponatremia: A Randomized Trial (INSIGHT). *American Journal of Kidney Diseases*, 67(6), 893–901. <https://doi.org/10.1053/j.ajkd.2015.12.024>
- Vincent, Jean-Louis, de Mendonca, A., Cantraine, F., Moreno, R., Takala, J., Suter, P. M., Sprung, C. L., Colardyn, F., & Blecher, S. (1998). Use of the SOFA score to assess the incidence of organ dysfunction/failure in intensive care units. *Critical Care Medicine*, 26(11), 1793–1800. <https://doi.org/10.1097/00003246-199811000-00016>
- Vincent, J.-L., Moreno, R., Takala, J., Willatts, S., Mendonça, A. D., Bruining, H., Reinhart, C. K., Suter, P. M., & Thijs, L. G. (1996). The SOFA (Sepsis-related Organ Failure Assessment) score to describe organ dysfunction/failure. *Intensive Care Medicine*, 22(7), 707–710. <https://doi.org/10.1007/bf01709751>

- von Elm, E., Altman, D. G., Egger, M., Pocock, S. J., Gøtzsche, P. C., & Vandenbroucke, J. P. (2007). The Strengthening of Reporting of Observational Studies in Epidemiology (STROBE) Statement: Guidelines for Reporting Observational Studies. *Annals of Internal Medicine*, 147(8), 573. <https://doi.org/10.7326/0003-4819-147-8-200710160-00010>
- Wainberg, M., Merico, D., Delong, A., & Frey, B. J. (2018). Deep learning in biomedicine. *Nat. Biotechnol.*, 36. <https://doi.org/10.1038/nbt.4233>
- Wald, R. (2010). Impact of Hospital-Associated Hyponatremia on Selected Outcomes. *Archives of Internal Medicine*, 170(3), 294. <https://doi.org/10.1001/archinternmed.2009.513>
- Walraven, C. van, Austin, P. C., Jennings, A., Quan, H., & Forster, A. J. (2009). A Modification of the Elixhauser Comorbidity Measures Into a Point System for Hospital Death Using Administrative Data. *Medical Care*, 47(6), 626–633. <https://doi.org/10.1097/mlr.0b013e31819432e5>
- Walsh, M., Devereaux, P. J., Garg, A. X., Kurz, A., Turan, A., Rodseth, R. N., Cywinski, J., Thabane, L., & Sessler, D. I. (2013). Relationship between Intraoperative Mean Arterial Pressure and Clinical Outcomes after Noncardiac Surgery. *Anesthesiology*, 119(3), 507–515. <https://doi.org/10.1097/aln.0b013e3182a10e26>
- Wang, S., Lv, S., Guan, Y., Gao, G., Li, J., Hei, F., & Long, C. (2011). Cardiopulmonary Bypass Techniques and Clinical Outcomes in Beijing Fuwai Hospital: A Brief Clinical Review. *ASAIO Journal*, 57(5), 414–420. <https://doi.org/10.1097/mat.0b013e318227fa72>
- Waudby-Smith, I. E. R., Tran, N., Dubin, J. A., & Lee, J. (2018). Sentiment in nursing notes as an indicator of out-of-hospital mortality in intensive care patients. *PLOS ONE*, 13(6), e0198687. <https://doi.org/10.1371/journal.pone.0198687>
- Weis, F., Kilger, E., Beiras-Fernandez, A., Nassau, K., Reuter, D., Goetz, A., Lamm, P., Reindl, L., & Briegel, J. (2006). Association between vasopressor dependence and early outcome in patients after cardiac surgery. *Anaesthesia*, 61(10), 938–942. <https://doi.org/10.1111/j.1365-2044.2006.04779.x>
- Wendt, D., Osswald, B. R., Kayser, K., Thielmann, M., Tossios, P., Massoudy, P., Kamler, M., & Jakob, H. (2009). Society of Thoracic Surgeons Score Is Superior to the EuroSCORE Determining Mortality in High Risk Patients Undergoing Isolated Aortic Valve Replacement. *The Annals of Thoracic Surgery*, 88(2), 468–475. <https://doi.org/10.1016/j.athoracsur.2009.04.059>
- Wesselink, E. M., Kappen, T. H., Torn, H. M., Slooter, A. J. C., & van Klei, W. A. (2018). Intraoperative hypotension and the risk of postoperative adverse outcomes: A systematic review. *British Journal of Anaesthesia*, 121(4), 706–721. <https://doi.org/10.1016/j.bja.2018.04.036>
- White, D. B., Cua, S. M., Walk, R., Pollice, L., Weissfeld, L., Hong, S., Landefeld, C. S., & Arnold, R. M. (2012). Nurse-Led Intervention to Improve Surrogate Decision Making for Patients with Advanced Critical Illness. *American Journal of Critical Care*, 21(6), 396–409. <https://doi.org/10.4037/ajcc2012223>
- White, Douglas B., Engelberg, R. A., Wenrich, M. D., Lo, B., & Curtis, J. R. (2008). The Language of Prognostication in Intensive Care Units. *Medical Decision Making*, 30(1), 76–83. <https://doi.org/10.1177/0272989x08317012>
- White, Douglas B., Ernecoff, N., Buddadhumaruk, P., Hong, S., Weissfeld, L., Curtis, J. R., Luce, J. M., & Lo, B. (2016). Prevalence of and Factors Related to Discordance About Prognosis Between Physicians and Surrogate Decision Makers of Critically Ill Patients. *JAMA*, 315(19), 2086. <https://doi.org/10.1001/jama.2016.5351>
- Williams, J. F., Zimmerman, J. E., Wagner, D. P., Hawkins, M., & Knaus, W. A. (1995). African-American and white patients admitted to the intensive care unit. *Critical Care Medicine*, 23(4), 626–636. <https://doi.org/10.1097/00003246-199504000-00009>
- Willms, D. C., & Brewer, J. A. (2005). Survey of Respiratory Therapists' Attitudes and Concerns Regarding Terminal Extubation. *Respiratory Care*, 50(8), 1046–1049.
- Wise, E. S., Hocking, K. M., & Brophy, C. M. (2015). Prediction of in-hospital mortality after ruptured abdominal aortic aneurysm repair using an artificial neural network. *Journal of Vascular Surgery*, 62(1), 8–15. <https://doi.org/10.1016/j.jvs.2015.02.038>
- Wolters Kluwer. (2016). *EHRs and healthcare information technology: The solution for smarter decisions and better bare in Brazi*. <http://bit.ly/2uzrvrA>

- Wu, J. T., Deroncourt, F., Gehrmann, S., Tyler, P. D., Moseley, E. T., Carlson, E. T., Grant, D. W., Li, Y., Welt, J., & Celi, L. A. (2018). Behind the scenes: A medical natural language processing project. *International Journal of Medical Informatics*, 112, 68–73. <https://doi.org/10.1016/j.ijmedinf.2017.12.003>
- Wu, X., Jiang, Z., Ying, J., Han, Y., & Chen, Z. (2017). Optimal blood pressure decreases acute kidney injury after gastrointestinal surgery in elderly hypertensive patients: A randomized study. *Journal of Clinical Anesthesia*, 43, 77–83. <https://doi.org/10.1016/j.jclinane.2017.09.004>
- Yadav, R. L., Yadav, P. K., Yadav, L. K., Agrawal, K., Sah, S. K., & Islam, M. N. (2017). Association between obesity and heart rate variability indices: An intuition toward cardiac autonomic alteration & a risk of CVD. *Diabetes, Metabolic Syndrome and Obesity: Targets and Therapy*, Volume 10, 57–64. <https://doi.org/10.2147/dmso.s123935>
- Yergan, J., Flood, A. B., LoGerfo, J. P., & Diehr, P. (1987). Relationship Between Patient Race and the Intensity of Hospital Services. *Medical Care*, 25(7), 592–603. <https://doi.org/10.1097/00005650-198707000-00003>
- Yessayan, L., Neyra, J. A., Canepa-Escaro, F., Vasquez-Rios, G., Heung, M., & Yee, J. (2017). Effect of hyperchloremia on acute kidney injury in critically ill septic patients: A retrospective cohort study. *BMC Nephrology*, 18(1). <https://doi.org/10.1186/s12882-017-0750-z>
- Yufeng, G. (2017, September 7). *The 7 Steps of Machine Learning*. Medium. <https://towardsdatascience.com/the-7-steps-of-machine-learning-2877d7e5548e>
- Zaccaro, S. J., Heinen, B., & Shuffler, M. (2009). Team leadership and team effectiveness. In *Team effectiveness in complex organizations: Cross-disciplinary perspectives and approaches* (pp. 83–111). Routledge/Taylor & Francis Group.
- Zaccaro, S. J., Rittman, A. L., & Marks, M. A. (2001). Team leadership. *The Leadership Quarterly*, 12(4), 451–483. [https://doi.org/10.1016/s1048-9843\(01\)00093-5](https://doi.org/10.1016/s1048-9843(01)00093-5)
- Zhang, Z., Xu, X., Fan, H., Li, D., & Deng, H. (2013). Higher serum chloride concentrations are associated with acute kidney injury in unselected critically ill patients. *BMC Nephrology*, 14(1). <https://doi.org/10.1186/1471-2369-14-235>
- Zhao, Y., Wang, Y., Wang, H., Yan, B., Shen, F., Peterson, K. J., Rocca, W. A., Sauver, J. S., & Liu, H. (2018, December). Annotating Cohort Data Elements with OHDSI Common Data Model to Promote Research Reproducibility. *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. <https://doi.org/10.1109/bibm.2018.8621269>
- Zier, L. S., Burack, J. H., Micco, G., Chipman, A. K., Frank, J. A., & White, D. B. (2009). Surrogate Decision Makers' Responses to Physicians' Predictions of Medical Futility. *Chest*, 136(1), 110–117. <https://doi.org/10.1378/chest.08-2753>
- Zimmerman, J. E., Kramer, A. A., McNair, D. S., & Malila, F. M. (2006). Acute Physiology and Chronic Health Evaluation (APACHE) IV: hospital mortality assessment for today's critically ill patients. *Crit. Care Med.*, 34. <https://doi.org/10.1097/01.CCM.0000215112.84523.F0>
- Zimmerman, J. E., Kramer, A. A., McNair, D. S., Malila, F. M., & Shaffer, V. L. (2006). Intensive care unit length of stay: Benchmarking based on Acute Physiology and Chronic Health Evaluation (APACHE) IV. *Crit. Care Med.*, 34. <https://doi.org/10.1097/01.CCM.0000240233.01711.D9>
- Zimmerman, Jack E, Kramer, A. A., & Knaus, W. A. (2013). Changes in hospital mortality for United States intensive care unit admissions from 1988 to 2012. *Critical Care*, 17(2), R81. <https://doi.org/10.1186/cc12695>

APPENDIX I: BIG DATA AND MACHINE LEARNING IN CRITICAL CARE: OPPORTUNITIES FOR COLLABORATIVE RESEARCH

List of authors who have participated in the Organizing Committee of Critical Care Datathon Madrid 2017.

Critical Care Department, Hospital Clínico San Carlos, Madrid

Antonio Núñez Reiz MD*, Miguel Sánchez García MD PhD*, Fernando Martínez Sagasti MD PhD, Manuel Álvarez González MD PhD, Antonio Blesa Malpica MD PhD, Juan Carlos Martín Benítez MD PhD, Mercedes Nieto Cabrera MD PhD, Ángela del Pino Ramírez MD, José Miguel Gil Perdomo MD, Jesús Prada Alonso

*Corresponding authors.

MIT Critical Data group, Massachusetts Institute of Technology, Boston

Leo Anthony Celi MD MS MPH, Miguel Ángel Armengol de la Hoz MBE, Rodrigo Deliberato MD PhD, Kenneth Paik MD MS MBA, Tom Pollard PhD, Jesse Raffa PhD, Felipe Torres

Unidad de Innovación, Instituto de Investigación Sanitaria San Carlos

Julio Mayol MD PhD, Joan Chafer MD, Arturo González Ferrer PhD, Angel Rey, Henar Gonzalez Luengo

Life Supporting Technologies, Universidad Politécnica Madrid

Giuseppe Fico PhD, Ivana Lombroni MsC, Liss Hernandez MsC, Laura Lopez MsC, Beatriz Merino MsC, Maria Fernanda Cabrera PhD, Maria Teresa Arredondo PhD.

Intensive Care Unit, Hospital Universitario Joan XXIII, Instituto de Investigación Sanitaria Pere Virgili, Rovira i Virgili University, Tarragona, Spain

María Bodí MD PhD, Josep Gómez PhD, Alejandro Rodríguez MD PhD

APPENDIX II: BIG DATA AND MACHINE LEARNING IN INTENSIVE CARE UNITS

Below is a brief overview of the most commonly used software:

Programming languages

R(48): is an open-source programming language specially designed for statistics and machine learning. There are free programming environments in this language, such as RStudio, which has functional and intuitive user interfaces that makes it much easier for the programmer to analyze the data and to build and test of the models.

Python(49): the alternative to the previous one (at least one of the two is necessary to master to be able to work in this field). Open source libraries for ML have been developed traditionally in this language, although today there are versions for other environments such as R,Java or C++. Its learning is somewhat more complex, but in return it is a more powerful language and efficient.

SQL,Structured Query Language(50): used for the exploration and exploitation of relational data, which is where structured clinical information from our patients. There are equivalent languages for database queries not SQL like Cassandra or HBase, which allow to store unstructured information in a accessible format for analysis. Code libraries for data analysis (libraries, frameworks).

Scikit-learn(51): is the most used code library in ML, with implementation for the main algorithms, which can be applied in a very simple and efficient way to our data. In the GitHub repository we have included several examples that use this library, which is originally programmed in Python, although there are versions for other languages.

NLTK(52) and tidytext(53) are two examples of libraries specifically designed for natural language processing.

Tensorflow(54) is Google's major contribution to machine learning, and is used primarily in the context of neural networks and "deep learning", although the library is being extended to other applications. It enables the highly efficient use of dedicated hardware such as GPUs and TPUs (specialized chips that allow many operations to be performed in parallel), which increase the speed of results by several orders of magnitude.

Spark MLib(55) and DeepLearning4j(56) are two of the contributions of the Java programmer community to machine learning. Although initially Java was considered an inappropriate language because it was less efficient, today these libraries provide a processing speed comparable to those previously considered faster. Auxiliary code libraries for data extraction, graphical presentation, vectorization and others: this section includes NumPy, SciPy, pandas, matplotlib and sciborn.

Table 1_Sup. Machine Learning Methodologies

Learning strategy	Methodology	Summary
Supervised	Perceptron, Adaline(57)	It uses artificial "neurons" that receive the values of the independent variables as inputs.
Supervised	Logistic Regression(58)	It uses linear functions of the independent variables with a sigmoid activation function (logit), very useful in classification of linearly separable cases.
Supervised	Support vector machine (SVM)(58)	It extends the algorithm used in Perceptron using the concept of maximizing the separation margins between the different classes. It allows the classification of non-linearly separable cases.
Supervised	Decision trees(58)	We use the independent variables to create a decision tree that maximizes the information gain in each branch until an optimal classification algorithm is reached.
Supervised	Random Forests(58), Random Forests, GBTs (Ensemble model)	Uses a set of decision trees and bootstrapping and gradient-enhancing techniques to improve the results of individual decision trees
Supervised	Adaptive enhancement(59),	Adaboost (Ensemble Model) uses simple classifiers that cascade through a learning-by-doing strategy without replacement.
Supervised	Bayesian networks(60)	Probabilistic approach that represents the interdependence between variables such as acyclic directional plots. They are used, for example, to estimate the presence of a disease as a function of a set of symptoms.
Supervised	Neural networks and deep learning, "deep learning"(61)	Based on the operational characteristics of biological

		neural networks, with layers of nodes that accumulate information and make inferences that are stored in their interconnections. They allow efficient classification of non-linearly separable cases. Useful for exploring complex relationships between system input and output variables.
Supervised	Natural Language Processing(62)	Interprets and analyzes natural human language to be processed by a computer in an efficient way, either in codification or classification of contents or for the extraction of relevant information.
Unsupervised	KNN, Nearby values, Knearest neighbors(63)	It is a technique of progressive grouping of records to obtain a classification of these according to the proximity of the values of the variables
Unsupervised	Association rules(64, 65)	It brings to light interesting relationships between variables in large databases.
Unsupervised	Inductive logic(66)	It uses functional and logical programming to derive software models that replicate the knowledge and relationships that exist in the system.
Supervised or unsupervised	Learning characteristics or representation(67)	Techniques to discover autonomously the type of representation necessary for the correct detection of data characteristics for classification from untreated or "raw" values.
For reinforcement	Monte Carlo, QLearning, SARSA(68)	The system must pursue an objective and learn as it explores the environment with which it interacts, which may not be known in advance.
Metaheuristics	Genetic Algorithms(69)	Solution optimization strategies applying natural selection, evolution, mutation and crossover mechanisms

		to other algorithms, representing the algorithm parameters as "genes" and the solutions as "phenotypes".

APPENDIX III: SEVERITY OF ILLNESS SCORES MAY MISCLASSIFY CRITICALLY ILL OBESE PATIENTS

Code

Description of the obesity project done with MIMIC-III database on March 2017.

Study question: "to compare the deviation of laboratory results utilized in scoring systems from baseline to ICU admission in both obese and normal weight patients, adjusted for the severity of score illness (SAPS-II or SOFA)"

Step by step:

- Cohort creation - note you need to create a new table with a specific name for each cohort (normal weight and obese).

```
-- Cohort creation
-- Our cohort: 1st ICU admission of patients >16 years old, with height, weight and laboratory
-- results at least 3 days and at most 1 year before the hospital admission
-- create table public.XXXX . as -- use this command to create a public table
WITH
t1 AS (
SELECT
  adm.subject_id,
  adm.hadm_id,
  adm.admittime,
  RANK() OVER (PARTITION BY adm.subject_id ORDER BY adm.admittime) AS
  hadm_id_order,
  EXTRACT('epoch'
FROM
  adm.admittime - pat.dob) / 60.0 / 60.0 / 24.0 / 365.242 AS age
FROM
  mimiciii.admissions adm,
  mimiciii.patients pat
WHERE
  adm.subject_id = pat.subject_id
ORDER BY
  subject_id
),
t2 AS -- first hospital admission
(
SELECT
  subject_id,
  hadm_id,
  EXTRACT
FROM
  (epoch
```

```

    admittance) AS admittance,
    age
FROM
    t1
WHERE
    hadm_id_order = 1),
    t3_lab AS -- all lab values (hemoglobin, white blood cells, platelets, sodium, potassium,
creatinine, BUN, bicarbonate, glucose and bilirubin)
(
    SELECT
        DISTINCT subject_id,
        hadm_id,
        EXTRACT(epoch
        FROM
            charttime) AS charttime -- , valuenum
FROM
    mimiciii.labevents
WHERE
    itemid IN (50912,
        50811,
        51222,
        51300,
        51301,
        51265,
        50983,
        50824,
        50971,
        50822,
        51006,
        51081,
        50882,
        50803,
        50902,
        50806,
        50809,
        50931,
        50960,
        50808)
    AND hadm_id IS NULL
ORDER BY
    subject_id,
    charttime),
    adultswithlabs AS (
        -- combining inclusion criteria
    SELECT
        DISTINCT t2.subject_id,

```



```

t2.hadm_id
FROM
t2,
t3_lab AS t3
WHERE
t2.age >= 16
AND t3.charttime BETWEEN t2.admittime-365.25*24*3600
AND t2.admittime-3*24*3600
AND t3.subject_id=t2.subject_id
ORDER BY
t2.hadm_id
t3_wt AS -- weight
(
SELECT
DISTINCT subject_id,
icustay_id,
EXTRACT(epoch
FROM
charttime) AS charttime,
valuenum AS weight
FROM
mimiciii.chartevents
WHERE
valuenum IS NOT NULL
AND itemid IN (581,
580,
224639,
226512)
ORDER BY
icustay_id,
charttime
) AS t1_icu
SELECT
icu.*,
RANK() OVER (PARTITION BY icu.subject_id ORDER BY icu.intime) AS icustay_id_order
FROM
mimiciii.icustays icu
ORDER BY
subject_id
t2_icu AS -- 1st icu stay
(
SELECT
subject_id,
hadm_id,
icustay_id,
EXTRACT
(
epoch

```

```

FROM
    intime)
    AS
    intime
FROM
    t1_icu
WHERE
    icustay_id_order = 1),
mean_weight AS -- mean weight between 24h before and 24 after the ICu admission
(
SELECT
    DISTINCT
    AVG(t3.weight)
    AS
    t3.subject_id,
    mean_weight
FROM
    t2_icu
    AS
    t2,
    t3_wt
    AS
    t3
WHERE
    t3.charttime BETWEEN t2.intime-01*24*3600
    AND t2.intime+1*24*3600
    AND t3.subject_id=t2.subject_id -- select weight within -1d / +1d from ICU admission
GROUP BY
    t3.subject_id
ORDER BY
    t3.subject_id
),
t0 AS -- height
(
SELECT
    DISTINCT
    EXTRACT(epoch
    FROM
    charttime)
    AS
    charttime,
    itemid,
    CASE
        WHEN itemid IN (920, 1394, 4187, 3486, 226707) THEN valuenum *2.54
    ELSE
        valuenum
    END
    AS
    height -- with conversion inches/cm
FROM
    mimiciii.chartevents
WHERE
    valuenum IS NOT NULL
    AND itemid IN (
        1394,
        4187,
        3486,
        3485,
        4188,

```

```

226707,
226730)
AND          valuenum          !=          0          )
--          select          max          available          height          per          subject_id
,
max_height          AS          (
SELECT
DISTINCT          subject_id,
MAX(height)          AS          height
FROM
t0
GROUP          BY
subject_id
tempo          AS          --          transforming          inches          in          cm
(
SELECT
ad.subject_id,
ad.hadm_id,
icu.icustay_id,
mw.mean_weight,
CASE
WHEN          mh.height          <          100          THEN          mh.height          *2.54
WHEN          mh.height>300          THEN          mh.height/2.54
ELSE
mh.height
END
AS          height
FROM
adultswithlabs          ad
LEFT          JOIN
mean_weight          mw
ON
ad.subject_id=mw.subject_id
LEFT          JOIN
max_height          mh
ON
ad.subject_id=mh.subject_id
LEFT          JOIN
t2_icu          icu
ON
ad.hadm_id          =          icu.hadm_id
ORDER          BY
ad.subject_id          ),
tempo2          AS          --          calculating          BMI
(
SELECT

```

```

tempo.*,
mean_weight/(height*height/10000) AS bmi
FROM
tempo
WHERE
mean_weight IS NOT NULL
AND height IS NOT NULL),
tempo3 AS -- defining BMI groups
(
SELECT
tempo2.*,
CASE
WHEN bmi < 18.5 THEN 1
WHEN bmi >= 18.5 AND bmi < 24.99999999 THEN 2
WHEN bmi >= 25 AND bmi < 30 THEN 3
WHEN bmi >= 30 THEN 4
ELSE 0
END
AS bmi_group
FROM
tempo2
)
SELECT
* -- selecting final cohort
FROM
tempo3
WHERE
icustay_id IS NOT NULL
AND bmi_group =2 -- (normal weight cohort) or bmi_group=4 (obese cohort)

```

- Demographics I and II.

```

-- code for:age,gender, ethnicity, marital status, insurance coverage, LOS hospital, LOS ICU,
admission type, ICU type
WITH
t1 AS (
SELECT
ie.subject_id,
ie.hadm_id,
ie.icustay_id
-- patient level factors
,
pat.gender
-- hospital level factors

```

```

,
adm.admittime,
adm.disctime,
adm.insurance,
adm.religion,
adm.marital_status,
adm.diagnosis,
ROUND( (CAST(adm.disctime AS DATE) - CAST(adm.admittime AS DATE)), 4) AS
los_hospital,
ROUND( (CAST(adm.admittime AS DATE) - CAST(pat.dob AS DATE)) / 365.242, 4) AS age,
adm.ethnicity,
adm.ADMISSION_TYPE,
adm.hospital_expire_flag,
DENSE_RANK() OVER (PARTITION BY adm.subject_id ORDER BY adm.admittime) AS
hospstay_seq,
CASE
  WHEN DENSE_RANK() OVER (PARTITION BY adm.subject_id ORDER BY
adm.admittime) = 1 THEN 'Y'
  ELSE 'N'
END
AS
-- icu level first_hosp_stay factors
,
ie.intime,
ie.outtime,
ROUND( (CAST(ie.outtime AS DATE) - CAST(ie.intime AS DATE)), 4) AS los_icu,
DENSE_RANK() OVER (
PARTITION BY
ie.hadm_id
ORDER BY
ie

```

```

-- Code for Elixhauser index
SELECT
  final_2.hadm_id,
  elixhauser_ahrq_score.elixhauser_vanwalraven AS elixhauser
FROM
  public.final_2 -- replace by the table name created by you in the cohort selection
LEFT JOIN
  elixhauser_ahrq_score
ON
  final_2.hadm_id = elixhauser_ahrq_score.hadm_id

```

```

ORDER
    hadm_id
-- Code for smoking status using Natural language processing searches:
WITH
    t1
    AS
    (
        WITH
            terms
            AS
            (
                SELECT
                    subject_id,
                    CASE
                        -- catches negative terms as 'never smoked', 'non-smoker', 'Pt is not a smoker'
                        WHEN ne.text ~* '(never|not|not a|none|non|no|no history of|no hVo of|denies|denies
any|negative)[\s-]?(smoke|smoking|tabacco|tobacco|cigar|cigs)'
                        -- catches negative terms as: 'Tobacco: denies.', 'Smoking: no;'
                        OR ne.text ~* '(smoke|smoking|tabacco|tobacco|tabacco abuse|tobacco
abuse|cigs|cigarettes):[\s]?(no|never|denies|negative)'
                        -- catches negative terms: 'Cigarettes: Smoked no [x]', 'No EtOH, tobacco', 'He
does not drink or smoke'
                        OR ne.text ~* 'smoked no \[x\]|no etoh, tobacco|not drink alcohol or smoke|not drink or
smoke|absence of current tobacco use|absence of tobacco use' THEN 1
                    ELSE
                        0
                    END
                AS
                nonsmoking,
                CASE
                    -- catches all terms related to smoking: 'Pt. with long history of smoking', 'Smokes
10 cigs/day', 'nicotine patch'
                    WHEN ne.text ~* '(smoke|smoking|tabacco|tobacco|cigar|cigs|marijuana|nicotine)' THEN
1
                    ELSE
                        0
                    END
                AS
                smokingterm
            FROM
                mimiciii.noteevents
                AS
                ne
            ),
            smoking_merged
            AS
            (
                SELECT
                    subject_id,
                    CASE
                        -- Define smoking variable:
                        WHEN nonsmoking = 1 THEN 0 -- 0 means patient doesn't smoke -> when there is a
negative terms present
                        WHEN smokingterm = 1 THEN 1 -- 1 means patient smokes -> when there is a
positive term present and no negative term
                    ELSE
                        2
                    END
            )
        END

```

```

    AS smoking -- 2 means unknown --> no negative or smoking terms mentioned
at
FROM
    terms
-- From multiple notes the query takes the min, so if it's mention the patient is a smoker (1)
-- and that the patient is a non-smoker (0), the function decides that the patient is a non-
smoker (0)
-- because the negative terms are more explicit and probably not triggered by accident.
SELECT
    subject_id,
    MIN(smoking) AS smoking
FROM
    smoking_merged
GROUP BY
    subject_id
ORDER BY
    subject_id
)
SELECT
    t1.subject_id,
    t1.smoking
FROM
    public.final_2 ---- replace by the table name created by you in the cohort selection
LEFT JOIN
    t1
ON
    final_2.subject_id = t1.subject_id
ORDER BY
    subject_id

```

- Primary ICD9.

```

-- Code for the primary ICD-9 diagnosis
SELECT
    final_2.hadm_id,
    primary_dx.icd9_code,
    primary_dx.short_title,
    primary_dx.long_title
FROM
    public.final_2 -- replace by the table name created by you in the cohort selection
LEFT JOIN
    public.primary_dx
ON
    public.final_2.hadm_id = public.primary_dx.hadm_id
ORDER BY

```

hadm_id

- ICU procedures in the first 24h.

```
-- code for mechanical ventilation in the first 24h of ICU
SELECT
  final_2.icustay_id,
  ventfirstday.mechvent                                AS MV_1stday
FROM
  public.final_2 -- replace by the table name created by you in the cohort selection
LEFT JOIN
  ventfirstday
ON
  final_2.icustay_id = ventfirstday.icustay_id
ORDER BY
  icustay_id
-- code for pressors in the first 24h of ICU
WITH
  t2                                AS (
  WITH
    t1                                AS (
      SELECT
        DISTINCT vasopressordurations.icustay_id,
        (vasopressordurations.starttime) AS starttime,
        rank () OVER (PARTITION BY vasopressordurations.icustay_id ORDER BY starttime) AS
pressor_order,
        CASE
          WHEN starttime >= icustays.intime - INTERVAL '1' day AND starttime <= icustays.intime
+          INTERVAL '1' day THEN 1
        ELSE
          0
        END
      AS pressor_1stday,
      icustays.intime
    FROM
      public.vasopressordurations,
      mimiciii.icustays
    WHERE
      vasopressordurations.icustay_id = icustays.icustay_id
    GROUP BY
      vasopressordurations.icustay_id,
      icustays.intime,
      starttime
    ORDER BY
      vasopressordurations.icustay_id )
  )
```



```

SELECT
  icustay_id,
  pressor_1stday,
  pressor_order
FROM
  t1
WHERE
  pressor_order = 1
)
SELECT
  public.final_2.icustay_id,
  t2.pressor_1stday
FROM
  public.final_2 -- replace by the table name created by you in the cohort selection
LEFT JOIN
  t2
ON
  public.final_2.icustay_id=t2.icustay_id
ORDER BY
  icustay_id
-- code for renal replacement therapy in the first 24h of ICU
SELECT
  final_2.icustay_id,
  rrtfirstday AS RRT_1stday
FROM
  public.final_2 -- replace by the table name created by you in the cohort selection
LEFT JOIN
  rrtfirstday
ON
  final_2.icustay_id = rrtfirstday.icustay_id
ORDER BY
  icustay_id

```

- ICU severity of illness score.

```

-- code for SAPS-II
SELECT
  final_2.icustay_id,
  sapsii.sapsii
FROM
  public.final_2 -- replace by the table name created by you in the cohort selection
LEFT JOIN
  sapsii
ON
  final_2.icustay_id = sapsii.icustay_id
ORDER BY
  BY

```

```

icustay_id
--code                                for                                SOFA
SELECT
final_2.icustay_id,
sofa.sofa                                AS                                sofa_score,
sofa.respiration,
sofa.coagulation,
sofa.liver,
sofa.cardiovascular,
sofa.cns,
sofa.renal
FROM
public.final_2 -- replace by the table name created by you in the cohort selection
LEFT JOIN
sapsii
LEFT JOIN
sofa
ON
final_2.icustay_id = sofa.icustay_id
ORDER BY
icustay_id

```

- Baseline lab results.

```

--                                code                                for                                the                                baseline                                laboratory                                result
SELECT
co.icustay_id,
AVG(CASE
    WHEN itemid IN (51300, 51301) THEN valuenum
    ELSE NULL
END
)
AS                                avgwbc_baseline,
AVG(CASE
    WHEN itemid = 51265 THEN valuenum
    ELSE NULL
END
)
AS                                avgplatelets_baseline,
avg
    WHEN itemid IN (50983, 50824) THEN valuenum
    ELSE NULL
END
)
AS                                avgsodium_baseline,

```

```

    avg
    WHEN      itemid IN      (50971,      50822)      THEN      (CASE
    ELSE
    NULL
    END
    )
    ) AS      avgpotassium_baseline,
    avg
    WHEN      itemid      =      51006      THEN      (CASE
    ELSE
    NULL
    END
    )
    ) AS      avgbun_baseline,
    avg
    WHEN      itemid IN      (50912,      51081)      THEN      (CASE
    ELSE
    NULL
    END
    )
    ) AS      avgcreatinine_baseline,
    avg
    WHEN      itemid IN      (50882,      50803)      THEN      (CASE
    ELSE
    NULL
    END
    )
    ) AS      avgbic_baseline
FROM
    final_2 co -- replace by the table name created by you in the cohort selection
INNER JOIN
    admissions adm
ON
    co.hadm_id = adm.hadm_id
LEFT JOIN
    labevents le
ON
    co.subject_id = le.subject_id
    AND le.charttime BETWEEN adm.admittime - INTERVAL '365' day
    AND adm.admittime - INTERVAL '3' day
GROUP BY
    co.subject_id,
    co.hadm_id,
    co.icustay_id,
    adm.admittime
ORDER BY
    icustay_id

```

- 1st24hICU_labresults.

```
-- code for the most abnormal lab result in the first 24h of the ICU admission
SELECT
final_2.icustay_id,
labsfirstday.wbc_max AS wbc_icu,
labsfirstday.platelet_min AS platelet_icu,
labsfirstday.sodium_min AS Na_icu,
labsfirstday.potassium_max AS k_icu,
labsfirstday.bun_max AS BUN_icu,
labsfirstday.creatinine_max AS Cr_icu,
labsfirstday.bicarbonate_min AS BIC_icu
FROM
public.final_2 -- replace by the table name created by you in the cohort selection
LEFT JOIN
labsfirstday
ON
final_2.icustay_id = labsfirstday.icustay_id
ORDER BY
icustay_id
```

- Statistical analyses code (in R).

Loading dataset

loading the 02 different final_2 (= normal weight) and final_4(= obese)

```
final_4 <- read.csv("~/Desktop/final_4/final_4.csv")
View(final_4)

final_2 <- read.csv("~/Desktop/final_2/final_2.csv")
View(final_2)
```

Data Transformation

Changing NA to 0 in the pressor_1stday column in the final dataset using the code below:

```
datasetcolumn[is.na(datasetcolumn[is.na(datasetcolumn)])<-0
final_4$pressor_1stday[is.na(final_4$pressor_1stday)]<-0

final_2$pressor_1stday[is.na(final_2$pressor_1stday)]<-0
```

Correcting age

MIMIC has changed the age of very old patients to 300.0011 to protect them. Here we are change it for 91.4 years

```
final_2$age[final_2$age == 300.0011] <- 91.4  
final_4$age[final_4$age == 300.0011] <- 91.4
```

Adjusting height and BMI

I looked back the database searching for the correct height based by the hadm_id. I looked the echodata (available on MIMIC database). After correcting the height I excluded 1 patient because of his new BMI and 1 patient because I couldn't find his/her correct height. This is the explanation why obese patients have 1258 individual

Adjusting some variables like Sodium

You may find one or two sodium measures like 12.6 which clear means a typo and the actually value is 126, if the value is too unusual like potassium of 126, we deleted the measure.

Data exploration

- data distribution in the variables (have a look at the example below)

```
hist(final_2$age)  
hist(final_4$age)
```

Statistical analysis according to the variables distribution (Table 1)

```
wilcox.test(final_2$age,final_4.16022017$age,correct = FALSE)  
wilcox.test(final_2$elixhauser,final_4.16022017$elixhauser,correct = FALSE)  
wilcox.test(final_2$bmi,final_4.16022017$bmi,correct = FALSE)  
wilcox.test(final_2$sapsii,final_4.16022017$sapsii,correct = FALSE)  
t.test(final_2$sofa,final_4$sofa, paired = FALSE)
```

Statistical analysis proportion comparisons (table 1)

- gender

```
x<-matrix(c(338,517,431,741),          nrow          =          2)
view(x)
chisq.test(x)
```

- ethnicity

```
w<-matrix      (c(583,53,17,25,91,935,95,38,0,190),      ncol          =          2)
View(w)
chisq.test(w)
```

- Marital Status

```
a<-matrix      (c(415,227,111,16,738,352,151,17),      ncol          =          2)
View(a)
chisq.test(a)
```

- Insurance

```
b<-matrix      (c(497,248,24,709,521,28),ncol          =          2)
chisq.test(b)
```

- smoking

```
y<-matrix      (c      (259,377,379,649,131,232),      nrow          =          2)
chisq.test(y)
```

- Admission type

```
q<-      matrix(c(433,15,321,491,15,752),ncol          =          2)
chisq.test(q)
```

- Source of Admission

```
r<-matrix(c(54,310,367,35,3,71,319,822,45,1), ncol=2)
chisq.test(r)
```

- ICU type

```
s<-matrix(c(73,300,286,110,87,726,272,173), ncol=2)
chisq.test(s)
```

- Primary diagnosis

```
e<-matrix(c(110,357,49,47,20,186,188,528,50,88,26,378), ncol=2)
chisq.test(e)
```

1. Mechanical ventilation first 24h

```
m<-matrix(c(453,316,937,321), ncol=2)
chisq.test(m)
```

- Vasopressors in the first 24h

```
v<-matrix(c(346,423,731,527), ncol=2)
chisq.test(v)
```

- renal replacement Therapy in the first 24h

```
r<-matrix(c(32,737,35,1221), ncol=2)
chisq.test(r)
```

QQ plots (to assess normality) for baseline labs on both obese and normal weight

```
with(final_4, qqnorm((avgwbc_baseline)))
with(final_2, qqnorm((avgwbc_baseline)))

with(final_4, qqnorm((avgsodium_baseline)))
with(final_2, qqnorm((avgsodium_baseline)))

with(final_4, qqnorm((avgpotassium_baseline)))
with(final_2, qqnorm((avgpotassium_baseline)))

with(final_4, qqnorm((avgbun_baseline)))
with(final_2, qqnorm((avgbun_baseline)))

with(final_4, qqnorm((avgbic_baseline)))
with(final_2, qqnorm((avgbic_baseline)))

with(final_4, qqnorm(avgcreatinine_baseline)))
with(final_2, qqnorm(avgcreatinine_baseline)))

with(final_4, qqnorm((avgplatelets_baseline)))
with(final_2, qqnorm((avgplatelets_baseline)))
```

QQ Plots for the difference (ICU-baseline)for obese and normal weight

```
with(final_4, qqnorm((avgwbc_baseline - wbc_icu)))
with(final_2, qqnorm((avgwbc_baseline - wbc_icu)))
with(final_4, qqnorm((avgsodium_baseline - na_icu)))
with(final_2, qqnorm((avgsodium_baseline - na_icu)))
with(final_4, qqnorm((avgpotassium_baseline - k_icu)))
with(final_2, qqnorm((avgpotassium_baseline - k_icu)))
with(final_4, qqnorm((avgbun_baseline - bun_icu)))
with(final_2, qqnorm((avgbun_baseline - bun_icu)))
with(final_4, qqnorm((avgbic_baseline - bic_icu)))
with(final_2, qqnorm((avgbic_baseline - bic_icu)))
with(final_4, qqnorm(avgcreatinine_baseline - cr_icu)))
with(final_2, qqnorm(avgcreatinine_baseline - cr_icu)))
with(final_4, qqnorm((avgplatelets_baseline - platelet_icu)))
with(final_2, qqnorm((avgplatelets_baseline - platelet_icu)))
```

Baseline comparison (lab results between normal weight and obese patients)

```
wilcox.test(final_2$avgwbc_baseline, final_4$avgwbc_baseline, correct = TRUE)
wilcox.test(final_2$avgsodium_baseline, final_4$avgsodium_baseline, correct = TRUE)
wilcox.test(final_2$avgpotassium_baseline, final_4$avgpotassium_baseline, correct = TRUE)
```



```
wilcox.test(final_2$avgbun_baseline, final_4$avgbun_baseline, correct = TRUE)
wilcox.test(final_2$avgbic_baseline, final_4$avgbic_baseline, correct = TRUE)
wilcox.test(final_2$avgcreatinine_baseline, final_4$avgcreatinine_baseline, correct = TRUE)
wilcox.test(final_2$avgplatelets_baseline, final_4$avgplatelets_baseline, correct = TRUE)
```

Creating the laboratory deviation column (ICU - baseline)

```
final_2 ["deviation_wbc"]<-NA
final_2 ["deviation_platelets"]<-NA
final_2 ["deviation_na"]<-NA
final_2 ["deviation_k"]<-NA
final_2 ["deviation_cr"]<-NA
final_2 ["deviation_bun"]<-NA
final_2 ["deviation_bic"]<-NA
final_4 ["deviation_wbc"]<-NA
final_4 ["deviation_platelets"]<-NA
final_4 ["deviation_na"]<-NA
final_4 ["deviation_k"]<-NA
final_4 ["deviation_cr"]<-NA
final_4 ["deviation_bun"]<-NA
final_4 ["deviation_bic"]<-NA
final_2$deviation_wbc<-final_2$wbc_icu - final_2$avgwbc_baseline
final_2$deviation_platelets <-final_2$platelet_icu - final_2$avgplatelets_baseline
final_2$deviation_na <-final_2$na_icu - final_2$avgsodium_baseline
final_2$deviation_k <-final_2$k_icu - final_2$avgpotassium_baseline
final_2$deviation_cr <-final_2$cr_icu - final_2$avgcreatinine_baseline
final_2$deviation_bun <-final_2$bun_icu - final_2$avgbun_baseline
final_2$deviation_bic <-final_2$bic_icu - final_2$avgbic_baseline
final_4$deviation_wbc<-final_4$wbc_icu - final_4$avgwbc_baseline
final_4$deviation_platelets <-final_4$platelet_icu - final_4$avgplatelets_baseline
final_4$deviation_na <-final_4$na_icu - final_4$avgsodium_baseline
final_4$deviation_k <-final_4$k_icu - final_4$avgpotassium_baseline
final_4$deviation_cr <-final_4$cr_icu - final_4$avgcreatinine_baseline
final_4$deviation_bun <-final_4$bun_icu - final_4$avgbun_baseline
final_4$deviation_bic <-final_4$bic_icu - final_4$avgbic_baseline
```

Unadjusted Statistical analysis of the deviation

```
t.test(final_2$deviation_wbc, final_4$deviation_wbc, paired = F)
t.test(final_2$deviation_platelets, final_4$deviation_platelets, paired = F)
t.test(final_2$deviation_na, final_4$deviation_na, paired = F)
t.test(final_2$deviation_k, final_4$deviation_k, paired = F)
t.test(final_2$deviation_bic, final_4$deviation_bic, paired = F)
```

Creating log_bun , log_creatinine , deviation and t test

```
final_4$log_bun_baseline<- log10(final_4$avgbun_baseline)
final_4$log_bun_icu<- log10(final_4$bun_icu)
final_4$deviation_log_bun<- final_4$log_bun_icu - final_4$log_bun_baseline
mean (final_4$deviation_log_bun, na.rm = TRUE)
sd(final_4$deviation_log_bun, na.rm = TRUE)
with(final_4 , qqnorm(log10(avgbun_baseline)))
with(final_4 , qqnorm(log10(avgbun_baseline) -log10(bun_icu)))

final_2$log_bun_baseline<- log10(final_2$avgbun_baseline)
final_2$log_bun_icu<- log10(final_2$bun_icu)
final_2$deviation_log_bun<-final_2$log_bun_icu - final_2$log_bun_baseline
mean (final_2$deviation_log_bun, na.rm = TRUE)
sd(final_2$deviation_log_bun, na.rm = TRUE)
with(final_2 , qqnorm(log10(avgbun_baseline)))
with(final_2 , qqnorm(log10(avgbun_baseline) -log10(bun_icu)))
t.test(final_4$deviation_log_bun,final_2$deviation_log_bun, paired= FALSE)

final_4$log_creatinine_baseline<- log10(final_4$avgcreatinine_baseline)
final_4$log_cr_icu<- log10(final_4$cr_icu)
final_4$deviation_log_creatinine<- final_4$log_cr_icu - final_4$log_creatinine_baseline
mean (final_4$deviation_log_creatinine, na.rm = TRUE)
sd(final_4$deviation_log_creatinine, na.rm = TRUE)
with(final_4 , qqnorm(log10(avgcreatinine_baseline)))
with(final_4 , qqnorm(log10(avgcreatinine_baseline) -log10(cr_icu)))
final_2$log_creatinine_baseline<- log10(final_2$avgcreatinine_baseline)
final_2$log_cr_icu<- log10(final_2$cr_icu)
final_2$deviation_log_creatinine<-final_2$log_cr_icu - final_2$log_creatinine_baseline
mean (final_2$deviation_log_creatinine, na.rm = TRUE)
sd(final_2$deviation_log_creatinine, na.rm = TRUE)
with(final_2 , qqnorm(log10(avgcreatinine_baseline)))
with(final_2 , qqnorm(log10(avgcreatinine_baseline) -log10(cr_icu)))
t.test(final_4$deviation_log_creatinine, final_2$deviation_log_creatinine, paired= FALSE)
```

Merging the two files normal weight(final_2) with obese (final_4)

```
total<-rbind(final_4,final_2)
```

Adjusted laboratory deviation analysis

The differences in deviation from baseline between both groups were compared using multivariable linear regression adjusted for age, gender, comorbidity index index, SAPS-II score or SOFA score, and type of ICU, and the relevant baseline laboratory result. A full model comprising the BMI status and all the covariates was initially fit, and subjected to stepwise backwards elimination retaining BMI status in the model, until a final model was obtained with only statistically significant variables.

multiple linear regression wbc

```
fit_wbc<-lm(deviation_wbc~age+gender+elixhauser+icu_type+bmi_group+avgwbc_baseline +sapsii,
data=total)
summary(fit_wbc)
fit_wbc<-lm(deviation_wbc~age+elixhauser+icu_type+bmi_group+avgwbc_baseline +sapsii, data=total)
summary(fit_wbc)
fit_wbc<-lm(deviation_wbc~age+icu_type+bmi_group+avgwbc_baseline +sapsii, data=total)
summary(fit_wbc)
drop1(fit_wbc, test="F")
sjPlot::sjt.lm(fit_wbc)
```

multiple linear regression platelets

```
fit_platelets<-
lm(deviation_platelets~age+gender+elixhauser+icu_type+bmi_group+avgplatelets_baseline
+sofa_score, data=total)
summary(fit_platelets)
fit_platelets<-lm(deviation_platelets~age+gender+icu_type+bmi_group+avgplatelets_baseline
+sofa_score, data=total)
summary(fit_platelets)
drop1(fit_platelets, test="F")
sjPlot::sjt.lm(fit_platelets)
```

multiple linear regression sodium

```
fit_na <-lm(deviation_na~age+gender+elixhauser+icu_type+bmi_group+avgsodium_baseline +sapsii,
data=total)
summary(fit_na)
fit_na <-lm(deviation_na~gender+elixhauser+icu_type+bmi_group+avgsodium_baseline +sapsii, data=
total)
summary(fit_na)
fit_na <-lm(deviation_na~elixhauser+icu_type+bmi_group+avgsodium_baseline +sapsii, data= total)
summary(fit_na)
drop1(fit_na, test="F")
sjPlot::sjt.lm(fit_na)
par(mfcol=c(2,2))
plot(fit_na)
```

multiple linear regression potassium

```
fit_k <- lm(deviation_k~age+gender+elixhauser+icu_type+bmi_group+avgpotassium_baseline +sapsii,
data=total)
```

```

summary(fit_k)
fit_k <- lm(deviation_k~age+elixhauser+icu_type+bmi_group+avgpotassium_baseline +sapsii,
data=total)
summary(fit_k)
fit_k <- lm(deviation_k~age+icu_type+bmi_group+avgpotassium_baseline +sapsii, data=total)
summary(fit_k)
drop1(fit_k, test="F")
sjPlot::sjt.lm(fit_k)
par(mfcol=c(2,2))
plot(fit_k)

```

multiple linear regression log creatinine

```

total$log_cr_baseline<- log10(total$avgcreatinine_baseline)
total$log_cr_icu<- log10(total$cr_icu)
total$deviation_log_creatinine <- total$log_cr_icu - total$log_cr_baseline
with(total,t.test(deviation_log_creatinine~bmi_group))
fit_log_creatinine <-
lm(deviation_log_creatinine~age+gender+elixhauser+icu_type+bmi_group+log_cr_baseline
+sofa_score, data=total)
summary(fit_log_creatinine)
fit_log_creatinine <-
lm(deviation_log_creatinine~gender+elixhauser+icu_type+bmi_group+log_cr_baseline +sofa_score,
data=total)
summary(fit_log_creatinine)
drop1(fit_log_creatinine, test="F")
sjPlot::sjt.lm(fit_log_creatinine)
par(mfcol=c(2,2))
plot(fit_log_creatinine)

```

multiple linear regression log BUN

```

total$log_bun_baseline<- log10(total$avgbun_baseline)
total$log_bun_icu<- log10(total$bun_icu)
total$deviation_log_bun <- total$log_bun_icu - total$log_bun_baseline
with(total,t.test(deviation_log_bun~bmi_group))
fit_log_bun <-lm(deviation_log_bun~age+gender+elixhauser+icu_type+bmi_group+log_bun_baseline
+sofa_score, data=total)
summary(fit_log_bun)
drop1(fit_log_bun, test="F")
sjPlot::sjt.lm(fit_log_bun)
par(mfcol=c(2,2))
plot(fit_log_bun)

```

multiple linear regression BIC

```
fit_bic <-lm(deviation_bic~age+gender+elixhauser+icu_type+bmi_group+avgbic_baseline+sapsii, data=
total)
summary(fit_bic)
fit_bic <-lm(deviation_bic~age+elixhauser+icu_type+bmi_group+avgbic_baseline+sapsii, data= total)
summary(fit_bic)
drop1(fit_bic, test="F")
fit_bic <-lm(deviation_bic~age+elixhauser+bmi_group+avgbic_baseline+sapsii, data= total)
summary(fit_bic)
sjPlot::sjt.lm(fit_bic)
par(mfcol=c(2,2))
plot(fit_bic)
```

EXTRACTING ALL OUTPUTS OF THE MULTIVARIATE LINEAR REGRESSION

```
sjPlot::sjt.lm(fit_wbc)
sjPlot::sjt.lm(fit_platelets)
sjPlot::sjt.lm(fit_na)
sjPlot::sjt.lm(fit_k)
sjPlot::sjt.lm(fit_log_creatinine)
sjPlot::sjt.lm(fit_log_bun)
sjPlot::sjt.lm(fit_bic)
```

Creation of subset without any missing variable and analyzing the impact of the statistically significant deviations in the hospital mortality

```
total_s<-subset(total, !is.na(deviation_wbc) & !is.na(deviation_log_bun) &
!is.na(deviation_log_creatinine))
nrow(total)
nrow(total_s)
lr<-glm (hospital_expire_flag ~ sapsii + sofa_score + age + wbc_icu + log_bun_icu + log_cr_icu,
data=total_s, family = binomial)
summary(lr)
lr1<-glm (hospital_expire_flag ~ sapsii + sofa_score + age + wbc_icu + deviation_wbc + log_bun_icu +
deviation_log_bun + log_cr_icu + deviation_log_creatinine, data=total_s, family = binomial)
summary(lr1)
anova(lr1, lr, test ="Chisq")
```

Supplemental Digital Content

Supplemental Table 1 provides the number of missing laboratory values:

Supplemental Table 1. Number of missing laboratory values

Variable	Obese (BMI >30) N = 1,258		Normal weight (BMI = 18.5 - 24.9) N = 769	
Variable	Baseline laboratory result	ICU laboratory result	Baseline laboratory result	ICU laboratory result
WBC	55	27	34	9
Sodium	114	2	93	4
Potassium	104	1	86	4
BUN	88	5	62	5
BIC	136	26	104	9
Creatinine	78	4	59	5
Platelets	54	18	32	7

WBC: white blood cells; BUN: Blood Urea Nitrogen; BIC: Bicarbonate

A) Complete multivariable linear regression analysis:

White Blood Cell:

		deviation_wbc		
		<i>B</i>	<i>CI</i>	<i>p</i>
(Intercept)		4.63	2.99 – 6.27	<.001
age		-0.03	-0.05 – -0.01	.009
icu_type				
<i>CSRU</i>		3.24	2.26 – 4.23	<.001
<i>MICU</i>		-0.44	-1.48 – 0.60	.405
<i>SICU</i>		1.69	0.48 – 2.90	.006
<i>TSICU</i>		0.06	-1.55 – 1.67	.943
bmi_group (obese)		0.80	0.27 – 1.33	.003
avgwbc_baseline		-0.25	-0.32 – -0.18	<.001
sapsii		0.08	0.06 – 0.10	<.001
Observations		1905		
R ² / adj. R ²		.121 / .117		

Platelet:

		deviation_platelets		
		<i>B</i>	<i>CI</i>	<i>p</i>
(Intercept)		76.44	51.28 – 101.59	<.001
age		0.27	0.02 – 0.52	.032
gender (M)		11.10	3.76 – 18.43	.003
icu_type				
<i>CSRU</i>		-53.75	-67.52 – -39.98	<.001
<i>MICU</i>		1.62	-12.91 – 16.16	.827
<i>SICU</i>		1.90	-14.98 – 18.78	.825
<i>TSICU</i>		-3.41	-25.90 – 19.07	.766
bmi_group (obese)		4.94	-2.48 – 12.36	.192
avgplatelets_baseline		-0.47	-0.51 – -0.44	<.001
sofa_score		-6.62	-7.85 – -5.38	<.001
Observations		1918		
R ² / adj. R ²		.342 / .339		

Sodium:

		deviation_na		
		<i>B</i>	<i>CI</i>	<i>p</i>
(Intercept)		68.02	61.26 – 74.79	<.001
Comorbidity index		-0.05	-0.08 – -0.02	.002
icu_type				
<i>CSRU</i>		-0.60	-1.23 – 0.02	.057
<i>MICU</i>		0.29	-0.37 – 0.95	.388
<i>SICU</i>		-0.02	-0.80 – 0.75	.953
<i>TSICU</i>		0.30	-0.79 – 1.39	.594
bmi_group (obese)		-0.06	-0.40 – 0.28	.712
avgsodium_baseline		-0.50	-0.55 – -0.45	<.001
sapsii		-0.02	-0.03 – -0.01	.002
Observations		1814		
R ² / adj. R ²		.210 / .206		

Potassium:

		deviation_k		
		<i>B</i>	<i>CI</i>	<i>p</i>
(Intercept)		2.61	2.20 – 3.02	<.001
age		-0.00	-0.01 – -0.00	.034
icu_type				
<i>CSRU</i>		0.81	0.66 – 0.95	<.001
<i>MICU</i>		-0.15	-0.30 – 0.00	.051
<i>SICU</i>		-0.12	-0.30 – 0.06	.177
<i>TSICU</i>		-0.29	-0.54 – -0.04	.024
bmi_group (obese)		0.01	-0.07 – 0.09	.857
avgpotassium_baseline		-0.58	-0.66 – -0.50	<.001
sapsii		0.02	0.01 – 0.02	<.001
Observations		1832		
R ² / adj. R ²		.312 / .309		

Creatinine:

		deviation_log_creatinine		
		<i>B</i>	<i>CI</i>	<i>p</i>
(Intercept)		-0.04	-0.06 – -0.01	.007
gender (M)		0.02	0.01 – 0.04	<.001
Comorbidity index		0.00	0.00 – 0.00	<.001
icu_type				
<i>CSRU</i>		-0.12	-0.14 – -0.09	<.001
<i>MICU</i>		-0.00	-0.03 – 0.02	.962
<i>SICU</i>		-0.04	-0.07 – -0.01	.004
<i>TSICU</i>		-0.09	-0.13 – -0.05	<.001
bmi_group (obese)		0.03	0.02 – 0.05	<.001
log_cr_baseline		-0.16	-0.20 – -0.13	<.001
sofa_score		0.02	0.02 – 0.02	<.001
Observations		1882		
R ² / adj. R ²		.277 / .273		

BUN:

		deviation_log_bun		
		<i>B</i>	<i>CI</i>	<i>p</i>
(Intercept)		0.29	0.22 – 0.35	<.001
age		0.00	0.00 – 0.00	<.001
gender (M)		0.04	0.03 – 0.06	<.001
elixhauser		0.00	0.00 – 0.00	<.001
icu_type				
<i>CSRU</i>		-0.17	-0.20 – - 0.14	<.001
<i>MICU</i>		-0.02	-0.05 – 0.01	.237
<i>SICU</i>		-0.06	-0.09 – - 0.02	.003
<i>TSICU</i>		-0.10	-0.15 – - 0.05	<.001
bmi_group (obese)		0.01	0.00 – 0.02	.014
log_bun_baseline		-0.30	-0.34 – - 0.26	<.001
sofa_score		0.02	0.02 – 0.02	<.001
Observations		1868		
R ² / adj. R ²		.295 / .291		

Bicarbonate:

		deviation_bic		
		<i>B</i>	<i>CI</i>	<i>p</i>
(Intercept)		11.9 1	10.46 – 13.37	<.001
age		0.03	0.02 – 0.04	<.001
Comorbidity index		-0.04	-0.07 – -0.02	<.001
bmi_group (obese)		-0.19	-0.50 – 0.13	.254
avgbic_baseline		-0.53	-0.58 – -0.48	<.001
sapsii		-0.09	-0.11 – -0.08	<.001
Observations		1757		
R ² / adj. R ²		.277 / .275		

APPENDIX IV: DEVELOPING WELL-CALIBRATED ILLNESS SEVERITY SCORES FOR DECISION SUPPORT IN THE CRITICALLY ILL

Cohort Extraction

In this notebook we will extract the necessary variables for feature engineering and cohort construction to build a set of severity score models for this study.

The initial variables we will extract are as follows:

- Age, gender, weight, ethnicity (patient table)
- Source of admission, unit type (patient table)
- Laboratory data on the first day (labsfirstday materialized view)
  - Blood gases: PaO₂, pH, base excess, bicarbonate,
  - Hematology: hematocrit, hemoglobin, lymphocytes, neutrophils, platelets, white cell count
  - Electrolytes: calcium, chloride, ionized calcium, magnesium, phosphate, sodium
  - Biochemistry: albumin, amylase, bilirubin, blood urea nitrogen (BUN), B-natriuretic peptide, creatine phosphokinase (cpk), creatinine, lactate, lipase, troponin I/T, pH, bicarbonate, base excess, glucose
  - Coagulation: PT/INR, fibrinogen
- Vital signs on the first day (vitalsfirstday materialized view)
  - Heart rate
  - Blood pressure
  - Respiratory rate
  - SpO₂
- Treatments (treatmentfirstday materialized view)
  - Antiarrhythmics
  - Antibiotics
  - Vasopressors
  - Sedatives
  - Diuretics
  - Blood products

- Ventilation status (apachepredvar table)
- Admission Dx (APACHE_GROUPS materialized view)
- APACHE IVa Features (apachepredvar table)

The label for our classifier as well as their APACHE IVa predicted mortality are located in apachepatientresult.

Environment

:

```
import pandas as pd
import numpy as np
import psycpg2
from sklearn.model_selection import train_test_split

dbname = 'eicu'
schema_name = 'eicu_crd'
query_schema = 'SET search_path TO ' + schema_name + ';'
con = psycpg2.connect(dbname=dbname)
```

Materialized Views

We will generate the requisite materialized views to aid in the extraction of the cohort features. We start by introducing helper functions for interacting with the eICU-CRD.

```
def execute_query_safely(sql, con):
    cur = con.cursor()
    try:
        cur.execute(sql)
    except:
        cur.execute('rollback;')
        raise
    finally:
        cur.close()
    return

def generate_materialized_view(query_file, con, query_schema):
    with open(query_file) as fp:
        query = "".join(fp.readlines())
    print('Generating materialized view using {} ...'.format(query_file), end=' ')
    execute_query_safely(query_schema + query, con)
    print('done.')
```

Now we generate the materialized views.


```

generate_materialized_view('./sql/vitalsfirstday.sql', con, query_schema)
generate_materialized_view('./sql/labsfirstday.sql', con, query_schema)
generate_materialized_view('./sql/treatmentfirstday.sql', con, query_schema)
generate_materialized_view('./sql/apache-groups.sql', con, query_schema)

```

Load Features

We begin by loading a base cohort. As our most import exclusion criteria is whether or not they had an APACHE IVa score (so we can fit models in the subpopulations), we can join the patient table on the apachepredvar and apachepatientresult table. There are 200,859 ICU stays, and so the number of rows returned will be the remainder after exclusion of patients for which APACHE data or an APACHE score is not available.

```

base_cohort_query = query_schema + '''
WITH
    (
        SELECT
            patientunitstayid,
            CAST(predictedhospitalmortality AS float) AS apache_prediction,
            actualhospitalmortality AS apache_patientresult,
        FROM
        WHERE
            apacheversion = 'IVa'
        AND
            apachescore > 0
        AND
            CAST(predictedhospitalmortality AS float) >= 0
    )
    ,
    admit_order AS
    (
        SELECT
            patientunitstayid,
            ROW_NUMBER() OVER (partition BY uniquepid ORDER BY hospitaladmitoffset DESC,
            patientunitstayid) AS admission_num
        FROM
            patient
    )

SELECT
    p.patientunitstayid, p.age, p.gender, p.ethnicity, p.admissionheight AS height,
    p.admissionweight AS weight, p.unittype AS unit_type, p.unitadmitsource,
    p.unitdischargeoffset AS unit_los, p.hospitaldischargeoffset AS hospital_los,
    a.day1meds AS gcs_meds, a.day1verbal AS gcs_verbal, a.day1motor AS gcs_motor,
    a.day1eyes AS gcs_eyes, a.admitDiagnosis AS admit_diagnosis, ag.apachedxgroup as
    adx_group,
    a.thrombolytics AS apache_thrombolytics, a.electivesurgery AS apache_elect_surg,
    a.readmit AS apache_readmit, ao.admission_num, a.ima AS apache_ima,
    a.midur AS apache_midur, a.oOBVentDay1 AS apache_ventday1, a.oOBIntubDay1 AS
    apache_intubday1

```

```

, a.day1fio2 AS apache_fio2, a.day1pao2 AS apache_pao2, (a.day1pao2 / a.day1fio2) AS
apache_o2ratio
, a.ejectfx AS apache_ejectfx, a.creatinine AS apache_creatinine
, a.graftcount AS apache_graftcount, o.apache_prediction
, o.apachescore AS apache_score, o.actualhospitalmortality AS hospital_expiration
FROM patient p
INNER JOIN apachepredvar a
ON p.patientunitstayid = a.patientunitstayid
INNER JOIN apacheIV o
ON p.patientunitstayid = o.patientunitstayid
INNER JOIN admit_order ao
on p.patientunitstayid = ao.patientunitstayid
INNER JOIN APACHE_GROUPS ag
on p.patientunitstayid = ag.patientunitstayid
ORDER BY patientunitstayid;
'''

base_cohort = pd.read_sql_query(base_cohort_query, con)
base_cohort.shape

```

(136231, 33)

Of the 200,859 patients in the database, 136,231 have APACHE IVa variables recorded as well as a hospital mortality prediction carried out. We'll then load the variables that will be used to derive the expanded feature set into a dataframe.

```

feature_set_query = query_schema + '''
SELECT v.patientunitstayid, v.HR_Mean, v.SBP_periodic_Mean, v.DBP_periodic_Mean
, v.MAP_periodic_Mean, v.SBP_aperiodic_Mean, v.DBP_aperiodic_Mean
, v.MAP_aperiodic_Mean, v.RR_Mean, v.SpO2_Mean, v.TempC_Mean
, ANIONGAP_min, ANIONGAP_max, ALBUMIN_min, ALBUMIN_max
, AMYLASE_min, AMYLASE_max, BASEEXCESS_min, BASEEXCESS_max
, BICARBONATE_min, BICARBONATE_max, BUN_min, BUN_max, BNP_min
, BNP_max, CPK_min, CPK_max, BILIRUBIN_min, BILIRUBIN_max
, CALCIUM_min, CALCIUM_max, IONCALCIUM_min, IONCALCIUM_max
, CREATININE_min, CREATININE_max, CHLORIDE_min, CHLORIDE_max
, GLUCOSE_min, GLUCOSE_max, HEMATOCRIT_min, HEMATOCRIT_max
, FIBRINOGEN_min, FIBRINOGEN_max, LIPASE_min, LIPASE_max
, HEMOGLOBIN_min, HEMOGLOBIN_max, LACTATE_min, LACTATE_max
, LYMPHS_min, LYMPHS_max, MAGNESIUM_min, MAGNESIUM_max
, PAO2_min, PAO2_max, PH_min, PH_max, PLATELET_min
, PLATELET_max, PMN_min, PMN_max, PHOSPHATE_min, PHOSPHATE_max
, POTASSIUM_min, POTASSIUM_max, PTT_min, PTT_max, INR_min
, INR_max, PT_min, PT_max, SODIUM_min, SODIUM_max
, TROPI_min, TROPI_max, TROPT_min, TROPT_max, WBC_min
'''

```

```

, WBC_max, t.abx, t.pressor, t.antiarr, t.sedative
, t.diuretic, t.blood_product
FROM vitalsfirstday v
LEFT JOIN labsfirstday l
ON v.patientunitstayid = l.patientunitstayid
LEFT OUTER JOIN treatmentfirstday t
ON v.patientunitstayid = t.patientunitstayid;
"""

feature_set = pd.read_sql_query(feature_set_query, con)
feature_set.shape

```

(192320, 85)

We can then merge the two dataframes. Note that we are using an inner join here and so if a patient did not have a vital sign record in vitalsperiodic they will not be included in the cohort.

```

cohort = pd.merge(left=base_cohort, right=feature_set, how='inner', on='patientunitstayid')
cohort.shape

```

(134946, 117)

Here we go from 136,231 to 134,946. The missing 1,285 did not have recorded vitals and their APACHE IVa score was likely derived from the nursecharting table which we are not utilizing in this study.

Inclusion / Exclusion

By nature of our SQL query, we have already excluded patients not eligible/capable of producing a valid score, and patients who lack all vital/lab/treatment data. We can then check the APACHE IVa criteria:

1. Not readmissions
2. Not admitted from another ICU
3. Admitted to ICU for hours ≥ 4 hours.
4. Not burn patients
5. Not transplant patients (other than hepatic renal)
6. Age ≥ 16 .

1 - Not Readmitted

```
cohort = cohort.loc[cohort.apache_readmit == 0, :]  
cohort.shape  
(134946, 117)
```

There are no readmissions.

2 - Not Admitted from ICU

```
cohort = cohort.loc[cohort.unitadmitsource != 'Other ICU', :]  
cohort.shape  
(134890, 117)
```

56 patients were admitted from another ICU, but still had scores calculated.

3 - ICU LoS ≥ 4 h

```
cohort      = cohort.loc[cohort.unit_los >= 240, :]
cohort.shape
```

(134890, 117)

No patients had a LoS <4h.

4 - Not Burn Patients

```
cohort      = cohort.loc[cohort.admit_diagnosis != 'BURN', :]
cohort.shape
```

(134890, 117)

There were no burn patients inappropriately kept in the cohort.

5 - Not Transplant Patients

```
cohort      = cohort.loc[cohort.admit_diagnosis != 'KIDPANTRAN', :]
cohort      = cohort.loc[cohort.admit_diagnosis != 'S-KIDPTRAN', :]
cohort      = cohort.loc[cohort.admit_diagnosis != 'S-HEARTRAN', :]
cohort      = cohort.loc[cohort.admit_diagnosis != 'HEARTRAN', :]
cohort      = cohort.loc[cohort.admit_diagnosis != 'S-LUNGTRAN', :]
cohort      = cohort.loc[cohort.admit_diagnosis != 'LUNGTRAN', :]
cohort      = cohort.loc[cohort.admit_diagnosis != 'LUNGSTRAN', :]
cohort.shape
```

(134890, 117)

There were also no transplant patients inappropriately kept in the cohort.

6 - Age >= 16, and Not Missing

```
# Per Rodrigo, the median age for >89 pt in eICU is 93
cohort.loc[cohort.age == '> 89', 'age'] = 93.0
cohort = cohort.loc[cohort.age != '']
cohort.age = cohort.age.astype('float64')
cohort = cohort.loc[cohort.age >= 16., :]
cohort.shape
```

(134890, 117)

No patients <16 years of age were in the dataset to be excluded.

Cleaning, Formatting and Feature Engineering

Before anything else, we can drop numerous variables from the pull as we won't need both min/max for various features (they'll often be the same value since many things are only measured once in 24 hours, and even if they're not, they'll be highly correlated). Instead we'll be going by the following principle: the most abnormal laboratory value in the first 24 hours of ICU stay will be included. That is we will use:

- The minimum value for: bicarbonate, chloride, calcium, magnesium, base excess (including negative values), platelets, hemoglobin, phosphate, fibrinogen, pH and hematocrit
- The maximum value for: creatinine, BUN, bilirubin, PT/INR, lactate, troponin I/T, amylase, lipase, B-natriuretic peptide and creatinine phosphokinase
- For sodium, which aberrantly deviates bidirectionally, the most abnormal value was defined as the value with the greatest deviation from the normal range boundaries.
  - This can be applied to glucose and potassium as well.
- For white blood cell and neutrophil counts, if any measurements were lower than the lower limit of normal, the minimum value was used; if the minimum was within normal range then the maximum was used as the most abnormal value.

```
# for the unidirectional aberrations, just drop what isn't needed
lab_drop = ['bicarbonate_max', 'chloride_max', 'calcium_max', 'magnesium_max',
            'baseexcess_max', 'platelet_max', 'hemoglobin_max', 'phosphate_max',
            'fibrinogen_max', 'ph_max', 'hematocrit_max', 'creatinine_min',
            'bun_min', 'bilirubin_min', 'pt_min', 'inr_min', 'lactate_min',
            'tropi_min', 'tropt_min', 'amylase_min', 'lipase_min', 'bnp_min',
            'cpk_min', 'albumin_max', 'ioncalcium_max', 'pao2_max', 'pt_min',
            'ptt_min', 'inr_min', 'aniongap_min']
cohort = cohort.drop(lab_drop, axis=1)

# sodium, deviates bidirectionally
sodium_check = abs(cohort.sodium_min - 135.) >= abs(cohort.sodium_max - 145.)
sodium = np.empty(len(cohort.index), dtype='float64')
sodium[sodium_check] = cohort.sodium_min[sodium_check]
sodium[~sodium_check] = cohort.sodium_max[~sodium_check]
cohort = cohort.assign(sodium=sodium)
cohort = cohort.drop(['sodium_min', 'sodium_max'], axis=1)

# potassium, deviates bidirectionally, same treatment then
potassium_check = abs(cohort.potassium_min - 3.5) >= abs(cohort.potassium_max - 5.0)
potassium = np.empty(len(cohort.index), dtype='float64')
```

```

potassium[potassium_check] = cohort.potassium_min[potassium_check]
potassium[~potassium_check] = cohort.potassium_max[~potassium_check]
cohort = cohort.assign(potassium=potassium)
cohort = cohort.drop(['potassium_min', 'potassium_max'], axis=1)

# similar treatment for glucose since hyperglycemia and hypoglycemia can both
# be important dependent on the clinical context.
glucose_check = abs(cohort.glucose_min - 70) >= abs(cohort.glucose_max - 130)
glucose = np.empty(len(cohort.index), dtype='float64')
glucose[glucose_check] = cohort.glucose_min[glucose_check]
glucose[~glucose_check] = cohort.glucose_max[~glucose_check]
cohort = cohort.assign(glucose=glucose)
cohort = cohort.drop(['glucose_min', 'glucose_max'], axis=1)

#
# wbc counts
wbc_check = cohort.wbc_min < 2
pmn_check = cohort.pmn_min < 45
lym_check = cohort.lymphs_min < 20

wbc = np.empty(len(cohort.index), dtype='float64')
wbc[wbc_check] = cohort.wbc_min[wbc_check]
wbc[~wbc_check] = cohort.wbc_max[~wbc_check]
cohort = cohort.assign(wbc=wbc)
cohort = cohort.drop(['wbc_min', 'wbc_max'], axis=1)

pmn = np.empty(len(cohort.index), dtype='float64')
pmn[pmn_check] = cohort.pmn_min[pmn_check]
pmn[~pmn_check] = cohort.pmn_max[~pmn_check]
cohort = cohort.assign(pmn=pmn)
cohort = cohort.drop(['pmn_min', 'pmn_max'], axis=1)

lym = np.empty(len(cohort.index), dtype='float64')
lym[lym_check] = cohort.lymphs_min[lym_check]
lym[~lym_check] = cohort.lymphs_max[~lym_check]
cohort = cohort.assign(lym=lym)
cohort = cohort.drop(['lymphs_min', 'lymphs_max'], axis=1)

```

Inspection of the data reveals some variables which are redundant and others which appear unreliable, and in some cases one source is better than another. As such we will:

- Use aperiodic BP instead of periodic
- Drop temperature; it is an unreliable signal when automatically captured

- Drop APACHE elective surgery, we will simply classify patients by their admission diagnosis
- Drop PaO2/FiO2 from APACHE and instead just use the PaO2 values derived directly from laboratory data
- Use sCr from labs instead of APACHE table
- Drop LoS variables since they would let our models peek into the future
- Drop unit admit source, it was only included for the exclusion criteria
- Drop APACHE score
- Drop APACHE readmit and admission number, as they were only used to examine exclusion criteria
- Missingness will be handled during modeling, but we should make sure that all missingness is labeled with np.nan and not -1 as is present in some of the eICU tables

```
cohort = cohort.replace(-1, np.nan)
cohort = cohort.drop(['sbp_periodic_mean', 'dbp_periodic_mean', 'map_periodic_mean',
                    'tempc_mean', 'apache_elect_surg', 'apache_creatinine', 'apache_pao2',
                    'apache_fio2', 'apache_o2ratio', 'hospital_los', 'unit_los',
                    'unitadmitsource', 'apache_score', 'apache_readmit', 'admission_num'], axis=1)
```

We next turn to formatting the data so that it will be amenable to modeling. This entails converting categorical variables into indicators, and thus we must first convert the strings composing the categories into good variable names.

We'll start with admission diagnoses.

```
cohort = cohort.drop('admit_diagnosis', axis=1)
adx_dummies = pd.get_dummies(cohort.adx_group, 'adx', drop_first=True)
cohort = pd.concat([cohort, adx_dummies], axis=1)
cohort = cohort.drop('adx_group', axis=1)
```

Next we turn to gender and ethnicity.

```
male_gender = (cohort.gender == 'Male').astype('int')
cohort = cohort.assign(male_gender=male_gender)
```

```

cohort = cohort.drop('gender', axis=1)

eth_map = {'Caucasian' : 'caucasian', 'Other/Unknown' : 'other',
           'Native American' : 'native_american', 'African American' : 'african_american',
           'Asian' : 'asian', 'Hispanic' : 'hispanic', " " : 'other'}
cohort.ethnicity = cohort.ethnicity.map(eth_map)
eth_dummies = pd.get_dummies(cohort.ethnicity, 'eth', drop_first=True)
cohort = pd.concat([cohort, eth_dummies], axis=1)
cohort = cohort.drop('ethnicity', axis=1)

```

This leaves unit type as a categorical variable.

```

cohort.unit_type = cohort.unit_type.str.replace('-', '_')
cohort.unit_type = cohort.unit_type.str.replace(' ', '_')
unit_dummies = pd.get_dummies(cohort.unit_type, 'unit', drop_first=True)
cohort = pd.concat([cohort, unit_dummies], axis=1)
cohort = cohort.drop('unit_type', axis=1)

```

Save Train/Test Split of Features and Label

We need to save the label and remove it from the features. We'll also need to save the APACHE prediction so as to incorporate the APACHE IVa model in our analysis.

```
label = (cohort.hospital_expiration == 'EXPIRED').astype('int')
apache_pred = cohort.apache_prediction
cohort = cohort.drop(['hospital_expiration', 'apache_prediction'], axis=1)
```

And now we can form a train test split.

```
train_X, test_X, train_y, test_y, train_apache, test_apache = train_test_split(cohort, label,
apache_pred, test_size=0.25, random_state=42)
```

With that, we can save the CSV files corresponding to data frames we generated.

```
train_X.to_csv('./data/train_X.csv', index=False)
train_y.to_csv('./data/train_y.csv', index=False, header=True)
train_apache.to_csv('./data/train_apache.csv', index=False, header=True)

test_X.to_csv('./data/test_X.csv', index=False)
test_y.to_csv('./data/test_y.csv', index=False, header=True)
test_apache.to_csv('./data/test_apache.csv', index=False, header=True)
```

With this portion complete, we can move onto the construction of our mortality models.

SQL Code for apache-groups

```
-- This query groups patients who have similar APACHE diagnoses into clinically meaningful
categories
-- for example, all the sepsis diagnoses are grouped into one group, 'Sepsis'
-- The remaining groups are categorized as 'other'
DROP MATERIALIZED VIEW IF EXISTS APACHE_GROUPS; CREATE MATERIALIZED
VIEW
APACHE_GROUPS
AS
SELECT
patientunitstayid,
CASE
WHEN apacheadmissiondx IN ('Angina, unstable (angina interferes w/quality of life or meds
are tolerated poorly)', 'Infarction, acute myocardial (MI)', 'MI admitted > 24 hrs after onset of
```

```

ischemia')
    THEN 'ACS'
    WHEN apacheadmissiondx IN ('Chest pain, atypical (noncardiac chest pain)',
    'Chest pain, epigastric',
    'Chest pain, musculoskeletal',
    'Chest pain, respiratory',
    'Chest pain, unknown origin') THEN 'ChestPainUnknown'
    WHEN apacheadmissiondx IN ('Cardiomyopathy', 'CHF, congestive heart failure', 'Shock,
cardiogenic') THEN 'CHF'
    WHEN apacheadmissiondx IN ('Angina, stable (asymptomatic or stable pattern of symptoms
w/meds)',
    'Anomaly, cardiac congenital',
    'Arteriovenous malformation, surgery for',
    'Atrial Septal Defect (ASD) Repair',
    'Cardiovascular medical, other',
    'Cardiovascular surgery, other',
    'Congenital Defect Repair (Other)') THEN 'CVOther'
    WHEN apacheadmissiondx IN ('Contusion, myocardial (include r/o)', 'Effusion, pericardial',
'Endocarditis', 'Hypertension-pulmonary, primary/idiopathic', 'Monitoring, hemodynamic (pre-
operative evaluation)', 'Pericardial effusion/tamponade', 'Pericardiectomy (total/subtotal)',
'Pericarditis', 'Tamponade, pericardial', 'Thrombus, arterial', 'Vascular medical, other',
'Vascular surgery, other') THEN 'CVOther'
    WHEN apacheadmissiondx IN ('Cardiac arrest (with or without respiratory arrest; for
respiratory arrest see Respiratory System)',
    'Rhythm disturbance (atrial, supraventricular)',
    'Rhythm disturbance (conduction defect)',
    'Rhythm disturbance (ventricular)') THEN 'CardiacArrest'
    WHEN apacheadmissiondx IN ('Ablation or mapping of cardiac conduction pathway',
'Defibrillator, automatic implantable cardiac; insertion of') THEN 'CVOther'
    WHEN apacheadmissiondx IN ('CABG alone, coronary artery bypass grafting',
'CABG alone, redo',
'CABG redo with other operation',
'CABG redo with valve repair/replacement',
'CABG with aortic valve replacement',
'CABG with double valve repair/replacement',
'CABG with mitral valve repair',
'CABG with mitral valve replacement',
'CABG with other operation',
'CABG with pulmonic or tricuspid valve repair or replacement ONLY.',
'CABG with single valve repair/replacement',
'CABG, minimally invasive; mid-CABG') THEN 'CABG'
    WHEN apacheadmissiondx IN ('Aortic and Mitral valve replacement', 'Aortic valve
replacement (isolated)', 'Mitral valve repair', 'Mitral valve replacement', 'Papillary muscle
rupture', 'Pulmonary valve surgery', 'Tricuspid valve surgery', 'Valve, double;
repair/replacement', 'Valve, redo, single', 'Valve, single; repair/replacement', 'Valve, triple;
repair/replacement') THEN 'ValveDz'
    WHEN apacheadmissiondx IN ('Pneumonia, aspiration',

```

'Pneumonia, bacterial',
 'Pneumonia, fungal',
 'Pneumonia, other',
 'Pneumonia, parasitic (i.e., Pneumocystic pneumonia)',
 'Pneumonia, viral') THEN 'PNA'
 WHEN apacheadmissiondx IN ('Apnea, sleep', 'Apnea-sleep; surgery for (i.e., UPPP - uvulopalatopharyngoplasty)', 'ARDS-adult respiratory distress syndrome, non-cardiogenic pulmonary edema', 'Arrest, respiratory (without cardiac arrest)') THEN 'RespMedOther'
 WHEN apacheadmissiondx IN ('Atelectasis',
 'Biopsy, open lung',
 'Effusions, pleural',
 'Embolus, pulmonary',
 'Guillain-Barre syndrome',
 'Hemorrhage/hemoptysis, pulmonary',
 'Hemothorax',
 'Obstruction-airway (i.e., acute epiglottitis, post-extubation edema, foreign body, etc)',
 'Pneumothorax',
 'Respiratory - medical, other',
 'Restrictive lung disease (i.e., Sarcoidosis, pulmonary fibrosis)',
 'Tracheostomy',
 'Weaning from mechanical ventilation (transfer from other unit or hospital only)') THEN
 'RespMedOther'
 WHEN apacheadmissiondx IN ('Asthma', 'Emphysema/bronchitis') THEN 'Asthma-Emphys'
 WHEN apacheadmissiondx IN ('Bleeding, GI from esophageal varices/portal hypertension',
 'Bleeding, GI-location unknown',
 'Bleeding, lower GI',
 'Bleeding, upper GI',
 'Bleeding-lower GI, surgery for',
 'Bleeding-other GI, surgery for',
 'Bleeding-upper GI, surgery for',
 'Bleeding-variceal, surgery for (excluding vascular shunting-see surgery for portosystemic shunt)',
 'GI perforation/rupture',
 'GI perforation/rupture, surgery for',
 'Hemorrhage, intra/retroperitoneal',
 'Ulcer disease, peptic') THEN 'GIBleed'
 WHEN apacheadmissiondx IN ('GI obstruction', 'GI obstruction, surgery for (including lysis of adhesions)') THEN 'GIObstruction'
 WHEN apacheadmissiondx IN ('CVA, cerebrovascular accident/stroke',
 'Hemorrhage/hematoma, intracranial',
 'Hemorrhage/hematoma-intracranial, surgery for',
 'Hypertension, uncontrolled (for cerebrovascular accident-see Neurological System)',
 'Subarachnoid hemorrhage/arteriovenous malformation',
 'Subarachnoid hemorrhage/intracranial aneurysm',
 'Subarachnoid hemorrhage/intracranial aneurysm, surgery for') THEN 'CVA'
 WHEN apacheadmissiondx IN ('Abscess, neurologic', 'Biopsy, brain', 'Hydrocephalus,

obstructive', 'Neoplasm, neurologic', 'Neoplasm-cranial, surgery for (excluding transphenoidal)', 'Neoplasm-spinal cord, surgery or other related procedures', 'Neurologic medical, other', 'Neuromuscular medical, other', 'Palsy, cranial nerve', 'Seizures (primary-no structural brain disease)', 'Seizures-intractable, surgery for') THEN 'Neuro'

WHEN apacheadmissiondx IN ('Coma/change in level of consciousness (for hepatic see GI, for diabetic see Endocrine, if related to cardiac arrest, see CV)', 'Nontraumatic coma due to anoxia/ischemia') THEN 'Coma'

WHEN apacheadmissiondx IN ('Overdose, alcohols (bethanol, methanol, ethylene glycol)', 'Overdose, analgesic (aspirin, acetaminophen)', 'Overdose, antidepressants (cyclic, lithium)', 'Overdose, other toxin, poison or drug', 'Overdose, sedatives, hypnotics, antipsychotics, benzodiazepines', 'Overdose, self-inflicted', 'Overdose, street drugs (opiates, cocaine, amphetamine)', 'Toxicity, drug (i.e., beta blockers, calcium channel blockers, etc.)') THEN 'Overdose'

WHEN apacheadmissiondx IN ('Sepsis, cutaneous/soft tissue', 'Sepsis, GI', 'Sepsis, gynecologic', 'Sepsis, other', 'Sepsis, pulmonary', 'Sepsis, renal/UTI (including bladder)', 'Sepsis, unknown') THEN 'Sepsis'

WHEN apacheadmissiondx IN ('Renal failure, acute', 'Renal obstruction') THEN 'ARF'

WHEN apacheadmissiondx IN ('Diabetic hyperglycemic hyperosmolar nonketotic coma (HHNC)', 'Diabetic ketoacidosis') THEN 'DKA'

WHEN apacheadmissiondx IN ('Abdomen only trauma', 'Abdomen/extremity trauma', 'Abdomen/face trauma', 'Abdomen/multiple trauma', 'Abdomen/pelvis trauma', 'Abdomen/spinal trauma', 'Chest thorax only trauma', 'Chest/abdomen trauma', 'Chest/extremity trauma', 'Chest/face trauma', 'Chest/multiple trauma', 'Chest/pelvis trauma', 'Chest/spinal trauma', 'Chest/thorax only trauma', 'Extremity only trauma') THEN 'Trauma'

WHEN apacheadmissiondx IN ('Extremity only trauma, surgery for', 'Extremity/face trauma', 'Extremity/face trauma, surgery for', 'Extremity/multiple trauma', 'Extremity/multiple trauma, surgery for', 'Face only trauma', 'Face only trauma, surgery for', 'Face/multiple trauma', 'Face/multiple trauma, surgery for', 'Facial surgery (if related to trauma, see Trauma)', 'Head only trauma', 'Head/abdomen trauma', 'Head/chest trauma') THEN 'Trauma'

WHEN apacheadmissiondx IN ('Head/extremity trauma', 'Head/face trauma', 'Head/multiple trauma', 'Head/pelvis trauma', 'Head/spinal trauma', 'Pelvis/extremity trauma', 'Pelvis/face trauma', 'Pelvis/hip trauma', 'Pelvis/multiple trauma', 'Pelvis/spinal trauma', 'Spinal cord only trauma', 'Spinal/extremity trauma', 'Spinal/face trauma', 'Spinal/multiple trauma', 'Trauma

```

medical,      other',      'Trauma      surgery,      other')      THEN      'Trauma'
ELSE
'Other'
END
AS
apachedxgroup,
apacheadmissiondx
FROM
patient;

```

SQL Code for labsfirstday

```

-- This query pivots lab values taken in the first 24 hours of a patient's ICU stay
-- Since all eICU stays are centered upon the ICU stay, this uses ICU entry offsets
--      not      admission      to      hospital      offsets.
DROP MATERIALIZED VIEW IF EXISTS labsfirstday CASCADE; CREATE materialized
VIEW
labsfirstday
AS
SELECT
pvt.uniquepid,
pvt.patienthealthsystemstayid,
pvt.patientunitstayid,
MIN(CASE
WHEN      labname      =      'anion      gap'      THEN      labresult
ELSE
NULL
END
)
AS
ANIONGAP_min,
MAX(CASE
WHEN      labname      =      'anion      gap'      THEN      labresult
ELSE
NULL
END
)
AS
ANIONGAP_max,
MIN(CASE
WHEN      labname      =      'albumin'      THEN      labresult
ELSE
NULL
END
)
AS
ALBUMIN_min,
MAX(CASE
WHEN      labname      =      'albumin'      THEN      labresult
ELSE
NULL
END
)
AS
ALBUMIN_max,

```

```

MIN(CASE
  WHEN      labname      =      'amylase'      THEN      labresult
  ELSE
  NULL
END
)
) AS AMYLASE_min,
MAX(CASE
  WHEN      labname      =      'amylase'      THEN      labresult
  ELSE
  NULL
END
)
) AS AMYLASE_max
-- , MIN(CASE WHEN labname = '-bands' THEN labresult ELSE null END) as BANDS_min
-- , MAX(CASE WHEN labname = '-bands' THEN labresult ELSE null END) as BANDS_max
,
MIN(CASE
  WHEN      labname      =      'Base      Excess'      THEN      labresult
  ELSE
  NULL
END
)
) AS BASEEXCESS_min,
MAX(CASE
  WHEN      labname      =      'Base      Excess'      THEN      labresult
  ELSE
  NULL
END
)
) AS BASEEXCESS_max,
MIN(CASE
  WHEN labname = 'bicarbonate' OR labname = 'HCO3' THEN labresult
  ELSE
  NULL
END
)
) AS BICARBONATE_min,
MAX(CASE
  WHEN labname = 'bicarbonate' OR labname = 'HCO3' THEN labresult
  ELSE
  NULL
END
)
) AS BICARBONATE_max,
MIN(CASE
  WHEN      labname      =      'BUN'      THEN      labresult
  ELSE
  NULL
END
)
) AS BUN_min,
MAX(CASE

```



```

    WHEN labname = 'BUN' THEN labresult
  ELSE
  NULL
END
)
AS BUN_max,
MIN(CASE
  WHEN labname = 'BNP' THEN labresult
  ELSE
  NULL
END
)
AS BNP_min,
MAX(CASE
  WHEN labname = 'BNP' THEN labresult
  ELSE
  NULL
END
)
AS BNP_max,
MIN(CASE
  WHEN labname = 'CPK' THEN labresult
  ELSE
  NULL
END
)
AS CPK_min,
MAX(CASE
  WHEN labname = 'CPK' THEN labresult
  ELSE
  NULL
END
)
AS CPK_max,
MIN(CASE
  WHEN labname = 'total bilirubin' THEN labresult
  ELSE
  NULL
END
)
AS BILIRUBIN_min,
MAX(CASE
  WHEN labname = 'total bilirubin' THEN labresult
  ELSE
  NULL
END
)
AS BILIRUBIN_max,
MIN(CASE
  WHEN labname = 'calcium' THEN labresult
  ELSE
  NULL
END

```

```

) AS CALCIUM_min,
MAX(CASE
  WHEN labname = 'calcium' THEN labresult
  ELSE
  NULL
END
) AS CALCIUM_max,
MIN(CASE
  WHEN labname = 'ionized calcium' THEN labresult
  ELSE
  NULL
END
) AS IONCALCIUM_min,
MAX(CASE
  WHEN labname = 'ionized calcium' THEN labresult
  ELSE
  NULL
END
) AS IONCALCIUM_max,
MIN(CASE
  WHEN labname = 'creatinine' THEN labresult
  ELSE
  NULL
END
) AS CREATININE_min,
MAX(CASE
  WHEN labname = 'creatinine' THEN labresult
  ELSE
  NULL
END
) AS CREATININE_max,
MIN(CASE
  WHEN labname = 'chloride' THEN labresult
  ELSE
  NULL
END
) AS CHLORIDE_min,
MAX(CASE
  WHEN labname = 'chloride' THEN labresult
  ELSE
  NULL
END
) AS CHLORIDE_max,
MIN(CASE
  WHEN labname = 'glucose' THEN labresult
  ELSE

```

```

NULL
END
)
AS
GLUCOSE_min,
MAX(CASE
  WHEN labname = 'glucose' THEN labresult
  ELSE
  NULL
END
)
AS
GLUCOSE_max,
MIN(CASE
  WHEN labname = 'fibrinogen' THEN labresult
  ELSE
  NULL
END
)
AS
FIBRINOGEN_min,
MAX(CASE
  WHEN labname = 'fibrinogen' THEN labresult
  ELSE
  NULL
END
)
AS
FIBRINOGEN_max,
MIN(CASE
  WHEN labname = 'Hct' THEN labresult
  ELSE
  NULL
END
)
AS
HEMATOCRIT_min,
MAX(CASE
  WHEN labname = 'Hct' THEN labresult
  ELSE
  NULL
END
)
AS
HEMATOCRIT_max,
MIN(CASE
  WHEN labname = 'Hgb' THEN labresult
  ELSE
  NULL
END
)
AS
HEMOGLOBIN_min,
MAX(CASE
  WHEN labname = 'Hgb' THEN labresult
  ELSE
  NULL
END
)
AS
HEMOGLOBIN_max,
MIN(CASE

```

```

    WHEN labname = 'lactate' THEN labresult
    ELSE
    NULL
END
)
AS LACTATE_min,
MAX(CASE
    WHEN labname = 'lactate' THEN labresult
    ELSE
    NULL
END
)
AS LACTATE_max,
MIN(CASE
    WHEN labname = 'lipase' THEN labresult
    ELSE
    NULL
END
)
AS LIPASE_min,
MAX(CASE
    WHEN labname = 'lipase' THEN labresult
    ELSE
    NULL
END
)
AS LIPASE_max,
MIN(CASE
    WHEN labname = '-lymphs' THEN labresult
    ELSE
    NULL
END
)
AS LYMPHS_min,
MAX(CASE
    WHEN labname = '-lymphs' THEN labresult
    ELSE
    NULL
END
)
AS LYMPHS_max,
MIN(CASE
    WHEN labname = 'magnesium' THEN labresult
    ELSE
    NULL
END
)
AS MAGNESIUM_min,
MAX(CASE
    WHEN labname = 'magnesium' THEN labresult
    ELSE
    NULL
END

```

```

) AS MAGNESIUM_max,
MIN(CASE
  WHEN labname = 'paO2' THEN labresult
  ELSE
  NULL
END
) AS PAO2_min,
MAX(CASE
  WHEN labname = 'paO2' THEN labresult
  ELSE
  NULL
END
) AS PAO2_max,
MIN(CASE
  WHEN labname = 'pH' THEN labresult
  ELSE
  NULL
END
) AS PH_min,
MAX(CASE
  WHEN labname = 'pH' THEN labresult
  ELSE
  NULL
END
) AS PH_max,
MIN(CASE
  WHEN labname = 'platelets x 1000' THEN labresult
  ELSE
  NULL
END
) AS PLATELET_min,
MAX(CASE
  WHEN labname = 'platelets x 1000' THEN labresult
  ELSE
  NULL
END
) AS PLATELET_max,
MIN(CASE
  WHEN labname = '-polys' THEN labresult
  ELSE
  NULL
END
) AS PMN_min,
MAX(CASE
  WHEN labname = '-polys' THEN labresult
  ELSE

```

```

    NULL
END
)
AS
PMN_max,
MIN(CASE
    WHEN labname = 'phosphate' THEN labresult
    ELSE
    NULL
END
)
AS
PHOSPHATE_min,
MAX(CASE
    WHEN labname = 'phosphate' THEN labresult
    ELSE
    NULL
END
)
AS
PHOSPHATE_max,
MIN(CASE
    WHEN labname = 'potassium' THEN labresult
    ELSE
    NULL
END
)
AS
POTASSIUM_min,
MAX(CASE
    WHEN labname = 'potassium' THEN labresult
    ELSE
    NULL
END
)
AS
POTASSIUM_max,
MIN(CASE
    WHEN labname = 'PTT' THEN labresult
    ELSE
    NULL
END
)
AS
PTT_min,
MAX(CASE
    WHEN labname = 'PTT' THEN labresult
    ELSE
    NULL
END
)
AS
PTT_max,
MIN(CASE
    WHEN labname = 'PT - INR' THEN labresult
    ELSE
    NULL
END
)
AS
INR_min,
MAX(CASE

```

```

    WHEN labname = 'PT - INR' THEN labresult
    ELSE
    NULL
END
)
AS INR_max,
MIN(CASE
    WHEN labname = 'PT' THEN labresult
    ELSE
    NULL
END
)
AS PT_min,
MAX(CASE
    WHEN labname = 'PT' THEN labresult
    ELSE
    NULL
END
)
AS PT_max,
MIN(CASE
    WHEN labname = 'sodium' THEN labresult
    ELSE
    NULL
END
)
AS SODIUM_min,
MAX(CASE
    WHEN labname = 'sodium' THEN labresult
    ELSE
    NULL
END
)
AS SODIUM_max,
MIN(CASE
    WHEN labname = 'troponin - I' THEN labresult
    ELSE
    NULL
END
)
AS TROPI_min,
MAX(CASE
    WHEN labname = 'troponin - I' THEN labresult
    ELSE
    NULL
END
)
AS TROPI_max,
MIN(CASE
    WHEN labname = 'troponin - T' THEN labresult
    ELSE
    NULL
END

```

```

    ) AS TROPT_min,
    MAX(CASE
      WHEN labname = 'troponin - T' THEN labresult
      ELSE
        NULL
      END
    ) AS TROPT_max,
    MIN(CASE
      WHEN labname = 'WBC x 1000' THEN labresult
      ELSE
        NULL
      END
    ) AS WBC_min,
    MAX(CASE
      WHEN labname = 'WBC x 1000' THEN labresult
      ELSE
        NULL
      END
    ) AS WBC_max
FROM ( -- begin query that extracts the data
  SELECT
    p.uniquepid,
    p.patienthealthsystemstayid,
    p.patientunitstayid,
    le.labname,
    le.labresult AS labresult
  FROM
    patient p
  LEFT JOIN
    lab le
  ON
    p.patientunitstayid = le.patientunitstayid
    AND le.labresultoffset <= 1440
    AND labresult IS NOT NULL
    AND labresult > 0 -- lab values cannot be 0 and cannot be negative
)
GROUP BY
  pvt.uniquepid,
  pvt.patienthealthsystemstayid,
  pvt.patientunitstayid
ORDER BY
  pvt.uniquepid,
  pvt.patienthealthsystemstayid,
  pvt.patientunitstayid;

```


SQL Code for treatmentfirstday

```
-- View for extraction treatment statuses on first day
DROP MATERIALIZED VIEW IF EXISTS treatmentfirstday CASCADE; CREATE
MATERIALIZED VIEW treatmentfirstday AS
SELECT
  trt.patientunitstayid,
  MAX(CASE
    WHEN LOWER(trt.treatmentstring) LIKE '%vasopressors%' THEN 1
    ELSE 0
  END) AS pressor,
  MAX(CASE
    WHEN LOWER(trt.treatmentstring) LIKE '%antiarrhythmics%' THEN 1
    ELSE 0
  END) AS antiarr,
  MAX(CASE
    WHEN LOWER(trt.treatmentstring) LIKE '%antibacterials%' THEN 1
    ELSE 0
  END) AS abx,
  MAX(CASE
    WHEN LOWER(trt.treatmentstring) LIKE '%sedative%' THEN 1
    ELSE 0
  END) AS sedative,
  MAX(CASE
    WHEN LOWER(trt.treatmentstring) LIKE '%diuretic%' THEN 1
    ELSE 0
  END) AS diuretic,
  MAX(CASE
    WHEN LOWER(trt.treatmentstring) LIKE '%blood product%' THEN 1
    ELSE 0
  END) AS blood_product
FROM
  treatment trt
WHERE
```

trt.treatmentoffset	<=	1440
GROUP		BY
patientunitstayid;		

SQL Code for vitalsfirstday

```
-- This query summarizes the patients first 24 hour vitals after admission to
-- the ICU. eICU contains both periodic and aperiodic vitals, and we will end
-- up wanting the aperiodic pressures because the patients have tons of missing
-- periodic pressures.
SET
search_path TO eicu_crd; DROP MATERIALIZED VIEW IF EXISTS vitalsfirstday CASCADE;
CREATE MATERIALIZED VIEW vitalsfirstday AS
WITH
periodic AS (
SELECT
vppvt.patientunitstayid,
MIN(vppvt.HR) AS HR_Min,
MAX(vppvt.HR) AS HR_Max,
AVG(vppvt.HR) AS HR_Mean,
MIN(vppvt.SBP) AS SBP_periodic_Min,
MAX(vppvt.SBP) AS SBP_periodic_Max,
AVG(vppvt.SBP) AS SBP_periodic_Mean,
MIN(vppvt.DBP) AS DBP_periodic_Min,
MAX(vppvt.DBP) AS DBP_periodic_Max,
AVG(vppvt.DBP) AS DBP_periodic_Mean,
MIN(vppvt.MAP) AS MAP_periodic_Min,
MAX(vppvt.MAP) AS MAP_periodic_Max,
AVG(vppvt.MAP) AS MAP_periodic_Mean,
MIN(vppvt.RR) AS RR_Min,
MAX(vppvt.RR) AS RR_Max,
AVG(vppvt.RR) AS RR_Mean,
MIN(vppvt.SpO2) AS SpO2_Min -- This is a data quality issue; they call this SaO2
,
MAX(vppvt.SpO2) AS SpO2_Max -- and we assume its SpO2. But SaO2 refers to ABG
,
AVG(vppvt.SpO2) AS SpO2_Mean -- in other places.
,
MIN(vppvt.TempC) AS TempC_Min,
MAX(vppvt.TempC) AS TempC_Max,
AVG(vppvt.TempC) AS TempC_Mean
-- Perfrom max sanity checks; we will assume under 0 is impossible, but
-- otherwise no lower bound checks since may be values collected during code.
FROM
```

```

SELECT
  vp.patientunitstayid,
  CASE
    WHEN vp.heartrate > 300 OR vp.heartrate < 0 THEN NULL
    ELSE
      vp.heartrate
  END
  AS
    HR,
  CASE
    WHEN vp.systemicsystolic > 200 OR vp.systemicsystolic < 0 THEN NULL
    ELSE
      vp.systemicsystolic
  END
  AS
    SBP,
  CASE
    WHEN vp.systemicdiastolic > 100 OR vp.systemicdiastolic < 0 THEN NULL
    ELSE
      vp.systemicdiastolic
  END
  AS
    DBP,
  CASE
    WHEN vp.systemicmean > 100 OR vp.systemicmean < 0 THEN NULL
    ELSE
      vp.systemicmean
  END
  AS
    MAP,
  CASE
    WHEN vp.respiration > 60 OR vp.respiration < 0 THEN NULL
    ELSE
      vp.respiration
  END
  AS
    RR,
  CASE
    WHEN vp.sao2 > 100 OR vp.sao2 < 0 THEN NULL
    ELSE
      vp.sao2
  END
  AS
    SpO2,
  CASE
    WHEN vp.temperature > 45 OR vp.temperature < 0 THEN NULL
    ELSE
      vp.temperature
  END
  AS
    tempC
FROM
  vitalperiodic
  vp

```

```

WHERE
  vp.observationoffset <= 1440 ) AS vppvt
GROUP BY
  vppvt.patientunitstayid
ORDER BY
  vppvt.patientunitstayid ),
aperiodic AS (
SELECT
  vapvt.patientunitstayid,
  MIN(vapvt.SBP) AS SBP_aperiodic_Min,
  MAX(vapvt.SBP) AS SBP_aperiodic_Max,
  AVG(vapvt.SBP) AS SBP_aperiodic_Mean,
  MIN(vapvt.DBP) AS DBP_aperiodic_Min,
  MAX(vapvt.DBP) AS DBP_aperiodic_Max,
  AVG(vapvt.DBP) AS DBP_aperiodic_Mean,
  MIN(vapvt.MAP) AS MAP_aperiodic_Min,
  MAX(vapvt.MAP) AS MAP_aperiodic_Max,
  AVG(vapvt.MAP) AS MAP_aperiodic_Mean
-- Perfrom max sanity checks; we will assume under 0 is impossible, but
-- otherwise no lower bound checks since may be values collected during code.
FROM (
  SELECT
    va.patientunitstayid,
    CASE
      WHEN va.noninvasivesystolic > 200 OR va.noninvasivesystolic < 0 THEN NULL
    ELSE
      va.noninvasivesystolic
    END
    AS SBP,
    CASE
      WHEN va.noninvasivediastolic > 100 OR va.noninvasivediastolic < 0 THEN NULL
    ELSE
      va.noninvasivediastolic
    END
    AS DBP,
    CASE
      WHEN va.noninvasivemean > 100 OR va.noninvasivemean < 0 THEN NULL
    ELSE
      va.noninvasivemean
    END
    AS MAP
  FROM
    vitalaperiodic va
  WHERE
    va.observationoffset <= 1440 ) AS vapvt
GROUP BY

```

```

    vapvt.patientunitstayid
ORDER
    vapvt.patientunitstayid
SELECT
    pp.patientunitstayid,
    pp.HR_Min,
    pp.HR_Max,
    pp.HR_Mean,
    pp.SBP_periodic_Min,
    pp.SBP_periodic_Max,
    pp.SBP_periodic_Mean,
    pp.DBP_periodic_Min,
    pp.DBP_periodic_Max,
    pp.DBP_periodic_Mean,
    pp.MAP_periodic_Min,
    pp.MAP_periodic_Max,
    pp.MAP_periodic_Mean,
    ap.SBP_aperiodic_Min,
    ap.SBP_aperiodic_Max,
    ap.SBP_aperiodic_Mean,
    ap.DBP_aperiodic_Min,
    ap.DBP_aperiodic_Max,
    ap.DBP_aperiodic_Mean,
    ap.MAP_aperiodic_Min,
    ap.MAP_aperiodic_Max,
    ap.MAP_aperiodic_Mean,
    pp.RR_Min,
    pp.RR_Max,
    pp.RR_Mean,
    pp.SpO2_Min,
    pp.SpO2_Max,
    pp.SpO2_Mean,
    pp.TempC_Min,
    pp.TempC_Max,
    pp.TempC_Mean
FROM
    periodic
LEFT
    aperiodic
ON
    pp.patientunitstayid = ap.patientunitstayid;

```

BY
)

pp
JOIN
ap

Modeling

Model Analysis

```

# modelanalysis.py
# Christopher V. Cosgriff, Harvard Chan School and NYU School of Medicine
# This file includes a number of helper functions for generating plots and
# metrics for the high-risk severity score project.

#: Document function inputs/outputs
#: Combine CI functions into one bootstrap function

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, roc_auc_score, precision_recall_curve,
average_precision_score
from sklearn.calibration import calibration_curve

import matplotlib.pyplot as plt
import seaborn as sns

# Functions for generating ROC curves and calculating AUROC
def auc_ci(f_hat, y_true, n_bootstraps=2000, ci_level=0.95):
    li = (1. - ci_level)/2
    ui = 1 - li

    rng = np.random.RandomState(seed=42)
    bootstrapped_auc = []

    for i in range(n_bootstraps):
        indices = rng.randint(0, len(f_hat), len(f_hat))
        auc = roc_auc_score(y_true[indices], f_hat[indices])
        bootstrapped_auc.append(auc)

    sorted_scores = np.array(bootstrapped_auc)
    sorted_scores.sort()
    confidence_lower = sorted_scores[int(li * len(sorted_scores))]
    confidence_upper = sorted_scores[int(ui * len(sorted_scores))]

    return confidence_lower, confidence_upper

def gen_auc_plot(models, names, title, X, y, ci_level=None, save_name=None):
    plt.figure(figsize=(10, 10))
    for i, model in enumerate(models):

```

```

f_hat = model.predict_proba(X)
roc = roc_curve(y, f_hat[:, 1])
auc = roc_auc_score(y, f_hat[:, 1])
if (ci_level != None):
    ci = auc_ci(f_hat=f_hat[:, 1], y_true=y, ci_level=ci_level)
    sns.lineplot(x=roc[0], y=roc[1], label='{0}\n(AUC = {1:.3f} [{2:.3f}, {3:.3f}])'.format(names[i], auc, *ci))
else:
    sns.lineplot(x=roc[0], y=roc[1], label='{0}\n(AUC = {1:.3f})'.format(names[i], auc))
plt.plot([0, 1], [0, 1], 'k:')
plt.xlabel('1 - Specificity')
plt.ylabel('Sensitivity')
plt.title(title)
if save_name != None:
    plt.savefig(save_name + '.svg', bbox_inches='tight')
plt.show()

# Functions for generating PRC curves and calculating AUPRC (AP)
def prc_ci(f_hat, y_true, n_bootstraps=2000, ci_level=0.95):
    li = (1. - ci_level)/2
    ui = 1 - li

    rng = np.random.RandomState(seed=42)
    bootstrapped_ap = []

    for i in range(n_bootstraps):
        indices = rng.randint(0, len(f_hat), len(f_hat))
        ap = average_precision_score(y_true[indices], f_hat[indices])
        bootstrapped_ap.append(ap)

    sorted_scores = np.array(bootstrapped_ap)
    sorted_scores.sort()
    confidence_lower = sorted_scores[int(li * len(sorted_scores))]
    confidence_upper = sorted_scores[int(ui * len(sorted_scores))]

    return confidence_lower, confidence_upper

def gen_prc_plot(models, names, title, X, y, ci_level=None, save_name=None):
    plt.figure(figsize=(10, 10))
    for i, model in enumerate(models):
        f_hat = model.predict_proba(X)
        precision, recall, _ = precision_recall_curve(y, f_hat[:, 1])
        ap = average_precision_score(y, f_hat[:, 1])
        if ci_level != None:
            ci = prc_ci(f_hat=f_hat[:, 1], y_true=y, ci_level=ci_level)
            sns.lineplot(x=recall, y=precision, label='{0}\n(AP = {1:.3f} [{2:.3f}, {3:.3f}])'.format(names[i], ap, *ci))

```

```

{3:.3f})).format(names[i], ap, *ci))
    else:
        sns.lineplot(x=recall, y=precision, label='{0}\n(AP = {1:.3f})'.format(names[i], ap))
        plt.xlabel('Recall')
        plt.ylabel('Precision')
        plt.title(title)
        if save_name != None:
            plt.savefig(save_name + '.svg', bbox_inches='tight')
        plt.show()

# Function for generating reliability curves
def gen_calib_plot(models, names, title, X, y, save_name=None):
    plt.figure(figsize=(10, 10))
    for i, model in enumerate(models):
        f_hat = model.predict_proba(X)
        fraction_of_positives, mean_predicted_value = calibration_curve(y, f_hat[:, 1], n_bins=10)
        sns.lineplot(x=mean_predicted_value, y=fraction_of_positives, label=names[i])
        plt.plot([0, 1], [0, 1], 'k:')
        plt.xlabel('Mean Predicted Value')
        plt.ylabel('Actual Probability of Outcome')
        plt.title(title)
        if save_name != None:
            plt.savefig(save_name + '.svg', bbox_inches='tight')
        plt.show()

# Functions for generating NRI
def nri(model1, model2, X, y):
    n = y.shape[0]
    n_event = y.sum()
    n_no_event = n - n_event

    pred_model1 = model1.predict_proba(X)[:, 1]
    pred_model2 = model2.predict_proba(X)[:, 1]
    z_pos = (pred_model2[y == 1] > pred_model1[y == 1]).sum() - (pred_model2[y == 1] <
pred_model1[y == 1]).sum()
    z_neg = (pred_model2[y == 0] < pred_model1[y == 0]).sum() - (pred_model2[y == 0] >
pred_model1[y == 0]).sum()

    additive_nri = (z_pos/n_event)*100 + (z_neg/n_no_event)*100
    absolute_nri = (z_pos + z_neg)/n
    return (additive_nri, absolute_nri)

def nri_grid(models, names, X, y, round=True):
    n_models = len(models)
    additive_grid = np.zeros(shape=(n_models, n_models), dtype=float)
    absolute_grid = np.zeros(shape=(n_models, n_models), dtype=float)

```



```

for i, model1 in enumerate(models):
    for j, model2 in enumerate(models):
        if i < j:
            additive_nri, absolute_nri = nri(model1=model1, model2=model2,
                                             X=X, y=y)
            additive_grid[i, j] = additive_nri
            absolute_grid[i, j] = absolute_nri
            additive_grid = additive_grid + np.flip(additive_grid)*-1
            absolute_grid = absolute_grid + np.flip(absolute_grid)*-1
            additive_grid = pd.DataFrame(data=additive_grid, index=names, columns=names)
            absolute_grid = pd.DataFrame(data=absolute_grid, index=names, columns=names)
            if round:
                return (additive_grid.round(0), absolute_grid.round(2))
            else:
                return (additive_grid, absolute_grid)

# Functions for generating OPR table
def op_ratio_ci(f_hat, y_true, n_bootstraps=2000, ci_level=0.95):
    li = (1. - ci_level)/2
    ui = 1. - li

    rng = np.random.RandomState(seed=42)
    bootstrapped_opr = []

    for i in range(n_bootstraps):
        indices = rng.randint(0, len(f_hat), len(f_hat))
        opr = y_true[indices].mean() / f_hat[indices].mean()
        bootstrapped_opr.append(opr)

    sorted_scores = np.array(bootstrapped_opr)
    sorted_scores.sort()
    confidence_lower = sorted_scores[int(li * len(sorted_scores))]
    confidence_upper = sorted_scores[int(ui * len(sorted_scores))]

    return confidence_lower, confidence_upper

def op_ratio(model, X, y):
    f_hat = model.predict_proba(X)
    observed = y.mean()
    predicted = f_hat[:, 1].mean()
    return (observed / predicted, *op_ratio_ci(f_hat[:, 1], y))

def opr_table(models, names, X, y):
    opr = [op_ratio(model, X, y) for model in models]
    opr_table = pd.DataFrame(data=opr, index=names, columns=['OPR', '2.5%', '97.5%'])
    return opr_table

```

```

# Feature analysis function for normalized regression model
def gen_logodds_plot(model, features, n_features=10, title='Log Odds Plot',
save_name=None):
    plt.figure(figsize=(10, 10))
    coef = {k:v for k, v in zip(features, model.named_steps['ridge'].coef_.ravel())}
    coef = pd.DataFrame.from_dict(coef, orient='index', columns=['log_odds'])
    coef = coef.reindex(coef.log_odds.abs().sort_values(ascending=False).index).iloc[0:n_features, :]
    coef = coef.reindex(index=coef.index[:-1])
    pos_neg = coef.log_odds > 0
    color_map = pos_neg.map({True: tableau20[8], False: tableau20[10]})
    coef.plot(kind='barh', grid=True, sort_columns=False,
            title=title,
            color=[color_map.values], ax=plt.axes(), width=0.20,
            legend=False)
    plt.xlabel('log(OR)')
    plt.ylabel('Features')
    if save_name != None:
        plt.savefig('./figures/' + save_name + '.svg', bbox_inches='tight')
    plt.show()

# Wrapper for working with APACHE model from eICU which just provides
# the probabilities from the model.
class APACHEWrapper:
    def __init__(self, pos_preds_proba):
        self.proba = pos_preds_proba

    def predict_proba(self, X):
        return np.array([1-self.proba, self.proba]).T

```

Model Metrics

Environement

```

import pandas as pd
import pickle

from sequential_model import SequentialModel
from model_analysis import APACHEWrapper
from model_analysis import gen_auc_plot, gen_calib_plot, gen_prc_plot
from model_analysis import import nri_grid, opr_table

# Load seaborn to set plot styles
import seaborn as sns

```

```
sns.set_style('whitegrid')
```

```
%matplotlib
```

```
inline
```

```
%config InlineBackend.figure_format = 'retina'
```

Models and Data

```
test_X      = pd.read_csv('../extraction/data/test_X.csv').set_index('patientunitstayid').values
test_y      = pd.read_csv('../extraction/data/test_y.csv').values.ravel()

test_apache = pd.read_csv('../extraction/data/test_apache.csv').values.ravel()
apache_model = APACHEWrapper(test_apache)

logit_full  = pickle.load(open('./logit_full', 'rb'))

logit_HR    = pickle.load(open('./logit_HR', 'rb'))
logit_hHR   = pickle.load(open('./logit_hHR', 'rb'))
seq_logit_HR = SequentialModel(logit_full, logit_HR, cutoff=0.10)
seq_logit_hHR = SequentialModel(logit_full, logit_hHR, cutoff=0.50)

xgb_full = pickle.load(open('./xgb_full', 'rb'))
```

Receiver Operator Characteristic and Precision-Recall Curves

```
gen_auc_plot(models=[xgb_full, logit_full, seq_logit_HR, seq_logit_hHR, apache_model],
names=['XGB', 'Logit', 'SeqLogit-0.10', 'SeqLogit-0.50', 'APACHE IVa'],
title='Receiver Operator Characteristic Curves', X=test_X, y=test_y, ci_level=0.95,
save_name='auroc_all_models')
```

```
gen_prc_plot(models=[xgb_full, logit_full, seq_logit_HR, seq_logit_hHR, apache_model],
names=['XGB', 'Logit', 'SeqLogit-0.10', 'SeqLogit-0.50', 'APACHE IVa'],
title='Precision-Recall Curves', X=test_X, y=test_y, ci_level=0.95,
save_name='auprc_all_models')
```

Reliability Curves

In [ ]:

```
gen_calib_plot(models=[apache_model, logit_full], names=['APACHE IVa', 'Logit'],
```

```
title='Reliability Curve: APACHE IVa', X=test_X, y=test_y)
```

```
gen_calib_plot(models=[seq_logit_HR, seq_logit_hHR], names=['SeqLogit-0.10', 'SeqLogit-0.50'],
               title='Reliability Curves: Linear and Sequential Models', X=test_X, y=test_y)
```

```
gen_calib_plot(models=[xgb_full], names=['XGB'], title='Reliability Curve: XGB Model', X=test_X,
               y=test_y)
```

Observed-to-Predicted Mortality Ratios

```
opr = opr_table(models=[apache_model, logit_full, seq_logit_HR, seq_logit_hHR, xgb_full],
                names=['APACHE IVa', 'Logit', 'SeqLogit-0.10', 'SeqLogit-0.50', 'XGB'],
                X=test_X, y=test_y)
opr.to_csv('./opr-table.csv')
```

Sequential Model

```
import numpy as np
import pandas as pd

#: Document functions in sequential model class
#: Write a fit function for class so not doing manually

class SequentialModel:
    def __init__(self, base_model, HR_model, cutoff=0.10, prefit=True):
        self.base_model = base_model
        self.HR_model = HR_model
        self.cutoff = cutoff
```

```

        self.prefit = prefit

    def predict_proba(self, X):
        initial_risk = self.base_model.predict_proba(X)
        hr_risk = self.HR_model.predict_proba(X)

        risk = np.where(initial_risk[:, 1] >= self.cutoff, hr_risk[:, 1], initial_risk[:, 1])
        risk = np.array([1-risk, risk]).T
        return risk

```

Train Models

```

import numpy as np
import pandas as pd

from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import BayesianRidge
from sklearn.impute import IterativeImputer
from sklearn.linear_model import LogisticRegressionCV, LogisticRegression
from sklearn.model_selection import StratifiedKFold, RandomizedSearchCV

from xgboost import XGBClassifier

import pickle

# Document functions
# Clean up sequential training

def train_logit(X, y):
    brm = BayesianRidge(n_iter=1000)
    iter_imp = IterativeImputer(estimator=brm, sample_posterior=True,
                                max_iter=25)
    logit_classifier = Pipeline([
        ('impute', IterativeImputer(estimator=brm, sample_posterior=True,
                                     max_iter=25)),
        ('center_scale', StandardScaler()),
        ('logit', LogisticRegression(C=10**10, solver='lbfgs', max_iter=1000))]
    )
    logit_classifier.fit(X, y)
    return logit_classifier

def train_xgb_rgridCV(X, y, params=None, K=10, n_iter=100, n_cpu=1, GPU=True):
    if params == None:
        params = {'objective': 'binary:logistic',

```

```

        'learning_rate': [0.01, 0.05, 0.10],
        'max_depth': [3, 6, 9, 12],
        'min_child_weight': [6, 8, 10, 12],
        'silent': [True],
        'subsample': [0.6, 0.8, 1],
        'colsample_bytree': [0.5, 0.75, 1],
        'n_estimators': [500, 1000]}

    if GPU:
        xgb_classifier = XGBClassifier(tree_method='gpu_hist',
predictor='gpu_predictor')
    else:
        xgb_classifier = XGBClassifier()

    skf = StratifiedKFold(n_splits=K, shuffle=True, random_state=42)
    cv_grid_search = RandomizedSearchCV(xgb_classifier, param_distributions=params,
        n_iter=n_iter, scoring='neg_log_loss', n_jobs=n_cpu, cv=skf.split(X, y),
        verbose=2, random_state=42)
    cv_grid_search.fit(X, y)
    xgb_classifier = cv_grid_search.best_estimator_
    xgb_classifier.fit(X, y)
    return xgb_classifier

def save_model(model, file_name):
    pickle.dump(model, open('{0}'.format(file_name), 'wb'))

# Load training data
train_X = pd.read_csv('./extraction/data/train_X.csv').set_index('patientunitstayid').values
train_y = pd.read_csv('./extraction/data/train_y.csv').values.ravel()

#First train full cohort models
print('Training logistic regression model.')
logit_full = train_logit(X=train_X, y=train_y)
save_model(logit_full, 'logit_full')
print('Done, model saved.')

print('Training XGB model.')
xgb_full = train_xgb_rgridCV(X=train_X, y=train_y, K=10, n_cpu=4, GPU=True)
save_model(xgb_full, 'xgb_full')
print('Done, model saved.')

# Then train HR logit model at 0.10 cutoff
print('Using base logit model to identify HR (0.10) patients.')
logit_full = pickle.load(open('./logit_full', 'rb'))
initial_risk = logit_full.predict_proba(train_X)
train_X_HR = train_X[initial_risk[:, 1] >= 0.10, :]

```

```

train_y_HR      =      train_y[initial_risk[:,      1]      >=      0.10]
print('Training      HR      logistic      regression      model.')
logit_HR        =      train_logit(X=train_X_HR,      y=train_y_HR)
save_model(logit_HR,
print('Done,      model      saved.')

#      Then      train      hHR      logit      model      at      0.50      cutoff
print('Using      base      logit      model      to      identify      hHR      patients.')
train_X_hHR      =      train_X[initial_risk[:,      1]      >=      0.50,      :]
train_y_hHR      =      train_y[initial_risk[:,      1]      >=      0.50]
print('Training      hHR      logistic      regression      model.')
logit_hHR        =      train_logit(X=train_X_hHR,      y=train_y_hHR)
save_model(logit_hHR,
print('Done, model saved.')

```

APPENDIX V: TEMPORAL TRENDS IN CRITICAL CARE OUTCOMES IN UNITED STATES MINORITY SERVING HOSPITALS

Data Supplement

Temporal trends of critical illness mortality rates according to hospital minority composition					
	2006-08	2009-10	2011-12	2013-14	2015-16
All hospitals	10.6	10.2	9.6	9.2	9.2
Minority Serving hospitals	11.1	10.5	10.9	10.0	10.3
African American	10.3	11.3	10.6	10.7	10.2
Hispanic	10.6	9.7	10.4	9.4	10.9
White	12.4	11.3	11.6	10.4	10.6
Non-minority serving hospitals	10.5	10.2	9.4	9.2	9.1
African American	10.1	10.1	8.8	8.3	8.4
Hispanic	7.3	8.1	7.0	7.6	8.0
White	10.7	10.3	9.5	9.3	9.3

Supplemental table I. Crude critical illness mortality rates (percentages) in minority and non-minority serving hospitals, stratified by ethnicity.

APPENDIX VI: INCIDENCE AND RISK MODEL DEVELOPMENT FOR SEVERE TACHYPNEA FOLLOWING TERMINAL EXTUBATION

Code

```
---
title: "Cohort Creation"
author: "Miguel Angel Armengol de la Hoz"
date: "August 1, 2018"
output: html_document
---

## Data extraction

### CMO

```{r}
code_status_cmo<-run_query(
#con,
'
with sq as(SELECT distinct subject_id, hadm_id, icustay_id
,COALESCE(timecmo_chart, timecmo_nursingnote, NULL ) AS codestatus_time
,Row_Number() Over (Partition By code_status.subject_id ORDER BY
COALESCE(timecmo_chart, timecmo_nursingnote, NULL ) Desc) As position
FROM mimiciii.code_status
WHERE
cmo_first = 1 OR
cmo_last = 1 OR
cmo = 1)
SELECT
subject_id
,hadm_id
,icustay_id
,codestatus_time
FROM sq
WHERE
position=1
')

noteevents_cmo<-dbGetQuery(con,"with sq AS(SELECT distinct subject_id, hadm_id
,Row_Number() Over (Partition By noteevents.subject_id ORDER BY noteevents.charttime
Desc) As position,
charttime AS charttime
FROM mimiciii.noteevents
WHERE
text ILIKE '%cmo%'
OR
text ILIKE '%comfort meas%'
OR
```

```

text ILIKE '%terminal ext%'
OR
text ILIKE '%terminally ext%'
)
SELECT subject_id, hadm_id, charttime
FROM sq
WHERE position =1 ")

noteevents_cmo<-unique(noteevents_cmo)

more_cmo_using_nlptool<-dbGetQuery(con,"with sq as(
SELECT DISTINCT subject_id
,hadm_id
,charttime
,Row_Number() Over (Partition By mimiciiii.noteevents.subject_id Order By
mimiciiii.noteevents.charttime DESC) As position
FROM mimiciiii.noteevents
WHERE
text similar to '%(made comfort measures only|confort measures|cmo status|confort
measures only|pursue comfort measures|dnr/dni/comfort|institute comfort
measures|comfort measures only|comfort measure|terminally extubated|comfort
focused|comfort-measures-only|made comfort measures|made cmo|cmo|comfort
measures|comfort measures only|confort care|comfort care|comofort)%'
)
SELECT
subject_id
,hadm_id
,charttime
FROM sq
WHERE position =1 --we only want the hadm_id associated with the very last note
ORDER BY subject_id")
...

#### Union

```{r}
#preparing datasets for union
library(dplyr)
# noteevents_cmo_notin_cs<-subset(noteevents_cmo, !(noteevents_cmo$subject_id %in%
code_status_cmo$subject_id) )
# colnames(noteevents_cmo_notin_cs)[3]<- 'charttime'

code_status_cmo$icustay_id<-NULL
code_status_cmo<- code_status_cmo %>%
select(subject_id, hadm_id, codestatus_time)
colnames(code_status_cmo)[3]<- 'charttime'

cmo_final<-union(code_status_cmo,noteevents_cmo)
cmo_final<-union(cmo_final,more_cmo_using_nlptool)

#we sort if descendand
cmo_final<-cmo_final[order(-cmo_final$subject_id,-unclass(cmo_final$charttime)), ]
#we take only the first (most recent) timestamp per hadm_id
cmo_final<-cmo_final[!duplicated(cmo_final$subject_id),]

```

```
```
```

```
### Extubated
```

```
```{r}
extubated_procedureevents_mv<-dbGetQuery(con,'
with select_sq AS(
SELECT subject_id
,hadm_id
,icustay_id
,starttime AS charttime,
  Row_Number() Over (PARTITION BY procedureevents_mv.subject_id ORDER BY
procedureevents_mv.starttime Desc) As position
  FROM mimiciii.procedureevents_mv
  where itemid = 227194 -- "Extubation"
)
SELECT subject_id, hadm_id, icustay_id,charttime
FROM select_sq
WHERE position =1
')
```

```
# we add this columns to identify which patients are gettin the extubation from the
most reliable extubation time source
```

```
extubated_procedureevents_mv['extubated_in_procedureevents']<-1
```

```
# we identified extubation using this table by considering those patients who were
receiving mechanical ventilation and suddently it stopped as extubated patients.
```

```
mv_chartevents<-dbGetQuery(con,"with sq AS(
SELECT distinct subject_id
,hadm_id
,icustay_id
,charttime AS charttime
,Row_Number() OVER (PARTITION BY chartevents.subject_id ORDER BY
chartevents.charttime DESC) As position
FROM mimiciii.chartevents
WHERE
(itemid = 720 and value != 'Other/Remarks') -- VentTypeRecorded
OR (itemid = 223848 and value != 'Other')
OR (itemid = 223849) -- ventilator mode
OR (itemid = 467 and value = 'Ventilator') -- O2 delivery device == ventilator
OR itemid in
(
  445, 448, 449, 450, 1340, 1486, 1600, 224687 -- minute volume
, 639, 654, 681, 682, 683, 684,224685,224684,224686 -- tidal volume
, 218,436,535,444,459,224697,224695,224696,224746,224747 --
High/Low/Peak/Mean/Neg insp force ('RespPressure')
, 221,1,1211,1655,2000,226873,224738,224419,224750,227187 -- Insp pressure
, 543 -- PlateauPressure
, 5865,5866,224707,224709,224705,224706 -- APRV pressure
, 60,437,505,506,686,220339,224700 -- PEEP
, 3459 -- high pressure relief
, 501,502,503,224702 -- PCV
)
```

```

        , 223,667,668,669,670,671,672 -- TCPCV
        , 224701 -- PSVlevel
    )
)
SELECT sq.subject_id
,sq.hadm_id
,sq.icustay_id
,sq.charttime
FROM sq
INNER JOIN mimicii.icustays
ON sq.hadm_id = icustays.hadm_id
WHERE POSITION = 1
AND
(dbsource = 'metavision' OR dbsource = 'both')
ORDER BY hadm_id ASC  ")

...

#### Union

```{r}

mv_chartevents<-unique(mv_chartevents)

#Performing union in both datasets

extubated_final<-union(extubated_procedureevents_mv,mv_chartevents)
#we sort if descendand
extubated_final<-extubated_final[order(-extubated_final$subject_id,-
unclass(extubated_final$charttime)), ]

#we take only the first (most recent) timestamp per hadm_id
extubated_final<-extubated_final[!duplicated(extubated_final$subject_id),]
```

CMO and extubation

```{r}
library(dplyr)

cmo_extubation<-inner_join(extubated_final, cmo_final, by = c("subject_id"))
colnames(cmo_extubation)<-
c("subject_id","hadm_id.extubation",'icustay_id',"charttime.extubation","hadm_id.cmo"
,"charttime.cmo")

cmo_extubation<-unique(cmo_extubation)

#we sort if descendand
cmo_extubation<-cmo_extubation[order(-cmo_extubation$subject_id,-
unclass(cmo_extubation$charttime.extubation)), ]

#we want only the last timestamp associated with the last hadm_id
cmo_extubation<-cmo_extubation[!duplicated(cmo_extubation$hadm_id.cmo),]

```

```

library(dplyr)
cmo_extubation['hadm_id']<-
  if_else(
    cmo_extubation$hadm_id.extubation <= cmo_extubation$hadm_id.cmo,
    cmo_extubation$hadm_id.cmo, cmo_extubation$hadm_id.extubation)

cmo_extubation['time_ext_to_cmo_hrs']<-round((cmo_extubation$charttime.cmo -
cmo_extubation$charttime.extubation)/3600,2)

#let's reorder the dataset and select only those columns we need
cmo_extubation<-cmo_extubation %>%

select(subject_id,hadm_id,icustay_id,charttime.extubation,charttime.cmo,time_ext_to_c
mo_hrs )
```

Deaths

```{r}
death_icu_or_hosp_query <-dbGetQuery(con,"select i.subject_id,i.hadm_id, i.icustay_id
, i.intime, i.outtime
, ad.deathtime
, extract(epoch from ad.deathtime-i.outtime)/60.0/60.0/24.0 as
time_to_deathtime_days
, p.dod
, extract(epoch from p.dod-date_trunc('day',i.outtime))/60.0/60.0/24.0 as
time_to_dod_days
, case when tr.intime = i.outtime then 1 else 0 end as icu_was_last_ward
--last ward means the patient last location of stay before discharge or death was the
icu
, case when extract(epoch from ad.deathtime-i.outtime)/60.0/60.0 < 2 then 1
when tr.intime = i.outtime and ad.deathtime is not null then 1
else 0 end as icu_mortality
,ad.hospital_expire_flag as hosp_mortality
FROM
mimiciii.icustays i
inner join mimiciii.admissions ad on i.hadm_id = ad.hadm_id
inner join mimiciii.patients p on i.subject_id = p.subject_id
left join ( select hadm_id, intime, outtime, row_number() over (partition by
hadm_id order by intime desc) rn from mimiciii.transfers where intime is not null )
tr
on i.hadm_id = tr.hadm_id
and tr.rn = 1
order by time_to_deathtime_days, time_to_dod_days
")
```

CMO, extubation and death

```{r}
detach("package:dplyr")
library(dplyr)

death_icu_or_hosp<-death_icu_or_hosp_query %>%
  filter(icu_mortality==1 | hosp_mortality==1) %>%

```

```

select(subject_id, deathtime,icu_mortality,hosp_mortality)

#we sort if descendand
death_icu_or_hosp<-death_icu_or_hosp[order(-death_icu_or_hosp$subject_id,-
unclass(death_icu_or_hosp$deathtime)), ]

#we want only the last timestamp associated with the last hadm_id
death_icu_or_hosp<-death_icu_or_hosp[!duplicated(death_icu_or_hosp$subject_id),]

cmo_and_extubation_and_death<-inner_join(cmo_extubation,death_icu_or_hosp)
...

# CMO, extubation before death

```{r}
library(dplyr)
cmo_and_extubation_and_death['time_from_ext_to_death_hrs']<-
  round(difftime(cmo_and_extubation_and_death$deathtime
                ,cmo_and_extubation_and_death$charttime.extubation
                ,units = 'hours'),2)

#converting to numeric
cmo_and_extubation_and_death$time_from_ext_to_death_hrs<-
round(as.double(cmo_and_extubation_and_death$time_from_ext_to_death_hrs),2)
cmo_and_extubation_and_death$time_ext_to_cmo_hrs<-
round(as.double(cmo_and_extubation_and_death$time_ext_to_cmo_hrs),2)

# q<-plot_ly(x = cmo_and_extubation_and_death$time_from_ext_to_death_hrs, type =
"histogram"
#           ,autobinx=FALSE
#           ,xbins=list(start=-14,end=30000,size=5))%>%
#   layout(title='time_from_ext_to_death_hrs')
#
# hist(as.numeric(cmo_and_extubation_and_death$time_from_ext_to_death_hrs))

cmo_and_extubation_and_death[cmo_and_extubation_and_death$subject_id==90078,]$hadm_id
<-163063

cmo_and_extubation_and_death_24<-cmo_and_extubation_and_death%>%
  filter(time_from_ext_to_death_hrs >= 0 & time_from_ext_to_death_hrs<=24)

cmo_and_extubation_and_death_48<-cmo_and_extubation_and_death%>%
  filter(time_from_ext_to_death_hrs >= 0 & time_from_ext_to_death_hrs<=48)
...

---
title: "Epoch Creation"
author: "Miguel Angel Armengol de la Hoz"
date: "August 1, 2018"
output: html_document
---
```

```

# Libraries and SQL connection

```{r}
library(summarytools)
library(bigrquery)
#library(RPostgreSQL)
library(DBI)
library(reshape)
library(dplyr)
```

```{r}
library(bigrquery)
project_id <- "hst-953-2018"
options(httr_oauth_cache=FALSE)
run_query <- function(query){
  data <- query_exec(query, project=project_id, use_legacy_sql=FALSE,max_pages = Inf)
  return(data)
}
```

Getting the list of ids from our previously generated cohort.

```{r}
subject_ids_fromourcohort24<-cmo_and_extubation_and_death_24$subject_id
subject_ids_fromourcohort24 <- toString(sprintf("%s", subject_ids_fromourcohort24))

hadm_ids_fromourcohort24<-cmo_and_extubation_and_death_24$hadm_id
hadm_ids_fromourcohort24 <- toString(sprintf("%s", hadm_ids_fromourcohort24))
```

Tachypnea Detection Binary

Data extraction

- ATTENTION: FOR SOME REASON THE LIBRARY IS NOT EXTRACTING TIME FROM THE DATETIME, only the data so I used pgadmin and exporter in csv.
- Extracting the data in csv was causing a lot of datetime to be NULL when importing them, so I used Big Query.
- Extracting the data in Big Query was changing the timezone so it needed to be reseted.

```{r}
query<-paste0("
SELECT
  chartevents.subject_id,
  chartevents.itemid,
  COALESCE (storetime, chartevents.charttime )AS charttime_rr,
  chartevents.valuenum
From
  `physionet-data.mimiciii_clinical.chartevents` chartevents
WHERE
  -- itemid = 220210 -- The most frequent RR item_id for our cohort

```

```

chartevents.itemid IN (220210, 224688, 224689, 618, 224690,
619,224422,7884,224718,227050,6106,3142,615,1635,219,223851,3603,614,8113,1884,653)
AND
subject_id IN ("subject_ids_fromourcohort24,")
AND
valuenum IS NOT NULL
")

#respiratory_rate<-dbGetQuery(con, query)
respiratory_rate_charttime_cohort_24<-query_exec(query, project =
project_HST,use_legacy_sql = F,max_pages = Inf)

# 822/822 patients with RR information
```

Epochs Creation

```{r}
library(dplyr)
rr_from_patients_tofilter<-inner_join(cmo_and_extubation_and_death_24[
,c('subject_id','icustay_id','charttime.extubation','deathtime')]
, respiratory_rate_charttime_cohort_24)

# we noticed some timezones are alteterd, so we reset them
rr_from_patients_tofilter$charttime_rr<-
as.character(rr_from_patients_tofilter$charttime_rr)
rr_from_patients_tofilter$charttime.extubation<-
as.character(rr_from_patients_tofilter$charttime.extubation)
rr_from_patients_tofilter$deathtime<-
as.character(rr_from_patients_tofilter$deathtime)

rr_from_patients_tofilter$charttime_rr<-
as.POSIXct(rr_from_patients_tofilter$charttime_rr, format='%Y-%m-%d %H:%M:%S',
tz="GMT")
rr_from_patients_tofilter$charttime.extubation<-
as.POSIXct(rr_from_patients_tofilter$charttime.extubation, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
rr_from_patients_tofilter$deathtime<-as.POSIXct(rr_from_patients_tofilter$deathtime,
format='%Y-%m-%d %H:%M:%S', tz="GMT")

rr_from_patients_tofilter['timefrom_ext_to_rr_mins']<-
difftime(rr_from_patients_tofilter$charttime_rr,rr_from_patients_tofilter$charttime.e
xtubation,units = 'mins')

# # Filtering begins
#
# post_ext_rr_patients<-rr_from_patients_tofilter %>%
#   filter( timefrom_ext_to_rr_mins >=0 )

#data availability epochs

#if there is no data in that epoch, write 0
cohort_with_rr_epoch<-rr_from_patients_tofilter %>%

```



```

mutate(
  rr_epoch_01=if_else(timefrom_ext_to_rr_mins>= 0 &
timefrom_ext_to_rr_mins<=60,valuenum,NULL)
,rr_epoch_02=if_else(timefrom_ext_to_rr_mins>60 &
timefrom_ext_to_rr_mins<=120,valuenum,NULL)
,rr_epoch_03=if_else(timefrom_ext_to_rr_mins>120 &
timefrom_ext_to_rr_mins<=180,valuenum,NULL)
,rr_epoch_04=if_else(timefrom_ext_to_rr_mins>180 &
timefrom_ext_to_rr_mins<=240,valuenum,NULL)
,rr_epoch_05=if_else(timefrom_ext_to_rr_mins>240 &
timefrom_ext_to_rr_mins<=300,valuenum,NULL)
,rr_epoch_06=if_else(timefrom_ext_to_rr_mins>300 &
timefrom_ext_to_rr_mins<=360,valuenum,NULL)
,rr_epoch_07=if_else(timefrom_ext_to_rr_mins>360 &
timefrom_ext_to_rr_mins<=420,valuenum,NULL)
,rr_epoch_08=if_else(timefrom_ext_to_rr_mins>420 &
timefrom_ext_to_rr_mins<=480,valuenum,NULL)
,rr_epoch_09=if_else(timefrom_ext_to_rr_mins>480 &
timefrom_ext_to_rr_mins<=540,valuenum,NULL)
,rr_epoch_10=if_else(timefrom_ext_to_rr_mins>540 &
timefrom_ext_to_rr_mins<=600,valuenum,NULL)
,rr_epoch_11=if_else(timefrom_ext_to_rr_mins>600 &
timefrom_ext_to_rr_mins<=660,valuenum,NULL)
,rr_epoch_12=if_else(timefrom_ext_to_rr_mins>660 &
timefrom_ext_to_rr_mins<=720,valuenum,NULL)
)

view(dfSummary(cohort_with_rr_epoch))

my.max <- function(x) ifelse( !all(is.na(x)), max(x, na.rm=T), NA)

cohort_with_rr_epoch<-cohort_with_rr_epoch%%
select(subject_id,rr_epoch_01,rr_epoch_02,rr_epoch_03,rr_epoch_04,rr_epoch_05,rr_epoc
h_06,rr_epoch_07,rr_epoch_08,rr_epoch_09,rr_epoch_10,rr_epoch_11,rr_epoch_12)%>%
  group_by(subject_id)%>%
  dplyr::summarize_all(my.max)

view(dfSummary(cohort_with_rr_epoch))

#0 means missing so replace 0 with NA
cohort_with_rr_epoch[cohort_with_rr_epoch==0]<-NA

#view(dfSummary(cohort_with_rr_epoch))

# Filtering begins

post_ext_rr_patients<-rr_from_patients_tofilter %>%
  filter(timefrom_ext_to_rr_mins > 0 )

n_distinct(post_ext_rr_patients$subject_id)
#771 patients have rr data after ext

tachypnea_patients<-post_ext_rr_patients%%
  filter(valuenum >= 30 & valuenum< 60)

```

```
n_distinct(tachypnea_patients$subject_id)
#248 patients have tachypnea
```

```
cohort_with_te_epoch<-rr_from_patients_tofilter %>%
mutate(
  te_epoch_01=case_when(timefrom_ext_to_rr_mins>= 0 & timefrom_ext_to_rr_mins<=60 &
valuenum >= 30 & valuenum <60 ~ 1
,timefrom_ext_to_rr_mins>= 0 & timefrom_ext_to_rr_mins<=60 ~ 0
)
,te_epoch_02=case_when(timefrom_ext_to_rr_mins>60 & timefrom_ext_to_rr_mins<=120 &
valuenum >= 30 & valuenum <60 ~ 1
,timefrom_ext_to_rr_mins>60 & timefrom_ext_to_rr_mins<=120 ~ 0
)
,te_epoch_03=case_when(timefrom_ext_to_rr_mins>120 & timefrom_ext_to_rr_mins<=180 &
valuenum >= 30 & valuenum <60 ~ 1
,timefrom_ext_to_rr_mins>120 & timefrom_ext_to_rr_mins<=180 ~ 0
)
,te_epoch_04=case_when(timefrom_ext_to_rr_mins>180 & timefrom_ext_to_rr_mins<=240 &
valuenum >= 30 & valuenum <60 ~ 1
,timefrom_ext_to_rr_mins>180 & timefrom_ext_to_rr_mins<=240 ~ 0
)
,te_epoch_05=case_when(timefrom_ext_to_rr_mins>240 & timefrom_ext_to_rr_mins<=300 &
valuenum >= 30 & valuenum <60 ~ 1
,timefrom_ext_to_rr_mins>240 & timefrom_ext_to_rr_mins<=300 ~ 0
)
,te_epoch_06=case_when(timefrom_ext_to_rr_mins>300 & timefrom_ext_to_rr_mins<=360 &
valuenum >= 30 & valuenum <60 ~ 1
,timefrom_ext_to_rr_mins>300 & timefrom_ext_to_rr_mins<=360 ~ 0
)
,te_epoch_07=case_when(timefrom_ext_to_rr_mins>360 & timefrom_ext_to_rr_mins<=420 &
valuenum >= 30 & valuenum <60 ~ 1
,timefrom_ext_to_rr_mins>360 & timefrom_ext_to_rr_mins<=420 ~ 0
)
,te_epoch_08=case_when(timefrom_ext_to_rr_mins>420 & timefrom_ext_to_rr_mins<=480 &
valuenum >= 30 & valuenum <60 ~ 1
,timefrom_ext_to_rr_mins>420 & timefrom_ext_to_rr_mins<=480 ~ 0
)
,te_epoch_09=case_when(timefrom_ext_to_rr_mins>480 & timefrom_ext_to_rr_mins<=540 &
valuenum >= 30 & valuenum <60 ~ 1
,timefrom_ext_to_rr_mins>480 & timefrom_ext_to_rr_mins<=540 ~ 0
)
,te_epoch_10=case_when(timefrom_ext_to_rr_mins>540 & timefrom_ext_to_rr_mins<=600 &
valuenum >= 30 & valuenum <60 ~ 1
,timefrom_ext_to_rr_mins>540 & timefrom_ext_to_rr_mins<=600 ~ 0
)
,te_epoch_11=case_when(timefrom_ext_to_rr_mins>600 & timefrom_ext_to_rr_mins<=660 &
valuenum >= 30 & valuenum <60 ~ 1
,timefrom_ext_to_rr_mins>600 & timefrom_ext_to_rr_mins<=660 ~ 0
)
,te_epoch_12=case_when(timefrom_ext_to_rr_mins>660 & timefrom_ext_to_rr_mins<=720 &
valuenum >= 30 & valuenum <60 ~ 1
,timefrom_ext_to_rr_mins>660 & timefrom_ext_to_rr_mins<=720 ~ 0
)
)
```

```

#view(dfSummary(cohort_with_te_epoch))

#this function returns NA only when all elements summed are NA
suma = function(x) if (all(is.na(x))) x[NA_integer_] else sum(x, na.rm = TRUE)

cohort_with_te_epoch<-cohort_with_te_epoch%>%
select(subject_id,te_epoch_01,te_epoch_02,te_epoch_03,te_epoch_04,te_epoch_05,te_epoc
h_06,te_epoch_07,te_epoch_08,te_epoch_09,te_epoch_10,te_epoch_11,te_epoch_12)%>%
group_by(subject_id)%>%
summarize_all(suma)

#view(dfSummary(cohort_with_te_epoch))

# Creating binary columns
subject_ids_fromourcohort24<-
as.data.frame(cmo_and_extubation_and_death_24$subject_id)
colnames(subject_ids_fromourcohort24)<- 'subject_id'

final_rr_te_epoch<-left_join(subject_ids_fromourcohort24, cohort_with_rr_epoch)
final_rr_te_epoch<-left_join(final_rr_te_epoch,cohort_with_te_epoch)

final_rr_te_epoch<-final_rr_te_epoch%>%
  mutate(has_rr_data=if_else(subject_id %in%
respiratory_rate_charttime_cohort_24$subject_id,1,0))%>%
  mutate(has_postext_rr_data=if_else(subject_id %in%
post_ext_rr_patients$subject_id,1,0))%>%
  mutate(has_postext_tachypnea=if_else(subject_id %in%
tachypnea_patients$subject_id,1,0))

#Create another dataset for data availability per epoch (all rr, not only te)
view(dfSummary(final_rr_te_epoch))

...

### Freq distribution of granularity

```{r}
#we order them before applying the lag
tachypnea_patients<-tachypnea_patients%>%arrange(subject_id,charttime_rr)

#we use the lag for calulating the resolution
tachypnea_patients<-tachypnea_patients%>% group_by(subject_id) %>%
mutate(lagged_charttime_rr = lag(charttime_rr))
tachypnea_patients['charttime_rr_resolution_min']<-(tachypnea_patients$charttime_rr-
tachypnea_patients$lagged_charttime_rr)/60

...

### First postext tachypnea episode time

```

```

"has post extubation tachypnea episode" with variable to contain the time to event
value.
(only the first one counts)

```{r}
rr_from_patients_tofilter_first<-rr_from_patients_tofilter%>%filter(
  timefrom_ext_to_rr_mins >=0 & valuenum >=30
)

rr_from_patients_tofilter_first<-rr_from_patients_tofilter_first
%>%arrange(subject_id,timefrom_ext_to_rr_mins)

rr_from_patients_tofilter_first_final<-
rr_from_patients_tofilter_first[!duplicated(rr_from_patients_tofilter_first$subject_i
d),]

rr_from_patients_tofilter_first_final<-rr_from_patients_tofilter_first_final%>%
  select(subject_id,charttime_rr,timefrom_ext_to_rr_mins)
rr_from_patients_tofilter_first_final<-
dplyr::rename(rr_from_patients_tofilter_first_final,
              charttime_first_te=charttime_rr,

timefrom_ext_to_first_te_mins=timefrom_ext_to_rr_mins)
```

DRUGS POSITIVE EPOCHS

Drugs Epoch Values Positive

* 225154 Morphine Sulfate
* 221744 Fentanyl
* 221668 Midazolam
* 221385 Ativan (Lorazepam)

opioids = 1 milligram of morphine(ref) IV (225154) is equivalent to .01mg Fentanyl
IV (221744).
benzodiazepines = 1 milligram of lorazepam (ativan)(ref) IV is equivalent to 2 mg
midazolam IV.
Ref= Barr J, Zomorodi K, Bertaccini EJ, et al. A double-blind, randomized comparison
of i.v. lorazepam versus midazolam for sedation of ICU patients via a pharmacologic
model. Anesthesiology. 2001 Aug;95(2):286-98. PMID 11506097

Infusion Drugs

Opioids

Data extraction

```{r eval=FALSE, include=FALSE}
query<-"
SELECT
  inputevents_mv.subject_id,

```

```

    inputevents_mv.hadm_id,
    inputevents_mv.icustay_id,
    inputevents_mv.itemid,
    COALESCE(inputevents_mv.storetime,inputevents_mv.starttime) AS
charttime_drugs_infusion,
    inputevents_mv.rate,
    inputevents_mv.rateuom,
    inputevents_mv.ordercategorydescription
From
  `physonet-data.mimiciii_clinical.inputevents_mv` inputevents_mv

WHERE
  inputevents_mv.itemid In (225154,221744)
  AND ordercategorydescription = 'Continuous Med'
"

# AND subject_id IN (" , subject_ids_fromourcohort24, "

opioids_infusion_cohort_24<-query_exec(query, project = project_HST,use_legacy_sql =
F,max_pages = Inf)

opioids_infusion_cohort_24$itemid[opioids_infusion_cohort_24$itemid==225154]<-
'Morphine'
opioids_infusion_cohort_24$itemid[opioids_infusion_cohort_24$itemid==221744]<-
'Fentanyl'
# in case we don't have enough add DILAUDID (hydromorphone)
# we tried hydromorphone and it increased in less than 10 the number of patients
1/7/18
```

Units equivalence

opioids = 1 milligram of morphine(ref) IV (225154) is equivalent to .01mg Fentanyl
IV (221744).
1 mcg of morphine = 0.001 mg of morphine
1 mcg of fentanyl = 0.1 mg of morphine
1 mg of fentanyl = 100 mg of morphine

```{r}
#nested if else function

i <- function(if_stat, then) {
  if_stat <- lazyeval::expr_text(if_stat)
  then <- lazyeval::expr_text(then)
  sprintf("ifelse(%s, %s, ", if_stat, then)
}

e <- function(else_ret) {
  else_ret <- lazyeval::expr_text(else_ret)
  else_ret
}

ie <- function(...) {
  args <- list(...)

```

```

for (i in 1:(length(args) - 1) ) {
  if (substr(args[[i]], 1, 6) != "ifelse") {
    stop("All but the last argument, need to be i functions.", call. = FALSE)
  }
}
if (substr(args[[length(args)]], 1, 6) == "ifelse"){
  stop("Last argument needs to be an e function.", call. = FALSE)
}
args$final <- paste(rep(')', length(args) - 1), collapse = '')
eval_string <- do.call('paste', args)
eval(parse(text = eval_string))
}

opioids_infusion_cohort_24['opioid_rate']<-NA
# * 225154 Morphine Sulfate
# * 221744 Fentanyl
opioids_infusion_cohort_24$opioid_rate <-
  ie(
    # 1 mcg of morphine = 0.001 mg of morphine
    i(opioids_infusion_cohort_24$itemid=='Morphine' &
opioids_infusion_cohort_24$rateuom=='mcg/hour',round(opioids_infusion_cohort_24$rate*
0.001,2)),
    #1 mcg of fentanyl = 0.1 mg of morphine
    i(opioids_infusion_cohort_24$itemid=='Fentanyl' &
opioids_infusion_cohort_24$rateuom=='mcg/hour',round(opioids_infusion_cohort_24$rate*
0.1,2)),
    #1 mg of fentanyl = 100 mg of morphine
    i(opioids_infusion_cohort_24$itemid=='Fentanyl' &
opioids_infusion_cohort_24$rateuom=='mg/hour',round(opioids_infusion_cohort_24$rate*1
00,2)),
    e(round(opioids_infusion_cohort_24$rate,2))
  )
...

##### Epochs Creation (max rate mg/h)

```{r eval=FALSE, include=FALSE}
library(dplyr)
opioids_infusion_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex
tubation','deathtime')], opioids_infusion_cohort_24)

# we noticed some timezones are alteterd, so we reset them
opioids_infusion_from_patients_tofilter$charttime_drugs_infusion<-
as.character(opioids_infusion_from_patients_tofilter$charttime_drugs_infusion)
opioids_infusion_from_patients_tofilter$charttime_extubation<-
as.character(opioids_infusion_from_patients_tofilter$charttime_extubation)
opioids_infusion_from_patients_tofilter$deathtime<-
as.character(opioids_infusion_from_patients_tofilter$deathtime)

```

```

opioids_infusion_from_patients_tofilter$charttime_drugs_infusion<-
as.POSIXct(opioids_infusion_from_patients_tofilter$charttime_drugs_infusion,
format='%Y-%m-%d %H:%M:%S', tz="GMT")
opioids_infusion_from_patients_tofilter$charttime_extubation<-
as.POSIXct(opioids_infusion_from_patients_tofilter$charttime_extubation, format='%Y-
%m-%d %H:%M:%S', tz="GMT")
opioids_infusion_from_patients_tofilter$deathtime<-
as.POSIXct(opioids_infusion_from_patients_tofilter$deathtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

opioids_infusion_from_patients_tofilter['time_from_ext_to_opioids_infusion_min']<-
difftime(opioids_infusion_from_patients_tofilter$charttime_drugs_infusion,opioids_inf
usion_from_patients_tofilter$charttime_extubation,units = 'mins')

#Drug Push is Bolus, so I filtered them out
opioids_infusion_after_ext<-opioids_infusion_from_patients_tofilter %>%
  filter(
    time_from_ext_to_opioids_infusion_min >=0
  )

# opioids = 1 milligram of morphine IV (225154) is equivalent to .01 mg Fentanyl IV
(221744).

n_distinct(opioids_infusion_after_ext$subject_id)
# 545 patients with time_from_ext_to_opioids_infusion_min >=-12*60 1/7/18
# 451 patients with time_from_ext_to_opioids_infusion_min >=0 2/13/19

library(dplyr)
## if(any(gdrgugs_infusionpl("package:plyr", search())) detach("package:plyr") else
message("plyr not loaded")
cohort_with_opioids_infusion_epoch<-opioids_infusion_after_ext %>%
  mutate(
    opioids_infusion_epoch_01=if_else(time_from_ext_to_opioids_infusion_min>=0 &
time_from_ext_to_opioids_infusion_min<=60,opioid_rate,NULL)
,opioids_infusion_epoch_02=if_else(time_from_ext_to_opioids_infusion_min>60 &
time_from_ext_to_opioids_infusion_min<=120,opioid_rate,NULL)
,opioids_infusion_epoch_03=if_else(time_from_ext_to_opioids_infusion_min>120 &
time_from_ext_to_opioids_infusion_min<=180,opioid_rate,NULL)
,opioids_infusion_epoch_04=if_else(time_from_ext_to_opioids_infusion_min>180 &
time_from_ext_to_opioids_infusion_min<=240,opioid_rate,NULL)
,opioids_infusion_epoch_05=if_else(time_from_ext_to_opioids_infusion_min>240 &
time_from_ext_to_opioids_infusion_min<=300,opioid_rate,NULL)
,opioids_infusion_epoch_06=if_else(time_from_ext_to_opioids_infusion_min>300 &
time_from_ext_to_opioids_infusion_min<=360,opioid_rate,NULL)
,opioids_infusion_epoch_07=if_else(time_from_ext_to_opioids_infusion_min>360 &
time_from_ext_to_opioids_infusion_min<=420,opioid_rate,NULL)
,opioids_infusion_epoch_08=if_else(time_from_ext_to_opioids_infusion_min>420 &
time_from_ext_to_opioids_infusion_min<=480,opioid_rate,NULL)
,opioids_infusion_epoch_09=if_else(time_from_ext_to_opioids_infusion_min>480 &
time_from_ext_to_opioids_infusion_min<=540,opioid_rate,NULL)
,opioids_infusion_epoch_10=if_else(time_from_ext_to_opioids_infusion_min>540 &
time_from_ext_to_opioids_infusion_min<=600,opioid_rate,NULL)
,opioids_infusion_epoch_11=if_else(time_from_ext_to_opioids_infusion_min>600 &
time_from_ext_to_opioids_infusion_min<=660,opioid_rate,NULL)

```

```
,opioids_infusion_epoch_12=if_else(time_from_ext_to_opioids_infusion_min>660 &
time_from_ext_to_opioids_infusion_min<=720,opioid_rate,NULL)
)
```

```
cohort_with_opioids_infusion_epoch<-cohort_with_opioids_infusion_epoch%>%
```

```
select(subject_id,opioids_infusion_epoch_01,opioids_infusion_epoch_02,opioids_infusion_epoch_03,opioids_infusion_epoch_04,opioids_infusion_epoch_05,opioids_infusion_epoch_06,opioids_infusion_epoch_07,opioids_infusion_epoch_08,opioids_infusion_epoch_09,opioids_infusion_epoch_10,opioids_infusion_epoch_11,opioids_infusion_epoch_12)%>%
  group_by(subject_id)%>%
  #summarize(mean(value,na.rm=TRUE))
  summarise_all(my.max)
...
```

```
#### Benzodiazepines
```

```
##### Data extraction
```

```
`r eval=FALSE, include=FALSE}
query<-"
SELECT
  inpatientevents_mv.subject_id,
  inpatientevents_mv.hadm_id,
  inpatientevents_mv.icustay_id,
  inpatientevents_mv.itemid,
  COALESCE(inpatientevents_mv.storetime,inpatientevents_mv.starttime) AS
charttime_drugs_infusion,
  inpatientevents_mv.rate,
  inpatientevents_mv.rateuom,
  inpatientevents_mv.ordercategorydescription
From
  `physionet-data.mimiciii_clinical.inpatientevents_mv` inpatientevents_mv

WHERE
  inpatientevents_mv.itemid In (221668,221385,221623)
  AND ordercategorydescription = 'Continuous Med'
"
```

```
benzodiazepines_infusion_cohort_24<-query_exec(query, project =
project_HST,use_legacy_sql = F,max_pages = Inf)
```

```
benzodiazepines_infusion_cohort_24$itemid[benzodiazepines_infusion_cohort_24$itemid==
221668]<-'Midazolam'
benzodiazepines_infusion_cohort_24$itemid[benzodiazepines_infusion_cohort_24$itemid==
221385]<-'Ativan'
#Valium (Diazepam)
#We tried valium and it didn't increase the number of patients, still 84 after ext
...
```

```
##### Units equivalence
```

```
benzodiazepines = 1 milligram of lorazepam (ativan)(ref) IV is equivalent to 2 mg
midazolam IV.
```


1 mg of midazolam = 0.5 mg of Ativan

```
```{r}
#nested if else function

i <- function(if_stat, then) {
  if_stat <- lazyeval::expr_text(if_stat)
  then <- lazyeval::expr_text(then)
  sprintf("ifelse(%s, %s, ", if_stat, then)
}

e <- function(else_ret) {
  else_ret <- lazyeval::expr_text(else_ret)
  else_ret
}

ie <- function(...) {
  args <- list(...)

  for (i in 1:(length(args) - 1) ) {
    if (substr(args[[i]], 1, 6) != "ifelse") {
      stop("All but the last argument, need to be i functions.", call. = FALSE)
    }
  }
  if (substr(args[[length(args)]], 1, 6) == "ifelse"){
    stop("Last argument needs to be an e function.", call. = FALSE)
  }
  args$final <- paste(rep(''), length(args) - 1), collapse = '')
  eval_string <- do.call('paste', args)
  eval(parse(text = eval_string))
}

benzodiazepines_infusion_cohort_24['opioid_rate']<-NA
benzodiazepines_infusion_cohort_24$opioid_rate <-
  ie(
#1 mg of midazolam = 0.5 mg of Ativan

i(benzodiazepines_infusion_cohort_24$itemid=='Midazolam',round(benzodiazepines_infusi
on_cohort_24$rate*0.5,2)),
  e(round(benzodiazepines_infusion_cohort_24$rate,2))
)
```

Epochs Creation (max rate mg/h)

```{r eval=FALSE, include=FALSE}
benzodiazepines_infusion_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex
tubation','deathtime')], benzodiazepines_infusion_cohort_24)

# we noticed some timezones adrug_infusion alteterd, so we drugs_infusionset them
```

```

benzodiazepines_infusion_from_patients_tofilter$charttime_drugs_infusion<-
as.character(benzodiazepines_infusion_from_patients_tofilter$charttime_drugs_infusion
)
benzodiazepines_infusion_from_patients_tofilter$charttime_extubation<-
as.character(benzodiazepines_infusion_from_patients_tofilter$charttime_extubation)
benzodiazepines_infusion_from_patients_tofilter$deathtime<-
as.character(benzodiazepines_infusion_from_patients_tofilter$deathtime)

```

```

benzodiazepines_infusion_from_patients_tofilter$charttime_drugs_infusion<-
as.POSIXct(benzodiazepines_infusion_from_patients_tofilter$charttime_drugs_infusion,
format='%Y-%m-%d %H:%M:%S', tz="GMT")
benzodiazepines_infusion_from_patients_tofilter$charttime_extubation<-
as.POSIXct(benzodiazepines_infusion_from_patients_tofilter$charttime_extubation,
format='%Y-%m-%d %H:%M:%S', tz="GMT")
benzodiazepines_infusion_from_patients_tofilter$deathtime<-
as.POSIXct(benzodiazepines_infusion_from_patients_tofilter$deathtime, format='%Y-%m-
%d %H:%M:%S', tz="GMT")

```

```

benzodiazepines_infusion_from_patients_tofilter['time_from_ext_to_benzodiazepines_inf
usion_min']<-
difftime(benzodiazepines_infusion_from_patients_tofilter$charttime_drugs_infusion,ben
zodiazepines_infusion_from_patients_tofilter$charttime_extubation,units = 'mins')

```

```

#Drug Push is Bolus, so I filtered them out
benzodiazepines_infusion_after_ext<-benzodiazepines_infusion_from_patients_tofilter
%>%
  filter(
    time_from_ext_to_benzodiazepines_infusion_min >=0
  )

```

```

# benzodiazepines = 1 milligram of morphine IV (225154) is equivalent to .01 mg
Fentanyl IV (221744).

```

```

n_distinct(benzodiazepines_infusion_after_ext$subject_id) #84 patients 1/7/18

```

```

library(dplyr)
## if(any(gdrugs_infusionpl("package:plyr", search())) detach("package:plyr") else
message("plyr not loaded")
cohort_with_benzodiazepines_infusion_epoch<-benzodiazepines_infusion_after_ext %>%
  mutate(

```

```

benzodiazepines_infusion_epoch_01=if_else(time_from_ext_to_benzodiazepines_infusion_m
in>=0 & time_from_ext_to_benzodiazepines_infusion_min<=60,opioid_rate,NULL)
,benzodiazepines_infusion_epoch_02=if_else(time_from_ext_to_benzodiazepines_infusion_
min>60 & time_from_ext_to_benzodiazepines_infusion_min<=120,opioid_rate,NULL)
,benzodiazepines_infusion_epoch_03=if_else(time_from_ext_to_benzodiazepines_infusion_
min>120 & time_from_ext_to_benzodiazepines_infusion_min<=180,opioid_rate,NULL)
,benzodiazepines_infusion_epoch_04=if_else(time_from_ext_to_benzodiazepines_infusion_
min>180 & time_from_ext_to_benzodiazepines_infusion_min<=240,opioid_rate,NULL)
,benzodiazepines_infusion_epoch_05=if_else(time_from_ext_to_benzodiazepines_infusion_
min>240 & time_from_ext_to_benzodiazepines_infusion_min<=300,opioid_rate,NULL)
,benzodiazepines_infusion_epoch_06=if_else(time_from_ext_to_benzodiazepines_infusion_
min>300 & time_from_ext_to_benzodiazepines_infusion_min<=360,opioid_rate,NULL)

```

```
,benzodiazepines_infusion_epoch_07=if_else(time_from_ext_to_benzodiazepines_infusion_
min>360 & time_from_ext_to_benzodiazepines_infusion_min<=420,opioid_rate,NULL)
,benzodiazepines_infusion_epoch_08=if_else(time_from_ext_to_benzodiazepines_infusion_
min>420 & time_from_ext_to_benzodiazepines_infusion_min<=480,opioid_rate,NULL)
,benzodiazepines_infusion_epoch_09=if_else(time_from_ext_to_benzodiazepines_infusion_
min>480 & time_from_ext_to_benzodiazepines_infusion_min<=540,opioid_rate,NULL)
,benzodiazepines_infusion_epoch_10=if_else(time_from_ext_to_benzodiazepines_infusion_
min>540 & time_from_ext_to_benzodiazepines_infusion_min<=600,opioid_rate,NULL)
,benzodiazepines_infusion_epoch_11=if_else(time_from_ext_to_benzodiazepines_infusion_
min>600 & time_from_ext_to_benzodiazepines_infusion_min<=660,opioid_rate,NULL)
,benzodiazepines_infusion_epoch_12=if_else(time_from_ext_to_benzodiazepines_infusion_
min>660 & time_from_ext_to_benzodiazepines_infusion_min<=720,opioid_rate,NULL)
)
```

```
cohort_with_benzodiazepines_infusion_epoch<-
cohort_with_benzodiazepines_infusion_epoch%>%
```

```
select(subject_id,benzodiazepines_infusion_epoch_01,benzodiazepines_infusion_epoch_02
,benzodiazepines_infusion_epoch_03,benzodiazepines_infusion_epoch_04,benzodiazepines_
infusion_epoch_05,benzodiazepines_infusion_epoch_06,benzodiazepines_infusion_epoch_07
,benzodiazepines_infusion_epoch_08,benzodiazepines_infusion_epoch_09,benzodiazepines_
infusion_epoch_10,benzodiazepines_infusion_epoch_11,benzodiazepines_infusion_epoch_12
)%>%
  group_by(subject_id)%>%
  #summarize(mean(value,na.rm=TRUE))
  summarise_all(my.max)
``
```

```
### Bolus Drugs
```

```
#### Opioids
```

```
##### Data extraction
```

```
```{r eval=FALSE, include=FALSE}
query<-"
SELECT
  inputevents_mv.subject_id,
  inputevents_mv.hadm_id,
  inputevents_mv.icustay_id,
  inputevents_mv.itemid,
  COALESCE(inputevents_mv.storetime,inputevents_mv.starttime) AS
charttime_drugs_bolus,
  inputevents_mv.amount,
  inputevents_mv.amountuom,
  inputevents_mv.ordercategorydescription
From
  `physionet-data.mimiciii_clinical.inputevents_mv` inputevents_mv

WHERE
  inputevents_mv.itemid In (225154,221744)
  AND ordercategorydescription = 'Drug Push' "
```

```

opioids_bolus_cohort_24<-query_exec(query, project = project_HST,use_legacy_sql =
F,max_pages = Inf)

opioids_bolus_cohort_24$itemid[opioids_bolus_cohort_24$itemid==225154]<-'Morphine'
opioids_bolus_cohort_24$itemid[opioids_bolus_cohort_24$itemid==221744]<-'Fentanyl'
```

Units equivalence

opioids = 1 milligram of morphine(ref) IV (225154) is equivalent to .01mg Fentanyl
IV (221744).
1 mcg of morphine = 0.001 mg of morphine
1 mcg of fentanyl = 0.1 mg of morphine
1 mg of fentanyl = 100 mg of morphine

```{r}
#nested if else function

i <- function(if_stat, then) {
  if_stat <- lazyeval::expr_text(if_stat)
  then <- lazyeval::expr_text(then)
  sprintf("ifelse(%s, %s, ", if_stat, then)
}

e <- function(else_ret) {
  else_ret <- lazyeval::expr_text(else_ret)
  else_ret
}

ie <- function(...) {
  args <- list(...)

  for (i in 1:(length(args) - 1) ) {
    if (substr(args[[i]], 1, 6) != "ifelse") {
      stop("All but the last argument, need to be i functions.", call. = FALSE)
    }
  }
  if (substr(args[[length(args)]], 1, 6) == "ifelse"){
    stop("Last argument needs to be an e function.", call. = FALSE)
  }
  args$final <- paste(rep(''), length(args) - 1), collapse = '')
  eval_string <- do.call('paste', args)
  eval(parse(text = eval_string))
}

opioids_bolus_cohort_24['opioid_amount']<-NA
# * 225154 Morphine Sulfate
# * 221744 Fentanyl
opioids_bolus_cohort_24$opioid_amount <-
  ie(
    # 1 mcg of morphine = 0.001 mg of morphine
    i(opioids_bolus_cohort_24$itemid=='Morphine' &
opioids_bolus_cohort_24$amountuom=='mcg',round(opioids_bolus_cohort_24$amount*0.001,2
)),
    #1 mcg of fentanyl = 0.1 mg of morphine

```

```

i(opioids_bolus_cohort_24$itemid=='Fentanyl' &
opioids_bolus_cohort_24$amountuom=='mcg',round(opioids_bolus_cohort_24$amount*0.1,2))
',
  #1 mg of fentanyl = 100 mg of morphine
  i(opioids_bolus_cohort_24$itemid=='Fentanyl' &
opioids_bolus_cohort_24$amountuom=='mg',round(opioids_bolus_cohort_24$amount*100,2)),
  e(round(opioids_bolus_cohort_24$amount,2))
...

```

Epochs Creation (total amount mg)

```

```{r eval=FALSE, include=FALSE}
library(dplyr)
opioids_bolus_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex
tubation','deathtime')], opioids_bolus_cohort_24)

```

```

# we noticed some timezones adrugs_bolus alteterd, so we drugs_bolusset them
opioids_bolus_from_patients_tofilter$charttime_drugs_bolus<-
as.character(opioids_bolus_from_patients_tofilter$charttime_drugs_bolus)
opioids_bolus_from_patients_tofilter$charttime_extubation<-
as.character(opioids_bolus_from_patients_tofilter$charttime_extubation)
opioids_bolus_from_patients_tofilter$deathtime<-
as.character(opioids_bolus_from_patients_tofilter$deathtime)

```

```

opioids_bolus_from_patients_tofilter$charttime_drugs_bolus<-
as.POSIXct(opioids_bolus_from_patients_tofilter$charttime_drugs_bolus, format='%Y-%m-
%d %H:%M:%S', tz="GMT")
opioids_bolus_from_patients_tofilter$charttime_extubation<-
as.POSIXct(opioids_bolus_from_patients_tofilter$charttime_extubation, format='%Y-%m-
%d %H:%M:%S', tz="GMT")
opioids_bolus_from_patients_tofilter$deathtime<-
as.POSIXct(opioids_bolus_from_patients_tofilter$deathtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

```

```

opioids_bolus_from_patients_tofilter['time_from_ext_to_opioids_bolus_min']<-
difftime(opioids_bolus_from_patients_tofilter$charttime_drugs_bolus,opioids_bolus_fro
m_patients_tofilter$charttime_extubation,units = 'mins')

```

```

#Drug Push is Bolus, so I filtered them out
opioids_bolus_after_ext<-opioids_bolus_from_patients_tofilter %>%
  filter(
    time_from_ext_to_opioids_bolus_min >=0
  )

```

```

# opioids = 1 milligram of morphine IV (225154) is equivalent to .01 mg Fentanyl IV
(221744).

```

```

n_distinct(opioids_bolus_after_ext$subject_id)
# 355 patients with time_from_ext_to_opioids_bolus_min >=-2*60 1/7/18

```

```
# 322 patients with time_from_ext_to_opioids_bolus_min >=0 2/12/19
```

```
library(dplyr)
## if(any(gdrugs_boluspl("package:plyr", search())) detach("package:plyr") else
message("plyr not loaded")
cohort_with_opioids_bolus_epoch<-opioids_bolus_after_ext %>%
  mutate(
    opioids_bolus_epoch_01=if_else(time_from_ext_to_opioids_bolus_min>=0 &
time_from_ext_to_opioids_bolus_min<=60,opioid_amount,NULL)
,opioids_bolus_epoch_02=if_else(time_from_ext_to_opioids_bolus_min>60 &
time_from_ext_to_opioids_bolus_min<=120,opioid_amount,NULL)
,opioids_bolus_epoch_03=if_else(time_from_ext_to_opioids_bolus_min>120 &
time_from_ext_to_opioids_bolus_min<=180,opioid_amount,NULL)
,opioids_bolus_epoch_04=if_else(time_from_ext_to_opioids_bolus_min>180 &
time_from_ext_to_opioids_bolus_min<=240,opioid_amount,NULL)
,opioids_bolus_epoch_05=if_else(time_from_ext_to_opioids_bolus_min>240 &
time_from_ext_to_opioids_bolus_min<=300,opioid_amount,NULL)
,opioids_bolus_epoch_06=if_else(time_from_ext_to_opioids_bolus_min>300 &
time_from_ext_to_opioids_bolus_min<=360,opioid_amount,NULL)
,opioids_bolus_epoch_07=if_else(time_from_ext_to_opioids_bolus_min>360 &
time_from_ext_to_opioids_bolus_min<=420,opioid_amount,NULL)
,opioids_bolus_epoch_08=if_else(time_from_ext_to_opioids_bolus_min>420 &
time_from_ext_to_opioids_bolus_min<=480,opioid_amount,NULL)
,opioids_bolus_epoch_09=if_else(time_from_ext_to_opioids_bolus_min>480 &
time_from_ext_to_opioids_bolus_min<=540,opioid_amount,NULL)
,opioids_bolus_epoch_10=if_else(time_from_ext_to_opioids_bolus_min>540 &
time_from_ext_to_opioids_bolus_min<=600,opioid_amount,NULL)
,opioids_bolus_epoch_11=if_else(time_from_ext_to_opioids_bolus_min>600 &
time_from_ext_to_opioids_bolus_min<=660,opioid_amount,NULL)
,opioids_bolus_epoch_12=if_else(time_from_ext_to_opioids_bolus_min>660 &
time_from_ext_to_opioids_bolus_min<=720,opioid_amount,NULL)
  )

my.sum <- function(x) ifelse( !all(is.na(x)), sum(x, na.rm=T), NA)

cohort_with_opioids_bolus_epoch<-cohort_with_opioids_bolus_epoch%>%

select(subject_id,opioids_bolus_epoch_01,opioids_bolus_epoch_02,opioids_bolus_epoch_0
3,opioids_bolus_epoch_04,opioids_bolus_epoch_05,opioids_bolus_epoch_06,opioids_bolus_
epoch_07,opioids_bolus_epoch_08,opioids_bolus_epoch_09,opioids_bolus_epoch_10,opioids
_bolus_epoch_11,opioids_bolus_epoch_12)%>%
  group_by(subject_id)%>%
  #summarize(mean(value,na.rm=TRUE))
  summarise_all(my.sum)
...

#### Benzodiazepines

##### Data extraction

```{r eval=FALSE, include=FALSE}
query<-"
SELECT
  inptevents_mv.subject_id,
```

```

    inputevents_mv.hadm_id,
    inputevents_mv.icustay_id,
    inputevents_mv.itemid,
    COALESCE(inputevents_mv.storetime,inputevents_mv.starttime) AS
charttime_drugs_bolus,
    inputevents_mv.amount,
    inputevents_mv.amountuom,
    inputevents_mv.ordercategorydescription
From
`physionet-data.mimiciii_clinical.inputevents_mv` inputevents_mv

WHERE
    inputevents_mv.itemid In (221668,221385)
    AND ordercategorydescription = 'Drug Push'"

benzodiazepines_bolus_cohort_24<-query_exec(query, project =
project_HST,use_legacy_sql = F,max_pages = Inf)

benzodiazepines_bolus_cohort_24$itemid[benzodiazepines_bolus_cohort_24$itemid==221668
]<-'Midazolam'
benzodiazepines_bolus_cohort_24$itemid[benzodiazepines_bolus_cohort_24$itemid==221385
]<-'Ativan'

...

##### Units equivalence

benzodiazepines = 1 milligram of lorazepam (ativan)(ref) IV is equivalent to 2 mg
midazolam IV.
1 mg of midazolam = 0.5 mg of Ativan

```{r}
#nested if else function

i <- function(if_stat, then) {
  if_stat <- lazyeval::expr_text(if_stat)
  then <- lazyeval::expr_text(then)
  sprintf("ifelse(%s, %s, ", if_stat, then)
}

e <- function(else_ret) {
  else_ret <- lazyeval::expr_text(else_ret)
  else_ret
}

ie <- function(...) {
  args <- list(...)

  for (i in 1:(length(args) - 1) ) {
    if (substr(args[[i]], 1, 6) != "ifelse") {
      stop("All but the last argument, need to be i functions.", call. = FALSE)
    }
  }
  if (substr(args[[length(args)]], 1, 6) == "ifelse"){

```

```

    stop("Last argument needs to be an e function.", call. = FALSE)
  }
  args$final <- paste(rep(''), length(args) - 1), collapse = '')
  eval_string <- do.call('paste', args)
  eval(parse(text = eval_string))
}

benzodiazepines_bolus_cohort_24['opioid_amount']<-NA
benzodiazepines_bolus_cohort_24$opioid_amount <-
  ie(
#1 mg of midazolam = 0.5 mg of Ativan

i(benzodiazepines_bolus_cohort_24$itemid=='Midazolam',round(benzodiazepines_bolus_coh
ort_24$amount*0.5,2)),
  e(round(benzodiazepines_bolus_cohort_24$amount,2))
  )
...

##### Epochs Creation (total amount mg/h)

```{r eval=FALSE, include=FALSE}
library(dplyr)
benzodiazepines_bolus_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex
tubation','deathtime')], benzodiazepines_bolus_cohort_24)

# we noticed some timezones adrug_bolus alteterd, so we drugs_bolusset them
benzodiazepines_bolus_from_patients_tofilter$charttime_drugs_bolus<-
as.character(benzodiazepines_bolus_from_patients_tofilter$charttime_drugs_bolus)
benzodiazepines_bolus_from_patients_tofilter$charttime_extubation<-
as.character(benzodiazepines_bolus_from_patients_tofilter$charttime_extubation)
benzodiazepines_bolus_from_patients_tofilter$deathtime<-
as.character(benzodiazepines_bolus_from_patients_tofilter$deathtime)

benzodiazepines_bolus_from_patients_tofilter$charttime_drugs_bolus<-
as.POSIXct(benzodiazepines_bolus_from_patients_tofilter$charttime_drugs_bolus,
format='%Y-%m-%d %H:%M:%S', tz="GMT")
benzodiazepines_bolus_from_patients_tofilter$charttime_extubation<-
as.POSIXct(benzodiazepines_bolus_from_patients_tofilter$charttime_extubation,
format='%Y-%m-%d %H:%M:%S', tz="GMT")
benzodiazepines_bolus_from_patients_tofilter$deathtime<-
as.POSIXct(benzodiazepines_bolus_from_patients_tofilter$deathtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

benzodiazepines_bolus_from_patients_tofilter['time_from_ext_to_benzodiazepines_bolus_
min']<-
difftime(benzodiazepines_bolus_from_patients_tofilter$charttime_drugs_bolus,benzodiaz
epines_bolus_from_patients_tofilter$charttime_extubation,units = 'mins')

#Drug Push is Bolus, so I filtered them out

```



```

benzodiazepines_bolus_after_ext<-benzodiazepines_bolus_from_patients_tofilter %>%
  filter(
    time_from_ext_to_benzodiazepines_bolus_min >=0
  )

# benzodiazepines = 1 milligram of morphine IV (225154) is equivalent to .01 mg
Fentanyl IV (221744).

n_distinct(benzodiazepines_bolus_after_ext$subject_id) #158 1/7/18

library(dplyr)
## if(any(gdrugs_boluspl("package:plyr", search())) detach("package:plyr") else
message("plyr not loaded")
cohort_with_benzodiazepines_bolus_epoch<-benzodiazepines_bolus_after_ext %>%
  mutate(
    benzodiazepines_bolus_epoch_01=if_else(time_from_ext_to_benzodiazepines_bolus_min>=0
& time_from_ext_to_benzodiazepines_bolus_min<=60,opioid_amount,NULL)
,benzodiazepines_bolus_epoch_02=if_else(time_from_ext_to_benzodiazepines_bolus_min>60
& time_from_ext_to_benzodiazepines_bolus_min<=120,opioid_amount,NULL)
,benzodiazepines_bolus_epoch_03=if_else(time_from_ext_to_benzodiazepines_bolus_min>12
0 & time_from_ext_to_benzodiazepines_bolus_min<=180,opioid_amount,NULL)
,benzodiazepines_bolus_epoch_04=if_else(time_from_ext_to_benzodiazepines_bolus_min>18
0 & time_from_ext_to_benzodiazepines_bolus_min<=240,opioid_amount,NULL)
,benzodiazepines_bolus_epoch_05=if_else(time_from_ext_to_benzodiazepines_bolus_min>24
0 & time_from_ext_to_benzodiazepines_bolus_min<=300,opioid_amount,NULL)
,benzodiazepines_bolus_epoch_06=if_else(time_from_ext_to_benzodiazepines_bolus_min>30
0 & time_from_ext_to_benzodiazepines_bolus_min<=360,opioid_amount,NULL)
,benzodiazepines_bolus_epoch_07=if_else(time_from_ext_to_benzodiazepines_bolus_min>36
0 & time_from_ext_to_benzodiazepines_bolus_min<=420,opioid_amount,NULL)
,benzodiazepines_bolus_epoch_08=if_else(time_from_ext_to_benzodiazepines_bolus_min>42
0 & time_from_ext_to_benzodiazepines_bolus_min<=480,opioid_amount,NULL)
,benzodiazepines_bolus_epoch_09=if_else(time_from_ext_to_benzodiazepines_bolus_min>48
0 & time_from_ext_to_benzodiazepines_bolus_min<=540,opioid_amount,NULL)
,benzodiazepines_bolus_epoch_10=if_else(time_from_ext_to_benzodiazepines_bolus_min>54
0 & time_from_ext_to_benzodiazepines_bolus_min<=600,opioid_amount,NULL)
,benzodiazepines_bolus_epoch_11=if_else(time_from_ext_to_benzodiazepines_bolus_min>60
0 & time_from_ext_to_benzodiazepines_bolus_min<=660,opioid_amount,NULL)
,benzodiazepines_bolus_epoch_12=if_else(time_from_ext_to_benzodiazepines_bolus_min>66
0 & time_from_ext_to_benzodiazepines_bolus_min<=720,opioid_amount,NULL)
  )

my.sum <- function(x) ifelse( !all(is.na(x)), sum(x, na.rm=T), NA)

cohort_with_benzodiazepines_bolus_epoch<-cohort_with_benzodiazepines_bolus_epoch%>%

select(subject_id,benzodiazepines_bolus_epoch_01,benzodiazepines_bolus_epoch_02,benzo
diazepines_bolus_epoch_03,benzodiazepines_bolus_epoch_04,benzodiazepines_bolus_epoch_
05,benzodiazepines_bolus_epoch_06,benzodiazepines_bolus_epoch_07,benzodiazepines_bolu
s_epoch_08,benzodiazepines_bolus_epoch_09,benzodiazepines_bolus_epoch_10,benzodiazepi
nes_bolus_epoch_11,benzodiazepines_bolus_epoch_12)%>%
  group_by(subject_id)%>%
  #summarize(mean(value,na.rm=TRUE))
  summarise_all(my.sum)
...

```

```

# DRUGS NEGATIVE EPOCHS

### Infusion Drugs Before Extubation

#### Opioids Before Extubation

##### Negative Epochs Creation (max rate mg/h)

```{r eval=FALSE, include=FALSE}

#Drug Push is Bolus, so I filtered them out
opiods_infusion_before_ext<-opiods_infusion_from_patients_tofilter %>%
  filter(
    time_from_ext_to_opiods_infusion_min >=-12*60
  )

# opiods = 1 milligram of morphine IV (225154) is equivalent to .01 mg Fentanyl IV
(221744).

n_distinct(opiods_infusion_before_ext$subject_id) #545 patients 1/7/18

library(dplyr)

cohort_with_opiods_infusion_neg_epoch<-opiods_infusion_before_ext %>%
  mutate(
    opiods_infusion_neg_epoch_01=if_else( between
(time_from_ext_to_opiods_infusion_min , -60, 0 ),opiod_rate,NULL)
,opiods_infusion_neg_epoch_02=if_else( between
(time_from_ext_to_opiods_infusion_min , -120, -60 ),opiod_rate,NULL)
,opiods_infusion_neg_epoch_03=if_else( between
(time_from_ext_to_opiods_infusion_min , -180, -120 ),opiod_rate,NULL)
,opiods_infusion_neg_epoch_04=if_else( between
(time_from_ext_to_opiods_infusion_min , -240, -180 ),opiod_rate,NULL)
,opiods_infusion_neg_epoch_05=if_else( between
(time_from_ext_to_opiods_infusion_min , -300, -240 ),opiod_rate,NULL)
,opiods_infusion_neg_epoch_06=if_else( between
(time_from_ext_to_opiods_infusion_min , -360, -300 ),opiod_rate,NULL)
,opiods_infusion_neg_epoch_07=if_else( between
(time_from_ext_to_opiods_infusion_min , -420, -360 ),opiod_rate,NULL)
,opiods_infusion_neg_epoch_08=if_else( between
(time_from_ext_to_opiods_infusion_min , -480, -420 ),opiod_rate,NULL)
,opiods_infusion_neg_epoch_09=if_else( between
(time_from_ext_to_opiods_infusion_min , -540, -480 ),opiod_rate,NULL)
,opiods_infusion_neg_epoch_10=if_else( between
(time_from_ext_to_opiods_infusion_min , -600, -540 ),opiod_rate,NULL)
,opiods_infusion_neg_epoch_11=if_else( between
(time_from_ext_to_opiods_infusion_min , -660, -600 ),opiod_rate,NULL)
,opiods_infusion_neg_epoch_12=if_else( between
(time_from_ext_to_opiods_infusion_min , -720, -660 ),opiod_rate,NULL)
  )

cohort_with_opiods_infusion_neg_epoch<-cohort_with_opiods_infusion_neg_epoch%>%
select(subject_id,opiods_infusion_neg_epoch_01,opiods_infusion_neg_epoch_02,opiods

```

```

_infusion_neg_epoch_03,opioids_infusion_neg_epoch_04,opioids_infusion_neg_epoch_05,opioids_infusion_neg_epoch_06,opioids_infusion_neg_epoch_07,opioids_infusion_neg_epoch_08,opioids_infusion_neg_epoch_09,opioids_infusion_neg_epoch_10,opioids_infusion_neg_epoch_11,opioids_infusion_neg_epoch_12)%>%
  group_by(subject_id)%>%
  summarise_all(my.max)
```

Benzodiazepines Before Extubation

Negative Epochs Creation (max rate mg/h)

```r eval=FALSE, include=FALSE}

#Drug Push is Bolus, so I filtered them out
benzodiazepines_infusion_before_ext<-benzodiazepines_infusion_from_patients_tofilter
%>%
  filter(
    time_from_ext_to_benzodiazepines_infusion_min >=-12*60
  )

n_distinct(benzodiazepines_infusion_before_ext$subject_id)
#240 patients 2/13/19

library(dplyr)

cohort_with_benzodiazepines_infusion_neg_epoch<-benzodiazepines_infusion_before_ext
%>%
  mutate(
    benzodiazepines_infusion_neg_epoch_01=if_else( between (
time_from_ext_to_benzodiazepines_infusion_min , -60, 0 ),opioid_rate,NULL)
,benzodiazepines_infusion_neg_epoch_02=if_else( between (
time_from_ext_to_benzodiazepines_infusion_min , -120, -60 ),opioid_rate,NULL)
,benzodiazepines_infusion_neg_epoch_03=if_else( between (
time_from_ext_to_benzodiazepines_infusion_min , -180, -120 ),opioid_rate,NULL)
,benzodiazepines_infusion_neg_epoch_04=if_else( between (
time_from_ext_to_benzodiazepines_infusion_min , -240, -180 ),opioid_rate,NULL)
,benzodiazepines_infusion_neg_epoch_05=if_else( between (
time_from_ext_to_benzodiazepines_infusion_min , -300, -240 ),opioid_rate,NULL)
,benzodiazepines_infusion_neg_epoch_06=if_else( between (
time_from_ext_to_benzodiazepines_infusion_min , -360, -300 ),opioid_rate,NULL)
,benzodiazepines_infusion_neg_epoch_07=if_else( between (
time_from_ext_to_benzodiazepines_infusion_min , -420, -360 ),opioid_rate,NULL)
,benzodiazepines_infusion_neg_epoch_08=if_else( between (
time_from_ext_to_benzodiazepines_infusion_min , -480, -420 ),opioid_rate,NULL)
,benzodiazepines_infusion_neg_epoch_09=if_else( between (
time_from_ext_to_benzodiazepines_infusion_min , -540, -480 ),opioid_rate,NULL)
,benzodiazepines_infusion_neg_epoch_10=if_else( between (
time_from_ext_to_benzodiazepines_infusion_min , -600, -540 ),opioid_rate,NULL)
,benzodiazepines_infusion_neg_epoch_11=if_else( between (
time_from_ext_to_benzodiazepines_infusion_min , -660, -600 ),opioid_rate,NULL)
,benzodiazepines_infusion_neg_epoch_12=if_else( between (
time_from_ext_to_benzodiazepines_infusion_min , -720, -660 ),opioid_rate,NULL)
  )

```

```

cohort_with_benzodiazepines_infusion_neg_epoch<-
cohort_with_benzodiazepines_infusion_neg_epoch%>%

select(subject_id,benzodiazepines_infusion_neg_epoch_01,benzodiazepines_infusion_neg_
epoch_02,benzodiazepines_infusion_neg_epoch_03,benzodiazepines_infusion_neg_epoch_04,
benzodiazepines_infusion_neg_epoch_05,benzodiazepines_infusion_neg_epoch_06,benzodiaz
epines_infusion_neg_epoch_07,benzodiazepines_infusion_neg_epoch_08,benzodiazepines_in
fusion_neg_epoch_09,benzodiazepines_infusion_neg_epoch_10,benzodiazepines_infusion_ne
g_epoch_11,benzodiazepines_infusion_neg_epoch_12)%>%
  group_by(subject_id)%>%
  summarise_all(my.max)
...

### Bolus Drugs Before Extubation

#### Opioids Before Extubation

##### Data extraction

##### Epochs Creation (total amount mg)

```{r eval=FALSE, include=FALSE}

#Drug Push is Bolus, so I filtered them out

opiods_bolus_before_ext<-opiods_bolus_from_patients_tofilter %>%
  filter(
    time_from_ext_to_opiods_bolus_min >=-12*60
  )

n_distinct(opiods_bolus_before_ext$subject_id) # 430 12/13/19

library(dplyr)

cohort_with_opiods_bolus_neg_epoch<-opiods_bolus_before_ext %>%
  mutate(
    opiods_bolus_neg_epoch_01=if_else(between(time_from_ext_to_opiods_bolus_min, -60,
0) ,opiod_amount,NULL)
,opiods_bolus_neg_epoch_02=if_else(between(time_from_ext_to_opiods_bolus_min,-120,
-60) ,opiod_amount,NULL)
,opiods_bolus_neg_epoch_03=if_else(between(time_from_ext_to_opiods_bolus_min,-180,-
120) ,opiod_amount,NULL)
,opiods_bolus_neg_epoch_04=if_else(between(time_from_ext_to_opiods_bolus_min,-240,-
180) ,opiod_amount,NULL)
,opiods_bolus_neg_epoch_05=if_else(between(time_from_ext_to_opiods_bolus_min,-300,-
240) ,opiod_amount,NULL)
,opiods_bolus_neg_epoch_06=if_else(between(time_from_ext_to_opiods_bolus_min,-360,-
300) ,opiod_amount,NULL)
,opiods_bolus_neg_epoch_07=if_else(between(time_from_ext_to_opiods_bolus_min,-420,-
360) ,opiod_amount,NULL)

```

```

,opioids_bolus_neg_epoch_08=if_else(between(time_from_ext_to_opioids_bolus_min,-480,-
420) ,opioid_amount,NULL)
,opioids_bolus_neg_epoch_09=if_else(between(time_from_ext_to_opioids_bolus_min,-540,-
480) ,opioid_amount,NULL)
,opioids_bolus_neg_epoch_10=if_else(between(time_from_ext_to_opioids_bolus_min,-600,-
540) ,opioid_amount,NULL)
,opioids_bolus_neg_epoch_11=if_else(between(time_from_ext_to_opioids_bolus_min,-660,-
600) ,opioid_amount,NULL)
,opioids_bolus_neg_epoch_12=if_else(between(time_from_ext_to_opioids_bolus_min,-720,-
660) ,opioid_amount,NULL)
)

my.sum <- function(x) ifelse( !all(is.na(x)), sum(x, na.rm=T), NA)

cohort_with_opioids_bolus_neg_epoch<-cohort_with_opioids_bolus_neg_epoch%>%

select(subject_id,opioids_bolus_neg_epoch_01,opioids_bolus_neg_epoch_02,opioids_bolus
_neg_epoch_03,opioids_bolus_neg_epoch_04,opioids_bolus_neg_epoch_05,opioids_bolus_neg
_epoch_06,opioids_bolus_neg_epoch_07,opioids_bolus_neg_epoch_08,opioids_bolus_neg_epo
ch_09,opioids_bolus_neg_epoch_10,opioids_bolus_neg_epoch_11,opioids_bolus_neg_epoch_1
2)%>%
  group_by(subject_id)%>%
  summarise_all(my.sum)
...

#### Benzodiazepines Before Extubation

##### Data extraction Before Extubation

##### Negative Epochs Creation (total amount mg/h)

```{r eval=FALSE, include=FALSE}

benzodiazepines_bolus_before_ext<-benzodiazepines_bolus_from_patients_tofilter %>%
  filter(
    time_from_ext_to_benzodiazepines_bolus_min >=-60*12
  )

n_distinct(benzodiazepines_bolus_before_ext$subject_id)
#252 12/13/19

library(dplyr)

cohort_with_benzodiazepines_bolus_neg_epoch<-benzodiazepines_bolus_before_ext %>%
  mutate(

benzodiazepines_bolus_neg_epoch_01=if_else(between(time_from_ext_to_benzodiazepines_b
olus_min, -60, 0),opioid_amount,NULL)
,benzodiazepines_bolus_neg_epoch_02=if_else(between(time_from_ext_to_benzodiazepines_
bolus_min,-120, -60),opioid_amount,NULL)
,benzodiazepines_bolus_neg_epoch_03=if_else(between(time_from_ext_to_benzodiazepines_
bolus_min,-180,-120),opioid_amount,NULL)
,benzodiazepines_bolus_neg_epoch_04=if_else(between(time_from_ext_to_benzodiazepines_
bolus_min,-240,-180),opioid_amount,NULL)

```

```
,benzodiazepines_bolus_neg_epoch_05=if_else(between(time_from_ext_to_benzodiazepines_
bolus_min,-300,-240),opioid_amount,NULL)
,benzodiazepines_bolus_neg_epoch_06=if_else(between(time_from_ext_to_benzodiazepines_
bolus_min,-360,-300),opioid_amount,NULL)
,benzodiazepines_bolus_neg_epoch_07=if_else(between(time_from_ext_to_benzodiazepines_
bolus_min,-420,-360),opioid_amount,NULL)
,benzodiazepines_bolus_neg_epoch_08=if_else(between(time_from_ext_to_benzodiazepines_
bolus_min,-480,-420),opioid_amount,NULL)
,benzodiazepines_bolus_neg_epoch_09=if_else(between(time_from_ext_to_benzodiazepines_
bolus_min,-540,-480),opioid_amount,NULL)
,benzodiazepines_bolus_neg_epoch_10=if_else(between(time_from_ext_to_benzodiazepines_
bolus_min,-600,-540),opioid_amount,NULL)
,benzodiazepines_bolus_neg_epoch_11=if_else(between(time_from_ext_to_benzodiazepines_
bolus_min,-660,-600),opioid_amount,NULL)
,benzodiazepines_bolus_neg_epoch_12=if_else(between(time_from_ext_to_benzodiazepines_
bolus_min,-720,-660),opioid_amount,NULL)
)
```

```
my.sum <- function(x) ifelse( !all(is.na(x)), sum(x, na.rm=T), NA)
```

```
cohort_with_benzodiazepines_bolus_neg_epoch<-
cohort_with_benzodiazepines_bolus_neg_epoch%>%
```

```
select(subject_id,benzodiazepines_bolus_neg_epoch_01,benzodiazepines_bolus_neg_epoch_
02,benzodiazepines_bolus_neg_epoch_03,benzodiazepines_bolus_neg_epoch_04,benzodiazepi
nes_bolus_neg_epoch_05,benzodiazepines_bolus_neg_epoch_06,benzodiazepines_bolus_neg_e
poch_07,benzodiazepines_bolus_neg_epoch_08,benzodiazepines_bolus_neg_epoch_09,benzodi
azepines_bolus_neg_epoch_10,benzodiazepines_bolus_neg_epoch_11,benzodiazepines_bolus_
neg_epoch_12)%>%
  group_by(subject_id)%>%
  summarise_all(my.sum)
...
```

Pain Scores Epochs Value

Take the max per EPOCH

-Let's also describe these variables from the list:

223781	Pain Present	Pain Present	metavision	chartevents
	Pain/Sedation			
223782	Pain Type	Pain Type	metavision	chartevents
	Pain/Sedation			
223783	Pain Location	Pain Location	metavision	chartevents
	Pain/Sedation			
223784	Pain Cause	Pain Cause	metavision	chartevents
	Pain/Sedation			
223794	Pain Level Acceptable	Pain Level Acceptable	metavision	
	chartevents	Pain/Sedation	Text	
223795	Pain Assessment Method	Pain Assessment Method	metavision	
	chartevents	Pain/Sedation	Text	

are we having more than 1 per epoch?

Data extraction

```

```{r}
query<-' SELECT chartevents.subject_id,
        chartevents.hadm_id,
        chartevents.icustay_id,
        chartevents.charttime AS charttime_ps,
        chartevents.itemid,
        label,
        chartevents.value
        FROM `physionet-data.mimiciii_clinical.chartevents` chartevents INNER JOIN
`physionet-data.mimiciii_clinical.d_items` d_items
ON chartevents.itemid = d_items.itemid
WHERE chartevents.itemid IN (1044, 1045, 225813, 227881, 223791, 223782, 223784,
223794, 223795)
'

painschores_cohort_24<-query_exec(query, project = project_HST,use_legacy_sql =
F,max_pages = Inf)
```

Epochs Creation

```{r}
ps_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex
tubation','deathtime')], painschores_cohort_24)

# we noticed some timezones are alteterd, so we reset them
ps_from_patients_tofilter$charttime_ps<-
as.character(ps_from_patients_tofilter$charttime_ps)
ps_from_patients_tofilter$charttime.extubation<-
as.character(ps_from_patients_tofilter$charttime.extubation)
ps_from_patients_tofilter$deathtime<-
as.character(ps_from_patients_tofilter$deathtime)

ps_from_patients_tofilter$charttime_ps<-
as.POSIXct(ps_from_patients_tofilter$charttime_ps, format='%Y-%m-%d %H:%M:%S',
tz="GMT")
ps_from_patients_tofilter$charttime.extubation<-
as.POSIXct(ps_from_patients_tofilter$charttime.extubation, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
ps_from_patients_tofilter$deathtime<-as.POSIXct(ps_from_patients_tofilter$deathtime,
format='%Y-%m-%d %H:%M:%S', tz="GMT")

ps_from_patients_tofilter['time_from_ext_to_ps_min']<-
difftime(ps_from_patients_tofilter$charttime_ps,ps_from_patients_tofilter$charttime.e
xtubation,units = 'mins')

pain_patients<-ps_from_patients_tofilter %>%
  filter(
    time_from_ext_to_ps_min >=0
  )

```

```

unique(pain_patients$value)
n_distinct(pain_patients$subject_id)

pain_patients<-pain_patients%>%filter(itemid !=223795)

pain_patients %>% group_by(itemid,label,value) %>% summarise(number_of_patients =
n_distinct(subject_id)) %>% arrange(label,desc(number_of_patients))

library(plyr)

#Mapping pain into numbers
#Reference
#https://www.disabled-world.com/pics/1/pain-scale-chart.gif

pain_patients$value<-revalue(pain_patients$value,
                             c( 'None to Mild'= 1
                                , 'None' = 0
                                , 'Mild to Moderate' = 3
                                , 'Unable to Score' = NA
                                , 'Moderate to Severe'= 5
                                , 'Moderate to Severe.'= 5
                                , 'Moderate'=4
                                , 'Worst'=10
                                , 'Mild '=2
                                , '0-None'=0
                                , 'Severe to Worse'=9
                                , 'Severe'=6
                                ))

pain_patients$value<-as.numeric(pain_patients$value)

#257 with storetime AS charttime_ps
if(any(grepl("package:plyr", search())))) detach("package:plyr") else message("plyr
not loaded")
## plyr not loaded
library("plyr")
## if(any(grepl("package:plyr", search())))) detach("package:plyr") else message("plyr
not loaded")

cohort_with_ps_epoch<-pain_patients %>%
  mutate(
    ps_epoch_01=if_else(time_from_ext_to_ps_min>=0 &
time_from_ext_to_ps_min<=60,value,NULL)
,ps_epoch_02=if_else(time_from_ext_to_ps_min>60 &
time_from_ext_to_ps_min<=120,value,NULL)
,ps_epoch_03=if_else(time_from_ext_to_ps_min>120 &
time_from_ext_to_ps_min<=180,value,NULL)
,ps_epoch_04=if_else(time_from_ext_to_ps_min>180 &
time_from_ext_to_ps_min<=240,value,NULL)
,ps_epoch_05=if_else(time_from_ext_to_ps_min>240 &
time_from_ext_to_ps_min<=300,value,NULL)

```



```
,ps_epoch_06=if_else(time_from_ext_to_ps_min>300 &
time_from_ext_to_ps_min<=360,value,NULL)
,ps_epoch_07=if_else(time_from_ext_to_ps_min>360 &
time_from_ext_to_ps_min<=420,value,NULL)
,ps_epoch_08=if_else(time_from_ext_to_ps_min>420 &
time_from_ext_to_ps_min<=480,value,NULL)
,ps_epoch_09=if_else(time_from_ext_to_ps_min>480 &
time_from_ext_to_ps_min<=540,value,NULL)
,ps_epoch_10=if_else(time_from_ext_to_ps_min>540 &
time_from_ext_to_ps_min<=600,value,NULL)
,ps_epoch_11=if_else(time_from_ext_to_ps_min>600 &
time_from_ext_to_ps_min<=660,value,NULL)
,ps_epoch_12=if_else(time_from_ext_to_ps_min>660 &
time_from_ext_to_ps_min<=720,value,NULL)
)
```

```
cohort_with_ps_epoch<-cohort_with_ps_epoch%>%
```

```
select(subject_id,ps_epoch_01,ps_epoch_02,ps_epoch_03,ps_epoch_04,ps_epoch_05,ps_epoch_06,ps_epoch_07,ps_epoch_08,ps_epoch_09,ps_epoch_10,ps_epoch_11,ps_epoch_12)%>%
  group_by(subject_id)%>%
  #summarize(mean(value,na.rm=TRUE))
  summarize_all(my.max)
```

```
#cohort_with_ps_epoch[cohort_with_ps_epoch== -Inf]<-NA
...
```

```
## Exploring Pain Scores
```

```
` `{r}
ps_from_patients_tofilter_short<-pain_patients%>%
  select(subject_id,charttime_ps,itemid, value)

# ps_from_patients_tofilter_summarized<-ps_from_patients_tofilter_short%>%
#   cast(subject_id+charttime_ps~itemid,value = 'value')
...
```

```
# Respiratory Effort Epochs Value
```

The respiratory effort variables look pretty good! You can't summarize categorical data with means etc. The data can be reported as a frequency distribution by Epoch after extubation. In instances where there is disagreement within the same Epoch, Coding for "Labored", "Splinting", or "Dyspneic" should "win" for the value of the cell in that Epoch. Make sense?

```
## Data Extraction
```

```
` `{r}
query<-(' SELECT chartevents.subject_id,
```

```

    chartevents.hadm_id,
    chartevents.icustay_id,
    chartevents.itemid,
    chartevents.charttime AS charttime_re,
    chartevents.value
  FROM `physionet-data.mimiciii_clinical.chartevents` chartevents
WHERE
chartevents.itemid IN (616,223990)')

re_cohort_24<-query_exec(query, project = project_HST,use_legacy_sql = F,max_pages =
Inf)

...

## Epochs Creation

```{r}
library(dplyr)
re_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex
tubation','deathtime')], re_cohort_24)

# we noticed some timezones are alteterd, so we reset them
re_from_patients_tofilter$charttime_re<-
as.character(re_from_patients_tofilter$charttime_re)
re_from_patients_tofilter$charttime.extubation<-
as.character(re_from_patients_tofilter$charttime.extubation)
re_from_patients_tofilter$deathtime<-
as.character(re_from_patients_tofilter$deathtime)

re_from_patients_tofilter$charttime_re<-
as.POSIXct(re_from_patients_tofilter$charttime_re, format='%Y-%m-%d %H:%M:%S',
tz="GMT")
re_from_patients_tofilter$charttime.extubation<-
as.POSIXct(re_from_patients_tofilter$charttime.extubation, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
re_from_patients_tofilter$deathtime<-as.POSIXct(re_from_patients_tofilter$deathtime,
format='%Y-%m-%d %H:%M:%S', tz="GMT")

re_from_patients_tofilter['time_from_ext_to_re_min']<-
difftime(re_from_patients_tofilter$charttime_re,re_from_patients_tofilter$charttime.e
xtubation,units = 'mins')

re_patients<-re_from_patients_tofilter %>%
  filter(
    time_from_ext_to_re_min >=0
  )

table(re_patients$value)

n_distinct(re_patients$subject_id)
#294 patients with storetime AS charttime_re

```

```

#priorizing Dyspneic and Labored
#Dyspneic 1 Labored 1 Agonal 0 Apneic 0 Normal 0 Shallow 0

#mapping strings to numbers so then we can prioritize
re_patients$value<-plyr::revalue(re_patients$value, c('Dyspneic' = 1
, 'Labored' = 1
, 'Agonal'= 0
, 'Apneic' = 0
, 'Normal'= 0
, 'Shallow'= 0
))

re_patients$value<-as.numeric(re_patients$value)

if(any(grepl("package:plyr", search())))) detach("package:plyr") else message("plyr
not loaded")
## plyr not loaded
library("plyr")
## if(any(grepl("package:plyr", search())))) detach("package:plyr") else message("plyr
not loaded")

cohort_with_re_epoch<-re_patients %>%
  mutate(
    re_epoch_01=if_else(time_from_ext_to_re_min>=0 &
time_from_ext_to_re_min<=60,value,NULL)
, re_epoch_02=if_else(time_from_ext_to_re_min>60 &
time_from_ext_to_re_min<=120,value,NULL)
, re_epoch_03=if_else(time_from_ext_to_re_min>120 &
time_from_ext_to_re_min<=180,value,NULL)
, re_epoch_04=if_else(time_from_ext_to_re_min>180 &
time_from_ext_to_re_min<=240,value,NULL)
, re_epoch_05=if_else(time_from_ext_to_re_min>240 &
time_from_ext_to_re_min<=300,value,NULL)
, re_epoch_06=if_else(time_from_ext_to_re_min>300 &
time_from_ext_to_re_min<=360,value,NULL)
, re_epoch_07=if_else(time_from_ext_to_re_min>360 &
time_from_ext_to_re_min<=420,value,NULL)
, re_epoch_08=if_else(time_from_ext_to_re_min>420 &
time_from_ext_to_re_min<=480,value,NULL)
, re_epoch_09=if_else(time_from_ext_to_re_min>480 &
time_from_ext_to_re_min<=540,value,NULL)
, re_epoch_10=if_else(time_from_ext_to_re_min>540 &
time_from_ext_to_re_min<=600,value,NULL)
, re_epoch_11=if_else(time_from_ext_to_re_min>600 &
time_from_ext_to_re_min<=660,value,NULL)
, re_epoch_12=if_else(time_from_ext_to_re_min>660 &
time_from_ext_to_re_min<=720,value,NULL)
  )

cohort_with_re_epoch<-cohort_with_re_epoch%>%

```

```

select(subject_id,re_epoch_01,re_epoch_02,re_epoch_03,re_epoch_04,re_epoch_05,re_epoch_06
, re_epoch_07,re_epoch_08,re_epoch_09,re_epoch_10,re_epoch_11,re_epoch_12)%>%
  group_by(subject_id)%>%
  dplyr::summarize_all(my.max)

#view(dfSummary(cohort_with_re_epoch))
...

# RASS effort Epochs Value

I'm taking the max per epoch

## Data Extraction

```{r}
query<-( ' SELECT chartevents.subject_id,
          chartevents.hadm_id,
          chartevents.icustay_id,
          chartevents.itemid,
          chartevents.charttime AS charttime_rass,
          chartevents.value
        FROM `physionet-data.mimiciii_clinical.chartevents` chartevents
        WHERE
        chartevents.itemid IN (228302,228299,228096)')

rass_cohort_24<-query_exec(query, project = project_HST,use_legacy_sql = F,max_pages
= Inf)
...

## Epochs Creation

```{r}
library(dplyr)
rass_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.extubation','deathtime')], rass_cohort_24)

# we noticed some timezones are alteterd, so we reset them
rass_from_patients_tofilter$charttime_rass<-
as.character(rass_from_patients_tofilter$charttime_rass)
rass_from_patients_tofilter$charttime.extubation<-
as.character(rass_from_patients_tofilter$charttime.extubation)
rass_from_patients_tofilter$deathtime<-
as.character(rass_from_patients_tofilter$deathtime)

```

```

rass_from_patients_tofilter$charttime_rass<-
as.POSIXct(rass_from_patients_tofilter$charttime_rass, format='%Y-%m-%d %H:%M:%S',
tz="GMT")
rass_from_patients_tofilter$charttime.extubation<-
as.POSIXct(rass_from_patients_tofilter$charttime.extubation, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
rass_from_patients_tofilter$deathtime<-
as.POSIXct(rass_from_patients_tofilter$deathtime, format='%Y-%m-%d %H:%M:%S',
tz="GMT")

rass_from_patients_tofilter['time_from_ext_to_rass_min']<-
difftime(rass_from_patients_tofilter$charttime_rass,rass_from_patients_tofilter$chart
time.extubation,units = 'mins')

rass_patients<-rass_from_patients_tofilter %>%
  filter(
    time_from_ext_to_rass_min >=0
  )

library(plyr)

#Mapping rass into numbers
#Reference
#httrass://www.disabled-world.com/pics/1/rass-scale-chart.gif

rass_patients$value<-revalue(rass_patients$value,
                             c("-1 Awakens to voice (eye
opening/contact) > 10 sec"= -1
                             , "-2 Light sedation, briefly awakens
to voice (eye opening/contact) < 10 sec" = -2
                             , "-3 Moderate sedation, movement or
eye opening; No eye contact" = -3
                             , "-4 Deep sedation, no response to
voice, but movement or eye opening to physical stimulation" = -4
                             , "-5 Unarousable, no response to
voice or physical stimulation"= -5
                             , " 0 Alert and calm"= 0
                             , "+1 Anxious, apprehensive, but not
aggressive"=1
                             , "+2 Frequent nonpurposeful
movement, fights ventilator"=2
                             , "+3 Pulls or removes tube(s) or
catheter(s); aggressive"=3
                             ))

rass_patients$value<-as.numeric(rass_patients$value)
n_distinct(rass_patients$subject_id)
#194 patients with storetime AS charttime_rass
if(any(grepl("package:plyr", search())) detach("package:plyr") else message("plyr
not loaded")
## plyr not loaded
library("plyr")

```

```
## if(any(grepl("package:plyr", search())) detach("package:plyr") else message("plyr
not loaded"))
```

```
my.max <- function(x) ifelse( !all(is.na(x)), max(x, na.rm=T), NA)
```

```
cohort_with_max_rass_epoch<-rass_patients %>%
  mutate(
    #change to rass epoch max and min
    max_rass_epoch_01=if_else(time_from_ext_to_rass_min>=0 &
time_from_ext_to_rass_min<=60,value,NULL)
,max_rass_epoch_02=if_else(time_from_ext_to_rass_min>60 &
time_from_ext_to_rass_min<=120,value,NULL)
,max_rass_epoch_03=if_else(time_from_ext_to_rass_min>120 &
time_from_ext_to_rass_min<=180,value,NULL)
,max_rass_epoch_04=if_else(time_from_ext_to_rass_min>180 &
time_from_ext_to_rass_min<=240,value,NULL)
,max_rass_epoch_05=if_else(time_from_ext_to_rass_min>240 &
time_from_ext_to_rass_min<=300,value,NULL)
,max_rass_epoch_06=if_else(time_from_ext_to_rass_min>300 &
time_from_ext_to_rass_min<=360,value,NULL)
,max_rass_epoch_07=if_else(time_from_ext_to_rass_min>360 &
time_from_ext_to_rass_min<=420,value,NULL)
,max_rass_epoch_08=if_else(time_from_ext_to_rass_min>420 &
time_from_ext_to_rass_min<=480,value,NULL)
,max_rass_epoch_09=if_else(time_from_ext_to_rass_min>480 &
time_from_ext_to_rass_min<=540,value,NULL)
,max_rass_epoch_10=if_else(time_from_ext_to_rass_min>540 &
time_from_ext_to_rass_min<=600,value,NULL)
,max_rass_epoch_11=if_else(time_from_ext_to_rass_min>600 &
time_from_ext_to_rass_min<=660,value,NULL)
,max_rass_epoch_12=if_else(time_from_ext_to_rass_min>660 &
time_from_ext_to_rass_min<=720,value,NULL)
  )
```

```
cohort_with_max_rass_epoch<-cohort_with_max_rass_epoch%>%
```

```
select(subject_id,max_rass_epoch_01,max_rass_epoch_02,max_rass_epoch_03,max_rass_epoc
h_04,max_rass_epoch_05,max_rass_epoch_06,max_rass_epoch_07,max_rass_epoch_08,max_rass
_epoch_09,max_rass_epoch_10,max_rass_epoch_11,max_rass_epoch_12)%>%
  group_by(subject_id)%>%
  summarize_all(my.max)
```

```
my.min <- function(x) ifelse( !all(is.na(x)), min(x, na.rm=T), NA)
```

```
cohort_with_min_rass_epoch<-rass_patients %>%
  mutate(
    #change to rass epoch min and min
    min_rass_epoch_01=if_else(time_from_ext_to_rass_min>=0 &
time_from_ext_to_rass_min<=60,value,NULL)
,min_rass_epoch_02=if_else(time_from_ext_to_rass_min>60 &
time_from_ext_to_rass_min<=120,value,NULL)
```

```

,min_rass_epoch_03=if_else(time_from_ext_to_rass_min>120 &
time_from_ext_to_rass_min<=180,value,NULL)
,min_rass_epoch_04=if_else(time_from_ext_to_rass_min>180 &
time_from_ext_to_rass_min<=240,value,NULL)
,min_rass_epoch_05=if_else(time_from_ext_to_rass_min>240 &
time_from_ext_to_rass_min<=300,value,NULL)
,min_rass_epoch_06=if_else(time_from_ext_to_rass_min>300 &
time_from_ext_to_rass_min<=360,value,NULL)
,min_rass_epoch_07=if_else(time_from_ext_to_rass_min>360 &
time_from_ext_to_rass_min<=420,value,NULL)
,min_rass_epoch_08=if_else(time_from_ext_to_rass_min>420 &
time_from_ext_to_rass_min<=480,value,NULL)
,min_rass_epoch_09=if_else(time_from_ext_to_rass_min>480 &
time_from_ext_to_rass_min<=540,value,NULL)
,min_rass_epoch_10=if_else(time_from_ext_to_rass_min>540 &
time_from_ext_to_rass_min<=600,value,NULL)
,min_rass_epoch_11=if_else(time_from_ext_to_rass_min>600 &
time_from_ext_to_rass_min<=660,value,NULL)
,min_rass_epoch_12=if_else(time_from_ext_to_rass_min>660 &
time_from_ext_to_rass_min<=720,value,NULL)
)

cohort_with_min_rass_epoch<-cohort_with_min_rass_epoch%>%

select(subject_id,min_rass_epoch_01,min_rass_epoch_02,min_rass_epoch_03,min_rass_epoc
h_04,min_rass_epoch_05,min_rass_epoch_06,min_rass_epoch_07,min_rass_epoch_08,min_rass
_epoch_09,min_rass_epoch_10,min_rass_epoch_11,min_rass_epoch_12)%>%
  group_by(subject_id)%>%
  summarize_all(my.min)
...

---
title: "Demographics"
author: "Miguel Angel Armengol de la Hoz"
date: "August 1, 2018"
output: html_document
---

# Libraries Data and sql connections load

```{r}
#library(RPostgreSQL)
library(DBI)
library(dplyr)
library(bigrquery)
library(reshape)

project_HST = ""
...

# Demographics

```{r}

```

```

demographics_all<-dbGetQuery(con,"with last_ventduration AS(
SELECT icustay_id,mech_vent_duration_hrs FROM
(
SELECT
    icustay_id
    ,ventdurations.duration_hours As mech_vent_duration_hrs
    ,Row_Number() Over (Partition By mimiciii.ventdurations.icustay_id Order By
mimiciii.ventdurations.ventnum DESC) As position
    FROM mimiciii.ventdurations
    ) As ssq
WHERE position = 1
)

,first_icd9 AS(
SELECT hadm_id, first_icd9 FROM
(
SELECT
    hadm_id
    ,icd9_code as first_icd9
    ,Row_Number() Over (Partition By mimiciii.diagnoses_icd.hadm_id Order By
mimiciii.diagnoses_icd.icd9_code ASC) As icd9_code_num
    FROM mimiciii.diagnoses_icd
    ) As ssq
WHERE icd9_code_num = 1
)

SELECT distinct
    mimiciii.icustays.subject_id
    ,mimiciii.icustays.hadm_id
    ,mimiciii.icustays.icustay_id
    ,mimiciii.patients.gender
    ,CASE
        WHEN Date_Part('year', Age(mimiciii.admissions.admittime, mimiciii.patients.dob)) >
89 THEN 91.4
        ELSE Date_Part('year', Age(mimiciii.admissions.admittime, mimiciii.patients.dob))
    END As age
    ,EXTRACT(epoch FROM mimiciii.admissions.disctime -
mimiciii.admissions.admittime)/60 AS length_of_stay_hrs
    ,mimiciii.admissions.insurance
    ,mimiciii.admissions.ethnicity
    ,mimiciii.sofa.sofa
    ,mimiciii.oasis.oasis
    ,round(mimiciii.oasis.oasis_prob, 2) as oasis_prob
    ,mimiciii.gcsfirstday.mingcs As mingcs_firstday
    ,mimiciii.gcsfirstday.gcsmotor As gcsmotor_firstday
    ,mimiciii.gcsfirstday.gcsverbal As gcsverbal_firstday
    ,mimiciii.gcsfirstday.gcseyes As gcseyes_firstday
    ,mimiciii.gcsfirstday.endotrachflag As endotrachflag_firstday
    ,mimiciii.smoker.smoking
    ,mimiciii.icustays.last_careunit
    ,last_ventduration.mech_vent_duration_hrs
FROM mimiciii.icustays
LEFT JOIN mimiciii.patients ON
mimiciii.icustays.subject_id = mimiciii.patients.subject_id

```



```

LEFT JOIN mimiciii.admissions ON
mimiciii.admissions.hadm_id = mimiciii.icustays.hadm_id
LEFT JOIN mimiciii.oasis ON
mimiciii.oasis.icustay_id = mimiciii.icustays.icustay_id
LEFT JOIN first_icd9 ON
first_icd9.hadm_id = mimiciii.icustays.hadm_id
LEFT JOIN mimiciii.sofa ON
mimiciii.sofa.icustay_id = mimiciii.icustays.icustay_id
LEFT JOIN mimiciii.gcsfirstday ON
mimiciii.gcsfirstday.icustay_id = mimiciii.icustays.icustay_id
LEFT JOIN mimiciii.smoker ON
mimiciii.smoker.subject_id = mimiciii.icustays.subject_id
LEFT JOIN mimiciii.ventdurations ON
mimiciii.icustays.icustay_id = mimiciii.ventdurations.icustay_id
LEFT JOIN last_ventduration ON
last_ventduration.icustay_id = mimiciii.icustays.icustay_id
ORDER BY hadm_id")

...

# GCS last day

```{r}
gcs_last_day<-dbGetQuery(con,"
SET
    search_path TO mimiciii;
-- itemids used:
-- CAREVUE
--    723 as GCSVerbal
--    454 as GCSMotor
--    184 as GCSEyes
-- METAVISION
--    223900 GCS - Verbal Response
--    223901 GCS - Motor Response
--    220739 GCS - Eye Opening
-- The code combines the itemids into the carevue itemids, then pivots those
-- So 223900 is changed to 723, then the itemid 723 is pivoted to form GCSVerbal
-- Note:
-- The GCS for sedated patients is defaulted to 15 in this code.
-- This is in line with how the data is meant to be collected.
-- e.g., from the SAPS II publication:
-- For sedated patients, the Glasgow Coma Score before sedation was used.
-- This was ascertained either from interviewing the physician who ordered the
sedation,
-- or by reviewing the patient's medical record.
WITH
    base AS (
    SELECT
        pvt.hadm_id,
        pvt.charttime
        -- Easier names - note we coalesced Metavision and CareVue IDs below
    ,
        MAX(CASE WHEN pvt.itemid = 454 THEN pvt.valuenum
            ELSE NULL END) AS GCSMotor,
        MAX(CASE WHEN pvt.itemid = 723 THEN pvt.valuenum

```

```

        ELSE NULL END) AS GCSVerbal,
MAX(CASE WHEN pvt.itemid = 184 THEN pvt.valuenum
    ELSE NULL END) AS GCSEyes
-- If verbal was set to 0 in the below select, then this is an intubated patient
, CASE WHEN MAX(CASE WHEN pvt.itemid = 723 THEN pvt.valuenum
    ELSE NULL END) = 0 THEN 1
    ELSE 0
END AS EndoTrachFlag,
--This retrieves the last value per hadm_id
ROW_NUMBER () OVER (PARTITION BY pvt.hadm_id ORDER BY pvt.charttime DESC) AS rn
FROM (
SELECT
    chartevents.hadm_id
    -- merge the itemids so that the pivot applies to both metavision/carevue data
    ,CASE WHEN chartevents.itemid IN (723,223900) THEN 723
        WHEN chartevents.itemid IN (454,223901) THEN 454
        WHEN chartevents.itemid IN (184, 220739) THEN 184
        ELSE chartevents.itemid
    END AS itemid
    -- convert the data into a number, reserving a value of 0 for ET/Trach
    ,CASE
    -- endotrach/vent is assigned a value of 0, later parsed specially
        WHEN chartevents.itemid = 723 AND chartevents.value = '1.0 ET/Trach' THEN 0 -
- carevue
        WHEN chartevents.itemid = 223900
        AND chartevents.value = 'No Response-ETT' THEN 0 -- metavision
        ELSE valuenum END AS valuenum,
    chartevents.CHARTTIME
FROM
    CHARTEVENTS
    -- get intime for charttime subselection
INNER JOIN admissions
ON CHARTEVENTS.hadm_id = admissions.hadm_id
    -- Isolate the desired GCS variables
WHERE
    chartevents.itemid IN (
        -- 198 -- GCS
        184, 454, 723 -- GCS components, CareVue
        , 223900, 223901, 220739 )-- GCS components, Metavision
    -- exclude rows marked as error
    AND CHARTEVENTS.error IS DISTINCT FROM 1 ) pvt
GROUP BY
    pvt.hadm_id,
    pvt.charttime ),
gcs AS (
SELECT
    b.*,
    b2.GCSVerbal AS GCSVerbalPrev,
    b2.GCSMotor AS GCSMotorPrev,
    b2.GCSEyes AS GCSEyesPrev
    -- Calculate GCS, factoring in special case when they are intubated and prev vals
    -- note that the coalesce are used to implement the following if:
    -- if current value exists, use it
    -- if previous value exists, use it
    -- otherwise, default to normal

```

```

CASE
-- replace GCS during sedation with 15
  WHEN b.GCSVerbal = 0 THEN 15
  WHEN b.GCSVerbal IS NULL AND b2.GCSVerbal = 0 THEN 15
-- if previously they were intub, but they aren't now, do not use previous GCS
values
  WHEN b2.GCSVerbal = 0 THEN coalesce(b.GCSMotor, 6) + coalesce(b.GCSVerbal, 5) +
coalesce(b.GCSEyes, 4)
-- otherwise, add up score normally, imputing previous value if none available at
current time
  ELSE coalesce(b.GCSMotor, coalesce(b2.GCSMotor, 6)) + coalesce(b.GCSVerbal,
coalesce(b2.GCSVerbal, 5)) + coalesce(b.GCSEyes, coalesce(b2.GCSEyes, 4))
END AS GCS
FROM
  base b
-- join to itself within 6 hours to get previous value
LEFT JOIN
  base b2
ON
  b.hadm_id = b2.hadm_id
  AND b.rn = b2.rn+1 ),
gcs_final AS (
SELECT
  gcs.* -- This sorts the data by GCS, so rn=1 is the the lowest GCS values to keep
,ROW_NUMBER () OVER (PARTITION BY gcs.hadm_id ORDER BY gcs.GCS ) AS IsMinGCS
FROM gcs
)
SELECT
  admissions.subject_id
,admissions.hadm_id
,gcs_final.charttime
,admissions.deathtime
, EXTRACT(epoch FROM admissions.deathtime - gcs_final.charttime)/3600 AS
hrs_from_gcs_to_death
-- The minimum GCS is determined by the above row partition, we only join if
IsMinGCS=1
,GCS AS last_MinGCS
,coalesce(GCSMotor, GCSMotorPrev) AS last_GCSMotor
,coalesce(GCSVerbal, GCSVerbalPrev) AS last_GCSVerbal
,coalesce(GCSEyes, GCSEyesPrev) AS last_GCSEyes
,EndoTrachFlag AS last_EndoTrachFlag
-- subselect down to the cohort of eligible patients
FROM
  admissions
LEFT JOIN
  gcs_final
ON
  admissions.hadm_id = gcs_final.hadm_id
  AND gcs_final.IsMinGCS = 1
AND admissions.deathtime IS NOT NULL
ORDER BY
  admissions.hadm_id ;
")

```

```

hadm_ids_fromourcohort24<-cmo_and_extubation_and_death_24$hadm_id

#hrs from gcs to extubation has to be positive.
gcs_last_day$hrs_from_gsc_to_death<-round(gcs_last_day$hrs_from_gcs_to_death,2)

gcs_last_day<-gcs_last_day %>%
  filter(!is.na(deathtime) & !is.na(charttime) & hadm_id %in%
hadm_ids_fromourcohort24)

gcs_last_day<-gcs_last_day %>%

select(subject_id,last_mingcs,last_gcsmotor,last_gcsverbal,last_gcseyes,last_endotrachealflag)

...

# Top 4 Last Vent Mode last day pre Ext

## Extraction

```{r}
query<-"
SELECT distinct subject_id
,charttime
,value
FROM `physionet-data.mimiciii_clinical.chartevents` chartevents
WHERE
itemid IN(223849,467,720)
AND value IS NOT NULL
AND value != 'Standby'
ORDER BY subject_id
"

vent_mode_last_pre_ext<-query_exec(query, project = project_HST,use_legacy_sql =
F,max_pages = Inf)

...

## Filter

```{r}

vent_mode_last_pre_ext_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.extubation','deathtime')], vent_mode_last_pre_ext)

vent_mode_last_pre_ext_tofilter<-
dplyr::rename(vent_mode_last_pre_ext_tofilter,charttime_last_pre_ext_ventmode=charttime)
vent_mode_last_pre_ext_tofilter<-
dplyr::rename(vent_mode_last_pre_ext_tofilter,last_vent_mode=value)

# we noticed some timezones are alteterd, so we reset them
vent_mode_last_pre_ext_tofilter$charttime_last_pre_ext_ventmode<-
as.character(vent_mode_last_pre_ext_tofilter$charttime_last_pre_ext_ventmode)

```

```

vent_mode_last_pre_ext_tofilter$charttime.extubation<-
as.character(vent_mode_last_pre_ext_tofilter$charttime.extubation)
vent_mode_last_pre_ext_tofilter$deathtime<-
as.character(vent_mode_last_pre_ext_tofilter$deathtime)

vent_mode_last_pre_ext_tofilter$charttime_last_pre_ext_ventmode<-
as.POSIXct(vent_mode_last_pre_ext_tofilter$charttime_last_pre_ext_ventmode,
format='%Y-%m-%d %H:%M:%S', tz="GMT")
vent_mode_last_pre_ext_tofilter$charttime.extubation<-
as.POSIXct(vent_mode_last_pre_ext_tofilter$charttime.extubation, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
vent_mode_last_pre_ext_tofilter$deathtime<-
as.POSIXct(vent_mode_last_pre_ext_tofilter$deathtime, format='%Y-%m-%d %H:%M:%S',
tz="GMT")

vent_mode_last_pre_ext_tofilter['time_from_last_vent_mode_to_ext_min']<-
difftime(vent_mode_last_pre_ext_tofilter$charttime.extubation,vent_mode_last_pre_ext_
tofilter$charttime_last_pre_ext_ventmode,units = 'mins')

vent_mode_last_pre_ext_tofilter<-vent_mode_last_pre_ext_tofilter %>%
  filter(
    time_from_last_vent_mode_to_ext_min >=0
  )

n_distinct(vent_mode_last_pre_ext_tofilter$subject_id)

vent_mode_last_pre_ext_tofilter<-
vent_mode_last_pre_ext_tofilter%>%arrange(subject_id,
time_from_last_vent_mode_to_ext_min)

vent_mode_last_pre_ext_final<-
vent_mode_last_pre_ext_tofilter[!duplicated(vent_mode_last_pre_ext_tofilter$subject_i
d),]
vent_mode_last_pre_ext_final<-
vent_mode_last_pre_ext_final%>%select(subject_id,time_from_last_vent_mode_to_ext_min,
last_vent_mode)
```

Top 15 ICD9 Codes

```{r}
query<-paste("SELECT subject_id, hadm_id, seq_num, icd9_code
FROM
`physionet-data.mimiciii_clinical.diagnoses_icd`diagnoses_icd
WHERE
seq_num <= 15
AND
hadm_id IN (",hadm_ids_fromourcohort24,")
")

top15icd9_codes<-query_exec(query, project = project_HST,use_legacy_sql = F,max_pages
= Inf)

```

```

top15icd9_codes %>% group_by(subject_id) %>% table(mutate(isconsecutive = seq_num -
lag(seq_num))) )

our_chohort_icd9_codes<-unique(top15icd9_codes$icd9_code)
our_chohort_icd9_codes<-paste0(sprintf("%s", our_chohort_icd9_codes), collapse = ",
")

query<-paste("SELECT icd9_code, short_title
FROM
`physionet-data.mimiciii_clinical.d_icd_diagnoses` d_icd_diagnoses
WHERE
icd9_code IN (",our_chohort_icd9_codes,")
")

our_chohort_icd9_codes_labels<-query_exec(query, project = project_HST,use_legacy_sql
= F,max_pages = Inf)

top15icd9_codes_with_labels<-
inner_join(top15icd9_codes,our_chohort_icd9_codes_labels)

top15icd9_codes_with_labels<-top15icd9_codes_with_labels %>%
select(subject_id,seq_num,short_title)

top15icd9_codes_with_labels_sumarized<-reshape( data =
top15icd9_codes_with_labels,direction = 'wide'
,idvar = 'subject_id'
,timevar = 'seq_num'
)
names(top15icd9_codes_with_labels_sumarized) <- gsub(x =
names(top15icd9_codes_with_labels_sumarized), pattern = "short_title", replacement =
"icd9_code_num")

...

# Last Pao2Fio2 pre Ext

## Extraction

```{r}
query<-paste("with pvt as
( -- begin query that extracts the data
select ie.subject_id, ie.hadm_id, ie.icustay_id
-- here we assign labels to ITEMIDs
-- this also fuses together multiple ITEMIDs containing the same data
, case
when itemid = 50800 then 'SPECIMEN'
when itemid = 50801 then 'AAD02'
when itemid = 50802 then 'BASEEXCESS'
when itemid = 50803 then 'BICARBONATE'
when itemid = 50804 then 'TOTALCO2'
when itemid = 50805 then 'CARBOXYHEMOGLOBIN'
when itemid = 50806 then 'CHLORIDE'
when itemid = 50808 then 'CALCIUM'
when itemid = 50809 then 'GLUCOSE'

```

```

when itemid = 50810 then 'HEMATOCRIT'
when itemid = 50811 then 'HEMOGLOBIN'
when itemid = 50812 then 'INTUBATED'
when itemid = 50813 then 'LACTATE'
when itemid = 50814 then 'METHEMOGLOBIN'
when itemid = 50815 then 'O2FLOW'
when itemid = 50816 then 'FIO2'
when itemid = 50817 then 'SO2' -- OXYGENSATURATION
when itemid = 50818 then 'PCO2'
when itemid = 50819 then 'PEEP'
when itemid = 50820 then 'PH'
when itemid = 50821 then 'PO2'
when itemid = 50822 then 'POTASSIUM'
when itemid = 50823 then 'REQUIREDO2'
when itemid = 50824 then 'SODIUM'
when itemid = 50825 then 'TEMPERATURE'
when itemid = 50826 then 'TIDALVOLUME'
when itemid = 50827 then 'VENTILATIONRATE'
when itemid = 50828 then 'VENTILATOR'
else null
end as label
, charttime
, value
-- add in some sanity checks on the values
, case
  when valuenum <= 0 then null
  when itemid = 50810 and valuenum > 100 then null -- hematocrit
  -- ensure FiO2 is a valid number between 21-100
  -- mistakes are rare (<100 obs out of ~100,000)
  -- there are 862 obs of valuenum == 20 - some people round down!
  -- rather than risk imputing garbage data for FiO2, we simply NULL invalid
values
  when itemid = 50816 and valuenum < 20 then null
  when itemid = 50816 and valuenum > 100 then null
  when itemid = 50817 and valuenum > 100 then null -- O2 sat
  when itemid = 50815 and valuenum > 70 then null -- O2 flow
  when itemid = 50821 and valuenum > 800 then null -- PO2
  -- conservative upper limit
else valuenum
end as valuenum

FROM `physionet-data.mimiciii_clinical.icustays` ie
left join `physionet-data.mimiciii_clinical.labevents` le
  on le.subject_id = ie.subject_id and le.hadm_id = ie.hadm_id
-- and le.charttime between (DATETIME_SUB(ie.intime, INTERVAL 6 HOUR)) and
(DATETIME_ADD(ie.intime, INTERVAL 1 DAY))
  and le.ITEMID in
  -- blood gases
  (
    50800, 50801, 50802, 50803, 50804, 50805, 50806, 50807, 50808, 50809
    , 50810, 50811, 50812, 50813, 50814, 50815, 50816, 50817, 50818, 50819
    , 50820, 50821, 50822, 50823, 50824, 50825, 50826, 50827, 50828
    , 51545
  )
), bloodgas_alldays AS(

```

```

select pvt.SUBJECT_ID, pvt.HADM_ID, pvt.ICUSTAY_ID, pvt.CHARTTIME
, max(case when label = 'SPECIMEN' then value else null end) as SPECIMEN
, max(case when label = 'AAD02' then valuenum else null end) as AAD02
, max(case when label = 'BASEEXCESS' then valuenum else null end) as BASEEXCESS
, max(case when label = 'BICARBONATE' then valuenum else null end) as BICARBONATE
, max(case when label = 'TOTALCO2' then valuenum else null end) as TOTALCO2
, max(case when label = 'CARBOXYHEMOGLOBIN' then valuenum else null end) as
CARBOXYHEMOGLOBIN
, max(case when label = 'CHLORIDE' then valuenum else null end) as CHLORIDE
, max(case when label = 'CALCIUM' then valuenum else null end) as CALCIUM
, max(case when label = 'GLUCOSE' then valuenum else null end) as GLUCOSE
, max(case when label = 'HEMATOCRIT' then valuenum else null end) as HEMATOCRIT
, max(case when label = 'HEMOGLOBIN' then valuenum else null end) as HEMOGLOBIN
, max(case when label = 'INTUBATED' then valuenum else null end) as INTUBATED
, max(case when label = 'LACTATE' then valuenum else null end) as LACTATE
, max(case when label = 'METHEMOGLOBIN' then valuenum else null end) as METHEMOGLOBIN
, max(case when label = 'O2FLOW' then valuenum else null end) as O2FLOW
, max(case when label = 'FIO2' then valuenum else null end) as FIO2
, max(case when label = 'SO2' then valuenum else null end) as SO2 -- OXYGENSATURATION
, max(case when label = 'PCO2' then valuenum else null end) as PCO2
, max(case when label = 'PEEP' then valuenum else null end) as PEEP
, max(case when label = 'PH' then valuenum else null end) as PH
, max(case when label = 'PO2' then valuenum else null end) as PO2
, max(case when label = 'POTASSIUM' then valuenum else null end) as POTASSIUM
, max(case when label = 'REQUIREDO2' then valuenum else null end) as REQUIREDO2
, max(case when label = 'SODIUM' then valuenum else null end) as SODIUM
, max(case when label = 'TEMPERATURE' then valuenum else null end) as TEMPERATURE
, max(case when label = 'TIDALVOLUME' then valuenum else null end) as TIDALVOLUME
, max(case when label = 'VENTILATIONRATE' then valuenum else null end) as
VENTILATIONRATE
, max(case when label = 'VENTILATOR' then valuenum else null end) as VENTILATOR
from pvt
group by pvt.subject_id, pvt.hadm_id, pvt.icustay_id, pvt.CHARTTIME
order by pvt.subject_id, pvt.hadm_id, pvt.icustay_id, pvt.CHARTTIME
),stg_spo2 AS (
    SELECT
        SUBJECT_ID,
        HADM_ID,
        ICUSTAY_ID,
        CHARTTIME
        -- max here is just used to group SpO2 by charttime
    ,
        MAX(CASE
            WHEN valuenum <= 0 OR valuenum > 100 THEN NULL
            ELSE valuenum END) AS SpO2
    FROM
        `physionet-data.mimiciii_clinical.chartevents`
        -- o2 sat
    WHERE
        ITEMID IN ( 646 -- SpO2
            ,
            220277 -- O2 saturation pulseoxymetry
        )
    GROUP BY
        SUBJECT_ID,

```



```

HADM_ID,
ICUSTAY_ID,
CHARTTIME ),
stg_fio2 AS (
SELECT
SUBJECT_ID,
HADM_ID,
ICUSTAY_ID,
CHARTTIME
-- pre-process the Fio2s to ensure they are between 21-100%
,
MAX(
CASE
WHEN itemid = 223835 THEN CASE
WHEN valuenum > 0
AND valuenum <= 1 THEN valuenum * 100
-- improperly input data - looks like O2 flow in litres
WHEN valuenum > 1 AND valuenum < 21 THEN NULL
WHEN valuenum >= 21
AND valuenum <= 100 THEN valuenum
ELSE NULL END -- unphysiological
WHEN itemid IN (3420, 3422)
-- all these values are well formatted
THEN valuenum
WHEN itemid = 190
AND valuenum > 0.20
AND valuenum < 1
-- well formatted but not in %
THEN valuenum * 100
ELSE NULL
END ) AS fio2_chartevents
FROM
`physionet-data.mimiciii_clinical.chartevents`
WHERE
ITEMID IN ( 3420 -- Fio2
,
190 -- Fio2 set
,
223835 -- Inspired O2 Fraction (Fio2)
,
3422 -- Fio2 [measured]
)
-- exclude rows marked as error
AND (error IS NULL
OR error = 1)
GROUP BY
SUBJECT_ID,
HADM_ID,
ICUSTAY_ID,
CHARTTIME ),
stg2 AS (
SELECT
bg.*,
ROW_NUMBER() OVER (PARTITION BY bg.icustay_id, bg.charttime ORDER BY s1.charttime
DESC) AS lastRowSpO2,

```

```

    s1.spo2
FROM
    bloodgas_alldays bg
LEFT JOIN
    stg_spo2 s1
    -- same patient
ON
    bg.icustay_id = s1.icustay_id
    -- spo2 occurred at most 2 hours before this blood gas
    AND s1.charttime >= DATETIME_SUB(bg.charttime,
        INTERVAL 2 HOUR)
    AND s1.charttime <= bg.charttime
WHERE
    bg.po2 IS NOT NULL ),
stg3 AS (
SELECT
    bg.*,
    ROW_NUMBER() OVER (PARTITION BY bg.icustay_id, bg.charttime ORDER BY s2.charttime
DESC) AS lastRowFiO2,
    s2.fio2_chartevents
    -- create our specimen prediction
,
    1/(1+EXP(-(-0.02544 + 0.04598 * po2 + coalesce(-0.15356 * spo2,
        -0.15356 * 97.49420 + 0.13429) + coalesce( 0.00621 * fio2_chartevents,
        0.00621 * 51.49550 + -0.24958) + coalesce( 0.10559 * hemoglobin,
        0.10559 * 10.32307 + 0.05954) + coalesce( 0.13251 * so2,
        0.13251 * 93.66539 + -0.23172) + coalesce(-0.01511 * pco2,
        -0.01511 * 42.08866 + -0.01630) + coalesce( 0.01480 * fio2,
        0.01480 * 63.97836 + -0.31142) + coalesce(-0.00200 * aado2,
        -0.00200 * 442.21186 + -0.01328) + coalesce(-0.03220 * bicarbonate,
        -0.03220 * 22.96894 + -0.06535) + coalesce( 0.05384 * totalco2,
        0.05384 * 24.72632 + -0.01405) + coalesce( 0.08202 * lactate,
        0.08202 * 3.06436 + 0.06038) + coalesce( 0.10956 * ph,
        0.10956 * 7.36233 + -0.00617) + coalesce( 0.00848 * o2flow,
        0.00848 * 7.59362 + -0.35803) ))) AS SPECIMEN_PROB
FROM
    stg2 bg
LEFT JOIN
    stg_fio2 s2
    -- same patient
ON
    bg.icustay_id = s2.icustay_id
    -- fio2 occurred at most 4 hours before this blood gas
    AND s2.charttime BETWEEN DATETIME_SUB(bg.charttime,
        INTERVAL 4 HOUR)
    AND bg.charttime
WHERE
    bg.lastRowSpO2 = 1 -- only the row with the most recent SpO2 (if no SpO2 found
lastRowSpO2 = 1)
)
SELECT
    subject_id,
    hadm_id,
    icustay_id,
    charttime,

```

```

SPECIMEN -- raw data indicating sample type, only present 80% of the time
-- prediction of specimen for missing data
-- oxygen related parameters
,
S02,
spo2 -- note spo2 is FROM `physionet-data.mimiciii_clinical.chartevents`
,
P02,
PC02,
fio2_chartevents,
FIO2,
AAD02
-- also calculate AAD02
,
CASE
  WHEN P02 IS NOT NULL AND pco2 IS NOT NULL AND coalesce(FIO2, fio2_chartevents)
IS NOT NULL
  -- multiple by 100 because FiO2 is in a % but should be a fraction
  THEN (coalesce(FIO2, fio2_chartevents)/100) * (760 - 47) - (pco2/0.8) - po2
  ELSE NULL
END AS AAD02_calc,
CASE
  WHEN P02 IS NOT NULL AND coalesce(FIO2, fio2_chartevents) IS NOT NULL
  -- multiply by 100 because FiO2 is in a % but should be a fraction
  THEN 100*P02/(coalesce(FIO2, fio2_chartevents))
  ELSE NULL
END AS PaO2FiO2
-- acid-base parameters
FROM
  stg3
WHERE
  lastRowFiO2 = 1 -- only the most recent FiO2
  -- restrict it to *only* arterial samples
  AND (SPECIMEN = 'ART'
  OR SPECIMEN_PROB > 0.75)
ORDER BY
  icustay_id,
  charttime
")

last_pao2fio2<-query_exec(query, project = project_HST,use_legacy_sql = F,max_pages =
Inf)
```

Filter

```{r}
last_pao2fio2<-
last_pao2fio2%>%select(subject_id,hadm_id,icustay_id,charttime,PaO2FiO2)

last_pao2fio2<-last_pao2fio2[complete.cases(last_pao2fio2),]

last_pao2fio2_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex
tubation','deathtime')], last_pao2fio2)

```

```

last_pao2fio2_from_patients_tofilter<-
dplyr::rename(last_pao2fio2_from_patients_tofilter,charftime_last_pao2fio2=charttime)

# we noticed some timezones are alteterd, so we reset them
last_pao2fio2_from_patients_tofilter$charttime_last_pao2fio2<-
as.character(last_pao2fio2_from_patients_tofilter$charttime_last_pao2fio2)
last_pao2fio2_from_patients_tofilter$charttime.extubation<-
as.character(last_pao2fio2_from_patients_tofilter$charttime.extubation)
last_pao2fio2_from_patients_tofilter$deathtime<-
as.character(last_pao2fio2_from_patients_tofilter$deathtime)

last_pao2fio2_from_patients_tofilter$charttime_last_pao2fio2<-
as.POSIXct(last_pao2fio2_from_patients_tofilter$charttime_last_pao2fio2, format='%Y-
%m-%d %H:%M:%S', tz="GMT")
last_pao2fio2_from_patients_tofilter$charttime.extubation<-
as.POSIXct(last_pao2fio2_from_patients_tofilter$charttime.extubation, format='%Y-%m-
%d %H:%M:%S', tz="GMT")
last_pao2fio2_from_patients_tofilter$deathtime<-
as.POSIXct(last_pao2fio2_from_patients_tofilter$deathtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

last_pao2fio2_from_patients_tofilter['time_from_last_pao2fio2_to_ext_min']<-
difftime(last_pao2fio2_from_patients_tofilter$charttime.extubation,last_pao2fio2_from
_patients_tofilter$charttime_last_pao2fio2,units = 'mins')

last_pao2fio2_patients<-last_pao2fio2_from_patients_tofilter %>%
  filter(
    time_from_last_pao2fio2_to_ext_min >=0
  )

n_distinct(last_pao2fio2_patients$subject_id)

last_pao2fio2_patients<-last_pao2fio2_patients%>%arrange(subject_id,
time_from_last_pao2fio2_to_ext_min)

last_pao2fio2_patients_final<-
last_pao2fio2_patients[!duplicated(last_pao2fio2_patients$subject_id),]
last_pao2fio2_patients_final<-
last_pao2fio2_patients_final%>%select(subject_id,time_from_last_pao2fio2_to_ext_min,P
aO2FiO2)
```

Environment

```{r}
library(summarytools)
library(bigrquery)
#library(RPostgreSQL)
library(DBI)
library(reshape)
library(dplyr)
```

```

```

```{r}
library(bigrquery)
project_id <- "hst-953-2018"
options(httr_oauth_cache=FALSE)
run_query <- function(query){
  data <- query_exec(query, project=project_id, use_legacy_sql=FALSE,max_pages = Inf)
  return(data)
}
```

POSTITIVE EPOCHS

propofol

Infusion Drugs

Data extraction

```{r eval=FALSE, include=FALSE}
query<-"
SELECT
  inpuvents_mv.subject_id,
  inpuvents_mv.hadm_id,
  inpuvents_mv.icustay_id,
  inpuvents_mv.itemid,
  inpuvents_mv.starttime,
  inpuvents_mv.endtime,
  inpuvents_mv.rate,
  inpuvents_mv.rateuom,
  inpuvents_mv.ordercategorydescription
From
  `physionet-data.mimiciii_clinical.inpuvents_mv` inpuvents_mv

WHERE
  inpuvents_mv.itemid IN (222168)
  AND ordercategorydescription = 'Continuous Med'
"

propofol_infusion_cohort_24<-query_exec(query, project = project_HST,use_legacy_sql =
F,max_pages = Inf)

propofol_infusion_cohort_24$itemid[propofol_infusion_cohort_24$itemid==222168]<-
'propofol'
#Valium (Diazepam)
#We tried valium and it didn't increase the number of patients, still 84 after ext
```

Epochs Creation (max rate mg/h)

```{r eval=FALSE, include=FALSE}
propofol_infusion_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex
tubation','deathtime')], propofol_infusion_cohort_24)

```

```

# we noticed some timezones ad drugs_infusion alteterd, so we drugs_infusionset them
propofol_infusion_from_patients_tofilter$starttime<-
as.character(propofol_infusion_from_patients_tofilter$starttime)
propofol_infusion_from_patients_tofilter$endtime<-
as.character(propofol_infusion_from_patients_tofilter$endtime)
propofol_infusion_from_patients_tofilter$charttime.extubation<-
as.character(propofol_infusion_from_patients_tofilter$charttime.extubation)
propofol_infusion_from_patients_tofilter$deathtime<-
as.character(propofol_infusion_from_patients_tofilter$deathtime)

propofol_infusion_from_patients_tofilter$starttime<-
as.POSIXct(propofol_infusion_from_patients_tofilter$starttime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
propofol_infusion_from_patients_tofilter$endtime<-
as.POSIXct(propofol_infusion_from_patients_tofilter$endtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
propofol_infusion_from_patients_tofilter$charttime.extubation<-
as.POSIXct(propofol_infusion_from_patients_tofilter$charttime.extubation, format='%Y-
%m-%d %H:%M:%S', tz="GMT")
propofol_infusion_from_patients_tofilter$deathtime<-
as.POSIXct(propofol_infusion_from_patients_tofilter$deathtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

propofol_infusion_from_patients_tofilter['time_from_ext_to_propofol_starttime_min']<-
difftime(propofol_infusion_from_patients_tofilter$starttime,propofol_infusion_from_pa
tients_tofilter$charttime.extubation,units = 'mins')

propofol_infusion_from_patients_tofilter['time_from_ext_to_propofol_endtime_min']<-
difftime(propofol_infusion_from_patients_tofilter$endtime,propofol_infusion_from_pati
ents_tofilter$charttime.extubation,units = 'mins')

#Drug Push is Bolus, so I filtered them out
propofol_infusion_after_ext<-propofol_infusion_from_patients_tofilter %>%
  filter(
    time_from_ext_to_propofol_starttime_min >= -30
    |
    time_from_ext_to_propofol_endtime_min >= -30
  )

n_distinct(propofol_infusion_after_ext$subject_id)

library(dplyr)
## if(any(gdrugs_infusionpl("package:plyr", search())) detach("package:plyr") else
message("plyr not loaded")
cohort_with_propofol_infusion_epoch<-propofol_infusion_after_ext %>%
  mutate(
    propofol_infusion_epoch_01=if_else(time_from_ext_to_propofol_starttime_min >= -30
& time_from_ext_to_propofol_starttime_min<= 90
    | time_from_ext_to_propofol_endtime_min >= -30
& time_from_ext_to_propofol_endtime_min<= 90
    ,rate,NULL)

```

```

        ,propofol_infusion_epoch_02=if_else(time_from_ext_to_propofol_starttime_min >90 &
time_from_ext_to_propofol_starttime_min<=150
        | time_from_ext_to_propofol_endtime_min >90 &
time_from_ext_to_propofol_endtime_min <= 150
        ,rate,NULL)
    )

```

```

cohort_with_propofol_infusion_epoch<-cohort_with_propofol_infusion_epoch%>%
  select(subject_id,propofol_infusion_epoch_01,propofol_infusion_epoch_02)%>%
  group_by(subject_id)%>%
  #summarize(mean(value,na.rm=TRUE))
  summarise_all(my.max)
...

```

Bolus Drugs

Data extraction

```

```{r eval=FALSE, include=FALSE}
query<-"
SELECT
  inputevents_mv.subject_id,
  inputevents_mv.hadm_id,
  inputevents_mv.icustay_id,
  inputevents_mv.itemid,
  COALESCE(inputevents_mv.starttime,inputevents_mv.storetime) AS
charttime_drugs_bolus,
  inputevents_mv.amount,
  inputevents_mv.amountuom,
  inputevents_mv.ordercategorydescription
From
  `physionet-data.mimiciii_clinical.inputevents_mv` inputevents_mv

```

```

WHERE
  inputevents_mv.itemid In (222168)
  AND ordercategorydescription = 'Drug Push' "

```

```

propofol_bolus_cohort_24<-query_exec(query, project = project_HST,use_legacy_sql =
F,max_pages = Inf)

```

```

propofol_bolus_cohort_24$itemid[propofol_bolus_cohort_24$itemid==222168]<-'propofol'
...

```

Epochs Creation (total amount mg)

```

```{r eval=FALSE, include=FALSE}
library(dplyr)
propofol_bolus_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex
tubation','deathtime')], propofol_bolus_cohort_24)

```

```

# we noticed some timezones adrugs_bolus alteterd, so we drugs_bolusset them
propofol_bolus_from_patients_tofilter$charttime_drugs_bolus<-
as.character(propofol_bolus_from_patients_tofilter$charttime_drugs_bolus)

```

```

propofol_bolus_from_patients_tofilter$charttime.extubation<-
as.character(propofol_bolus_from_patients_tofilter$charttime.extubation)
propofol_bolus_from_patients_tofilter$deathtime<-
as.character(propofol_bolus_from_patients_tofilter$deathtime)

propofol_bolus_from_patients_tofilter$charttime_drugs_bolus<-
as.POSIXct(propofol_bolus_from_patients_tofilter$charttime_drugs_bolus, format='%Y-
%m-%d %H:%M:%S', tz="GMT")
propofol_bolus_from_patients_tofilter$charttime.extubation<-
as.POSIXct(propofol_bolus_from_patients_tofilter$charttime.extubation, format='%Y-%m-
%d %H:%M:%S', tz="GMT")
propofol_bolus_from_patients_tofilter$deathtime<-
as.POSIXct(propofol_bolus_from_patients_tofilter$deathtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

propofol_bolus_from_patients_tofilter['time_from_ext_to_propofol_bolus_min']<-
difftime(propofol_bolus_from_patients_tofilter$charttime_drugs_bolus,propofol_bolus_f
rom_patients_tofilter$charttime.extubation,units = 'mins')

#Drug Push is Bolus, so I filtered them out
propofol_bolus_after_ext<-propofol_bolus_from_patients_tofilter %>%
  filter(
    time_from_ext_to_propofol_bolus_min >=-30
  )

n_distinct(propofol_bolus_after_ext$subject_id)
# 355 patients with time_from_ext_to_propofol_bolus_min >=-2*60 1/7/18
# 322 patients with time_from_ext_to_propofol_bolus_min >=-30 2/12/19

library(dplyr)
## if(any(gdrugs_boluspl("package:plyr", search())) detach("package:plyr") else
message("plyr not loaded")
cohort_with_propofol_bolus_epoch<-propofol_bolus_after_ext %>%
  mutate(
    propofol_bolus_epoch_01=if_else(time_from_ext_to_propofol_bolus_min>=-30 &
time_from_ext_to_propofol_bolus_min<=90,amount,NULL)
,propofol_bolus_epoch_02=if_else(time_from_ext_to_propofol_bolus_min>90 &
time_from_ext_to_propofol_bolus_min<=150,amount,NULL)
  )

my.sum <- function(x) ifelse( !all(is.na(x)), sum(x, na.rm=T), NA)

cohort_with_propofol_bolus_epoch<-cohort_with_propofol_bolus_epoch%>%
  select(subject_id,propofol_bolus_epoch_01,propofol_bolus_epoch_02)%>%
  group_by(subject_id)%>%
  #summarize(mean(value,na.rm=TRUE))
  summarise_all(my.sum)
...

## diazepam

```


Infusion Drugs

Data extraction

There are only Drug Push of diazepam, not continuous infusions.

```
```{r eval=FALSE, include=FALSE}
```

```
query<-"
```

```
SELECT
```

```
  inputevents_mv.subject_id,  
  inputevents_mv.hadm_id,  
  inputevents_mv.icustay_id,  
  inputevents_mv.itemid,  
  inputevents_mv.starttime,  
  inputevents_mv.endtime,  
  inputevents_mv.rate,  
  inputevents_mv.rateuom,  
  inputevents_mv.ordercategorydescription
```

```
From
```

```
  `physionet-data.mimiciii_clinical.inputevents_mv` inputevents_mv
```

```
WHERE
```

```
  inputevents_mv.itemid IN (221623)  
  AND ordercategorydescription = 'Continuous Med'  
"
```

```
diazepam_infusion_cohort_24<-query_exec(query, project = project_HST,use_legacy_sql =  
F,max_pages = Inf)
```

```
diazepam_infusion_cohort_24$itemid[diazepam_infusion_cohort_24$itemid==221623]<-  
'diazepam'
```

```
#Valium (Diazepam)
```

```
#We tried valium and it didn't increase the number of patients, still 84 after ext
```

```
```
```

##### Epochs Creation (max rate mg/h)

```
```{r eval=FALSE, include=FALSE}
```

```
diazepam_infusion_from_patients_tofilter<-
```

```
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex  
tubation','deathtime')], diazepam_infusion_cohort_24)
```

```
# we noticed some timezones adrug_infusion alteterd, so we drugs_infusionset them
```

```
diazepam_infusion_from_patients_tofilter$starttime<-
```

```
as.character(diazepam_infusion_from_patients_tofilter$starttime)
```

```
diazepam_infusion_from_patients_tofilter$endtime<-
```

```
as.character(diazepam_infusion_from_patients_tofilter$endtime)
```

```
diazepam_infusion_from_patients_tofilter$charttime.extubation<-
```

```
as.character(diazepam_infusion_from_patients_tofilter$charttime.extubation)
```

```
diazepam_infusion_from_patients_tofilter$deathtime<-
```

```
as.character(diazepam_infusion_from_patients_tofilter$deathtime)
```

```

diazepam_infusion_from_patients_tofilter$starttime<-
as.POSIXct(diazepam_infusion_from_patients_tofilter$starttime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
diazepam_infusion_from_patients_tofilter$endtime<-
as.POSIXct(diazepam_infusion_from_patients_tofilter$endtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
diazepam_infusion_from_patients_tofilter$charttime.extubation<-
as.POSIXct(diazepam_infusion_from_patients_tofilter$charttime.extubation, format='%Y-
%m-%d %H:%M:%S', tz="GMT")
diazepam_infusion_from_patients_tofilter$deathtime<-
as.POSIXct(diazepam_infusion_from_patients_tofilter$deathtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

diazepam_infusion_from_patients_tofilter['time_from_ext_to_diazepam_starttime_min']<-
difftime(diazepam_infusion_from_patients_tofilter$starttime,diazepam_infusion_from_pa
tients_tofilter$charttime.extubation,units = 'mins')

diazepam_infusion_from_patients_tofilter['time_from_ext_to_diazepam_endtime_min']<-
difftime(diazepam_infusion_from_patients_tofilter$starttime,diazepam_infusion_from_pa
tients_tofilter$charttime.extubation,units = 'mins')

#Drug Push is Bolus, so I filtered them out
diazepam_infusion_after_ext<-diazepam_infusion_from_patients_tofilter %>%
  filter(
    time_from_ext_to_diazepam_starttime_min >=-30
    |
    time_from_ext_to_diazepam_endtime_min >=-30
  )

n_distinct(diazepam_infusion_after_ext$subject_id)

library(dplyr)
## if(any(gdrugs_infusionpl("package:plyr", search())) detach("package:plyr") else
message("plyr not loaded")
cohort_with_diazepam_infusion_epoch<-diazepam_infusion_after_ext %>%
  mutate(
    diazepam_infusion_epoch_01=if_else(time_from_ext_to_diazepam_starttime_min>=-30 &
time_from_ext_to_diazepam_starttime_min<=90
    |
    time_from_ext_to_diazepam_endtime_min>=-30 &
time_from_ext_to_diazepam_endtime_min<=90
    ,rate,NULL)
    ,diazepam_infusion_epoch_02=if_else(time_from_ext_to_diazepam_starttime_min>90 &
time_from_ext_to_diazepam_starttime_min<=150
    |
    time_from_ext_to_diazepam_endtime_min>90 &
time_from_ext_to_diazepam_endtime_min<=150
    ,rate,NULL)
  )

cohort_with_diazepam_infusion_epoch<-cohort_with_diazepam_infusion_epoch%>%

```

```

    select(subject_id,diazepam_infusion_epoch_01,diazepam_infusion_epoch_02)%>%
    group_by(subject_id)%>%
    #summarize(mean(value,na.rm=TRUE))
    summarise_all(my.max)
  ```

Bolus Drugs

Data extraction

```{r eval=FALSE, include=FALSE}
query<-"
SELECT
  inpatientevents_mv.subject_id,
  inpatientevents_mv.hadm_id,
  inpatientevents_mv.icustay_id,
  inpatientevents_mv.itemid,
  COALESCE(inpatientevents_mv.starttime,inpatientevents_mv.storetime) AS
charttime_drugs_bolus,
  inpatientevents_mv.amount,
  inpatientevents_mv.amountuom,
  inpatientevents_mv.ordercategorydescription
From
  `physionet-data.mimiciii_clinical.inpatientevents_mv` inpatientevents_mv

WHERE
  inpatientevents_mv.itemid In (221623)
  AND ordercategorydescription = 'Drug Push' "

diazepam_bolus_cohort_24<-query_exec(query, project = project_HST,use_legacy_sql =
F,max_pages = Inf)

diazepam_bolus_cohort_24$itemid[diazepam_bolus_cohort_24$itemid==221623]<-'diazepam'
```

Epochs Creation (total amount mg)

```{r eval=FALSE, include=FALSE}
library(dplyr)
diazepam_bolus_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex
tubation','deathtime')], diazepam_bolus_cohort_24)

# we noticed some timezones adrug_bolus alteterd, so we drug_bolusset them
diazepam_bolus_from_patients_tofilter$charttime_drugs_bolus<-
as.character(diazepam_bolus_from_patients_tofilter$charttime_drugs_bolus)
diazepam_bolus_from_patients_tofilter$charttime_extubation<-
as.character(diazepam_bolus_from_patients_tofilter$charttime_extubation)
diazepam_bolus_from_patients_tofilter$deathtime<-
as.character(diazepam_bolus_from_patients_tofilter$deathtime)

diazepam_bolus_from_patients_tofilter$charttime_drugs_bolus<-
as.POSIXct(diazepam_bolus_from_patients_tofilter$charttime_drugs_bolus, format='%Y-
%m-%d %H:%M:%S', tz="GMT")

```

```

diazepam_bolus_from_patients_tofilter$charttime.extubation<-
as.POSIXct(diazepam_bolus_from_patients_tofilter$charttime.extubation, format='%Y-%m-
%d %H:%M:%S', tz="GMT")
diazepam_bolus_from_patients_tofilter$deathtime<-
as.POSIXct(diazepam_bolus_from_patients_tofilter$deathtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

diazepam_bolus_from_patients_tofilter['time_from_ext_to_diazepam_bolus_min']<-
difftime(diazepam_bolus_from_patients_tofilter$charttime_drugs_bolus,diazepam_bolus_f
rom_patients_tofilter$charttime.extubation,units = 'mins')

#Drug Push is Bolus, so I filtered them out
diazepam_bolus_after_ext<-diazepam_bolus_from_patients_tofilter %>%
  filter(
    time_from_ext_to_diazepam_bolus_min >=-30
  )

n_distinct(diazepam_bolus_after_ext$subject_id)
# 355 patients with time_from_ext_to_diazepam_bolus_min >=-2*60 1/7/18
# 322 patients with time_from_ext_to_diazepam_bolus_min >=-30 2/12/19

library(dplyr)
## if(any(gdrugs_boluspl("package:plyr", search())) detach("package:plyr") else
message("plyr not loaded")
cohort_with_diazepam_bolus_epoch<-diazepam_bolus_after_ext %>%
  mutate(
    diazepam_bolus_epoch_01=if_else(time_from_ext_to_diazepam_bolus_min>=-30 &
time_from_ext_to_diazepam_bolus_min<=90,amount,NULL)
,diazepam_bolus_epoch_02=if_else(time_from_ext_to_diazepam_bolus_min>90 &
time_from_ext_to_diazepam_bolus_min<=150,amount,NULL)
  )

my.sum <- function(x) ifelse( !all(is.na(x)), sum(x, na.rm=T), NA)

cohort_with_diazepam_bolus_epoch<-cohort_with_diazepam_bolus_epoch%>%
  select(subject_id,diazepam_bolus_epoch_01,diazepam_bolus_epoch_02)%>%
  group_by(subject_id)%>%
  #summarize(mean(value,na.rm=TRUE))
  summarise_all(my.sum)
...

# midazolam

## Infusion Drugs

### Data extraction

```{r eval=FALSE, include=FALSE}
query<-"
SELECT
  inputevents_mv.subject_id,

```

```

    inputevents_mv.hadm_id,
    inputevents_mv.icustay_id,
    inputevents_mv.itemid,
    inputevents_mv.starttime,
    inputevents_mv.endtime,
    inputevents_mv.rate,
    inputevents_mv.rateuom,
    inputevents_mv.ordercategorydescription
From
  `physionet-data.mimiciii_clinical.inputevents_mv` inputevents_mv

WHERE
  inputevents_mv.itemid IN (221668)
  AND ordercategorydescription = 'Continuous Med'
"
```

```

midazolam_infusion_cohort_24<-query_exec(query, project = project_HST,use_legacy_sql
= F,max_pages = Inf)

```

```

midazolam_infusion_cohort_24$itemid[midazolam_infusion_cohort_24$itemid==221668]<-
'midazolam'
```

```

```

Epochs Creation (max rate mg/h)

```

```

```{r eval=FALSE, include=FALSE}
midazolam_infusion_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex
tubation','deathtime')], midazolam_infusion_cohort_24)

```

```

# we noticed some timezones adrug_infusion alteterd, so we drug_infusionset them
midazolam_infusion_from_patients_tofilter$starttime<-
as.character(midazolam_infusion_from_patients_tofilter$starttime)
midazolam_infusion_from_patients_tofilter$endtime<-
as.character(midazolam_infusion_from_patients_tofilter$endtime)
midazolam_infusion_from_patients_tofilter$charttime.extubation<-
as.character(midazolam_infusion_from_patients_tofilter$charttime.extubation)
midazolam_infusion_from_patients_tofilter$deathtime<-
as.character(midazolam_infusion_from_patients_tofilter$deathtime)

```

```

midazolam_infusion_from_patients_tofilter$starttime<-
as.POSIXct(midazolam_infusion_from_patients_tofilter$starttime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
midazolam_infusion_from_patients_tofilter$endtime<-
as.POSIXct(midazolam_infusion_from_patients_tofilter$endtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
midazolam_infusion_from_patients_tofilter$charttime.extubation<-
as.POSIXct(midazolam_infusion_from_patients_tofilter$charttime.extubation,
format='%Y-%m-%d %H:%M:%S', tz="GMT")
midazolam_infusion_from_patients_tofilter$deathtime<-
as.POSIXct(midazolam_infusion_from_patients_tofilter$deathtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

```

```

midazolam_infusion_from_patients_tofilter['time_from_ext_to_midazolam_starttime_min']
<-
difftime(midazolam_infusion_from_patients_tofilter$starttime,midazolam_infusion_from_
patients_tofilter$charttime.extubation,units = 'mins')
midazolam_infusion_from_patients_tofilter['time_from_ext_to_midazolam_endtime_min']<-
difftime(midazolam_infusion_from_patients_tofilter$endtime,midazolam_infusion_from_pa
tients_tofilter$charttime.extubation,units = 'mins')

#Drug Push is Bolus, so I filtered them out
midazolam_infusion_after_ext<-midazolam_infusion_from_patients_tofilter %>%
  filter(
    time_from_ext_to_midazolam_starttime_min >=-30
    |
    time_from_ext_to_midazolam_endtime_min >=-30
  )

n_distinct(midazolam_infusion_after_ext$subject_id)

library(dplyr)
## if(any(gdrugs_infusionpl("package:plyr", search())) detach("package:plyr") else
message("plyr not loaded")
cohort_with_midazolam_infusion_epoch<-midazolam_infusion_after_ext %>%
  mutate(
    midazolam_infusion_epoch_01=if_else(
      time_from_ext_to_midazolam_starttime_min>=-30 &
time_from_ext_to_midazolam_starttime_min<=90
      |
      time_from_ext_to_midazolam_endtime_min>=-30 &
time_from_ext_to_midazolam_endtime_min<=90
      ,rate,NULL)
    ,midazolam_infusion_epoch_02=if_else(
      time_from_ext_to_midazolam_starttime_min> 90 &
time_from_ext_to_midazolam_starttime_min<=150
      |
      time_from_ext_to_midazolam_endtime_min>90 &
time_from_ext_to_midazolam_endtime_min<=150
      ,rate,NULL)
  )

cohort_with_midazolam_infusion_epoch<-cohort_with_midazolam_infusion_epoch%>%
  select(subject_id,midazolam_infusion_epoch_01,midazolam_infusion_epoch_02)%>%
  group_by(subject_id)%>%
  #summarize(mean(value,na.rm=TRUE))
  summarise_all(my.max)
...

## Bolus Drugs

### Data extraction

```{r eval=FALSE, include=FALSE}

```

```

query<-"
SELECT
  inpuvents_mv.subject_id,
  inpuvents_mv.hadm_id,
  inpuvents_mv.icustay_id,
  inpuvents_mv.itemid,
  COALESCE(inpuvents_mv.starttime,inpuvents_mv.storetime) AS
charttime_drugs_bolus,
  inpuvents_mv.amount,
  inpuvents_mv.amountuom,
  inpuvents_mv.ordercategorydescription
From
  `physionet-data.mimiciii_clinical.inpuvents_mv` inpuvents_mv

WHERE
  inpuvents_mv.itemid In (221668)
  AND ordercategorydescription = 'Drug Push' "

midazolam_bolus_cohort_24<-query_exec(query, project = project_HST,use_legacy_sql =
F,max_pages = Inf)

midazolam_bolus_cohort_24$itemid[midazolam_bolus_cohort_24$itemid==221668]<-
'midazolam'
```

Epochs Creation (total amount mg)

```
```{r eval=FALSE, include=FALSE}
library(dplyr)
midazolam_bolus_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex
tubation','deathtime')], midazolam_bolus_cohort_24)

# we noticed some timezones adrugs_bolus alteterd, so we drugs_bolusset them
midazolam_bolus_from_patients_tofilter$charttime_drugs_bolus<-
as.character(midazolam_bolus_from_patients_tofilter$charttime_drugs_bolus)
midazolam_bolus_from_patients_tofilter$charttime.extubation<-
as.character(midazolam_bolus_from_patients_tofilter$charttime.extubation)
midazolam_bolus_from_patients_tofilter$deathtime<-
as.character(midazolam_bolus_from_patients_tofilter$deathtime)

midazolam_bolus_from_patients_tofilter$charttime_drugs_bolus<-
as.POSIXct(midazolam_bolus_from_patients_tofilter$charttime_drugs_bolus, format='%Y-
%m-%d %H:%M:%S', tz="GMT")
midazolam_bolus_from_patients_tofilter$charttime.extubation<-
as.POSIXct(midazolam_bolus_from_patients_tofilter$charttime.extubation, format='%Y-
%m-%d %H:%M:%S', tz="GMT")
midazolam_bolus_from_patients_tofilter$deathtime<-
as.POSIXct(midazolam_bolus_from_patients_tofilter$deathtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

midazolam_bolus_from_patients_tofilter['time_from_ext_to_midazolam_bolus_min']<-
difftime(midazolam_bolus_from_patients_tofilter$charttime_drugs_bolus,midazolam_bolus
_from_patients_tofilter$charttime.extubation,units = 'mins')

```

```

#Drug Push is Bolus, so I filtered them out
midazolam_bolus_after_ext<-midazolam_bolus_from_patients_tofilter %>%
  filter(
    time_from_ext_to_midazolam_bolus_min >=-30
  )

n_distinct(midazolam_bolus_after_ext$subject_id)
# 355 patients with time_from_ext_to_midazolam_bolus_min >=-2*60 1/7/18
# 322 patients with time_from_ext_to_midazolam_bolus_min >=-30 2/12/19

library(dplyr)
## if(any(gdrugs_boluspl("package:plyr", search())) detach("package:plyr") else
message("plyr not loaded")
cohort_with_midazolam_bolus_epoch<-midazolam_bolus_after_ext %>%
  mutate(
    midazolam_bolus_epoch_01=if_else(time_from_ext_to_midazolam_bolus_min>=-30 &
time_from_ext_to_midazolam_bolus_min<=90,amount,NULL)
,midazolam_bolus_epoch_02=if_else(time_from_ext_to_midazolam_bolus_min>90 &
time_from_ext_to_midazolam_bolus_min<=150,amount,NULL)
  )

my.sum <- function(x) ifelse( !all(is.na(x)), sum(x, na.rm=T), NA)

cohort_with_midazolam_bolus_epoch<-cohort_with_midazolam_bolus_epoch%>%
  select(subject_id,midazolam_bolus_epoch_01,midazolam_bolus_epoch_02)%>%
  group_by(subject_id)%>%
  #summarize(mean(value,na.rm=TRUE))
  summarise_all(my.sum)
``

## lorazepam

### Infusion Drugs

#### Data extraction

``{r eval=FALSE, include=FALSE}
query<-"
SELECT
  inpuvents_mv.subject_id,
  inpuvents_mv.hadm_id,
  inpuvents_mv.icustay_id,
  inpuvents_mv.itemid,
  inpuvents_mv.starttime,
  inpuvents_mv.endtime,
  inpuvents_mv.rate,
  inpuvents_mv.rateuom,
  inpuvents_mv.ordercategorydescription
From
`physionet-data.mimiciii_clinical.inpuvents_mv` inpuvents_mv

```



```

WHERE
  inpuतेvents_mv.itemid IN (221385)
  AND ordercategorydescription = 'Continuous Med'
"

lorazepam_infusion_cohort_24<-query_exec(query, project = project_HST,use_legacy_sql
= F,max_pages = Inf)

lorazepam_infusion_cohort_24$itemid[lorazepam_infusion_cohort_24$itemid==221385]<-
'lorazepam'
```

Epochs Creation (max rate mg/h)

```r eval=FALSE, include=FALSE}
lorazepam_infusion_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex
tubation','deathtime')], lorazepam_infusion_cohort_24)

# we noticed some timezones adrugs_infusion alteterd, so we drugs_infusionset them
lorazepam_infusion_from_patients_tofilter$starttime<-
as.character(lorazepam_infusion_from_patients_tofilter$starttime)
lorazepam_infusion_from_patients_tofilter$endtime<-
as.character(lorazepam_infusion_from_patients_tofilter$endtime)
lorazepam_infusion_from_patients_tofilter$charttime.extubation<-
as.character(lorazepam_infusion_from_patients_tofilter$charttime.extubation)
lorazepam_infusion_from_patients_tofilter$deathtime<-
as.character(lorazepam_infusion_from_patients_tofilter$deathtime)

lorazepam_infusion_from_patients_tofilter$starttime<-
as.POSIXct(lorazepam_infusion_from_patients_tofilter$starttime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
lorazepam_infusion_from_patients_tofilter$endtime<-
as.POSIXct(lorazepam_infusion_from_patients_tofilter$endtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
lorazepam_infusion_from_patients_tofilter$charttime.extubation<-
as.POSIXct(lorazepam_infusion_from_patients_tofilter$charttime.extubation,
format='%Y-%m-%d %H:%M:%S', tz="GMT")
lorazepam_infusion_from_patients_tofilter$deathtime<-
as.POSIXct(lorazepam_infusion_from_patients_tofilter$deathtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

lorazepam_infusion_from_patients_tofilter['time_from_ext_to_lorazepam_starttime_min']
<-
difftime(lorazepam_infusion_from_patients_tofilter$starttime,lorazepam_infusion_from
patients_tofilter$charttime.extubation,units = 'mins')
lorazepam_infusion_from_patients_tofilter['time_from_ext_to_lorazepam_endtime_min']<-
difftime(lorazepam_infusion_from_patients_tofilter$endtime,lorazepam_infusion_from_pa
tients_tofilter$charttime.extubation,units = 'mins')

#Drug Push is Bolus, so I filtered them out
lorazepam_infusion_after_ext<-lorazepam_infusion_from_patients_tofilter %>%

```

```

filter(
  time_from_ext_to_lorazepam_starttime_min >=-30
  |
  time_from_ext_to_lorazepam_endtime_min >=-30
)

n_distinct(lorazepam_infusion_after_ext$subject_id)

library(dplyr)
## if(any(gdrugs_infusionpl("package:plyr", search())) detach("package:plyr") else
message("plyr not loaded")
cohort_with_lorazepam_infusion_epoch<-lorazepam_infusion_after_ext %>%
  mutate(
    lorazepam_infusion_epoch_01=if_else(
      time_from_ext_to_lorazepam_starttime_min>=-30 &
time_from_ext_to_lorazepam_starttime_min<=90
      |
      time_from_ext_to_lorazepam_endtime_min>=-30 &
time_from_ext_to_lorazepam_endtime_min<=90
      ,rate,NULL)
    ,lorazepam_infusion_epoch_02=if_else(
      time_from_ext_to_lorazepam_starttime_min>90 &
time_from_ext_to_lorazepam_starttime_min<=150
      |
      time_from_ext_to_lorazepam_endtime_min>90 &
time_from_ext_to_lorazepam_endtime_min<=150
      ,rate,NULL)
  )

cohort_with_lorazepam_infusion_epoch<-cohort_with_lorazepam_infusion_epoch%>%
  select(subject_id,lorazepam_infusion_epoch_01,lorazepam_infusion_epoch_02)%>%
  group_by(subject_id)%>%
  #summarize(mean(value,na.rm=TRUE))
  summarise_all(my.max)
...

### Bolus Drugs

##### Data extraction

```{r eval=FALSE, include=FALSE}
query<-"
SELECT
  inpatientevents_mv.subject_id,
  inpatientevents_mv.hadm_id,
  inpatientevents_mv.icustay_id,
  inpatientevents_mv.itemid,
  inpatientevents_mv.starttime,
  inpatientevents_mv.endtime,
  inpatientevents_mv.amount,
  inpatientevents_mv.amountuom,
  inpatientevents_mv.ordercategorydescription
From

```

```

`physionet-data.mimiciii_clinical.inpatevents_mv` inpatevents_mv

WHERE
  inpatevents_mv.itemid In (221385)
  AND ordercategorydescription = 'Drug Push' "

lorazepam_bolus_cohort_24<-query_exec(query, project = project_HST,use_legacy_sql =
F,max_pages = Inf)

lorazepam_bolus_cohort_24$itemid[lorazepam_bolus_cohort_24$itemid==221385]<-
'lorazepam'
```

Epochs Creation (total amount mg)

```{r eval=FALSE, include=FALSE}
library(dplyr)
lorazepam_bolus_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex
tubation','deathtime')], lorazepam_bolus_cohort_24)

# we noticed some timezones adrugs_bolus alteterd, so we drugs_bolusset them
lorazepam_bolus_from_patients_tofilter$charttime_drugs_bolus<-
as.character(lorazepam_bolus_from_patients_tofilter$charttime_drugs_bolus)
lorazepam_bolus_from_patients_tofilter$charttime_extubation<-
as.character(lorazepam_bolus_from_patients_tofilter$charttime_extubation)
lorazepam_bolus_from_patients_tofilter$deathtime<-
as.character(lorazepam_bolus_from_patients_tofilter$deathtime)

lorazepam_bolus_from_patients_tofilter$charttime_drugs_bolus<-
as.POSIXct(lorazepam_bolus_from_patients_tofilter$charttime_drugs_bolus, format='%Y-
%m-%d %H:%M:%S', tz="GMT")
lorazepam_bolus_from_patients_tofilter$charttime_extubation<-
as.POSIXct(lorazepam_bolus_from_patients_tofilter$charttime_extubation, format='%Y-
%m-%d %H:%M:%S', tz="GMT")
lorazepam_bolus_from_patients_tofilter$deathtime<-
as.POSIXct(lorazepam_bolus_from_patients_tofilter$deathtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

lorazepam_bolus_from_patients_tofilter['time_from_ext_to_lorazepam_bolus_min']<-
difftime(lorazepam_bolus_from_patients_tofilter$charttime_drugs_bolus,lorazepam_bolus
_from_patients_tofilter$charttime_extubation,units = 'mins')

#Drug Push is Bolus, so I filtered them out
lorazepam_bolus_after_ext<-lorazepam_bolus_from_patients_tofilter %>%
  filter(
    time_from_ext_to_lorazepam_bolus_min >=-30
  )

# lorazepam = 1 milligram of morphine IV (225154) is equivalent to .01 mg Fentanyl
IV (221744).

n_distinct(lorazepam_bolus_after_ext$subject_id)

```

```

# 355 patients with time_from_ext_to_lorazepam_bolus_min >=-2*60 1/7/18
# 322 patients with time_from_ext_to_lorazepam_bolus_min >=-30 2/12/19

library(dplyr)
## if(any(gdrugs_boluspl("package:plyr", search())) detach("package:plyr") else
message("plyr not loaded")
cohort_with_lorazepam_bolus_epoch<-lorazepam_bolus_after_ext %>%
  mutate(
    lorazepam_bolus_epoch_01=if_else(time_from_ext_to_lorazepam_bolus_min>=-30 &
time_from_ext_to_lorazepam_bolus_min<=90,amount,NULL)
,lorazepam_bolus_epoch_02=if_else(time_from_ext_to_lorazepam_bolus_min>90 &
time_from_ext_to_lorazepam_bolus_min<=150,amount,NULL)
  )

my.sum <- function(x) ifelse( !all(is.na(x)), sum(x, na.rm=T), NA)

cohort_with_lorazepam_bolus_epoch<-cohort_with_lorazepam_bolus_epoch%>%
  select(subject_id,lorazepam_bolus_epoch_01,lorazepam_bolus_epoch_02)%>%
  group_by(subject_id)%>%
  #summarize(mean(value,na.rm=TRUE))
  summarise_all(my.sum)
```

dexmedetomidie

Infusion Drugs

Data extraction

```{r eval=FALSE, include=FALSE}
query<-"
SELECT
  inputevents_mv.subject_id,
  inputevents_mv.hadm_id,
  inputevents_mv.icustay_id,
  inputevents_mv.itemid,
  inputevents_mv.starttime,
  inputevents_mv.endtime,
  inputevents_mv.rate,
  inputevents_mv.rateuom,
  inputevents_mv.ordercategorydescription
From
  `physionet-data.mimiciii_clinical.inputevents_mv` inputevents_mv

WHERE
  inputevents_mv.itemid IN (225150)
  AND ordercategorydescription = 'Continuous Med'
"

dexmedetomidie_infusion_cohort_24<-query_exec(query, project =
project_HST,use_legacy_sql = F,max_pages = Inf)

dexmedetomidie_infusion_cohort_24$itemid[dexmedetomidie_infusion_cohort_24$itemid==22
5150]<-'dexmedetomidie'

```

```

#Valium (Diazepam)
#We tried valium and it didn't increase the number of patients, still 84 after ext
```

Epochs Creation (max rate mg/h)

```{r eval=FALSE, include=FALSE}
dexmedetomedie_infusion_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex
tubation','deathtime')], dexmedetomedie_infusion_cohort_24)

# we noticed some timezones adrugs_infusion alteterd, so we drugs_infusionset them
dexmedetomedie_infusion_from_patients_tofilter$starttime<-
as.character(dexmedetomedie_infusion_from_patients_tofilter$starttime)
dexmedetomedie_infusion_from_patients_tofilter$endtime<-
as.character(dexmedetomedie_infusion_from_patients_tofilter$endtime)
dexmedetomedie_infusion_from_patients_tofilter$charttime.extubation<-
as.character(dexmedetomedie_infusion_from_patients_tofilter$charttime.extubation)
dexmedetomedie_infusion_from_patients_tofilter$deathtime<-
as.character(dexmedetomedie_infusion_from_patients_tofilter$deathtime)

dexmedetomedie_infusion_from_patients_tofilter$starttime<-
as.POSIXct(dexmedetomedie_infusion_from_patients_tofilter$starttime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
dexmedetomedie_infusion_from_patients_tofilter$endtime<-
as.POSIXct(dexmedetomedie_infusion_from_patients_tofilter$endtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
dexmedetomedie_infusion_from_patients_tofilter$charttime.extubation<-
as.POSIXct(dexmedetomedie_infusion_from_patients_tofilter$charttime.extubation,
format='%Y-%m-%d %H:%M:%S', tz="GMT")
dexmedetomedie_infusion_from_patients_tofilter$deathtime<-
as.POSIXct(dexmedetomedie_infusion_from_patients_tofilter$deathtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

dexmedetomedie_infusion_from_patients_tofilter['time_from_ext_to_dexmedetomedie_start
time_min']<-
difftime(dexmedetomedie_infusion_from_patients_tofilter$starttime,dexmedetomedie_infu
sion_from_patients_tofilter$charttime.extubation,units = 'mins')

dexmedetomedie_infusion_from_patients_tofilter['time_from_ext_to_dexmedetomedie_endti
me_min']<-
difftime(dexmedetomedie_infusion_from_patients_tofilter$endtime,dexmedetomedie_infusi
on_from_patients_tofilter$charttime.extubation,units = 'mins')

#Drug Push is Bolus, so I filtered them out
dexmedetomedie_infusion_after_ext<-dexmedetomedie_infusion_from_patients_tofilter %>%
  filter(
    time_from_ext_to_dexmedetomedie_starttime_min >=-30
    |
    time_from_ext_to_dexmedetomedie_endtime_min >=-30  )

n_distinct(dexmedetomedie_infusion_after_ext$subject_id)

```

```

library(dplyr)
## if(any(gdrugs_infusionpl("package:plyr", search())) detach("package:plyr") else
message("plyr not loaded")
cohort_with_dexmedetomidie_infusion_epoch<-dexmedetomidie_infusion_after_ext %>%
  mutate(
    dexmedetomidie_infusion_epoch_01=if_else(
      time_from_ext_to_dexmedetomidie_starttime_min>=-30 &
time_from_ext_to_dexmedetomidie_starttime_min<=90
      |
      time_from_ext_to_dexmedetomidie_endtime_min>=-30 &
time_from_ext_to_dexmedetomidie_endtime_min<=90
      ,rate,NULL)
    ,dexmedetomidie_infusion_epoch_02=if_else(
      time_from_ext_to_dexmedetomidie_starttime_min>90 &
time_from_ext_to_dexmedetomidie_starttime_min<=150
      |
      time_from_ext_to_dexmedetomidie_endtime_min>90 &
time_from_ext_to_dexmedetomidie_endtime_min<=150
      ,rate,NULL)
    )

cohort_with_dexmedetomidie_infusion_epoch<-
cohort_with_dexmedetomidie_infusion_epoch%>%

select(subject_id,dexmedetomidie_infusion_epoch_01,dexmedetomidie_infusion_epoch_02)%
>%
  group_by(subject_id)%>%
  #summarize(mean(value,na.rm=TRUE))
  summarise_all(my.max)
...

### Bolus Drugs

##### Data extraction

```{r eval=FALSE, include=FALSE}
query<-"
SELECT
  inputevents_mv.subject_id,
  inputevents_mv.hadm_id,
  inputevents_mv.icustay_id,
  inputevents_mv.itemid,
  COALESCE(inputevents_mv.starttime,inputevents_mv.storetime) AS
charttime_drugs_bolus,
  inputevents_mv.amount,
  inputevents_mv.amountuom,
  inputevents_mv.ordercategorydescription
From
  `physionet-data.mimiciii_clinical.inputevents_mv` inputevents_mv

WHERE
  inputevents_mv.itemid In (225150)
  AND ordercategorydescription = 'Drug Push' "

```

```

dexmedetomedie_bolus_cohort_24<-query_exec(query, project =
project_HST,use_legacy_sql = F,max_pages = Inf)

dexmedetomedie_bolus_cohort_24$itemid[dexmedetomedie_bolus_cohort_24$itemid==225150]<
-'dexmedetomedie'
```

Epochs Creation (total amount mg)

```{r eval=FALSE, include=FALSE}
library(dplyr)
dexmedetomedie_bolus_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex
tubation','deathtime')], dexmedetomedie_bolus_cohort_24)

# we noticed some timezones adrugs_bolus alteterd, so we drugs_bolusset them
dexmedetomedie_bolus_from_patients_tofilter$charttime_drugs_bolus<-
as.character(dexmedetomedie_bolus_from_patients_tofilter$charttime_drugs_bolus)
dexmedetomedie_bolus_from_patients_tofilter$charttime_extubation<-
as.character(dexmedetomedie_bolus_from_patients_tofilter$charttime_extubation)
dexmedetomedie_bolus_from_patients_tofilter$deathtime<-
as.character(dexmedetomedie_bolus_from_patients_tofilter$deathtime)

dexmedetomedie_bolus_from_patients_tofilter$charttime_drugs_bolus<-
as.POSIXct(dexmedetomedie_bolus_from_patients_tofilter$charttime_drugs_bolus,
format='%Y-%m-%d %H:%M:%S', tz="GMT")
dexmedetomedie_bolus_from_patients_tofilter$charttime_extubation<-
as.POSIXct(dexmedetomedie_bolus_from_patients_tofilter$charttime_extubation,
format='%Y-%m-%d %H:%M:%S', tz="GMT")
dexmedetomedie_bolus_from_patients_tofilter$deathtime<-
as.POSIXct(dexmedetomedie_bolus_from_patients_tofilter$deathtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

dexmedetomedie_bolus_from_patients_tofilter['time_from_ext_to_dexmedetomedie_bolus_mi
n']<-
difftime(dexmedetomedie_bolus_from_patients_tofilter$charttime_drugs_bolus,dexmedetom
edie_bolus_from_patients_tofilter$charttime_extubation,units = 'mins')

#Drug Push is Bolus, so I filtered them out
dexmedetomedie_bolus_after_ext<-dexmedetomedie_bolus_from_patients_tofilter %>%
  filter(
    time_from_ext_to_dexmedetomedie_bolus_min >=-30
  )

n_distinct(dexmedetomedie_bolus_after_ext$subject_id)
# 355 patients with time_from_ext_to_dexmedetomedie_bolus_min >=-2*60 1/7/18
# 322 patients with time_from_ext_to_dexmedetomedie_bolus_min >=-30 2/12/19

library(dplyr)

```

```

## if(any(gdrugs_boluspl("package:plyr", search())) detach("package:plyr") else
message("plyr not loaded")
cohort_with_dexmedetomedie_bolus_epoch<-dexmedetomedie_bolus_after_ext %>%
  mutate(
    dexmedetomedie_bolus_epoch_01=if_else(time_from_ext_to_dexmedetomedie_bolus_min>=-30
& time_from_ext_to_dexmedetomedie_bolus_min<=90,amount,NULL)
,dexmedetomedie_bolus_epoch_02=if_else(time_from_ext_to_dexmedetomedie_bolus_min>90 &
time_from_ext_to_dexmedetomedie_bolus_min<=150,amount,NULL)
  )

my.sum <- function(x) ifelse( !all(is.na(x)), sum(x, na.rm=T), NA)

cohort_with_dexmedetomedie_bolus_epoch<-cohort_with_dexmedetomedie_bolus_epoch%>%
  select(subject_id,dexmedetomedie_bolus_epoch_01,dexmedetomedie_bolus_epoch_02)%>%
  group_by(subject_id)%>%
  #summarize(mean(value,na.rm=TRUE))
  summarise_all(my.sum)
` ``

# NEGATIVE EPOCHS

Please note the following drugs were processed differently:

- phenylephrine 221749
- norepinephrine 221906
- dopamine 221662
- epinephrine 221289
- vasopressin 222315

## propofol

### Infusion Drugs

#### Negative Epochs Creation (max rate mg/h)

``{r eval=FALSE, include=FALSE}

#Drug Push is Bolus, so I filtered them out
propofol_infusion_before_ext<-propofol_infusion_from_patients_tofilter %>%
  filter(
    time_from_ext_to_propofol_starttime_min >= -150
    |
    time_from_ext_to_propofol_endtime_min >= -150
  )

n_distinct(propofol_infusion_before_ext$subject_id)

library(dplyr)

cohort_with_propofol_infusion_neg_epoch<-propofol_infusion_before_ext %>%
  mutate(
    propofol_infusion_neg_epoch_01=if_else( between
(time_from_ext_to_propofol_starttime_min , -90, 30 )

```



```

                                |
                                between
(time_from_ext_to_propofol_endtime_min , -90, 30 )
                                ,rate,NULL)
,propofol_infusion_neg_epoch_02=if_else( between
(time_from_ext_to_propofol_starttime_min , -150, -90 )
                                |
                                between
(time_from_ext_to_propofol_endtime_min , -150, -90 )
                                ,rate,NULL)
)

cohort_with_propofol_infusion_neg_epoch<-cohort_with_propofol_infusion_neg_epoch%%
  select(subject_id,propofol_infusion_neg_epoch_01,propofol_infusion_neg_epoch_02)%>%
  group_by(subject_id)%>%
  summarise_all(my.max)
...

### Bolus Drugs

#### Negative Epochs Creation (total amount mg)

```{r eval=FALSE, include=FALSE}

#Drug Push is Bolus, so I filtered them out

propofol_bolus_before_ext<-propofol_bolus_from_patients_tofilter %>%
  filter(
    time_from_ext_to_propofol_bolus_min >= -150
  )

n_distinct(propofol_bolus_before_ext$subject_id) # 430 12/13/19

library(dplyr)

cohort_with_propofol_bolus_neg_epoch<-propofol_bolus_before_ext %>%
  mutate(
    propofol_bolus_neg_epoch_01=if_else(between(time_from_ext_to_propofol_bolus_min, -
90, 30) ,amount,NULL)
    ,propofol_bolus_neg_epoch_02=if_else(between(time_from_ext_to_propofol_bolus_min, -
150, -90) ,amount,NULL)
  )

my.sum <- function(x) ifelse( !all(is.na(x)), sum(x, na.rm=T), NA)

cohort_with_propofol_bolus_neg_epoch<-cohort_with_propofol_bolus_neg_epoch%%
  select(subject_id,propofol_bolus_neg_epoch_01,propofol_bolus_neg_epoch_02)%>%
  group_by(subject_id)%>%
  summarise_all(my.sum)
...

## diazepam

```

```

### Infusion Drugs

#### Negative Epochs Creation (max rate mg/h)

```{r eval=FALSE, include=FALSE}

#Drug Push is Bolus, so I filtered them out
diazepam_infusion_before_ext<-diazepam_infusion_from_patients_tofilter %>%
  filter(
    time_from_ext_to_diazepam_starttime_min >= -150
    |
    time_from_ext_to_diazepam_endtime_min >= -150
  )

n_distinct(diazepam_infusion_before_ext$subject_id)

library(dplyr)

cohort_with_diazepam_infusion_neg_epoch<-diazepam_infusion_before_ext %>%
  mutate(
    diazepam_infusion_neg_epoch_01=if_else(
      between (time_from_ext_to_diazepam_starttime_min , -90, 30 )
      |
      between (time_from_ext_to_diazepam_endtime_min , -90, 30 )
      ,rate,NULL)
    ,diazepam_infusion_neg_epoch_02=if_else(
      between (time_from_ext_to_diazepam_starttime_min ,-150, -90 )
      |
      between (time_from_ext_to_diazepam_endtime_min ,-150, -90 )
      ,rate,NULL)
  )

cohort_with_diazepam_infusion_neg_epoch<-cohort_with_diazepam_infusion_neg_epoch%>%
  select(subject_id,diazepam_infusion_neg_epoch_01,diazepam_infusion_neg_epoch_02)%>%
  group_by(subject_id)%>%
  summarise_all(my.max)
```

Bolus Drugs

Negative Epochs Creation (total amount mg)

```{r eval=FALSE, include=FALSE}

#Drug Push is Bolus, so I filtered them out

diazepam_bolus_before_ext<-diazepam_bolus_from_patients_tofilter %>%
  filter(
    time_from_ext_to_diazepam_bolus_min >= -150
  )

```

```

n_distinct(diazepam_bolus_before_ext$subject_id) # 430 12/13/19

library(dplyr)

cohort_with_diazepam_bolus_neg_epoch<-diazepam_bolus_before_ext %>%
  mutate(
    diazepam_bolus_neg_epoch_01=if_else(between(time_from_ext_to_diazepam_bolus_min, -
90, 30), amount, NULL)
    ,diazepam_bolus_neg_epoch_02=if_else(between(time_from_ext_to_diazepam_bolus_min, -
150, -90), amount, NULL)
  )

my.sum <- function(x) ifelse( !all(is.na(x)), sum(x, na.rm=T), NA)

cohort_with_diazepam_bolus_neg_epoch<-cohort_with_diazepam_bolus_neg_epoch%>%
  select(subject_id,diazepam_bolus_neg_epoch_01,diazepam_bolus_neg_epoch_02)%>%
  group_by(subject_id)%>%
  summarise_all(my.sum)
...

## midazolam

### Infusion Drugs

#### Negative Epochs Creation (max rate mg/h)

```{r eval=FALSE, include=FALSE}

#Drug Push is Bolus, so I filtered them out
midazolam_infusion_before_ext<-midazolam_infusion_from_patients_tofilter %>%
  filter(
    time_from_ext_to_midazolam_starttime_min >= -150
    |
    time_from_ext_to_midazolam_endtime_min >= -150
  )

n_distinct(midazolam_infusion_before_ext$subject_id)

library(dplyr)

cohort_with_midazolam_infusion_neg_epoch<-midazolam_infusion_before_ext %>%
  mutate(
    midazolam_infusion_neg_epoch_01=if_else(
      between (time_from_ext_to_midazolam_starttime_min , -90, 30 )
      |
      between (time_from_ext_to_midazolam_endtime_min , -90, 30 )
      ,rate,NULL)
    ,midazolam_infusion_neg_epoch_02=if_else(
      between (time_from_ext_to_midazolam_starttime_min , -150, -90 )
      |
      between (time_from_ext_to_midazolam_endtime_min , -150, -90 )
      ,rate,NULL)
  )

```

```

cohort_with_midazolam_infusion_neg_epoch<-cohort_with_midazolam_infusion_neg_epoch%>%

select(subject_id,midazolam_infusion_neg_epoch_01,midazolam_infusion_neg_epoch_02)%>%
  group_by(subject_id)%>%
  summarise_all(my.max)
...

### Bolus Drugs

#### Negative Epochs Creation (total amount mg)

```{r eval=FALSE, include=FALSE}

#Drug Push is Bolus, so I filtered them out

midazolam_bolus_before_ext<-midazolam_bolus_from_patients_tofilter %>%
  filter(
    time_from_ext_to_midazolam_bolus_min >= -150
  )

n_distinct(midazolam_bolus_before_ext$subject_id) # 430 12/13/19

library(dplyr)

cohort_with_midazolam_bolus_neg_epoch<-midazolam_bolus_before_ext %>%
  mutate(
    midazolam_bolus_neg_epoch_01=if_else(between(time_from_ext_to_midazolam_bolus_min, -
90, 30), amount, NULL)
, midazolam_bolus_neg_epoch_02=if_else(between(time_from_ext_to_midazolam_bolus_min, -
150, -90), amount, NULL)
  )

my.sum <- function(x) ifelse( !all(is.na(x)), sum(x, na.rm=T), NA)

cohort_with_midazolam_bolus_neg_epoch<-cohort_with_midazolam_bolus_neg_epoch%>%
  select(subject_id,midazolam_bolus_neg_epoch_01,midazolam_bolus_neg_epoch_02)%>%
  group_by(subject_id)%>%
  summarise_all(my.sum)
...

## lorazepam

### Infusion Drugs

#### Negative Epochs Creation (max rate mg/h)

```{r eval=FALSE, include=FALSE}

#Drug Push is Bolus, so I filtered them out
lorazepam_infusion_before_ext<-lorazepam_infusion_from_patients_tofilter %>%
  filter(
    time_from_ext_to_lorazepam_starttime_min >= -150

```

```

      |
      time_from_ext_to_lorazepam_endtime_min >= -150
    )

# lorazepam = 1 milligram of morphine IV (225154) is equivalent to .01 mg Fentanyl
IV (221744).

n_distinct(lorazepam_infusion_before_ext$subject_id)

library(dplyr)

cohort_with_lorazepam_infusion_neg_epoch<-lorazepam_infusion_before_ext %>%
  mutate(
    lorazepam_infusion_neg_epoch_01=if_else(
      between (time_from_ext_to_lorazepam_starttime_min , -90, 30 )
      |
      between (time_from_ext_to_lorazepam_endtime_min , -90, 30 )
      ,rate,NULL)
    ,lorazepam_infusion_neg_epoch_02=if_else(
      between (time_from_ext_to_lorazepam_starttime_min , -150, -90 )
      |
      between (time_from_ext_to_lorazepam_endtime_min , -150, -90 )
      ,rate,NULL)
    )

cohort_with_lorazepam_infusion_neg_epoch<-cohort_with_lorazepam_infusion_neg_epoch%>%
  select(subject_id,lorazepam_infusion_neg_epoch_01,lorazepam_infusion_neg_epoch_02)%>%
  group_by(subject_id)%>%
  summarise_all(my.max)
...

### Bolus Drugs

#### Negative Epochs Creation (total amount mg)

```{r eval=FALSE, include=FALSE}

#Drug Push is Bolus, so I filtered them out

lorazepam_bolus_before_ext<-lorazepam_bolus_from_patients_tofilter %>%
  filter(
    time_from_ext_to_lorazepam_bolus_min >= -150
  )

n_distinct(lorazepam_bolus_before_ext$subject_id) # 430 12/13/19

library(dplyr)

cohort_with_lorazepam_bolus_neg_epoch<-lorazepam_bolus_before_ext %>%
  mutate(
    lorazepam_bolus_neg_epoch_01=if_else(between(time_from_ext_to_lorazepam_bolus_min, -
90, 30) ,amount,NULL)

```

```

,lorazepam_bolus_neg_epoch_02=if_else(between(time_from_ext_to_lorazepam_bolus_min,-
150, -90) ,amount,NULL)
)

my.sum <- function(x) ifelse( !all(is.na(x)), sum(x, na.rm=T), NA)

cohort_with_lorazepam_bolus_neg_epoch<-cohort_with_lorazepam_bolus_neg_epoch%>%
  select(subject_id,lorazepam_bolus_neg_epoch_01,lorazepam_bolus_neg_epoch_02)%>%
  group_by(subject_id)%>%
  summarise_all(my.sum)
...

## dexmedetomidie

### Infusion Drugs

#### Negative Epochs Creation (max rate mg/h)

```{r eval=FALSE, include=FALSE}

#Drug Push is Bolus, so I filtered them out
dexmedetomidie_infusion_before_ext<-dexmedetomidie_infusion_from_patients_tofilter
%>%
  filter(
    time_from_ext_to_dexmedetomidie_starttime_min >= -150
    |
    time_from_ext_to_dexmedetomidie_endtime_min >= -150
  )

n_distinct(dexmedetomidie_infusion_before_ext$subject_id)

library(dplyr)

cohort_with_dexmedetomidie_infusion_neg_epoch<-dexmedetomidie_infusion_before_ext %>%
  mutate(
    dexmedetomidie_infusion_neg_epoch_01=if_else(
      between (time_from_ext_to_dexmedetomidie_starttime_min , -90, 30 )
      |
      between (time_from_ext_to_dexmedetomidie_endtime_min , -90, 30 )
      ,rate,NULL)
    ,dexmedetomidie_infusion_neg_epoch_02=if_else(
      between (time_from_ext_to_dexmedetomidie_starttime_min , -150, -90 )
      |
      between (time_from_ext_to_dexmedetomidie_endtime_min , -150, -90 )
      ,rate,NULL)
    )

cohort_with_dexmedetomidie_infusion_neg_epoch<-
cohort_with_dexmedetomidie_infusion_neg_epoch%>%

select(subject_id,dexmedetomidie_infusion_neg_epoch_01,dexmedetomidie_infusion_neg_ep
och_02)%>%

```

```

    group_by(subject_id)%>%
    summarise_all(my.max)
  ...

### Bolus Drugs

#### Negative Epochs Creation (total amount mg)

```{r eval=FALSE, include=FALSE}

#Drug Push is Bolus, so I filtered them out

dexmedetomidie_bolus_before_ext<-dexmedetomidie_bolus_from_patients_tofilter %>%
  filter(
    time_from_ext_to_dexmedetomidie_bolus_min >= -150
  )

n_distinct(dexmedetomidie_bolus_before_ext$subject_id) # 430 12/13/19

library(dplyr)

cohort_with_dexmedetomidie_bolus_neg_epoch<-dexmedetomidie_bolus_before_ext %>%
  mutate(

    dexmedetomidie_bolus_neg_epoch_01=if_else(between(time_from_ext_to_dexmedetomidie_bolus_min, -90, 30) ,amount,NULL)
    ,dexmedetomidie_bolus_neg_epoch_02=if_else(between(time_from_ext_to_dexmedetomidie_bolus_min,-150, -90) ,amount,NULL)
  )

my.sum <- function(x) ifelse( !all(is.na(x)), sum(x, na.rm=T), NA)

cohort_with_dexmedetomidie_bolus_neg_epoch<-
cohort_with_dexmedetomidie_bolus_neg_epoch%>%

select(subject_id,dexmedetomidie_bolus_neg_epoch_01,dexmedetomidie_bolus_neg_epoch_02
)%>%
  group_by(subject_id)%>%
  summarise_all(my.sum)
...

# DRUGS WITH ONLY NEG EPOCHS INFUSIONS

## phenylephrine

### Infusion Drugs

#### Data extraction

```{r eval=FALSE, include=FALSE}
query<-"
SELECT
  inputevents_mv.subject_id,
  inputevents_mv.hadm_id,

```

```

    inputevents_mv.icustay_id,
    inputevents_mv.itemid,
    inputevents_mv.starttime,
    inputevents_mv.endtime,
    inputevents_mv.rate,
    inputevents_mv.rateuom,
    inputevents_mv.ordercategorydescription
From
  `physionet-data.mimiciii_clinical.inputevents_mv` inputevents_mv

```

```

WHERE
  inputevents_mv.itemid IN (221749)
  AND ordercategorydescription = 'Continuous Med'
"

```

```

phenylephrine_infusion_cohort_24<-query_exec(query, project =
project_HST,use_legacy_sql = F,max_pages = Inf)

```

```

phenylephrine_infusion_cohort_24$itemid<-'phenylephrine'
```

```

```

Creating the dataset

```

```

```{r}
phenylephrine_infusion_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex
tubation','deathtime')], phenylephrine_infusion_cohort_24)

```

```

# we noticed some timezones adrug_infusion alteterd, so we drugs_infusionset them
phenylephrine_infusion_from_patients_tofilter$starttime<-
as.character(phenylephrine_infusion_from_patients_tofilter$starttime)
phenylephrine_infusion_from_patients_tofilter$endtime<-
as.character(phenylephrine_infusion_from_patients_tofilter$endtime)
phenylephrine_infusion_from_patients_tofilter$charttime.extubation<-
as.character(phenylephrine_infusion_from_patients_tofilter$charttime.extubation)
phenylephrine_infusion_from_patients_tofilter$deathtime<-
as.character(phenylephrine_infusion_from_patients_tofilter$deathtime)

```

```

phenylephrine_infusion_from_patients_tofilter$starttime<-
as.POSIXct(phenylephrine_infusion_from_patients_tofilter$starttime,format='%Y-%m-%d
%H:%M:%S', tz="GMT")
phenylephrine_infusion_from_patients_tofilter$endtime<-
as.POSIXct(phenylephrine_infusion_from_patients_tofilter$endtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
phenylephrine_infusion_from_patients_tofilter$charttime.extubation<-
as.POSIXct(phenylephrine_infusion_from_patients_tofilter$charttime.extubation,
format='%Y-%m-%d %H:%M:%S', tz="GMT")
phenylephrine_infusion_from_patients_tofilter$deathtime<-
as.POSIXct(phenylephrine_infusion_from_patients_tofilter$deathtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

```

```

phenylephrine_infusion_from_patients_tofilter['time_from_ext_to_phenylephrine_startti
me_min']<-

```



```

difftime(phenylephrine_infusion_from_patients_tofilter$starttime,phenylephrine_infusi
on_from_patients_tofilter$charttime.extubation,units = 'mins')
phenylephrine_infusion_from_patients_tofilter['time_from_ext_to_phenylephrine_endtime
_min']<-
difftime(phenylephrine_infusion_from_patients_tofilter$endtime,phenylephrine_infusion
_from_patients_tofilter$charttime.extubation,units = 'mins')
```

Negative Epochs Creation (max rate mg/h)

```{r eval=FALSE, include=FALSE}

#Drug Push is Bolus, so I filtered them out
phenylephrine_infusion_before_ext<-phenylephrine_infusion_from_patients_tofilter %>%
  filter(
    time_from_ext_to_phenylephrine_starttime_min >= -150
    |
    time_from_ext_to_phenylephrine_endtime_min >= -150
  )

# phenylephrine = 1 milligram of morphine IV (225154) is equivalent to .01 mg
Fentanyl IV (221744).

n_distinct(phenylephrine_infusion_before_ext$subject_id)

library(dplyr)

cohort_with_phenylephrine_infusion_neg_epoch<-phenylephrine_infusion_before_ext %>%
  mutate(
    phenylephrine_infusion_neg_epoch_01=if_else(
      between (time_from_ext_to_phenylephrine_starttime_min , -90, 30 )
      |
      between (time_from_ext_to_phenylephrine_endtime_min , -90, 30 )
      ,rate,NULL)
    ,phenylephrine_infusion_neg_epoch_02=if_else(
      between (time_from_ext_to_phenylephrine_starttime_min ,-150, -90 )
      |
      between (time_from_ext_to_phenylephrine_endtime_min ,-150, -90 )
      ,rate,NULL)
    )

cohort_with_phenylephrine_infusion_neg_epoch<-
cohort_with_phenylephrine_infusion_neg_epoch%>%

select(subject_id,phenylephrine_infusion_neg_epoch_01,phenylephrine_infusion_neg_epoc
h_02)%>%
  group_by(subject_id)%>%
  summarise_all(my.max)
```

norepinephrine

Infusion Drugs

```

#### Data extraction

```
```{r eval=FALSE, include=FALSE}
query<-"
SELECT
  inpuvents_mv.subject_id,
  inpuvents_mv.hadm_id,
  inpuvents_mv.icustay_id,
  inpuvents_mv.itemid,
  inpuvents_mv.starttime,
  inpuvents_mv.endtime,
  inpuvents_mv.rate,
  inpuvents_mv.rateuom,
  inpuvents_mv.ordercategorydescription
From
  `physionet-data.mimiciii_clinical.inpuvents_mv` inpuvents_mv

WHERE
  inpuvents_mv.itemid IN (221906)
  AND ordercategorydescription = 'Continuous Med'
"

norepinephrine_infusion_cohort_24<-query_exec(query, project =
project_HST,use_legacy_sql = F,max_pages = Inf)

norepinephrine_infusion_cohort_24$itemid<-'norepinephrine'
```
```

#### Creating the dataset

```
```{r}
norepinephrine_infusion_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.extubation','deathtime')], norepinephrine_infusion_cohort_24)

# we noticed some timezones adnugs_infusion alteterd, so we drugs_infusionset them
norepinephrine_infusion_from_patients_tofilter$starttime<-
as.character(norepinephrine_infusion_from_patients_tofilter$starttime)
norepinephrine_infusion_from_patients_tofilter$endtime<-
as.character(norepinephrine_infusion_from_patients_tofilter$endtime)
norepinephrine_infusion_from_patients_tofilter$charttime.extubation<-
as.character(norepinephrine_infusion_from_patients_tofilter$charttime.extubation)
norepinephrine_infusion_from_patients_tofilter$deathtime<-
as.character(norepinephrine_infusion_from_patients_tofilter$deathtime)

norepinephrine_infusion_from_patients_tofilter$starttime<-
as.POSIXct(norepinephrine_infusion_from_patients_tofilter$starttime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
norepinephrine_infusion_from_patients_tofilter$endtime<-
as.POSIXct(norepinephrine_infusion_from_patients_tofilter$endtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
norepinephrine_infusion_from_patients_tofilter$charttime.extubation<-
as.POSIXct(norepinephrine_infusion_from_patients_tofilter$charttime.extubation,
format='%Y-%m-%d %H:%M:%S', tz="GMT")
```

```

norepinephrine_infusion_from_patients_tofilter$deathtime<-
as.POSIXct(norepinephrine_infusion_from_patients_tofilter$deathtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

norepinephrine_infusion_from_patients_tofilter['time_from_ext_to_norepinephrine_start
time_min']<-
difftime(norepinephrine_infusion_from_patients_tofilter$starttime,norepinephrine_infu
sion_from_patients_tofilter$charttime.extubation,units = 'mins')
norepinephrine_infusion_from_patients_tofilter['time_from_ext_to_norepinephrine_endti
me_min']<-
difftime(norepinephrine_infusion_from_patients_tofilter$endtime,norepinephrine_infusi
on_from_patients_tofilter$charttime.extubation,units = 'mins')
``

#### Negative Epochs Creation (max rate mg/h)

``{r eval=FALSE, include=FALSE}

#Drug Push is Bolus, so I filtered them out
norepinephrine_infusion_before_ext<-norepinephrine_infusion_from_patients_tofilter
%>%
  filter(
    time_from_ext_to_norepinephrine_starttime_min >= -150
    |
    time_from_ext_to_norepinephrine_endtime_min >= -150
  )

# norepinephrine = 1 milligram of morphine IV (225154) is equivalent to .01 mg
Fentanyl IV (221744).

n_distinct(norepinephrine_infusion_before_ext$subject_id)

library(dplyr)

cohort_with_norepinephrine_infusion_neg_epoch<-norepinephrine_infusion_before_ext %>%
  mutate(
    norepinephrine_infusion_neg_epoch_01=if_else(
      between (time_from_ext_to_norepinephrine_starttime_min , -90, 30 )
      |
      between (time_from_ext_to_norepinephrine_endtime_min , -90, 30 )
      ,rate,NULL)
    ,norepinephrine_infusion_neg_epoch_02=if_else(
      between (time_from_ext_to_norepinephrine_starttime_min , -150, -90 )
      |
      between (time_from_ext_to_norepinephrine_endtime_min , -90, 30 )
      ,rate,NULL)
    )

cohort_with_norepinephrine_infusion_neg_epoch<-
cohort_with_norepinephrine_infusion_neg_epoch%>%

select(subject_id,norepinephrine_infusion_neg_epoch_01,norepinephrine_infusion_neg_ep
och_02)%>%
  group_by(subject_id)%>%

```

```

    summarise_all(my.max)
  })

## dopamine

### Infusion Drugs

#### Data extraction

```{r eval=FALSE, include=FALSE}
query<-"
SELECT
  inputevents_mv.subject_id,
  inputevents_mv.hadm_id,
  inputevents_mv.icustay_id,
  inputevents_mv.itemid,
  inputevents_mv.starttime,
  inputevents_mv.endtime,
  inputevents_mv.rate,
  inputevents_mv.rateuom,
  inputevents_mv.ordercategorydescription
From
  `physionet-data.mimiciii_clinical.inputevents_mv` inputevents_mv

WHERE
  inputevents_mv.itemid IN (221662)
  AND ordercategorydescription = 'Continuous Med'
"

dopamine_infusion_cohort_24<-query_exec(query, project = project_HST,use_legacy_sql =
F,max_pages = Inf)

dopamine_infusion_cohort_24$itemid<-'dopamine'
```

Creating the dataset

```{r}
dopamine_infusion_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex
tubation','deathtime')], dopamine_infusion_cohort_24)

# we noticed some timezones adrug_infusion alteterd, so we drugs_infusionset them
dopamine_infusion_from_patients_tofilter$starttime<-
as.character(dopamine_infusion_from_patients_tofilter$starttime)
dopamine_infusion_from_patients_tofilter$endtime<-
as.character(dopamine_infusion_from_patients_tofilter$endtime)
dopamine_infusion_from_patients_tofilter$charttime.extubation<-
as.character(dopamine_infusion_from_patients_tofilter$charttime.extubation)
dopamine_infusion_from_patients_tofilter$deathtime<-
as.character(dopamine_infusion_from_patients_tofilter$deathtime)

```

```

dopamine_infusion_from_patients_tofilter$starttime<-
as.POSIXct(dopamine_infusion_from_patients_tofilter$starttime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
dopamine_infusion_from_patients_tofilter$endtime<-
as.POSIXct(dopamine_infusion_from_patients_tofilter$endtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
dopamine_infusion_from_patients_tofilter$charttime.extubation<-
as.POSIXct(dopamine_infusion_from_patients_tofilter$charttime.extubation, format='%Y-
%m-%d %H:%M:%S', tz="GMT")
dopamine_infusion_from_patients_tofilter$deathtime<-
as.POSIXct(dopamine_infusion_from_patients_tofilter$deathtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

dopamine_infusion_from_patients_tofilter['time_from_ext_to_dopamine_starttime_min']<-
difftime(dopamine_infusion_from_patients_tofilter$starttime,dopamine_infusion_from_pa
tients_tofilter$charttime.extubation,units = 'mins')
dopamine_infusion_from_patients_tofilter['time_from_ext_to_dopamine_endtime_min']<-
difftime(dopamine_infusion_from_patients_tofilter$endtime,dopamine_infusion_from_pati
ents_tofilter$charttime.extubation,units = 'mins')

...

#### Negative Epochs Creation (max rate mg/h)

```{r eval=FALSE, include=FALSE}

#Drug Push is Bolus, so I filtered them out
dopamine_infusion_before_ext<-dopamine_infusion_from_patients_tofilter %>%
  filter(
    time_from_ext_to_dopamine_starttime_min >= -150
    |
    time_from_ext_to_dopamine_endtime_min >= -150
  )

n_distinct(dopamine_infusion_before_ext$subject_id)

library(dplyr)

cohort_with_dopamine_infusion_neg_epoch<-dopamine_infusion_before_ext %>%
  mutate(
    dopamine_infusion_neg_epoch_01=if_else(
      between (time_from_ext_to_dopamine_starttime_min , -90, 30 )
      |
      between (time_from_ext_to_dopamine_endtime_min , -90, 30 )
      ,rate,NULL)
    ,dopamine_infusion_neg_epoch_02=if_else(
      between (time_from_ext_to_dopamine_starttime_min , -150, -90 )
      |
      between (time_from_ext_to_dopamine_endtime_min , -150, -90 )
      ,rate,NULL)
    )

cohort_with_dopamine_infusion_neg_epoch<-cohort_with_dopamine_infusion_neg_epoch%>%
  select(subject_id,dopamine_infusion_neg_epoch_01,dopamine_infusion_neg_epoch_02)%>%

```

```

    group_by(subject_id)%>%
    summarise_all(my.max)
  })

## epinephrine

### Infusion Drugs

#### Data extraction

```{r eval=FALSE, include=FALSE}
query<-"
SELECT
  inpuvents_mv.subject_id,
  inpuvents_mv.hadm_id,
  inpuvents_mv.icustay_id,
  inpuvents_mv.itemid,
  inpuvents_mv.starttime,
  inpuvents_mv.endtime,
  inpuvents_mv.rate,
  inpuvents_mv.rateuom,
  inpuvents_mv.ordercategorydescription
From
  `physionet-data.mimiciii_clinical.inpuvents_mv` inpuvents_mv

WHERE
  inpuvents_mv.itemid IN (221289)
  AND ordercategorydescription = 'Continuous Med'
"

epinephrine_infusion_cohort_24<-query_exec(query, project =
project_HST,use_legacy_sql = F,max_pages = Inf)

epinephrine_infusion_cohort_24$itemid<-'epinephrine'
#Valium (Diazepam)
#We tried valium and it didn't increase the number of patients, still 84 after ext
```

Creating the dataset

```{r}
epinephrine_infusion_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex
tubation','deathtime')], epinephrine_infusion_cohort_24)

# we noticed some timezones adrug_infusion alteterd, so we drugs_infusionset them
epinephrine_infusion_from_patients_tofilter$starttime<-
as.character(epinephrine_infusion_from_patients_tofilter$starttime)
epinephrine_infusion_from_patients_tofilter$endtime<-
as.character(epinephrine_infusion_from_patients_tofilter$endtime)
epinephrine_infusion_from_patients_tofilter$charttime.extubation<-
as.character(epinephrine_infusion_from_patients_tofilter$charttime.extubation)
epinephrine_infusion_from_patients_tofilter$deathtime<-
as.character(epinephrine_infusion_from_patients_tofilter$deathtime)

```

```

epinephrine_infusion_from_patients_tofilter$starttime<-
as.POSIXct(epinephrine_infusion_from_patients_tofilter$starttime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
epinephrine_infusion_from_patients_tofilter$endtime<-
as.POSIXct(epinephrine_infusion_from_patients_tofilter$endtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
epinephrine_infusion_from_patients_tofilter$charttime.extubation<-
as.POSIXct(epinephrine_infusion_from_patients_tofilter$charttime.extubation,
format='%Y-%m-%d %H:%M:%S', tz="GMT")
epinephrine_infusion_from_patients_tofilter$deathtime<-
as.POSIXct(epinephrine_infusion_from_patients_tofilter$deathtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

epinephrine_infusion_from_patients_tofilter['time_from_ext_to_epinephrine_starttime_m
in']<-
difftime(epinephrine_infusion_from_patients_tofilter$starttime,epinephrine_infusion_f
rom_patients_tofilter$charttime.extubation,units = 'mins')
epinephrine_infusion_from_patients_tofilter['time_from_ext_to_epinephrine_endtime_min
']<-
difftime(epinephrine_infusion_from_patients_tofilter$endtime,epinephrine_infusion_fro
m_patients_tofilter$charttime.extubation,units = 'mins')
```

Negative Epochs Creation (max rate mg/h)

```{r eval=FALSE, include=FALSE}

#Drug Push is Bolus, so I filtered them out
epinephrine_infusion_before_ext<-epinephrine_infusion_from_patients_tofilter %>%
  filter(
    time_from_ext_to_epinephrine_starttime_min >= -150
    |
    time_from_ext_to_epinephrine_endtime_min >= -150
  )

# epinephrine = 1 milligram of morphine IV (225154) is equivalent to .01 mg Fentanyl
IV (221744).

n_distinct(epinephrine_infusion_before_ext$subject_id)

library(dplyr)

cohort_with_epinephrine_infusion_neg_epoch<-epinephrine_infusion_before_ext %>%
  mutate(
    epinephrine_infusion_neg_epoch_01=if_else(
      between (time_from_ext_to_epinephrine_starttime_min , -90, 30 )
      |
      between (time_from_ext_to_epinephrine_endtime_min , -90, 30 )
      ,rate,NULL)
    ,epinephrine_infusion_neg_epoch_02=if_else(
      between (time_from_ext_to_epinephrine_starttime_min , -150, -90 )
      |
      between (time_from_ext_to_epinephrine_endtime_min , -150, -90 )
      ,rate,NULL)
  )

```

```

)

cohort_with_epinephrine_infusion_neg_epoch<-
cohort_with_epinephrine_infusion_neg_epoch%>%

select(subject_id,epinephrine_infusion_neg_epoch_01,epinephrine_infusion_neg_epoch_02
)%>%
  group_by(subject_id)%>%
  summarise_all(my.max)
...

## vasopressin

### Infusion Drugs

#### Data extraction

```{r eval=FALSE, include=FALSE}
query<-"
SELECT
  inpatientevents_mv.subject_id,
  inpatientevents_mv.hadm_id,
  inpatientevents_mv.icustay_id,
  inpatientevents_mv.itemid,
  inpatientevents_mv.starttime,
  inpatientevents_mv.endtime,
  inpatientevents_mv.rate,
  inpatientevents_mv.rateuom,
  inpatientevents_mv.ordercategorydescription
From
  `physionet-data.mimiciii_clinical.inpatientevents_mv` inpatientevents_mv

WHERE
  inpatientevents_mv.itemid IN (222315)
  AND ordercategorydescription = 'Continuous Med'
"

vasopressin_infusion_cohort_24<-query_exec(query, project =
project_HST,use_legacy_sql = F,max_pages = Inf)

vasopressin_infusion_cohort_24$itemid<-'vasopressin'
#Valium (Diazepam)
#We tried valium and it didn't increase the number of patients, still 84 after ext
...

#### Creating the dataset

```{r}
vasopressin_infusion_from_patients_tofilter<-
inner_join(cmo_and_extubation_and_death_24[,c('subject_id','icustay_id','charttime.ex
tubation','deathtime')], vasopressin_infusion_cohort_24)

```



```

# we noticed some timezones ad drugs_infusion alteterd, so we drugs_infusionset them
vasopressin_infusion_from_patients_tofilter$starttime<-
as.character(vasopressin_infusion_from_patients_tofilter$starttime)
vasopressin_infusion_from_patients_tofilter$endtime<-
as.character(vasopressin_infusion_from_patients_tofilter$endtime)
vasopressin_infusion_from_patients_tofilter$charttime.extubation<-
as.character(vasopressin_infusion_from_patients_tofilter$charttime.extubation)
vasopressin_infusion_from_patients_tofilter$deathtime<-
as.character(vasopressin_infusion_from_patients_tofilter$deathtime)

vasopressin_infusion_from_patients_tofilter$starttime<-
as.POSIXct(vasopressin_infusion_from_patients_tofilter$starttime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
vasopressin_infusion_from_patients_tofilter$endtime<-
as.POSIXct(vasopressin_infusion_from_patients_tofilter$endtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")
vasopressin_infusion_from_patients_tofilter$charttime.extubation<-
as.POSIXct(vasopressin_infusion_from_patients_tofilter$charttime.extubation,
format='%Y-%m-%d %H:%M:%S', tz="GMT")
vasopressin_infusion_from_patients_tofilter$deathtime<-
as.POSIXct(vasopressin_infusion_from_patients_tofilter$deathtime, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

vasopressin_infusion_from_patients_tofilter['time_from_ext_to_vasopressin_starttime_m
in']<-
difftime(vasopressin_infusion_from_patients_tofilter$starttime,vasopressin_infusion_f
rom_patients_tofilter$charttime.extubation,units = 'mins')
vasopressin_infusion_from_patients_tofilter['time_from_ext_to_vasopressin_endtime_min
']<-
difftime(vasopressin_infusion_from_patients_tofilter$endtime,vasopressin_infusion_fro
m_patients_tofilter$charttime.extubation,units = 'mins')
``

#### Negative Epochs Creation (max rate mg/h)

```{r eval=FALSE, include=FALSE}

#Drug Push is Bolus, so I filtered them out
vasopressin_infusion_before_ext<-vasopressin_infusion_from_patients_tofilter %>%
  filter(
    time_from_ext_to_vasopressin_starttime_min >= -150
    |
    time_from_ext_to_vasopressin_endtime_min >= -150
  )

n_distinct(vasopressin_infusion_before_ext$subject_id)

library(dplyr)

cohort_with_vasopressin_infusion_neg_epoch<-vasopressin_infusion_before_ext %>%
  mutate(
    vasopressin_infusion_neg_epoch_01=if_else(

```

```

      between (time_from_ext_to_vasopressin_starttime_min , -90, 30 )
      |
      between (time_from_ext_to_vasopressin_endtime_min , -90, 30 )
      ,rate,NULL)
,vasopressin_infusion_neg_epoch_02=if_else(
  between (time_from_ext_to_vasopressin_starttime_min , -150, -90 )
  |
  between (time_from_ext_to_vasopressin_endtime_min , -150, -90 )
  ,rate,NULL)
)

cohort_with_vasopressin_infusion_neg_epoch<-
cohort_with_vasopressin_infusion_neg_epoch%>%

select(subject_id,vasopressin_infusion_neg_epoch_01,vasopressin_infusion_neg_epoch_02
)%>%
  group_by(subject_id)%>%
  summarise_all(my.max)
...

---
title: "Final Join"
author: "Miguel Angel Armengol de la Hoz"
date: "August 1, 2018"
output: html_document
---

# Environment

```{r}
library(RPostgreSQL)
library(DBI)
library(dplyr)
library(summarytools)
...

## Data join

### Final Dataset
```{r}

#Demographics joins
final_dataset<-left_join(cmo_and_extubation_and_death_24,
extubated_procedureevents_mv[, c('subject_id','extubated_in_procedureevents')])
final_dataset<-left_join(final_dataset, demographics_all)
final_dataset<-left_join(final_dataset, gcs_last_day)
final_dataset<-left_join(final_dataset, vent_mode_last_pre_ext_final)
final_dataset<-left_join(final_dataset, top15icd9_codes_with_labels_sumarized)
final_dataset<-left_join(final_dataset,last_pao2fio2_patients_final)

#Epochs joins
final_dataset<-left_join(final_dataset,final_rr_te_epoch)
final_dataset<-left_join(final_dataset,rr_from_patients_tofilter_first_final)
final_dataset<-left_join(final_dataset,cohort_with_opioids_infusion_epoch)

```

```

final_dataset<-left_join(final_dataset,cohort_with_benzodiazepines_infusion_epoch)
final_dataset<-left_join(final_dataset,cohort_with_opioids_bolus_epoch)
final_dataset<-left_join(final_dataset,cohort_with_benzodiazepines_bolus_epoch)
final_dataset<-left_join(final_dataset,cohort_with_opioids_infusion_neg_epoch)
final_dataset<-
left_join(final_dataset,cohort_with_benzodiazepines_infusion_neg_epoch)
final_dataset<-left_join(final_dataset,cohort_with_opioids_bolus_neg_epoch)
final_dataset<-left_join(final_dataset,cohort_with_benzodiazepines_bolus_neg_epoch)
final_dataset<-left_join(final_dataset,cohort_with_re_epoch)
final_dataset<-left_join(final_dataset,cohort_with_ps_epoch)
final_dataset<-left_join(final_dataset,cohort_with_min_rass_epoch)
final_dataset<-left_join(final_dataset,cohort_with_max_rass_epoch)

#new drugs Epochs joins
final_dataset<-left_join(final_dataset,cohort_with_propofol_bolus_epoch)
final_dataset<-left_join(final_dataset,cohort_with_diazepam_bolus_epoch)
final_dataset<-left_join(final_dataset,cohort_with_midazolam_bolus_epoch)
final_dataset<-left_join(final_dataset,cohort_with_lorazepam_bolus_epoch)
final_dataset<-left_join(final_dataset,cohort_with_dexmedetomidie_bolus_epoch)

final_dataset<-left_join(final_dataset,cohort_with_propofol_infusion_epoch)
final_dataset<-left_join(final_dataset,cohort_with_diazepam_infusion_epoch)
final_dataset<-left_join(final_dataset,cohort_with_midazolam_infusion_epoch)
final_dataset<-left_join(final_dataset,cohort_with_lorazepam_infusion_epoch)
final_dataset<-left_join(final_dataset,cohort_with_dexmedetomidie_infusion_epoch)

#new drugs neg Epochs joins
final_dataset<-left_join(final_dataset,cohort_with_propofol_bolus_neg_epoch)
final_dataset<-left_join(final_dataset,cohort_with_diazepam_bolus_neg_epoch)
final_dataset<-left_join(final_dataset,cohort_with_midazolam_bolus_neg_epoch)
final_dataset<-left_join(final_dataset,cohort_with_lorazepam_bolus_neg_epoch)
final_dataset<-left_join(final_dataset,cohort_with_dexmedetomidie_bolus_neg_epoch)

final_dataset<-left_join(final_dataset,cohort_with_propofol_infusion_neg_epoch)
final_dataset<-left_join(final_dataset,cohort_with_diazepam_infusion_neg_epoch)
final_dataset<-left_join(final_dataset,cohort_with_midazolam_infusion_neg_epoch)
final_dataset<-left_join(final_dataset,cohort_with_lorazepam_infusion_neg_epoch)
final_dataset<-left_join(final_dataset,cohort_with_dexmedetomidie_infusion_neg_epoch)

#new drugs with only neg infusions Epochs joins
final_dataset<-left_join(final_dataset,cohort_with_phenylephrine_infusion_neg_epoch)
final_dataset<-left_join(final_dataset,cohort_with_norepinephrine_infusion_neg_epoch)
final_dataset<-left_join(final_dataset,cohort_with_dopamine_infusion_neg_epoch)
final_dataset<-left_join(final_dataset,cohort_with_epinephrine_infusion_neg_epoch)
final_dataset<-left_join(final_dataset,cohort_with_vasopressin_infusion_neg_epoch)

library(foreign)
write.dta(final_dataset, "final_dataset.dta")
cat(names(final_dataset),sep = '\n')
#view(dfSummary(final_dataset))
``,`

# Variables distribution

```

```

```{r}
view(dfSummary(final_dataset%>%select(
propofol_bolus_epoch_01,
propofol_bolus_epoch_02,
diazepam_bolus_epoch_01,
diazepam_bolus_epoch_02,
midazolam_bolus_epoch_01,
midazolam_bolus_epoch_02,
lorazepam_bolus_epoch_01,
lorazepam_bolus_epoch_02,
dexmedetomidie_bolus_epoch_01,
dexmedetomidie_bolus_epoch_02,
propofol_infusion_epoch_01,
propofol_infusion_epoch_02,
diazepam_infusion_epoch_01,
diazepam_infusion_epoch_02,
midazolam_infusion_epoch_01,
midazolam_infusion_epoch_02,
lorazepam_infusion_epoch_01,
lorazepam_infusion_epoch_02,
dexmedetomidie_infusion_epoch_01,
dexmedetomidie_infusion_epoch_02,
propofol_bolus_neg_epoch_01,
propofol_bolus_neg_epoch_02,
diazepam_bolus_neg_epoch_01,
diazepam_bolus_neg_epoch_02,
midazolam_bolus_neg_epoch_01,
midazolam_bolus_neg_epoch_02,
lorazepam_bolus_neg_epoch_01,
lorazepam_bolus_neg_epoch_02,
dexmedetomidie_bolus_neg_epoch_01,
dexmedetomidie_bolus_neg_epoch_02,
propofol_infusion_neg_epoch_01,
propofol_infusion_neg_epoch_02,
diazepam_infusion_neg_epoch_01,
diazepam_infusion_neg_epoch_02,
midazolam_infusion_neg_epoch_01,
midazolam_infusion_neg_epoch_02,
lorazepam_infusion_neg_epoch_01,
lorazepam_infusion_neg_epoch_02,
dexmedetomidie_infusion_neg_epoch_01,
dexmedetomidie_infusion_neg_epoch_02,
phenylephrine_infusion_neg_epoch_01,
phenylephrine_infusion_neg_epoch_02,
norepinephrine_infusion_neg_epoch_01,
norepinephrine_infusion_neg_epoch_02,
dopamine_infusion_neg_epoch_01,
dopamine_infusion_neg_epoch_02,
epinephrine_infusion_neg_epoch_01,
epinephrine_infusion_neg_epoch_02,
vasopressin_infusion_neg_epoch_01,
vasopressin_infusion_neg_epoch_02
)))
```

```

# APPENDIX VII: ASSOCIATION OF INTRAOPERATIVE HYPOTENSION WITH POSTOPERATIVE STROKE, ACUTE KIDNEY INJURY, AND DEATH IN PATIENTS HAVING CARDIAC SURGERY WITH CARDIOPULMONARY BYPASS

**Supplemental Table 1.** Postoperative Major Adverse Events (MAE)\*

| In-hospital complications | N = 274 (5.3) |
|---------------------------|---------------|
| Stroke                    | 69 (1.3)      |
| Renal Failure             | 134 (2.6)     |
| Death                     | 117 (2.3)     |

*Values are presented as numbers (percentages) unless otherwise indicated*

*Major Adverse Events (MAE)\*, composite of renal failure, stroke and death*

*<sup>a</sup>Reexploration for bleeding complications*

**Supplemental Table 2.** Intraoperative hypotension duration, AUC-MAP < 65 mm Hg , vasopressor dose categorized by quartiles.

| Quartiles | Duration MAP<br>< 65 mmHg<br>(minutes) | AUC-MAP<br>< 65 mmHg<br>(mm Hg.min) | Vasopressor-inotrope dose<br>(mg) |
|-----------|----------------------------------------|-------------------------------------|-----------------------------------|
| 1         | 0 - 72                                 | 0 - 660                             | < 0.38                            |
| 2         | 72.25 - 96.75                          | 660.25 - 941.50                     | 0.38 - 0.68                       |
| 3         | 97.00- 128.25                          | 941.75 - 1338.75                    | 0.68 - 1.17                       |
| 4         | > 128.5                                | > 1339                              | > 1.17                            |

*Abbreviations: AUC, Area Under the Curve; MAP, Mean arterial Pressure;*

## STS (Society of Thoracic Surgeons) VERSION 2.81

### Definitions for postoperative outcomes

#### Complications

Indicate whether a postoperative event occurred during the hospitalization for surgery. This includes the entire postoperative period up to discharge, even if over 30 days.

#### Renal Failure

Indicate whether the patient had acute renal failure or worsening renal function resulting in one or both of the following

1. Increase in serum creatinine level 3.0 x greater than baseline
2. serum creatinine level  $\geq 4$  mg/dL
3. Acute rise must be at least 0.5 mg/dl 2
4. A new requirement for dialysis postoperatively.

#### Stroke

Indicate whether the patient has a postoperative stroke (i.e., any confirmed neurological deficit of abrupt onset caused by a disturbance in cerebral blood supply) that did not resolve within 24 hours.

#### Death

Operative Death includes: (1) all deaths, regardless of cause, occurring during the hospitalization in which the operation was performed, even if after 30 days (including patients transferred to other acute care facilities); and (2) all deaths, regardless of cause, occurring after discharge from the hospital, but before the end of the thirtieth postoperative day.

#### STS Dataset Notations

As a reference and in order to verify every variable in the STS dataset, we used the manuscript 'Increased Glycemic Variability in Patients with Elevated Preoperative HbA1C Predicts Adverse Outcomes Following Coronary Artery Bypass Grafting Surgery', this paper used the 2014 dataset and our current dataset contains data up to 2017.

Perioperative\_MI was discontinued in July 2011 thus it cannot be found anywhere else since the lab data to calculate it has not been collected, thus, we excluded it from the study.

Infect\_Sternum\_Deep was overcoded before 2011, so after 2011, when it was redefined to PostOpDeepSternalMediastin there are no cases of Infect\_Sternum\_Deep, so we included any kind of infection in the study.

Neuro\_Stroke\_Permanent changed to PostOpNeuroStrokePermanent in the version dataset STS\_2.81\_July\_2014\_June\_2017

Smoking is the most inconsistent variable throughout the database due to changing definitions version to version. It has been consistent since July 2014.

CHF changes in 2014 to HeartFailure2wks

But it is since 2013 when Percentages are multiplied by 2

The variable was better documented since 2013. They drastically undercoded this variable in the past.

In June 2014 PreviousMI changed to PriorMI

Hypertension changes to RFHypertension in 2014.

In the version dataset STS\_2.73\_July\_2011\_Jun\_2014 PreOpMedACE\_ARBIInhibitors changed to PreOpMedACEIARBI48hrs.

## Code

### Dataset creation

```
---
title: "DataLoad_and_optimization_PhenotypingJuly17"
output: html_document

#Loading data from MS Access database

-----VERY
IMPORTANT!!!!!!!!!!!!!!
Steps to use, assuming 32 bit Access on windows

Select 32 bit R (is just a setting in R studio)
search on windows for Set up ODBC data sources (32 bit)
Go to System DSN>Add
```

```

Choose Driver do Microsoft Access (*.mdb) > Finish
Data source name: ProjecnameAcc
Description: ProjectnameAcc
Make sure to actually select the database > OK
Now I could run the code that I liked

```

```

channel <- odbcConnect("ProjectnameAcc")
Table1Dat <- sqlFetch(channel, "Table1")

##Load Light Tables

```{r setup, include=FALSE}
#install.packages("RODBC")
library(RODBC)

library(readr)
hemodynamics_source_name_codes <- read_delim("C:/BIG
FILES/Phenotyping/hemodynamics_source-name_codes.csv",
      ";", escape_double = FALSE, trim_ws = TRUE)

for (i in 8:16){
  if (i<10) {
    # Connect to Access db
    channel <- odbcConnect (paste("Hemodynamics0", i, sep=""))
    # Load tables
    tablename<-(paste("Text0", i, sep=""))
    assign( tablename,sqlFetch(channel, "Text"))
    tablename<-(paste("InfusionDrugs0", i, sep=""))
    assign( tablename,sqlFetch(channel, "InfusionDrugs"))
    tablename<-(paste("Events0", i, sep=""))
    assign( tablename,sqlFetch(channel, "Events"))
    tablename<-(paste("Date0", i, sep=""))
    assign( tablename,sqlFetch(channel, "Date"))
    tablename<-(paste("Combo0", i, sep=""))
    assign( tablename,sqlFetch(channel, "Combo"))
    tablename<-(paste("BolusDrugs0", i, sep=""))
    assign( tablename,sqlFetch(channel, "BolusDrugs"))
    tablename<-(paste("InfusionDrugs0", i, sep=""))
    assign( tablename,sqlFetch(channel, "InfusionDrugs"))
  }
  else{
    # Connect to Access db
    channel <- odbcConnect (paste("Hemodynamics", i, sep=""))
    # Load tables
    tablename<-(paste("Text", i, sep=""))
    assign( tablename,sqlFetch(channel, "Text"))
    tablename<-(paste("InfusionDrugs", i, sep=""))
    assign( tablename,sqlFetch(channel, "InfusionDrugs"))
    tablename<-(paste("Events", i, sep=""))
    assign( tablename,sqlFetch(channel, "Events"))
    tablename<-(paste("Date", i, sep=""))
    assign( tablename,sqlFetch(channel, "Date"))
    tablename<-(paste("Combo", i, sep=""))
  }
}

```



```

    assign( tablename,sqlFetch(channel, "Combo"))
    tablename<-(paste("BolusDrugs", i, sep=""))
    assign( tablename,sqlFetch(channel, "BolusDrugs"))
    tablename<-(paste("InfusionDrugs", i, sep=""))
    assign( tablename,sqlFetch(channel, "InfusionDrugs"))
  }
}

```

```

...

```

```

##Load Heavy Tables

```

```

```{r}
library(RODBC)

for (i in 16){
  if (i<10) {
    # Connect to Access db
    channel <- odbcConnect (paste("Hemodynamics0", i, sep=""))
    # Load tables
    tablename<-(paste("Vitalsigns0", i, sep=""))
    assign( tablename,sqlFetch(channel, "Vitalsigns"))

  }
  else{
    # Connect to Access db
    channel <- odbcConnect (paste("Hemodynamics", i, sep=""))
    # Load tables
    tablename<-(paste("Vitalsigns", i, sep=""))
    assign( tablename,sqlFetch(channel, "Vitalsigns"))
  }
}
...

```

```

#Cleaning and optimizing the data

```

This chunk is removing not used columns and mapping the BP tags to a numerical code so the final dataset is lighter.

```

```{r}

dropList <-
c("ItemID","Label","Units","Priority","Device_Name","Port#","Display_Value","Valid","
User_Invalid","Limits_Invalid","Value_Questionable")
HDC_source_code<- hemodynamics_source_name_codes[,c(3,4)]

Vitalsigns08 <- Vitalsigns08[, !colnames(Vitalsigns08) %in% dropList]
Vitalsigns08$`Port#`<-NULL
Vitalsigns08<- merge(Vitalsigns08,  HDC_source_code, by='Source_Name')
Vitalsigns08$Source_Name<-NULL

Vitalsigns09 <- Vitalsigns09[, !colnames(Vitalsigns09) %in% dropList]
Vitalsigns09$`Port#`<-NULL
Vitalsigns09<- merge(Vitalsigns09,  HDC_source_code, by='Source_Name')

```

```

Vitalsigns09$Source_Name<-NULL

Vitalsigns10 <- Vitalsigns10[, !colnames(Vitalsigns10) %in% dropList]
Vitalsigns10$`Port#`<-NULL
Vitalsigns10<- merge(Vitalsigns10, HDC_source_code, by='Source_Name')
Vitalsigns10$Source_Name<-NULL

Vitalsigns11 <- Vitalsigns11[, !colnames(Vitalsigns11) %in% dropList]
Vitalsigns11$`Port#`<-NULL
Vitalsigns11<- merge(Vitalsigns11, HDC_source_code, by='Source_Name')
Vitalsigns11$Source_Name<-NULL

Vitalsigns12 <- Vitalsigns12[, !colnames(Vitalsigns12) %in% dropList]
Vitalsigns12$`Port#`<-NULL
Vitalsigns12<- merge(Vitalsigns12, HDC_source_code, by='Source_Name')
Vitalsigns12$Source_Name<-NULL

Vitalsigns13 <- Vitalsigns13[, !colnames(Vitalsigns13) %in% dropList]
Vitalsigns13$`Port#`<-NULL
Vitalsigns13<- merge(Vitalsigns13, HDC_source_code, by='Source_Name')
Vitalsigns13$Source_Name<-NULL

Vitalsigns14 <- Vitalsigns14[, !colnames(Vitalsigns14) %in% dropList]
Vitalsigns14$`Port#`<-NULL
Vitalsigns14<- merge(Vitalsigns14, HDC_source_code, by='Source_Name')
Vitalsigns14$Source_Name<-NULL

Vitalsigns15 <- Vitalsigns15[, !colnames(Vitalsigns15) %in% dropList]
Vitalsigns15$`Port#`<-NULL
Vitalsigns15<- merge(Vitalsigns15, HDC_source_code, by='Source_Name')
Vitalsigns15$Source_Name<-NULL

Vitalsigns16 <- Vitalsigns16[, !colnames(Vitalsigns16) %in% dropList]
Vitalsigns16$`Port#`<-NULL
Vitalsigns16<- merge(Vitalsigns16, HDC_source_code, by='Source_Name')
Vitalsigns16$Source_Name<-NULL

...

---
title: "Phenotyping_Demographic_Analysis"
output: html_document
author: "Miguel Angel Armengol de la Hoz"
date: "27 de febrero de 2018"
toc: TRUE
---

``{r}
library(dplyr)
library(summarytools)
library(readr)

```

```

```{r setup, include=FALSE}

#load("C:/BIG FILES/Phenotyping/HemodynamicProjectsDatasetFeb18.RData")

##IMPORTANT
#before converting date, run:
Sys.setlocale("LC_ALL","English")

load("C:/BIG FILES/Phenotyping/STS.All.Status.Uni.v3.Rda")
STS_All_Status_Uni <- STS.All.Status.Uni.v3

STS_All_Status_Uni$Surgery_Date<-as.Date(STS_All_Status_Uni$Surgery_Date, format =
"%d-%b-%y")

STS_2_73_July_2011_Jun_2014 <- read_delim("C:/BIG FILES/Phenotyping/new STS Generator
and README Feb18/STS_2_73_July_2011_Jun_2014.csv",
";", escape_double = FALSE, col_types = cols(Admission_Date = col_date(format =
"%m/%d/%y"),
Discharge_Date = col_date(format = "%m/%d/%y"),
Surgery_Date = col_date(format = "%m/%d/%y")),
trim_ws = TRUE)

STS_2_81_July_2014_June_2017 <- read_delim("C:/BIG FILES/Phenotyping/new STS
Generator and README Feb18/STS_2_81_July_2014_June_2017.csv",
";", escape_double = FALSE, col_types = cols(Admission_Date = col_date(format =
"%m/%d/%y"),
Discharge_Date = col_date(format = "%m/%d/%y"),
Surgery_Date = col_date(format = "%m/%d/%y")),
trim_ws = TRUE)

library(readxl)
data_superficial_wound_270_281_July2011_Jun_2014 <- read_excel("C:/BIG
FILES/Phenotyping/new STS Generator and README Feb18/Bala_HgbA1C data update with
superficial wound _270_281_dec 2016.xlsx",
col_types = c("text", "date", "date",
"date", "numeric", "numeric", "numeric",
"text", "numeric", "numeric", "numeric",
"numeric", "numeric", "numeric",
"text", "numeric", "text", "numeric",
"numeric", "numeric", "numeric",
"text", "numeric", "text", "text",
"text", "text", "numeric", "text",
"text", "text", "numeric", "numeric",
"text", "numeric", "numeric", "numeric",
"numeric", "numeric", "text", "numeric",
"numeric", "numeric", "numeric",
"numeric", "text", "numeric", "numeric",
"numeric", "numeric", "numeric",
"numeric", "numeric", "numeric",
"numeric", "text", "text", "text",
"text", "text", "text", "numeric",

```

```

      "numeric", "numeric", "numeric",
      "numeric", "numeric", "numeric",
      "numeric", "numeric", "text", "numeric",
      "numeric", "numeric", "numeric",
      "numeric", "numeric", "numeric",
      "numeric", "text"))

data_superficial_wound_270_281_July2014_Dec_2016 <- read_excel("C:/BIG
FILES/Phenotyping/new STS Generator and README Feb18/Bala_HgbA1C data update with
superficial wound _270_281_dec 2016.xlsx", sheet = "Jul 14-Dec16", col_types =
c("text",
  "date", "date", "date", "numeric",
  "text", "numeric", "text", "numeric",
  "numeric", "numeric", "numeric",
  "numeric", "numeric", "numeric",
  "numeric", "numeric", "numeric",
  "numeric", "text", "numeric", "text",
  "text", "text", "text", "text", "text",
  "text", "numeric", "text", "numeric",
  "text", "text", "text", "numeric",
  "text", "text", "text", "text", "text",
  "text", "text", "text", "text", "text",
  "text", "text", "numeric", "numeric",
  "text", "numeric", "text", "text",
  "text", "text", "text", "numeric",
  "numeric", "numeric", "numeric",
  "numeric", "text", "text",
  "text", "text", "text", "numeric",
  "numeric", "numeric", "numeric",
  "numeric", "numeric", "text", "text",
  "numeric", "numeric", "numeric",
  "numeric", "numeric", "numeric",
  "numeric", "numeric", "text"))
...

# Creating final uncorrupted full dataset

There are still some duplicated patients we need to remove

```{r}
library(dplyr)
library(plyr)
#In the plyr package there is a function rbind.fill that will merge data.frames and
introduce NA for empty cells

#the original dataset needs to be filtered until the next one is starting.
STS_All_Status_Uni_Jan_2008_Jun_2011<-subset(STS_All_Status_Uni,
format(as.Date(Surgery_Date))<='2011-06-31')

STS_full_dataset<-rbind.fill(STS_All_Status_Uni_Jan_2008_Jun_2011,
STS_2_73_July_2011_Jun_2014,STS_2_81_July_2014_June_2017)

```

```

#formatting dates properly for the left join
#to be runned only the first time
STS_full_dataset$Surgery_Date<-format(STS_full_dataset$Surgery_Date, "%Y-%m-%d")
data_superficial_wound_270_281_July2011_Jun_2014$Surgery_Date<-
format(data_superficial_wound_270_281_July2011_Jun_2014$Surgery_Date, "%Y-%m-%d")
data_superficial_wound_270_281_July2014_Dec_2016$Surgery_Date<-
format(data_superficial_wound_270_281_July2014_Dec_2016$Surgery_Date, "%Y-%m-%d")

#Cleaning PostOpSternalSuperficialInfect in
data_superficial_wound_270_281_July2011_Jun_2014
data_superficial_wound_270_281_July2011_Jun_2014$PostOpSternalSuperficialInfect[is.na
(data_superficial_wound_270_281_July2011_Jun_2014$PostOpSternalSuperficialInfect)]<-0
data_superficial_wound_270_281_July2011_Jun_2014$PostOpSternalSuperficialInfect[data_
superficial_wound_270_281_July2011_Jun_2014$PostOpSternalSuperficialInfect=='No']<-0
data_superficial_wound_270_281_July2011_Jun_2014$PostOpSternalSuperficialInfect[data_
superficial_wound_270_281_July2011_Jun_2014$PostOpSternalSuperficialInfect=='Yes']<-1

#Cleaning PostOpSternalSuperficialInfect in
data_superficial_wound_270_281_July2014_Dec_2016

data_superficial_wound_270_281_July2014_Dec_2016$PostOpSternalSuperficialInfect[is.na
(data_superficial_wound_270_281_July2014_Dec_2016$PostOpSternalSuperficialInfect)]<-0
data_superficial_wound_270_281_July2014_Dec_2016$PostOpSternalSuperficialInfect[data_
superficial_wound_270_281_July2014_Dec_2016$PostOpSternalSuperficialInfect=='No']<-0
data_superficial_wound_270_281_July2014_Dec_2016$PostOpSternalSuperficialInfect[data_
superficial_wound_270_281_July2014_Dec_2016$PostOpSternalSuperficialInfect=='Yes,
within 30 days of procedure']<-1

#performing left joins with the datasets containing wounds info.
STS_full_dataset<-left_join(STS_full_dataset,
subset(data_superficial_wound_270_281_July2011_Jun_2014,
select =
c("Surgery_Date", "Patient_ID", "PostOpSternalSuperficialInfect", "PostOpSternalMediasti
nitis", "Infect_Thoracotomy", "Infect_Leg")),
by=c("Surgery_Date", "Patient_ID"))

STS_full_dataset<-left_join(STS_full_dataset,
subset(data_superficial_wound_270_281_July2014_Dec_2016,
select =
c("Surgery_Date", "Patient_ID", "PostOpSternalSuperficialInfect", "PostOpCannulationSite
", "PostOpConduitHarvest")),
by=c("Surgery_Date", "Patient_ID"))

#Formating dates

STS_full_dataset$DateOfDeath<-as.Date(STS_full_dataset$DateOfDeath, "%d-%B-%y")
STS_full_dataset$Surgery_Date<-as.Date(STS_full_dataset$Surgery_Date, "%Y-%m-%d")
STS_full_dataset$Admission_Date<-as.Date(STS_full_dataset$Admission_Date, "%Y-%m-%d")
STS_full_dataset$Discharge_Date<-as.Date(STS_full_dataset$Discharge_Date, "%Y-%m-%d")

write.table(STS_full_dataset, 'STS_full_dataset.txt')
view(dfSummary(STS_full_dataset))

```

```

#AIMS dataset goes up to June 2016
STS_full_dataset_2008_2016<-subset(STS_full_dataset,
format(as.Date(Surgery_Date))<='2016-06-31')

...

# Removing duplications

There are still some duplicated patients we need to remove

```{r}
STS_full_dataset_2008_2016 <- STS_full_dataset_2008_2016[ !
rownames(STS_full_dataset_2008_2016) %in%
      which( duplicated(STS_full_dataset_2008_2016$Patient_ID)), ]
...

# STS

## Columns removing

```{r}
#some columns are causing problems, we remove them for the final dataset
#STS_full_dataset_2008_2016$actualsurgery<-NULL
STS_full_dataset_2008_2016$Comments<-NULL

#we remove columns we don't need
STS_full_dataset_2008_2016$morbmort<-NULL
STS_full_dataset_2008_2016$morbmortbinary<-NULL
STS_full_dataset_2008_2016$Surgery_Type_Code<-NULL
# STS_full_dataset_2008_2016$Surgery_Type_Descript<-NULL
STS_full_dataset_2008_2016$Num_Distals<-NULL

#some .bin columns might be incomplete so we remove them all
BINcolumns<-grep("bin", names(STS_full_dataset_2008_2016), value = TRUE)

for (name in names(STS_full_dataset_2008_2016)){
if (name %in% BINcolumns){
  STS_full_dataset_2008_2016[name]<-NULL
}
}

}

write.csv(STS_full_dataset_2008_2016,'STS_full_dataset_2008_2016.csv')
...

STS

```

```

## Filtering STS dataset

```{r}
#we exclude those patients not in our study Category and status
library(dplyr)
STS_filtered<-STS_full_dataset_2008_2016 %>%
#filter(grepl('CAB',Category)) %>%
filter(!grepl('Cancelled case',Category)) %>%
filter(grepl('Urgen|Elective', Status))

...

## Data Mapping STS dataset

### Exposures

```{r}
#we merge CAB categories

#Combine CABG and CABG +Other,
# Combine VALVE and VALVE + other
# Combine CABG +Valve and CABG+Valve+other

#Cleaning Diabetes
STS_filtered$Diabetes[is.na(STS_filtered$Diabetes)]<-0
STS_filtered$Diabetes[STS_filtered$Diabetes=='']<-0
STS_filtered$Diabetes[STS_filtered$Diabetes=='No']<-0
STS_filtered$Diabetes[STS_filtered$Diabetes=='Yes']<-1

#Cleaning RFDyslipidemia
STS_filtered$RFDyslipidemia[is.na(STS_filtered$RFDyslipidemia)]<-0
STS_filtered$RFDyslipidemia[STS_filtered$RFDyslipidemia=='']<-0
STS_filtered$RFDyslipidemia[STS_filtered$RFDyslipidemia=='No']<-0
STS_filtered$RFDyslipidemia[STS_filtered$RFDyslipidemia=='Unknown']<-0
STS_filtered$RFDyslipidemia[STS_filtered$RFDyslipidemia=='Yes']<-1

#Cleaning ChronicLungDisease
STS_filtered$ChronicLungDisease[is.na(STS_filtered$ChronicLungDisease)]<-0
STS_filtered$ChronicLungDisease[STS_filtered$ChronicLungDisease=='Lung disease
documen']<-1
STS_filtered$ChronicLungDisease[STS_filtered$ChronicLungDisease=='Mild']<-1
STS_filtered$ChronicLungDisease[STS_filtered$ChronicLungDisease=='Moderate']<-1
STS_filtered$ChronicLungDisease[STS_filtered$ChronicLungDisease=='Severe']<-1
STS_filtered$ChronicLungDisease[STS_filtered$ChronicLungDisease=='No']<-0
STS_filtered$ChronicLungDisease[STS_filtered$ChronicLungDisease=='']<-0
STS_filtered$ChronicLungDisease[STS_filtered$ChronicLungDisease=='Unknown']<-0

#Cleaning Dialysis
STS_filtered$Dialysis[STS_filtered$Dialysis=='No']<-0
STS_filtered$Dialysis[STS_filtered$Dialysis=='']<-0
STS_filtered$Dialysis[STS_filtered$Dialysis=='Yes']<-1
STS_filtered$Dialysis[is.na(STS_filtered$Dialysis)]<-0

```

```
#Cleaning PreOpMedBetaBlockers
STS_filtered$PreOpMedBetaBlockers[is.na(STS_filtered$PreOpMedBetaBlockers)]<-0
STS_filtered$PreOpMedBetaBlockers[STS_filtered$PreOpMedBetaBlockers=='']<-0
STS_filtered$PreOpMedBetaBlockers[STS_filtered$PreOpMedBetaBlockers=='Contraindicated']<-0
STS_filtered$PreOpMedBetaBlockers[STS_filtered$PreOpMedBetaBlockers=='Contraindicated / 0t Indicated']<-0
STS_filtered$PreOpMedBetaBlockers[STS_filtered$PreOpMedBetaBlockers=='Contraindicated / Not Indicated']<-0
STS_filtered$PreOpMedBetaBlockers[STS_filtered$PreOpMedBetaBlockers=='No']<-0
STS_filtered$PreOpMedBetaBlockers[STS_filtered$PreOpMedBetaBlockers=='']<-0
STS_filtered$PreOpMedBetaBlockers[STS_filtered$PreOpMedBetaBlockers=='Yes']<-1
STS_filtered$PreOpMedBetaBlockers<-as.factor(STS_filtered$PreOpMedBetaBlockers)
```

```
#Cleaning PreOpMedInotropes
STS_filtered$PreOpMedInotropes[is.na(STS_filtered$PreOpMedInotropes)]<-0
STS_filtered$PreOpMedInotropes[STS_filtered$PreOpMedInotropes=='']<-0
STS_filtered$PreOpMedInotropes[STS_filtered$PreOpMedInotropes=='Contraindicated']<-0
STS_filtered$PreOpMedInotropes[STS_filtered$PreOpMedInotropes=='Contraindicated / 0t Indicated']<-0
STS_filtered$PreOpMedInotropes[STS_filtered$PreOpMedInotropes=='Contraindicated / Not Indicated']<-0
STS_filtered$PreOpMedInotropes[STS_filtered$PreOpMedInotropes=='No']<-0
STS_filtered$PreOpMedInotropes[STS_filtered$PreOpMedInotropes=='Yes']<-1
STS_filtered$PreOpMedInotropes<-as.factor(STS_filtered$PreOpMedInotropes)
```

```
#Cleaning PreOpMedSteroids
STS_filtered$PreOpMedSteroids[is.na(STS_filtered$PreOpMedSteroids)]<-0
STS_filtered$PreOpMedSteroids[STS_filtered$PreOpMedSteroids=='']<-0
STS_filtered$PreOpMedSteroids[STS_filtered$PreOpMedSteroids=='Contraindicated']<-0
STS_filtered$PreOpMedSteroids[STS_filtered$PreOpMedSteroids=='Contraindicated / 0t Indicated']<-0
STS_filtered$PreOpMedSteroids[STS_filtered$PreOpMedSteroids=='Contraindicated / Not Indicated']<-0
STS_filtered$PreOpMedSteroids[STS_filtered$PreOpMedSteroids=='No']<-0
STS_filtered$PreOpMedSteroids[STS_filtered$PreOpMedSteroids=='Yes']<-1
STS_filtered$PreOpMedSteroids<-as.factor(STS_filtered$PreOpMedSteroids)
```

```
#Cleaning PreOpMedAspirin
STS_filtered$PreOpMedAspirin[is.na(STS_filtered$PreOpMedAspirin)]<-0
STS_filtered$PreOpMedAspirin[STS_filtered$PreOpMedAspirin=='']<-0
STS_filtered$PreOpMedAspirin[STS_filtered$PreOpMedAspirin=='Contraindicated']<-0
STS_filtered$PreOpMedAspirin[STS_filtered$PreOpMedAspirin=='Contraindicated / 0t Indicated']<-0
STS_filtered$PreOpMedAspirin[STS_filtered$PreOpMedAspirin=='Contraindicated / Not Indicated']<-0
STS_filtered$PreOpMedAspirin[STS_filtered$PreOpMedAspirin=='No']<-0
STS_filtered$PreOpMedAspirin[STS_filtered$PreOpMedAspirin=='Yes']<-1
```

```
#Cleaning PreOpMedLipidLowering
# AKA statins
STS_filtered$PreOpMedLipidLowering[is.na(STS_filtered$PreOpMedLipidLowering)]<-0
STS_filtered$PreOpMedLipidLowering[STS_filtered$PreOpMedLipidLowering=='Contraindicated']<-0
```



```

STS_filtered$PreOpMedLipidLowering[STS_filtered$PreOpMedLipidLowering=='Contraindicat
ed / 0t Indicated']<-0
STS_filtered$PreOpMedLipidLowering[STS_filtered$PreOpMedLipidLowering=='Contraindicat
ed / Not Indicated']<-0
STS_filtered$PreOpMedLipidLowering[STS_filtered$PreOpMedLipidLowering=='No']<-0
STS_filtered$PreOpMedLipidLowering[STS_filtered$PreOpMedLipidLowering=='Yes']<-1
STS_filtered$PreOpMedLipidLowering[STS_filtered$PreOpMedLipidLowering=='Unknown']<-0
STS_filtered$PreOpMedLipidLowering[STS_filtered$PreOpMedLipidLowering=='']<-0
STS_filtered$PreOpMedLipidLowering<-as.factor(STS_filtered$PreOpMedLipidLowering)

#Cleaning TobaccoUse
STS_filtered$TobaccoUse[is.na(STS_filtered$TobaccoUse)]<-0
STS_filtered$TobaccoUse[STS_filtered$TobaccoUse=='']<- 0
STS_filtered$TobaccoUse<-as.character(STS_filtered$TobaccoUse)
STS_filtered$TobaccoUse[STS_filtered$TobaccoUse=='Current every day smoker'|
STS_filtered$TobaccoUse=='Current some day smoker'|
STS_filtered$TobaccoUse=='Former smoker' |
STS_filtered$TobaccoUse=='Smoker, current status (frequency) unknown']<- 1
STS_filtered$TobaccoUse[STS_filtered$TobaccoUse=='Never smoker']<- 0
STS_filtered$TobaccoUse[STS_filtered$TobaccoUse=='Smoking status unknown']<-0

#Cleaning CHF
STS_filtered$CHF[is.na(STS_filtered$CHF)]<-0
STS_filtered$CHF[STS_filtered$CHF=='']<-0
STS_filtered$CHF[STS_filtered$CHF=='No']<-0
STS_filtered$CHF[STS_filtered$CHF=='Yes']<-1

#Cleaning HeartFailure2wks
STS_filtered$HeartFailure2wks[is.na(STS_filtered$HeartFailure2wks)]<-0
STS_filtered$HeartFailure2wks[STS_filtered$HeartFailure2wks=='']<-0
STS_filtered$HeartFailure2wks[STS_filtered$HeartFailure2wks=='No']<-0
STS_filtered$HeartFailure2wks[STS_filtered$HeartFailure2wks=='Yes']<-1

#Cleaning Hypertension
STS_filtered$Hypertension[is.na(STS_filtered$Hypertension)]<-0
STS_filtered$Hypertension[STS_filtered$Hypertension=='']<-0
STS_filtered$Hypertension[STS_filtered$Hypertension=='No']<-0
STS_filtered$Hypertension[STS_filtered$Hypertension=='Yes']<-1

#Cleaning RFHypertension
STS_filtered$RFHypertension[is.na(STS_filtered$RFHypertension)]<-0
STS_filtered$RFHypertension[STS_filtered$RFHypertension=='']<-0
STS_filtered$RFHypertension[STS_filtered$RFHypertension=='Unknown']<-0
STS_filtered$RFHypertension[STS_filtered$RFHypertension=='No']<-0
STS_filtered$RFHypertension[STS_filtered$RFHypertension=='Yes']<-1

#Cleaning PreviousMI
STS_filtered$PreviousMI[is.na(STS_filtered$PreviousMI)]<-0
STS_filtered$PreviousMI[STS_filtered$PreviousMI=='']<-0
STS_filtered$PreviousMI[STS_filtered$PreviousMI=='No']<-0
STS_filtered$PreviousMI[STS_filtered$PreviousMI=='Yes']<-1

#Cleaning PriorMI
STS_filtered$PriorMI[is.na(STS_filtered$PriorMI)]<-0
STS_filtered$PriorMI[STS_filtered$PriorMI=='']<-0

```

```

STS_filtered$PriorMI[STS_filtered$PriorMI=='No']<-0
STS_filtered$PriorMI[STS_filtered$PriorMI=='Yes']<-1

#Drugs

#Cleaning PreOpMedACE_ARBINhibitors
STS_filtered$PreOpMedACE_ARBINhibitors[STS_filtered$PreOpMedACE_ARBINhibitors=='Contraindicated']<-0
STS_filtered$PreOpMedACE_ARBINhibitors[STS_filtered$PreOpMedACE_ARBINhibitors=='']<-0
STS_filtered$PreOpMedACE_ARBINhibitors[STS_filtered$PreOpMedACE_ARBINhibitors=='Contraindicated / 0t Indicated']<-0
STS_filtered$PreOpMedACE_ARBINhibitors[STS_filtered$PreOpMedACE_ARBINhibitors=='Contraindicated / Not Indicated']<-0
STS_filtered$PreOpMedACE_ARBINhibitors[STS_filtered$PreOpMedACE_ARBINhibitors=='No']<-0
STS_filtered$PreOpMedACE_ARBINhibitors[STS_filtered$PreOpMedACE_ARBINhibitors=='Yes']<-1
STS_filtered$PreOpMedACE_ARBINhibitors<-
as.factor(STS_filtered$PreOpMedACE_ARBINhibitors)
STS_filtered$PreOpMedACE_ARBINhibitors[is.na(STS_filtered$PreOpMedACE_ARBINhibitors)]<-0

#Cleaning PreOpMedACEIARBI48hrs
STS_filtered$PreOpMedACEIARBI48hrs[STS_filtered$PreOpMedACEIARBI48hrs=='Contraindicated']<-0
STS_filtered$PreOpMedACEIARBI48hrs[STS_filtered$PreOpMedACEIARBI48hrs=='']<-0
STS_filtered$PreOpMedACEIARBI48hrs[STS_filtered$PreOpMedACEIARBI48hrs=='No']<-0
STS_filtered$PreOpMedACEIARBI48hrs[STS_filtered$PreOpMedACEIARBI48hrs=='Yes']<-1
STS_filtered$PreOpMedACEIARBI48hrs[is.na(STS_filtered$PreOpMedACEIARBI48hrs)]<-0
STS_filtered$PreOpMedACEIARBI48hrs<-as.factor(STS_filtered$PreOpMedACEIARBI48hrs)

...

###Outcomes
```{r}

#Cleaning Reop_Bleeding
STS_filtered$Reop_Bleeding[is.na(STS_filtered$Reop_Bleeding)]<-0

#Cleaning Infect_Sternum_Deep
STS_filtered$Infect_Sternum_Deep[is.na(STS_filtered$Infect_Sternum_Deep)]<-0
STS_filtered$Infect_Sternum_Deep[STS_filtered$Infect_Sternum_Deep=='']<-0
STS_filtered$Infect_Sternum_Deep[STS_filtered$Infect_Sternum_Deep=='No']<-0
STS_filtered$Infect_Sternum_Deep[STS_filtered$Infect_Sternum_Deep=='Yes']<-1

#Cleaning PostOpSternalMediastinitis
STS_filtered$PostOpSternalMediastinitis[is.na(STS_filtered$PostOpSternalMediastinitis)]<-0
STS_filtered$PostOpSternalMediastinitis[STS_filtered$PostOpSternalMediastinitis=='']<-0
STS_filtered$PostOpSternalMediastinitis[STS_filtered$PostOpSternalMediastinitis=='No']<-0
STS_filtered$PostOpSternalMediastinitis[STS_filtered$PostOpSternalMediastinitis=='Yes']<-1

```

```

#Cleaning Infect_Thoracotomy
STS_filtered$Infect_Thoracotomy[is.na(STS_filtered$Infect_Thoracotomy)]<-0
STS_filtered$Infect_Thoracotomy[STS_filtered$Infect_Thoracotomy=='']<-0
STS_filtered$Infect_Thoracotomy[STS_filtered$Infect_Thoracotomy=='No']<-0
STS_filtered$Infect_Thoracotomy[STS_filtered$Infect_Thoracotomy=='Yes']<-1

#Cleaning Infect_Leg
STS_filtered$Infect_Leg[is.na(STS_filtered$Infect_Leg)]<-0
STS_filtered$Infect_Leg[STS_filtered$Infect_Leg=='']<-0
STS_filtered$Infect_Leg[STS_filtered$Infect_Leg=='No']<-0
STS_filtered$Infect_Leg[STS_filtered$Infect_Leg=='Yes']<-1

#Cleaning PostOpSternalSuperficialInfect.x
STS_filtered$PostOpSternalSuperficialInfect.x[is.na(STS_filtered$PostOpSternalSuperfi
cialInfect.x)]<-0
STS_filtered$PostOpSternalSuperficialInfect.x[STS_filtered$PostOpSternalSuperficialIn
fect.x=='']<-0
STS_filtered$PostOpSternalSuperficialInfect.x[STS_filtered$PostOpSternalSuperficialIn
fect.x=='No']<-0
STS_filtered$PostOpSternalSuperficialInfect.x[STS_filtered$PostOpSternalSuperficialIn
fect.x=='Yes']<-1

#Cleaning PostOpSternalSuperficialInfect.y
STS_filtered$PostOpSternalSuperficialInfect.y[is.na(STS_filtered$PostOpSternalSuperfi
cialInfect.y)]<-0
STS_filtered$PostOpSternalSuperficialInfect.y[STS_filtered$PostOpSternalSuperficialIn
fect.y=='']<-0
STS_filtered$PostOpSternalSuperficialInfect.y[STS_filtered$PostOpSternalSuperficialIn
fect.y=='No']<-0
STS_filtered$PostOpSternalSuperficialInfect.y[STS_filtered$PostOpSternalSuperficialIn
fect.y=='Yes']<-1
STS_filtered$PostOpSternalSuperficialInfect.y<-
as.numeric(STS_filtered$PostOpSternalSuperficialInfect.y)

#Cleaning PostOpDeepSternalMediastin
STS_filtered$PostOpDeepSternalMediastin[is.na(STS_filtered$PostOpDeepSternalMediastin
)]<-0
STS_filtered$PostOpDeepSternalMediastin[STS_filtered$PostOpDeepSternalMediastin=='']<
-0
STS_filtered$PostOpDeepSternalMediastin[STS_filtered$PostOpDeepSternalMediastin=='No'
]<-0
STS_filtered$PostOpDeepSternalMediastin[STS_filtered$PostOpDeepSternalMediastin=='Yes
']<-1
STS_filtered$PostOpDeepSternalMediastin<-
as.numeric(STS_filtered$PostOpDeepSternalMediastin)

#Cleaning PostOpCannulationSite
STS_filtered$PostOpCannulationSite[is.na(STS_filtered$PostOpCannulationSite)]<-0
STS_filtered$PostOpCannulationSite[STS_filtered$PostOpCannulationSite=='']<-0
STS_filtered$PostOpCannulationSite[STS_filtered$PostOpCannulationSite=='No']<-0
STS_filtered$PostOpCannulationSite[STS_filtered$PostOpCannulationSite=='Yes']<-1
STS_filtered$PostOpCannulationSite<-as.numeric(STS_filtered$PostOpCannulationSite)

```

```

#Cleaning PostOpConduitHarvest
STS_filtered$PostOpConduitHarvest[is.na(STS_filtered$PostOpConduitHarvest)]<-0
STS_filtered$PostOpConduitHarvest[STS_filtered$PostOpConduitHarvest=='']<-0
STS_filtered$PostOpConduitHarvest[STS_filtered$PostOpConduitHarvest=='No']<-0
STS_filtered$PostOpConduitHarvest[STS_filtered$PostOpConduitHarvest=='Yes, within 30
days of procedure']<-1
STS_filtered$PostOpConduitHarvest<-as.numeric(STS_filtered$PostOpConduitHarvest)

#Cleaning Neuro_Stroke_Permanent
STS_filtered$Neuro_Stroke_Permanent[is.na(STS_filtered$Neuro_Stroke_Permanent)]<-0
STS_filtered$Neuro_Stroke_Permanent[STS_filtered$Neuro_Stroke_Permanent=='']<-0
STS_filtered$Neuro_Stroke_Permanent[STS_filtered$Neuro_Stroke_Permanent=='No']<-0
STS_filtered$Neuro_Stroke_Permanent[STS_filtered$Neuro_Stroke_Permanent=='Yes']<-1

#Cleaning PostOpNeuroStrokePermanent
STS_filtered$PostOpNeuroStrokePermanent[is.na(STS_filtered$PostOpNeuroStrokePermanent
)]<-0
STS_filtered$PostOpNeuroStrokePermanent[STS_filtered$PostOpNeuroStrokePermanent=='']<
-0
STS_filtered$PostOpNeuroStrokePermanent[STS_filtered$PostOpNeuroStrokePermanent=='No'
]<-0
STS_filtered$PostOpNeuroStrokePermanent[STS_filtered$PostOpNeuroStrokePermanent=='Yes
, embolic']<-1
STS_filtered$PostOpNeuroStrokePermanent[STS_filtered$PostOpNeuroStrokePermanent=='Yes
, hemorrhagic']<-1

#Cleaning Pulm_Pneumonia
STS_filtered$Pulm_Pneumonia[is.na(STS_filtered$Pulm_Pneumonia)]<-0
STS_filtered$Pulm_Pneumonia[STS_filtered$Pulm_Pneumonia=='']<-0

#Cleaning Renal_Failure
STS_filtered$Renal_Failure[is.na(STS_filtered$Renal_Failure)]<-0
STS_filtered$Renal_Failure[STS_filtered$Renal_Failure=='']<-0

#Cleaning Oth_Tamponade
STS_filtered$Oth_Tamponade[is.na(STS_filtered$Oth_Tamponade)]<-0
STS_filtered$Oth_Tamponade[STS_filtered$Oth_Tamponade=='']<-0

#Cleaning Death
STS_filtered$Death[is.na(STS_filtered$Death)]<-0
STS_filtered$Death[STS_filtered$Death=='']<-0

#Cleaning Oth_AFib
STS_filtered$Oth_AFib[is.na(STS_filtered$Oth_AFib)]<-0
STS_filtered$Oth_AFib[STS_filtered$Oth_AFib=='']<-0

listVars<-c("Gender",
            "Age",
            "Category",

            #Background diseases
            "Diabetes",

```

```

"RFDyslipidemia",
"Hypertension_final",
"Smoking",
"Congestive_Heart_Failure",
"Previous_myocardial_infarction",
"ChronicLungDisease",
"Dialysis",

#Preoperative medications
"PreOpMedBetaBlockers",
"PreOpMedACEI_or_ARBs",
"PreOpMedInotropes",
"PreOpMedSteroids",
"PreOpMedAspirin",
"PreOpMedLipidLowering"
)

# # There are 9 patients always missing baseline variables we are going to remove
them
# patientsmissingbaselinevars<-STS_filtered$Patient_ID[is.na(STS_filtered$Diabetes)]
#
# STS_filtered<-STS_filtered[!STS_filtered$Patient_ID %in%
patientsmissingbaselinevars, ]

# #we correct missing values
# #STS columns NA=NULL, In the rest NA=0
#
# #we are going to store the columns id which contains the string 'STS' because in
these columns NA means unknown
# STScolumns<-grep("STS", names(STS_filtered), value = TRUE)
# #other Columns where NA means unknown
# STScolumns<-append(STScolumns, c('Diabetes', 'PreviousMI',
#
# "PreOpMedBetaBlockers",
# "PreOpMedACE_ARBInhibitors",
# "PreOpMedInotropes",
# "PreOpMedSteroids",
# "PreOpMedAspirin",
# "PreOpMedLipidLowering"))
# STScolumns<-which(names(STS_filtered) %in% (STScolumns))
#
# #Columns where NA means unknown
#
# STS_filtered$Diabetes[is.na(STS_filtered$Diabetes)]<-'Unknown'
# STS_filtered$ChronicLungDisease[is.na(STS_filtered$ChronicLungDisease)]<-'Unknown'
# STS_filtered$TobaccoUse[is.na(STS_filtered$TobaccoUse)]<-'Smoking status unknown'
# STS_filtered$RFDyslipidemia[is.na(STS_filtered$RFDyslipidemia)]<-'Unknown'

#For those columns where NA doesn't mean unknown replace NA with 0

# for (i in 1:length(colnames(STS_filtered))){
#   if (i %in% STScolumns){
#

```

```

#   }
#   else(STS_filtered[,i][is.na(STS_filtered[,i])] <- 0)
# }

#convert character columns to factors
#charactercolumns<-sapply(STS_filtered, is.character)

#STS_filtered[charactercolumns] <- lapply(STS_filtered[charactercolumns], factor)
```

Converting columns to the right format

```{r}
STS_filtered$PreOpMedACE_ARBIInhibitors<-
as.character(STS_filtered$PreOpMedACE_ARBIInhibitors)
STS_filtered$PreOpMedBetaBlockers<-as.character(STS_filtered$PreOpMedBetaBlockers)
STS_filtered$PreOpMedLipidLowering<-as.character(STS_filtered$PreOpMedLipidLowering)
```

Creating new columns with constant data for the period 2008-2016

```{r}
#AIMS dataset goes up to June 2016

#Creating Stroke column

STS_filtered['Stroke']<-NA

STS_filtered$Stroke<-if_else(
  format(as.Date(STS_filtered$Surgery_Date))<='2014-06-31',
  STS_filtered$Neuro_Stroke_Permanent ,
  STS_filtered$PostOpNeuroStrokePermanent
)

STS_filtered$Stroke[is.na(STS_filtered$Stroke)]<-0

#Creating Smoking column

STS_filtered['Smoking']<-NA

STS_filtered$Smoking<-if_else(
  format(as.Date(STS_filtered$Surgery_Date))<='2014-06-31',
  STS_filtered$RFCurrentRecentCigarette ,
  as.integer(STS_filtered$TobaccoUse)
)
STS_filtered$Smoking[is.na(STS_filtered$Smoking)]<-0

#Creating Congestive_Heart_Failure column

STS_filtered['Congestive_Heart_Failure']<-NA

STS_filtered$Congestive_Heart_Failure<-if_else(
  format(as.Date(STS_filtered$Surgery_Date))<='2014-06-31',

```

```

    STS_filtered$CHF      ,
    STS_filtered$HeartFailure2wks
  )
STS_filtered$Congestive_Heart_Failure[is.na(STS_filtered$Congestive_Heart_Failure)]<-
0
#Creating Hypertension_final column

STS_filtered['Hypertension_final']<-NA

STS_filtered$Hypertension_final<-if_else(
  format(as.Date(STS_filtered$Surgery_Date))<='2014-06-31',
  STS_filtered$Hypertension      ,
  STS_filtered$RFHypertension
)
STS_filtered$Hypertension_final[is.na(STS_filtered$Hypertension_final)]<-0
#Creating Previous_myocardial_infarction column

STS_filtered['Previous_myocardial_infarction']<-NA

STS_filtered$Previous_myocardial_infarction<-if_else(
  format(as.Date(STS_filtered$Surgery_Date))<='2014-06-31',
  STS_filtered$PreviousMI      ,
  STS_filtered$PriorMI
)
STS_filtered$Previous_myocardial_infarction[is.na(STS_filtered$Previous_myocardial_in
farction)]<-0

#Creating PreOpMedACEI_or_ARBs column

STS_filtered['PreOpMedACEI_or_ARBs']<-NA

STS_filtered$PreOpMedACEI_or_ARBs<-if_else(
  format(as.Date(STS_filtered$Surgery_Date))<='2011-06-31',
  STS_filtered$PreOpMedACE_ARBINhibitors      ,
  as.character(STS_filtered$PreOpMedACEIARBI48hrs)
)
STS_filtered$PreOpMedACEI_or_ARBs[is.na(STS_filtered$PreOpMedACEI_or_ARBs)]<-0

#Creating Infection column

# Post-Op-Sternal-Superficial Wound Infection
# Short Name: CSternalSupInf

#PostOpDeepSternalMediastin only contains data from 2014
STS_filtered['Infection']<-NA

STS_filtered$Infection<-if_else(

```

```

format(as.Date(STS_filtered$Surgery_Date))<='2011-06-
31',as.numeric(STS_filtered$Infect_Sternum_Deep) ,
  if_else(
    format(as.Date(STS_filtered$Surgery_Date))<='2014-06-31',

if_else(STS_filtered$PostOpSternalSuperficialInfect.x+STS_filtered$Infect_Thoracotomy
+ STS_filtered$PostOpSternalMediastinitis>0,1,0) ,

if_else(STS_filtered$PostOpSternalSuperficialInfect.y+STS_filtered$PostOpCannulationS
ite+ STS_filtered$PostOpConduitHarvest>0,1,0)
)
)

STS_filtered$Infection[is.na(STS_filtered$Infection)]<-0

#Cleaning
#STS_filtered$Deep_Sternal_Infection[is.na(STS_filtered$Deep_Sternal_Infection)]<-0
...

##Data Mapping STS new created columns
```{r}
#Cleaning Congestive_Heart_Failure
STS_filtered$Congestive_Heart_Failure[is.na(STS_filtered$Congestive_Heart_Failure)]<-
0
STS_filtered$Congestive_Heart_Failure[STS_filtered$Congestive_Heart_Failure=='']<-0
STS_filtered$Congestive_Heart_Failure[STS_filtered$Congestive_Heart_Failure=='Unknown
']<-0

#Cleaning Previous_myocardial_infarction
STS_filtered$Previous_myocardial_infarction[is.na(STS_filtered$Previous_myocardial_in
farction)]<-0
STS_filtered$Previous_myocardial_infarction[STS_filtered$Previous_myocardial_infarcti
on=='']<-0
STS_filtered$Previous_myocardial_infarction[STS_filtered$Previous_myocardial_infarcti
on=='Unknown']<-0

...

#Processing Outcomes
```{r}

#We remove Oth_AFib since it is too big compared to other outcomes hence our model
would only be predicting it among all the others outcomes composing AnyMAE

listOuts<-
c("Reop_Bleeding","Infection","Stroke","Pulm_Pneumonia","Renal_Failure","Oth_Tamponad
e","Death")
#all Outs need to be numeric
for (name in names(STS_filtered[,listOuts])) {
  print(name)

```



```

STS_filtered[,name]<-as.integer(STS_filtered[,name])
}
STS_filtered$Stroke[is.na(STS_filtered$Stroke)]<-0
...

# Any Mayor Adverse Event Creation
```{r}
#We create an extra column in case any Mayor Adverse Event is ocuring during surgery

#Those events are:
#
"Reop_Bleeding","Infection","Stroke","Pulm_Pneumonia","Renal_Failure","Oth_Tamponade"
#,"Death"

multi.fun <- function(x) {cbind(freq = table(x), percentage =
round(prop.table(table(x))*100,2))}

STS_filtered['Any_MAE']<-rowSums(STS_filtered[,listOuts])

STS_filtered$Any_MAE<-ifelse(STS_filtered$Any_MAE==0,0,1)

STS_filtered['Total_MAE']<-rowSums(STS_filtered[,listOuts])
...

#initial tests
```{r}
multi.fun <- function(x) {cbind(freq = table(x), percentage =
round(prop.table(table(x))*100,2))}

for (name in listVars){
  print(name)
  print(multi.fun(STS_filtered[, name]))
}

# Check if there is any empty column for a certain year
for (name in names(STS_filtered[,c(listOuts)])) {
  for (year in 2008:2016){
    filtereddata<-subset(STS_filtered, format(as.Date(Surgery_Date),"%Y")==year)
    cat('\n')
    cat(year)
    cat('\n')
    cat(name)
    print(multi.fun(filtereddata[name]))
  }
}
# write(name,file="printingMissingDataColumns.txt",append=TRUE)
# write(year,file="printingMissingDataColumns.txt",append=TRUE)
# write("\n",file="printingMissingDataColumns.txt",append=TRUE)

```

```

# write.table(tablessummary,file="printingMissingDataColumns.txt",append=TRUE,sep =
"\t",quote = FALSE, row.names = FALSE)
    }

    }

    for (year in 2008:2016){
      filtereddata<-subset(STS_filtered, format(as.Date(Surgery_Date),"%Y")==year)
      cat('\n')
      cat(year)
      cat('\n')
      print(multi.fun(filtereddata$Infection ))
    }

for (name in names(STS_filtered[,c(listVars)])) {
  cat('\n')
  cat(name)
  print(multi.fun(STS_filtered[name]))
  print(summary(STS_filtered[name]))
# write(name,file="printingMissingDataColumns.txt",append=TRUE)
# write(year,file="printingMissingDataColumns.txt",append=TRUE)
# write("\n",file="printingMissingDataColumns.txt",append=TRUE)
# write.table(tablessummary,file="printingMissingDataColumns.txt",append=TRUE,sep =
"\t",quote = FALSE, row.names = FALSE)
    }

...

---
title: "3_HemodynamicAIMS_datasets_creation"
author: "Miguel Ángel Armengol de la Hoz"
date: "27 de febrero de 2018"
output: html_document
toc: TRUE
---

```{r setup, include=FALSE}
library(dplyr)
#load("C:/BIG FILES/Phenotyping/HemodynamicProjectsDatasetFeb18.RData")
```

#AIMS

#Creatintg Lists to analyze
```{r}
ComboList <- list(Combo08, Combo09,
Combo10,Combo11,Combo12,Combo13,Combo14,Combo15,Combo16)
names(ComboList)<- c("Combo08", "Combo09", "Combo10", "Combo11", "Combo12",
"Combo13","Combo14","Combo15", "Combo16")

TextList <- list(Text08, Text09, Text10,Text11,Text12,Text13,Text14,Text15,Text16)

```

```
names(TextList)<- c("Text08", "Text09", "Text10", "Text11", "Text12",
"Text13","Text14","Text15", "Text16")
```

```
VitalsignsList <- list(Vitalsigns08, Vitalsigns09,
Vitalsigns10,Vitalsigns11,Vitalsigns12,Vitalsigns13,Vitalsigns14,Vitalsigns15,Vitalsigns16)
names( VitalsignsList)<- c("Vitalsigns08", "Vitalsigns09", "Vitalsigns10",
"Vitalsigns11", "Vitalsigns12", "Vitalsigns13","Vitalsigns14","Vitalsigns15",
"Vitalsigns16")
```

```
EventsList <- list(Events08, Events09,
Events10,Events11,Events12,Events13,Events14,Events15,Events16)
names( EventsList)<- c("Events08", "Events09", "Events10", "Events11", "Events12",
"Events13","Events14","Events15", "Events16")
```

```
InfusionDrugsList <- list(InfusionDrugs08, InfusionDrugs09,
InfusionDrugs10,InfusionDrugs11,InfusionDrugs12,InfusionDrugs13,InfusionDrugs14,InfusionDrugs15,InfusionDrugs16)
names( InfusionDrugsList)<- c("InfusionDrugs08", "InfusionDrugs09",
"InfusionDrugs10", "InfusionDrugs11", "InfusionDrugs12",
"InfusionDrugs13","InfusionDrugs14","InfusionDrugs15", "InfusionDrugs16")
```

```
DateList <- list(Date08, Date09, Date10,Date11,Date12,Date13,Date14,Date15,Date16)
names( DateList)<- c("Date08", "Date09", "Date10", "Date11", "Date12",
"Date13","Date14","Date15", "Date16")
```

```
BolusDrugsList <- list(BolusDrugs08, BolusDrugs09,
BolusDrugs10,BolusDrugs11,BolusDrugs12,BolusDrugs13,BolusDrugs14,BolusDrugs15,BolusDrugs16)
names( BolusDrugsList)<- c("BolusDrugs08", "BolusDrugs09", "BolusDrugs10",
"BolusDrugs11", "BolusDrugs12", "BolusDrugs13","BolusDrugs14","BolusDrugs15",
"BolusDrugs16")
```

```
...
```

```
#Anesthesiologists
```

```
``{r}
```

```
totalnumberOFAnesthesiologists<-union_all(
  Combo08$C_Item[Combo08$Label=='Attending Anesthesiologist'],
  Combo09$C_Item[Combo09$Label=='Attending Anesthesiologist'],
  Combo09$C_Item[Combo10$Label=='Attending Anesthesiologist'],
  Combo09$C_Item[Combo11$Label=='Attending Anesthesiologist'],
  Combo09$C_Item[Combo12$Label=='Attending Anesthesiologist'],
  Combo09$C_Item[Combo13$Label=='Attending Anesthesiologist'],
  Combo09$C_Item[Combo14$Label=='Attending Anesthesiologist'],
  Combo09$C_Item[Combo15$Label=='Attending Anesthesiologist'],
  Combo09$C_Item[Combo16$Label=='Attending Anesthesiologist'])
```

```
totalnumberOFResidents<-union_all(
  Combo08$C_Item[Combo08$Label=='Resident/CRNA'],
  Combo09$C_Item[Combo09$Label=='Resident/CRNA'],
  Combo09$C_Item[Combo10$Label=='Resident/CRNA'],
  Combo09$C_Item[Combo11$Label=='Resident/CRNA'],
  Combo09$C_Item[Combo12$Label=='Resident/CRNA'],
```

```

Combo09$C_Item[Combo13$Label=='Resident/CRNA'],
Combo09$C_Item[Combo14$Label=='Resident/CRNA'],
Combo09$C_Item[Combo15$Label=='Resident/CRNA'],
Combo09$C_Item[Combo16$Label=='Resident/CRNA'])

##Surgeon

totalnumberOFSurgeons<-union_all(
  Combo08$C_Item[Combo08$Label=='Surgeon'],
  Combo09$C_Item[Combo09$Label=='Surgeon'],
  Combo09$C_Item[Combo10$Label=='Surgeon'],
  Combo09$C_Item[Combo11$Label=='Surgeon'],
  Combo09$C_Item[Combo12$Label=='Surgeon'],
  Combo09$C_Item[Combo13$Label=='Surgeon'],
  Combo09$C_Item[Combo14$Label=='Surgeon'],
  Combo09$C_Item[Combo15$Label=='Surgeon'],
  Combo09$C_Item[Combo16$Label=='Surgeon'])

...

---
title: "4_STS_and_AIMS_merge CasesofInterestDF creation"
author: "Miguel Angel Armengol de la Hoz"
date: "27 de febrero de 2018"
output: html_document
toc: TRUE
---

```{r setup, include=FALSE}
#Data Load
#load("C:/BIG FILES/Phenotyping/HemodynamicProjectsDatasetFeb18.RData")
#setwd("C:/Users/Miguel_Angel/Documents/MEGA/Boston/BIDMC - Harvard/Phenotyping
Feb18")
```

Cases of interest

```{r eval=FALSE, include=FALSE}
#Already formatted Date of Surgery to Join with STS
Text08$Time<-format(Text08$Time, "%Y-%m-%d")
Text09$Time<-format(Text09$Time, "%Y-%m-%d")
Text10$Time<-format(Text10$Time, "%Y-%m-%d")
Text11$Time<-format(Text11$Time, "%Y-%m-%d")
Text12$Time<-format(Text12$Time, "%Y-%m-%d")
Text13$Time<-format(Text13$Time, "%Y-%m-%d")
Text14$Time<-format(Text14$Time, "%Y-%m-%d")
Text15$Time<-format(Text15$Time, "%Y-%m-%d")
Text16$Time<-format(Text16$Time, "%Y-%m-%d")
```

STS and AIMS join

```

We need the table Text to be able to get the Case\_Name for each procedure so we are later able to join by Vital\_Signs.  
This way we can link 5724 Cases between STS and AIMS.

```
``{r}
```

```
Cases_of_Interest08<-merge(x=STS_filtered, y= select(      Text08,-Label,-ItemID) ,
by.x=c( "Surgery_Date","Patient_ID"), by.y=c("Time","Text"))
Cases_of_Interest09<-merge(x=STS_filtered, y= select(      Text09,-Label,-ItemID) ,
by.x=c( "Surgery_Date","Patient_ID"), by.y=c("Time","Text"))
Cases_of_Interest10<-merge(x=STS_filtered, y= select(      Text10,-Label,-ItemID) ,
by.x=c( "Surgery_Date","Patient_ID"), by.y=c("Time","Text"))
Cases_of_Interest11<-merge(x=STS_filtered, y= select(      Text11,-Label,-ItemID) ,
by.x=c( "Surgery_Date","Patient_ID"), by.y=c("Time","Text"))
Cases_of_Interest12<-merge(x=STS_filtered, y= select(      Text12,-Label,-ItemID) ,
by.x=c( "Surgery_Date","Patient_ID"), by.y=c("Time","Text"))
Cases_of_Interest13<-merge(x=STS_filtered, y= select(      Text13,-Label,-ItemID) ,
by.x=c( "Surgery_Date","Patient_ID"), by.y=c("Time","Text"))
Cases_of_Interest14<-merge(x=STS_filtered, y= select(      Text14,-Label,-ItemID) ,
by.x=c( "Surgery_Date","Patient_ID"), by.y=c("Time","Text"))
Cases_of_Interest15<-merge(x=STS_filtered, y= select(      Text15,-Label,-ItemID) ,
by.x=c( "Surgery_Date","Patient_ID"), by.y=c("Time","Text"))
Cases_of_Interest16<-merge(x=STS_filtered, y= select(      Text16,-Label,-ItemID) ,
by.x=c( "Surgery_Date","Patient_ID"), by.y=c("Time","Text"))

# Cases_of_InteresList <- list(Cases_of_Interest08, Cases_of_Interest09,
Cases_of_Interest10,Cases_of_Interest11,Cases_of_Interest12,Cases_of_Interest13,Cases
_of_Interest14,Cases_of_Interest15,Cases_of_Interest16)
# names( Cases_of_InteresList)<- c("Cases_of_Interest08", "Cases_of_Interest09",
"Cases_of_Interest10", "Cases_of_Interest11", "Cases_of_Interest12",
"Cases_of_Interest13","Cases_of_Interest14","Cases_of_Interest15",
"Cases_of_Interest16")

VectorwithCasesofInterest<-union_all(Cases_of_Interest08$Case_Name,
      Cases_of_Interest09$Case_Name,
      Cases_of_Interest10$Case_Name,
      Cases_of_Interest11$Case_Name,
      Cases_of_Interest12$Case_Name,
      Cases_of_Interest13$Case_Name,
      Cases_of_Interest14$Case_Name,
      Cases_of_Interest15$Case_Name,
      Cases_of_Interest16$Case_Name
)

VectorwithCasesofInterest<-unique(VectorwithCasesofInterest)

CasesofInterestDF<-union_all(Cases_of_Interest08, Cases_of_Interest09)
CasesofInterestDF<-union_all(CasesofInterestDF, Cases_of_Interest10)
CasesofInterestDF<-union_all(CasesofInterestDF, Cases_of_Interest11)
CasesofInterestDF<-union_all(CasesofInterestDF, Cases_of_Interest12)
CasesofInterestDF<-union_all(CasesofInterestDF, Cases_of_Interest13)
CasesofInterestDF<-union_all(CasesofInterestDF, Cases_of_Interest14)
CasesofInterestDF<-union_all(CasesofInterestDF, Cases_of_Interest15)
CasesofInterestDF<-union_all(CasesofInterestDF, Cases_of_Interest16)
```

```

#write.csv(CasesofInterestDF, 'CasesofInterestDF.csv')

write.xlsx(CasesofInterestDF[,c('Patient_ID','Surgery_Date')], 'patientslist.xlsx')
```

---
title: "5_Hemodynamic_Artifact_Removal_algorithms and MAP recalculation"
author: "Miguel Angel Armengol de la Hoz"
output:
  html_document:
    toc: true
    theme: united

```{r}
#Data load

library(readr)
#load("C:/BIG FILES/Phenotyping/HemodynamicProjectsDatasetFeb18.RData")
```

After doing some research in the department, we found out ABP-Mean is being
calculated using an integral directly in the monitor, thus is our most reliable
source; therefore, we are using the original ABP-Mean since it is more accurate than
using a formula like $\text{MAP} = \text{DBP} + \frac{1}{3} (\text{SBP} - \text{DBP})$.

#Hemodynamic codes

art_sys_codes<-c(8,9,10,11,12,13)

The most numerous SBP code is 8: ABP-Systolic

art_dias_codes<-c(14:19)

The most numerous DBP code is 14: ABP-Diastolic

art_mean_codes<-c(20:25)

The most numerous MAP code is 20: ABP-Mean

Artifact Removed Datasets

Remove artifacts using the following rules, in order:

(a) SBP (Systolic Blood Pressure) greater than or equal to 300 or SBP less than or
equal to 20 mmhg.
(b) SBP less than or equal to DBP + 5 mmhg.
(c) DBP (Diastolic Blood Pressure) less than or equal to 5 mmhg or DBP greater than
or equal to 225 mmhg.

2008

####New datasets Creation

```

```

```{r}
library(dplyr)
Vitalsigns08SPB<-subset(Vitalsigns08,Vitalsigns08$Code==8)
Vitalsigns08DBP<-subset(Vitalsigns08,Vitalsigns08$Code==14)
Vitalsigns08MAP<-subset(Vitalsigns08,Vitalsigns08$Code==20)
Vitalsigns08j<-full_join(Vitalsigns08SPB,Vitalsigns08DBP,by=c('Case_Name','Time'))
Vitalsigns08j<-full_join(Vitalsigns08j,Vitalsigns08MAP,by=c('Case_Name','Time'))
Vitalsigns08j<-Vitalsigns08j[order(Vitalsigns08j$Case_Name,Vitalsigns08j$Time),]
colnames(Vitalsigns08j)<-
c("Index.SBP","Case_Name","Time","SBP","Code.SBP","Index.DBP","DBP","Code.DBP",
  "Index.MAP","MAP","Code.MAP")

```

#MAP Formula, not needed in the end.

```

MAP <- function(SBP,DBP){
  MAP<-DBP + 1/3 * (SBP - DBP)
  roundMAP<-round(MAP,0)
  return(roundMAP)
}
```

```

#### Artifact Removal

```

```{r}
library(plotly)
library(shiny)
library(ggplot2)
library(dplyr)

#SBP artifact removal
Vitalsigns08j$SBP[Vitalsigns08j$SBP >= 300]<-NA
Vitalsigns08j$SBP[Vitalsigns08j$SBP<= 20]<-NA
Vitalsigns08j$SBP[Vitalsigns08j$SBP<= (Vitalsigns08j$DBP+5)]<-NA

#DBP artifact removal
Vitalsigns08j$DBP[Vitalsigns08j$DBP <= 5]<-NA
Vitalsigns08j$DBP[Vitalsigns08j$DBP >= 255]<-NA

#MAP artifact removal
Vitalsigns08j$MAP[Vitalsigns08j$MAP <= 0]<-NA
Vitalsigns08j$MAP[Vitalsigns08j$MAP >= 255]<-NA

```

```

### 2009

####New datasets Creation

```

```{r}
library(dplyr)
Vitalsigns09SPB<-subset(Vitalsigns09,Vitalsigns09$Code==8)
Vitalsigns09DBP<-subset(Vitalsigns09,Vitalsigns09$Code==14)

```

```

Vitalsigns09MAP<-subset(Vitalsigns09,Vitalsigns09$Code==20)
Vitalsigns09j<-full_join(Vitalsigns09SPB,Vitalsigns09DBP,by=c('Case_Name','Time'))
Vitalsigns09j<-full_join(Vitalsigns09j,Vitalsigns09MAP,by=c('Case_Name','Time'))
Vitalsigns09j<-Vitalsigns09j[order(Vitalsigns09j$Case_Name,Vitalsigns09j$Time),]
colnames(Vitalsigns09j)<-
c("Index.SBP","Case_Name","Time","SBP","Code.SBP","Index.DBP","DBP","Code.DBP",
  "Index.MAP","MAP","Code.MAP")
...

```

Artifact Removal

```

```{r}
library(plotly)
library(shiny)
library(ggplot2)
library(dplyr)

#SBP artifact removal
Vitalsigns09j$SBP[Vitalsigns09j$SBP >= 300]<-NA
Vitalsigns09j$SBP[Vitalsigns09j$SBP<=20]<-NA
Vitalsigns09j$SBP[Vitalsigns09j$SBP<=(Vitalsigns09j$DBP+5)]<-NA

#DBP artifact removal
Vitalsigns09j$DBP[Vitalsigns09j$DBP <= 5]<-NA
Vitalsigns09j$DBP[Vitalsigns09j$DBP >= 255]<-NA

#MAP artifact removal
Vitalsigns09j$MAP[Vitalsigns09j$MAP < 0]<-NA
Vitalsigns09j$MAP[Vitalsigns09j$MAP >= 255]<-NA
...

```

2010

####New datasets Creation

```

```{r}
library(dplyr)
Vitalsigns10SPB<-subset(Vitalsigns10,Vitalsigns10$Code==8)
Vitalsigns10DBP<-subset(Vitalsigns10,Vitalsigns10$Code==14)
Vitalsigns10MAP<-subset(Vitalsigns10,Vitalsigns10$Code==20)
Vitalsigns10j<-full_join(Vitalsigns10SPB,Vitalsigns10DBP,by=c('Case_Name','Time'))
Vitalsigns10j<-full_join(Vitalsigns10j,Vitalsigns10MAP,by=c('Case_Name','Time'))
Vitalsigns10j<-Vitalsigns10j[order(Vitalsigns10j$Case_Name,Vitalsigns10j$Time),]
colnames(Vitalsigns10j)<-
c("Index.SBP","Case_Name","Time","SBP","Code.SBP","Index.DBP","DBP","Code.DBP",
  "Index.MAP","MAP","Code.MAP")
...

```

Artifact Removal

```

```{r}
library(plotly)
library(shiny)

```



```

library(ggplot2)
library(dplyr)

#SBP artifact removal
Vitalsigns10j$SBP[Vitalsigns10j$SBP >= 300]<-NA
Vitalsigns10j$SBP[Vitalsigns10j$SBP<=20]<-NA
Vitalsigns10j$SBP[Vitalsigns10j$SBP<=(Vitalsigns10j$DBP+5)]<-NA

#DBP artifact removal
Vitalsigns10j$DBP[Vitalsigns10j$DBP <= 5]<-NA
Vitalsigns10j$DBP[Vitalsigns10j$DBP >= 255]<-NA

#MAP artifact removal
Vitalsigns10j$MAP[Vitalsigns10j$MAP < 0]<-NA
Vitalsigns10j$MAP[Vitalsigns10j$MAP >= 255]<-NA

...

### 2011

####New datasets Creation

```{r}
library(dplyr)
Vitalsigns11SPB<-subset(Vitalsigns11,Vitalsigns11$Code==8)
Vitalsigns11DBP<-subset(Vitalsigns11,Vitalsigns11$Code==14)
Vitalsigns11MAP<-subset(Vitalsigns11,Vitalsigns11$Code==20)
Vitalsigns11j<-full_join(Vitalsigns11SPB,Vitalsigns11DBP,by=c('Case_Name','Time'))
Vitalsigns11j<-full_join(Vitalsigns11j,Vitalsigns11MAP,by=c('Case_Name','Time'))
Vitalsigns11j<-Vitalsigns11j[order(Vitalsigns11j$Case_Name,Vitalsigns11j$Time),]
colnames(Vitalsigns11j)<-
c("Index.SBP","Case_Name","Time","SBP","Code.SBP","Index.DBP","DBP","Code.DBP",
  "Index.MAP","MAP","Code.MAP")
...

#### Artifact Removal

```{r}
library(plotly)
library(shiny)
library(ggplot2)
library(dplyr)

#SBP artifact removal
Vitalsigns11j$SBP[Vitalsigns11j$SBP >= 300]<-NA
Vitalsigns11j$SBP[Vitalsigns11j$SBP<=20]<-NA
Vitalsigns11j$SBP[Vitalsigns11j$SBP<=(Vitalsigns11j$DBP+5)]<-NA

#DBP artifact removal
Vitalsigns11j$DBP[Vitalsigns11j$DBP <= 5]<-NA
Vitalsigns11j$DBP[Vitalsigns11j$DBP >= 255]<-NA

#MAP artifact removal
Vitalsigns11j$MAP[Vitalsigns11j$MAP < 0]<-NA

```

```

Vitalsigns11j$MAP[Vitalsigns11j$MAP >= 255]<-NA
```

2012

New datasets Creation

```{r}
library(dplyr)
Vitalsigns12SPB<-subset(Vitalsigns12,Vitalsigns12$Code==8)
Vitalsigns12DBP<-subset(Vitalsigns12,Vitalsigns12$Code==14)
Vitalsigns12MAP<-subset(Vitalsigns12,Vitalsigns12$Code==20)
Vitalsigns12j<-full_join(Vitalsigns12SPB,Vitalsigns12DBP,by=c('Case_Name','Time'))
Vitalsigns12j<-full_join(Vitalsigns12j,Vitalsigns12MAP,by=c('Case_Name','Time'))
Vitalsigns12j<-Vitalsigns12j[order(Vitalsigns12j$Case_Name,Vitalsigns12j$Time),]
colnames(Vitalsigns12j)<-
c("Index.SBP","Case_Name","Time","SBP","Code.SBP","Index.DBP","DBP","Code.DBP",
  "Index.MAP","MAP","Code.MAP")
```

Artifact Removal

```{r}
library(plotly)
library(shiny)
library(ggplot2)
library(dplyr)

#SBP artifact removal
Vitalsigns12j$SBP[Vitalsigns12j$SBP >= 300]<-NA
Vitalsigns12j$SBP[Vitalsigns12j$SBP<=20]<-NA
Vitalsigns12j$SBP[Vitalsigns12j$SBP<=(Vitalsigns12j$DBP+5)]<-NA

#DBP artifact removal
Vitalsigns12j$DBP[Vitalsigns12j$DBP <= 5]<-NA
Vitalsigns12j$DBP[Vitalsigns12j$DBP >= 255]<-NA

#MAP artifact removal
Vitalsigns12j$MAP[Vitalsigns12j$MAP < 0]<-NA
Vitalsigns12j$MAP[Vitalsigns12j$MAP >= 255]<-NA
```

2013

New datasets Creation

```{r}
library(dplyr)
Vitalsigns13SPB<-subset(Vitalsigns13,Vitalsigns13$Code==8)
Vitalsigns13DBP<-subset(Vitalsigns13,Vitalsigns13$Code==14)
Vitalsigns13MAP<-subset(Vitalsigns13,Vitalsigns13$Code==20)
Vitalsigns13j<-full_join(Vitalsigns13SPB,Vitalsigns13DBP,by=c('Case_Name','Time'))

```

```

Vitalsigns13j<-full_join(Vitalsigns13j,Vitalsigns13MAP,by=c('Case_Name','Time'))
Vitalsigns13j<-Vitalsigns13j[order(Vitalsigns13j$Case_Name,Vitalsigns13j$Time),]
colnames(Vitalsigns13j)<-
c("Index.SBP","Case_Name","Time","SBP","Code.SBP","Index.DBP","DBP","Code.DBP",
  "Index.MAP","MAP","Code.MAP")
...

#### Artifact Removal

```{r}
library(plotly)
library(shiny)
library(ggplot2)
library(dplyr)

#SBP artifact removal
Vitalsigns13j$SBP[Vitalsigns13j$SBP >= 300]<-NA
Vitalsigns13j$SBP[Vitalsigns13j$SBP<=20]<-NA
Vitalsigns13j$SBP[Vitalsigns13j$SBP<=(Vitalsigns13j$DBP+5)]<-NA

#DBP artifact removal
Vitalsigns13j$DBP[Vitalsigns13j$DBP <= 5]<-NA
Vitalsigns13j$DBP[Vitalsigns13j$DBP >= 255]<-NA

#MAP artifact removal
Vitalsigns13j$MAP[Vitalsigns13j$MAP < 0]<-NA
Vitalsigns13j$MAP[Vitalsigns13j$MAP >= 255]<-NA
...

### 2014

####New datasets Creation

```{r}
library(dplyr)
Vitalsigns14SPB<-subset(Vitalsigns14,Vitalsigns14$Code==8)
Vitalsigns14DBP<-subset(Vitalsigns14,Vitalsigns14$Code==14)
Vitalsigns14MAP<-subset(Vitalsigns14,Vitalsigns14$Code==20)
Vitalsigns14j<-full_join(Vitalsigns14SPB,Vitalsigns14DBP,by=c('Case_Name','Time'))
Vitalsigns14j<-full_join(Vitalsigns14j,Vitalsigns14MAP,by=c('Case_Name','Time'))
Vitalsigns14j<-Vitalsigns14j[order(Vitalsigns14j$Case_Name,Vitalsigns14j$Time),]
colnames(Vitalsigns14j)<-
c("Index.SBP","Case_Name","Time","SBP","Code.SBP","Index.DBP","DBP","Code.DBP",
  "Index.MAP","MAP","Code.MAP")
...

#### Artifact Removal

```{r}
library(plotly)
library(shiny)
library(ggplot2)
library(dplyr)

```

```

#SBP artifact removal
Vitalsigns14j$SBP[Vitalsigns14j$SBP >= 300]<-NA
Vitalsigns14j$SBP[Vitalsigns14j$SBP<=20]<-NA
Vitalsigns14j$SBP[Vitalsigns14j$SBP<=(Vitalsigns14j$DBP+5)]<-NA

#DBP artifact removal
Vitalsigns14j$DBP[Vitalsigns14j$DBP <= 5]<-NA
Vitalsigns14j$DBP[Vitalsigns14j$DBP >= 255]<-NA

#MAP artifact removal
Vitalsigns14j$MAP[Vitalsigns14j$MAP < 0]<-NA
Vitalsigns14j$MAP[Vitalsigns14j$MAP >= 255]<-NA

...

### 2015

####New datasets Creation

```{r}
library(dplyr)
Vitalsigns15SPB<-subset(Vitalsigns15,Vitalsigns15$Code==8)
Vitalsigns15DBP<-subset(Vitalsigns15,Vitalsigns15$Code==14)
Vitalsigns15MAP<-subset(Vitalsigns15,Vitalsigns15$Code==20)
Vitalsigns15j<-full_join(Vitalsigns15SPB,Vitalsigns15DBP,by=c('Case_Name','Time'))
Vitalsigns15j<-full_join(Vitalsigns15j,Vitalsigns15MAP,by=c('Case_Name','Time'))
Vitalsigns15j<-Vitalsigns15j[order(Vitalsigns15j$Case_Name,Vitalsigns15j$Time),]
colnames(Vitalsigns15j)<-
c("Index.SBP","Case_Name","Time","SBP","Code.SBP","Index.DBP","DBP","Code.DBP",
  "Index.MAP","MAP","Code.MAP")
...

#### Artifact Removal

```{r}
library(plotly)
library(shiny)
library(ggplot2)
library(dplyr)

#SBP artifact removal
Vitalsigns15j$SBP[Vitalsigns15j$SBP >= 300]<-NA
Vitalsigns15j$SBP[Vitalsigns15j$SBP<=20]<-NA
Vitalsigns15j$SBP[Vitalsigns15j$SBP<=(Vitalsigns15j$DBP+5)]<-NA

#DBP artifact removal
Vitalsigns15j$DBP[Vitalsigns15j$DBP <= 5]<-NA
Vitalsigns15j$DBP[Vitalsigns15j$DBP >= 255]<-NA

#MAP artifact removal
Vitalsigns15j$MAP[Vitalsigns15j$MAP < 0]<-NA
Vitalsigns15j$MAP[Vitalsigns15j$MAP >= 255]<-NA

...

```

```

### 2016

####New datasets Creation

```{r}
library(dplyr)
Vitalsigns16SPB<-subset(Vitalsigns16,Vitalsigns16$Code==8)
Vitalsigns16DBP<-subset(Vitalsigns16,Vitalsigns16$Code==14)
Vitalsigns16MAP<-subset(Vitalsigns16,Vitalsigns16$Code==20)
Vitalsigns16j<-full_join(Vitalsigns16SPB,Vitalsigns16DBP,by=c('Case_Name','Time'))
Vitalsigns16j<-full_join(Vitalsigns16j,Vitalsigns16MAP,by=c('Case_Name','Time'))
Vitalsigns16j<-Vitalsigns16j[order(Vitalsigns16j$Case_Name,Vitalsigns16j$Time),]
colnames(Vitalsigns16j)<-
c("Index.SBP","Case_Name","Time","SBP","Code.SBP","Index.DBP","DBP","Code.DBP",
  "Index.MAP","MAP","Code.MAP")
...

#### Artifact Removal

```{r}
library(plotly)
library(shiny)
library(ggplot2)
library(dplyr)

#SBP artifact removal
Vitalsigns16j$SBP[Vitalsigns16j$SBP >= 300]<-NA
Vitalsigns16j$SBP[Vitalsigns16j$SBP<=20]<-NA
Vitalsigns16j$SBP[Vitalsigns16j$SBP<=(Vitalsigns16j$DBP+5)]<-NA

#DBP artifact removal
Vitalsigns16j$DBP[Vitalsigns16j$DBP <= 5]<-NA
Vitalsigns16j$DBP[Vitalsigns16j$DBP >= 255]<-NA

#MAP artifact removal
Vitalsigns16j$MAP[Vitalsigns16j$MAP < 0]<-NA
Vitalsigns16j$MAP[Vitalsigns16j$MAP >= 255]<-NA
...

---
title: "6_CPBPeriods_Surg_Times_and_VitalsignsF_creation"
author: "Miguel Angel Armengol de la Hoz"
output:
  html_document:
    toc: true
    theme: united
---

```{r}
#Load the enviroment

#load("C:/BIG FILES/Phenotyping/HemodynamicProjectsDatasetFeb18.RData")
...

```

```

# Creating VitalsignsF: Merging all years of vitalsigns in one dataset

```{r}
VitalsignsF <- rbind(Vitalsigns08j, Vitalsigns09j,
Vitalsigns10j,Vitalsigns11j,Vitalsigns12j,Vitalsigns13j,Vitalsigns14j,Vitalsigns15j,V
italsigns16j)
```

Importing data for Anesthesia Induction (AI) and Anesthesia Finish (AF) Times

```{r}
library(readxl)
Anesthesia_Finish_Times <- read_excel("C:/BIG
FILES/Phenotyping/hemodynamics/Anesthesia Finish Times.xlsx",
  col_types = c("blank", "text", "text",
    "date"))

Anesthesia_Induction_Times <- read_excel("C:/BIG
FILES/Phenotyping/hemodynamics/Anesthesia Induction Times.xlsx",
  col_types = c("text", "date"))
```

Studying data available between tables

```{r}
Case_Name_List_in_Text<-as.vector(unlist(unname(sapply(TextList, `[`, 1))))
Case_Name_List_in_Text<-unique(Case_Name_List_in_Text)
#Text has 5997 Cases

Case_Name_List_in_Vitalsigns<-as.vector(VitalsignsF$Case_Name)
Case_Name_List_in_Vitalsigns<-as.character(unique(VitalsignsF$Case_Name))
#Vitalsigns has 5708 Cases

table(Case_Name_List_in_Vitalsigns %in% Case_Name_List_in_Text)
# All 5708 Cases from Vitalsigns are in Text

Case_Name_List_in_STS_Text_JOIN<-unique(CasesofInterestDF$Case_Name)
# The Join between STS and Text has 5724 Cases

VitalsignsF<-VitalsignsF %>%
  filter(Case_Name %in% Case_Name_List_in_STS_Text_JOIN)

#There are 5708 with both VitalSigns and STS information.

#From our Vitalsign table, there are 390 Cases with no CPB Times Info.
#Meaning there are 5318 Cases with all the info we need
```

Anesthesia Induction (AI) and Anesthesia Finish (AF), Anesthesia Times Creation

```{r}
library(dplyr)

```

```

Anesthesia_Times<-
inner_join(Anesthesia_Induction_Times,Anesthesia_Finish_Times[,c('Case_Name','RECORDTIME')])

Anesthesia_Times<-rename(Anesthesia_Times,"AF_Time"="RECORDTIME")
Anesthesia_Times<-rename(Anesthesia_Times,"AI_Time"="Anesthesia Induction ( I )")
```

First and End Times of VitalSigns, VitalSigns_Times creation

```{r}
library(dplyr)

VitalSigns_First <- VitalsignsF %>%
  group_by(Case_Name) %>%
  dplyr::select(c(Case_Name,Time)) %>%
  arrange(Time)%>%
  filter(!duplicated(Case_Name))
colnames(VitalSigns_First) <- c('Case_Name', 'VitalSigns_First_Time')

VitalSigns_Last <- VitalsignsF %>%
  group_by(Case_Name) %>%
  dplyr::select(c(Case_Name,Time)) %>%
  arrange(desc(Time))%>%
  filter(!duplicated(Case_Name))
colnames(VitalSigns_Last) <- c('Case_Name', 'VitalSigns_Last_Time')

VitalSigns_Times<-merge(VitalSigns_First, VitalSigns_Last)
```

Studying Surgery Begin and End Times discrepancies

```{r}
discrepancies_check<-inner_join(VitalSigns_Times,Anesthesia_Times)

discrepancies_check$VitalSigns_First_Time<-
as.character(discrepancies_check$VitalSigns_First_Time)
discrepancies_check$VitalSigns_Last_Time<-
as.character(discrepancies_check$VitalSigns_Last_Time)
discrepancies_check$AI_Time<-as.character(discrepancies_check$AI_Time)
discrepancies_check$AF_Time<-as.character(discrepancies_check$AF_Time)

discrepancies_check$VitalSigns_First_Time<-
as.POSIXct(discrepancies_check$VitalSigns_First_Time, format='%Y-%m-%d %H:%M:%S',
tz="GMT")
discrepancies_check$VitalSigns_Last_Time<-
as.POSIXct(discrepancies_check$VitalSigns_Last_Time, format='%Y-%m-%d %H:%M:%S',
tz="GMT")
discrepancies_check$AI_Time<-as.POSIXct(discrepancies_check$AI_Time, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

```

```

discrepancies_check$AF_Time<-as.POSIXct(discrepancies_check$AF_Time, format='%Y-%m-%d
%H:%M:%S', tz="GMT")

discrepancies_check['First_Difference']<-
difftime(discrepancies_check$AI_Time,discrepancies_check$VitalSigns_First_Time,units
= 'mins')
discrepancies_check['Last_Difference']<-
difftime(discrepancies_check$AF_Time,discrepancies_check$VitalSigns_Last_Time,units =
'mins')

discrepancies_check$First_Difference<-
as.numeric(discrepancies_check$First_Difference)
discrepancies_check$Last_Difference<-as.numeric(discrepancies_check$Last_Difference)

summary(discrepancies_check$First_Difference)
hist(discrepancies_check$First_Difference,breaks = 100)

summary(discrepancies_check$Last_Difference)
hist(discrepancies_check$Last_Difference,breaks = 100)

FinalHemodynamicsDataset%>%filter(duration_Surgery_min<=60)
```

```

# Start and End Times of CPB, rCPBEventsDF creation

In this part of the code we filter the Event table in order to extract only CPB related information, then we reshape the columns in order to have the begin and end time information available per case.

```

```{r}
library(dplyr)

#We create a new dataset only containing CPB information

for (name in names(EventsList)){
  #EventCount<-paste('CPB',name, sep='')
  EventCount<-EventsList[[name]]%>%
    filter(Label == 'Bypass Begin' |
           Label == 'Bypass End')
  assign(paste('CPB',name, sep=''), EventCount)
}

#we store them in a list fot easily handling them in the future.
CPBEventsList <- list(CPBEvents08, CPBEvents09,
CPBEvents10,CPBEvents11,CPBEvents12,CPBEvents13,CPBEvents14,CPBEvents15,CPBEvents16)
names( CPBEventsList)<- c("CPBEvents08", "CPBEvents09", "CPBEvents10", "CPBEvents11",
"CPBEvents12", "CPBEvents13", "CPBEvents14", "CPBEvents15", "CPBEvents16")

#we get rid of the columns we are not going to use.

for (name in names(CPBEventsList)){
  #EventCount<-paste('CPB',name, sep='')
  CPBEventsList[[name]]$ItemID<-NULL
  CPBEventsList[[name]]$ESIG<-NULL
}

```



```

        CPBEventsList[[name]]$ESIGNAMETEXT<-NULL
        CPBEventsList[[name]]$ESIGDATEANDTIME<-NULL
    }

library(reshape)

for (name in names(CPBEventsList)){

#Sometimes there are several CPB events during a case, we filter the dataframe taking
only the first CPB_Begin timestamp and the last CPB_End timestamp in order to get the
whole length.
CPBEventsb <-subset(CPBEventsList[[name]], CPBEventsList[[name]]$Label=='Bypass
Begin')
CPBEventsb<-CPBEventsb[order(CPBEventsb$Case_Name, CPBEventsb$Time,
decreasing=FALSE),]
CPBEventse <-subset(CPBEventsList[[name]], CPBEventsList[[name]]$Label=='Bypass End')
CPBEventse<-CPBEventse[order(CPBEventse$Case_Name, CPBEventse$Time,
decreasing=TRUE),]

CPBEventsf<-union_all(CPBEventsb, CPBEventse)

#we create new dataframes for the rSHAPED tables
BPE<-reshape(CPBEventsf, timevar = 'Label' , idvar = 'Case_Name', direction = 'wide')
colnames(BPE)<-c('Case_Name','CPB_Begin_Time','CPB_End_Time')
BPE<-subset(BPE,complete.cases(BPE))
assign(paste('r',name, sep=''), BPE)
}

rCPBEventsDF <- rbind(rCPBEvents08, rCPBEvents09,
rCPBEvents10,rCPBEvents11,rCPBEvents12,rCPBEvents13,rCPBEvents14,rCPBEvents15,rCPBEve
nts16)

#there are 5541 cases with CPB times information.
...

# All duration_periods_min columns and AllPeriodsDF creation
```{r}
AllPeriodsDF<-inner_join(VitalSigns_Times,rCPBEventsDF)
#From our Vitalsign table, there are 390 Cases with no CPB Times Info.
#Meaning there are 5318 Cases with all the info we need
# We are not interested on Cases with NO CPB informations, thus we exclude this
cases.

AllPeriodsDF['duration_preCPB_min']<-
as.numeric(difftime(AllPeriodsDF$CPB_Begin_Time,AllPeriodsDF$VitalSigns_First_Time),
units="mins")
AllPeriodsDF["duration_CPB_min"] <-
as.numeric(difftime(AllPeriodsDF$CPB_End_Time,AllPeriodsDF$CPB_Begin_Time),
units="mins")
BPE<-subset(BPE,complete.cases(BPE))
AllPeriodsDF['duration_postCPB_min']<-
as.numeric(difftime(AllPeriodsDF$VitalSigns_Last_Time,AllPeriodsDF$CPB_End_Time),
units="mins")

```

```

AllPeriodsDF['duration_Surgery_min']<-
as.numeric(difftime(AllPeriodsDF$`VitalSigns_Last_Time`,AllPeriodsDF$`VitalSigns_First_Time`), units="mins")
```

---
title: "7_HPE_Study_AUT_AWI_and_ratios"
author: "Miguel Angel Armengol de la Hoz"
output:
  html_document:
    toc: true
    theme: united

Load the enviroment and data

```{r}

#load("C:/BIG FILES/Phenotyping/HemodynamicProjectsDatasetFeb18.RData")

#load libs
library(dplyr)
library(xlsx)
library(compare)
```

Hypothension episodes location dataset

We need a dataset to know in which period of the surgery (preCPB|CPB|postCPB) our timestamps are taking place.

```{r eval=FALSE, include=FALSE}
All_Cases<-data.frame(unique(VitalsignsF$Case_Name))
colnames(All_Cases)[1]<-'Case_Name'

VitalsignsandCPBTimes <- merge(VitalsignsF, rCPBEventsDF, by = c('Case_Name'), all = FALSE)
#In fVitalsignsDF index is useless and we need to order by Case_Name and Time!
VitalsignsandCPBTimes<-VitalsignsandCPBTimes%>%
arrange(Case_Name, Time)

#here we identify in which period of the CPB our timestamp is taking place
VitalsignsandCPBTimes<-VitalsignsandCPBTimes %>% mutate(preCPB=
if_else(Time<CPB_Begin_Time,1,0))
VitalsignsandCPBTimes<-VitalsignsandCPBTimes %>% mutate(CPB=
if_else(Time>=CPB_Begin_Time & Time<=CPB_End_Time,1,0))
VitalsignsandCPBTimes<-VitalsignsandCPBTimes %>% mutate(postCPB=
if_else(Time>CPB_End_Time,1,0))
```

MAP

- AWI: Area within the interval.
- AUT: Area under the threshold.
- AOT: Area over the threshold.

```

```

AUT MAP <= 50 mmHg length per period and ratios

```{r}

#Solving select problem!!!!!!:

#select causes problem since it is also in package MASS
#use dplyr::select( to solve it

#AUT50MAP_total_min calculation

#detach(package:plyr)

#detach("package:plyr", unload=TRUE)

AUT50MAP<-VitalsignsF%%
dplyr::select(Case_Name, MAP)%>%
group_by(Case_Name)%>%
dplyr::filter(MAP<=50)%>%
dplyr::summarise(AUT50MAP_total_min = n()*15/60)
AUT50MAP<-left_join(All_Cases, AUT50MAP)
AUT50MAP[is.na(AUT50MAP)]<-0

#AUT50MAP_total_min calculation

AUT50MAP_preCPB<-VitalsignsandCPBTimes %% dplyr::group_by(Case_Name) %>%
dplyr::filter(preCPB==1 & MAP <=50) %>% dplyr::summarise(AUT50MAP_preCPB_min =
n()*15/60)

AUT50MAP_CPB<-VitalsignsandCPBTimes %% dplyr::group_by(Case_Name)
%>%dplyr::filter(CPB==1 & MAP <=50) %>% dplyr::summarise(AUT50MAP_CPB_min =
n()*15/60)

AUT50MAP_postCPB<-VitalsignsandCPBTimes %% dplyr::group_by(Case_Name) %>%
dplyr::filter(postCPB==1 & MAP <=50) %>% dplyr::summarise(AUT50MAP_postCPB_min =
n()*15/60)

require(plyr)
AUT50MAP<-join_all(list(AUT50MAP,AUT50MAP_preCPB,AUT50MAP_CPB,AUT50MAP_postCPB), by
='Case_Name')

#now we add the ratios, for that we need duration info thus we merge our AUT dataset
with AllPeriodsDF

AllPeriodswith_AUT_AOT_and_AWI_DF<-left_join(AllPeriodsDF,AUT50MAP)

AllPeriodswith_AUT_AOT_and_AWI_DF<-AllPeriodswith_AUT_AOT_and_AWI_DF%%
mutate(AUT50MAP_total_ratio=round(AUT50MAP_total_min/duration_Surgery_min,2),
      AUT50MAP_preCPB_ratio=round(AUT50MAP_preCPB_min/duration_preCPB_min,2),
      AUT50MAP_CPB_ratio=round(AUT50MAP_CPB_min/duration_CPB_min,2),

```

```

        AUT50MAP_postCPB_ratio=round(AUT50MAP_postCPB_min/duration_postCPB_min,2)
    ...
)

##AWI MAP >= 51 <=60 mmHg length per period and ratios

```{r}

#AUT50MAP_total_min calculation

detach(package:plyr)
AWI51_60MAP<-VitalsignsF%>%
dplyr::select(Case_Name, MAP)%>%
group_by(Case_Name)%>%
filter(MAP >= 51 & MAP <=60 ) %>%
summarise(AWI51_60MAP_total_min = n()*15/60)
AWI51_60MAP<-left_join(All_Cases, AWI51_60MAP)
AWI51_60MAP[is.na(AWI51_60MAP)]<-0

#AUT50MAP_total_min calculation
library(compare)

AWI51_60MAP_preCPB<-VitalsignsandCPBTimes %>% dplyr::group_by(Case_Name) %>%
dplyr::filter(preCPB==1 & MAP >= 51 & MAP <=60 ) %>%
dplyr::summarise(AWI51_60MAP_preCPB_min = n()*15/60)

AWI51_60MAP_CPB<-VitalsignsandCPBTimes %>% dplyr::group_by(Case_Name)
%>%dplyr::filter(CPB==1 & MAP >= 51 & MAP <=60 ) %>%
dplyr::summarise(AWI51_60MAP_CPB_min = n()*15/60)

AWI51_60MAP_postCPB<-VitalsignsandCPBTimes %>% dplyr::group_by(Case_Name) %>%
dplyr::filter(postCPB==1 & MAP >= 51 & MAP <=60 ) %>%
dplyr::summarise(AWI51_60MAP_postCPB_min = n()*15/60)

require(plyr)
AWI51_60MAP<-
join_all(list(AWI51_60MAP,AWI51_60MAP_preCPB,AWI51_60MAP_CPB,AWI51_60MAP_postCPB), by
='Case_Name')

#now we add the ratios, for that we need duration info thus we merge our AUT dataset
with AllPeriodsDF

AllPeriodswith_AUT_AOT_and_AWI_DF<-
left_join(AllPeriodswith_AUT_AOT_and_AWI_DF,AWI51_60MAP)

AllPeriodswith_AUT_AOT_and_AWI_DF<-AllPeriodswith_AUT_AOT_and_AWI_DF%>%
mutate(AWI51_60MAP_total_ratio=round(AWI51_60MAP_total_min/duration_Surgery_min,2),
        AWI51_60MAP_preCPB_ratio=round(AWI51_60MAP_preCPB_min/duration_preCPB_min,2),
        AWI51_60MAP_CPB_ratio=round(AWI51_60MAP_CPB_min/duration_CPB_min,2),

AWI51_60MAP_postCPB_ratio=round(AWI51_60MAP_postCPB_min/duration_postCPB_min,2)
    ...
)

## AWI MAP >= 61 <65 mmHg length per period and ratios

```

```

```{r}

#AUT50MAP_total_min calculation

detach(package:plyr)
AWI61_65MAP<-VitalsignsF%>%
dplyr::select(Case_Name, MAP)%>%
group_by(Case_Name)%>%
filter(MAP >= 61 & MAP <65 ) %>%
summarise(AWI61_65MAP_total_min = n()*15/60)
AWI61_65MAP<-left_join(All_Cases, AWI61_65MAP)
AWI61_65MAP[is.na(AWI61_65MAP)]<-0

#AUT50MAP_total_min calculation
library(compare)

AWI61_65MAP_preCPB<-VitalsignsandCPBTimes %>% dplyr::group_by(Case_Name) %>%
dplyr::filter(preCPB==1 & MAP >= 61 & MAP <65 ) %>%
dplyr::summarise(AWI61_65MAP_preCPB_min = n()*15/60)

AWI61_65MAP_CPB<-VitalsignsandCPBTimes %>% dplyr::group_by(Case_Name)
%>%dplyr::filter(CPB==1 & MAP >= 61 & MAP <65 ) %>%
dplyr::summarise(AWI61_65MAP_CPB_min = n()*15/60)

AWI61_65MAP_postCPB<-VitalsignsandCPBTimes %>% dplyr::group_by(Case_Name) %>%
dplyr::filter(postCPB==1 & MAP >= 61 & MAP <65 ) %>%
dplyr::summarise(AWI61_65MAP_postCPB_min = n()*15/60)

require(plyr)
AWI61_65MAP<-
join_all(list(AWI61_65MAP,AWI61_65MAP_preCPB,AWI61_65MAP_CPB,AWI61_65MAP_postCPB), by
='Case_Name')

#now we add the ratios, for that we need duration info thus we merge our AUT dataset
with AllPeriodsDF

AllPeriodswith_AUT_AOT_and_AWI_DF<-
left_join(AllPeriodswith_AUT_AOT_and_AWI_DF,AWI61_65MAP)

AllPeriodswith_AUT_AOT_and_AWI_DF<-AllPeriodswith_AUT_AOT_and_AWI_DF%>%
mutate(AWI61_65MAP_total_ratio=round(AWI61_65MAP_total_min/duration_Surgery_min,2),
      AWI61_65MAP_preCPB_ratio=round(AWI61_65MAP_preCPB_min/duration_preCPB_min,2),
      AWI61_65MAP_CPB_ratio=round(AWI61_65MAP_CPB_min/duration_CPB_min,2),

AWI61_65MAP_postCPB_ratio=round(AWI61_65MAP_postCPB_min/duration_postCPB_min,2)
...
)

## AOT MAP >= 65 mmHg length per period and ratios

```{r}

#AOT65MAP_total_min calculation

```

```

detach(package:plyr)
AOT65MAP<-VitalsignsF%>%
dplyr::select(Case_Name, MAP)%>%
group_by(Case_Name)%>%
filter(MAP >=65) %>%
summarise(AOT65MAP_total_min = n()*15/60)
AOT65MAP<-left_join(All_Cases, AOT65MAP)
AOT65MAP[is.na(AOT65MAP)]<-0

#AOT65MAP_total_min calculation
library(compare)

AOT65MAP_preCPB<-VitalsignsandCPBTimes %>% dplyr::group_by(Case_Name) %>%
dplyr::filter(preCPB==1 & MAP >=65) %>% dplyr::summarise(AOT65MAP_preCPB_min =
n()*15/60)

AOT65MAP_CPB<-VitalsignsandCPBTimes %>% dplyr::group_by(Case_Name)
%>%dplyr::filter(CPB==1 & MAP >=65) %>% dplyr::summarise(AOT65MAP_CPB_min =
n()*15/60)

AOT65MAP_postCPB<-VitalsignsandCPBTimes %>% dplyr::group_by(Case_Name) %>%
dplyr::filter(postCPB==1 & MAP >=65) %>% dplyr::summarise(AOT65MAP_postCPB_min =
n()*15/60)

require(plyr)
AOT65MAP<-join_all(list(AOT65MAP,AOT65MAP_preCPB,AOT65MAP_CPB,AOT65MAP_postCPB), by
='Case_Name')

#now we add the ratios, for that we need duration info thus we merge our AUT dataset
with AllPeriodsDF

AllPeriodswith_AUT_AOT_and_AWI_DF<-
left_join(AllPeriodswith_AUT_AOT_and_AWI_DF,AOT65MAP)

AllPeriodswith_AUT_AOT_and_AWI_DF<-AllPeriodswith_AUT_AOT_and_AWI_DF%>%
mutate(AOT65MAP_total_ratio=round(AOT65MAP_total_min/duration_Surgery_min,2),
      AOT65MAP_preCPB_ratio=round(AOT65MAP_preCPB_min/duration_preCPB_min,2),
      AOT65MAP_CPB_ratio=round(AOT65MAP_CPB_min/duration_CPB_min,2),
      AOT65MAP_postCPB_ratio=round(AOT65MAP_postCPB_min/duration_postCPB_min,2)
      )
...

## AUT MAP < 65 mmHg length per period and ratios

```{r}

#AUT65MAP_total_min calculation
detach("package:plyr")
AUT65MAP<-VitalsignsF%>%
dplyr::select(Case_Name, MAP)%>%
group_by(Case_Name)%>%
filter(MAP <65) %>%
summarise(AUT65MAP_total_min = n()*15/60)
AUT65MAP<-left_join(All_Cases, AUT65MAP)

```

```

AUT65MAP[is.na(AUT65MAP)]<-0

#AUT65MAP_total_min calculation
library(compare)

AUT65MAP_preCPB<-VitalsignsandCPBTimes %>% dplyr::group_by(Case_Name) %>%
dplyr::filter(preCPB==1 & MAP <65) %>% dplyr::summarise(AUT65MAP_preCPB_min =
n()*15/60)

AUT65MAP_CPB<-VitalsignsandCPBTimes %>% dplyr::group_by(Case_Name)
%>%dplyr::filter(CPB==1 & MAP <65) %>% dplyr::summarise(AUT65MAP_CPB_min = n()*15/60)

AUT65MAP_postCPB<-VitalsignsandCPBTimes %>% dplyr::group_by(Case_Name) %>%
dplyr::filter(postCPB==1 & MAP <65) %>% dplyr::summarise(AUT65MAP_postCPB_min =
n()*15/60)

require(plyr)
AUT65MAP<-join_all(list(AUT65MAP,AUT65MAP_preCPB,AUT65MAP_CPB,AUT65MAP_postCPB), by
='Case_Name')

#now we add the ratios, for that we need duration info thus we merge our AUT dataset
with AllPeriodsDF

AllPeriodswith_AUT_AOT_and_AWI_DF<-
left_join(AllPeriodswith_AUT_AOT_and_AWI_DF,AUT65MAP)

AllPeriodswith_AUT_AOT_and_AWI_DF<-AllPeriodswith_AUT_AOT_and_AWI_DF%>%
mutate(AUT65MAP_total_ratio=round(AUT65MAP_total_min/duration_Surgery_min,2),
      AUT65MAP_preCPB_ratio=round(AUT65MAP_preCPB_min/duration_preCPB_min,2),
      AUT65MAP_CPB_ratio=round(AUT65MAP_CPB_min/duration_CPB_min,2),
      AUT65MAP_postCPB_ratio=round(AUT65MAP_postCPB_min/duration_postCPB_min,2)
      )
...

# SBP

## AUT SBP <= 80 mmHg length per period and ratios

```{r}

#AUT80SBP_total_min calculation

#detach(package:plyr)
AUT80SBP<-VitalsignsF%>%
dplyr::select(Case_Name, SBP)%>%
group_by(Case_Name)%>%
filter(SBP <=80) %>%
dplyr::summarise(AUT80SBP_total_min = n()*15/60)
AUT80SBP<-left_join(All_Cases, AUT80SBP)
AUT80SBP[is.na(AUT80SBP)]<-0

#AUT80SBP_total preCPB and postCPB min calculation
library(compare)

```

```

AUT80SBP_preCPB<-VitalsignsandCPBTimes %>% dplyr::group_by(Case_Name) %>%
dplyr::filter(preCPB==1 & SBP <=80) %>% dplyr::summarise(AUT80SBP_preCPB_min =
n()*15/60)

AUT80SBP_postCPB<-VitalsignsandCPBTimes %>% dplyr::group_by(Case_Name) %>%
dplyr::filter(postCPB==1 & SBP <=80) %>% dplyr::summarise(AUT80SBP_postCPB_min =
n()*15/60)

require(plyr)
AUT80SBP<-join_all(list(AUT80SBP,AUT80SBP_preCPB,AUT80SBP_postCPB), by ='Case_Name')

#now we add the ratios, for that we need duration info thus we merge our AUT dataset
with AllPeriodsDF

AllPeriodswith_AUT_AOT_and_AWI_DF<-
left_join(AllPeriodswith_AUT_AOT_and_AWI_DF,AUT80SBP)

AllPeriodswith_AUT_AOT_and_AWI_DF<-AllPeriodswith_AUT_AOT_and_AWI_DF%>%
mutate(AUT80SBP_total_ratio=round(AUT80SBP_total_min/duration_Surgery_min,2),
      AUT80SBP_preCPB_ratio=round(AUT80SBP_preCPB_min/duration_preCPB_min,2),
      AUT80SBP_postCPB_ratio=round(AUT80SBP_postCPB_min/duration_postCPB_min,2)
      )
...

## AOT SBP >80 mmHg length per period and ratios

```{r}

#AUT80SBP_total_min calculation

detach(package:plyr)
AOT80SBP<-VitalsignsF%>%
dplyr::select(Case_Name, SBP)%>%
group_by(Case_Name)%>%
filter(SBP >80) %>%
dplyr::summarise(AOT80SBP_total_min = n()*15/60)
AOT80SBP<-left_join(All_Cases, AOT80SBP)
AOT80SBP[is.na(AOT80SBP)]<-0

#AOT80SBP_total preCPB and postCPB min calculation
library(compare)

AOT80SBP_preCPB<-VitalsignsandCPBTimes %>% dplyr::group_by(Case_Name) %>%
dplyr::filter(preCPB==1 & SBP >80) %>% dplyr::summarise(AOT80SBP_preCPB_min =
n()*15/60)

AOT80SBP_postCPB<-VitalsignsandCPBTimes %>% dplyr::group_by(Case_Name) %>%
dplyr::filter(postCPB==1 & SBP >80) %>% dplyr::summarise(AOT80SBP_postCPB_min =
n()*15/60)

require(plyr)
AOT80SBP<-join_all(list(AOT80SBP,AOT80SBP_preCPB,AOT80SBP_postCPB), by ='Case_Name')

```



```
#now we add the ratios, for that we need duration info thus we merge our AOT dataset
with AllPeriodsDF
```

```
AllPeriodswith_AUT_AOT_and_AWI_DF<-
left_join(AllPeriodswith_AUT_AOT_and_AWI_DF,AOT80SBP)
```

```
AllPeriodswith_AUT_AOT_and_AWI_DF<-AllPeriodswith_AUT_AOT_and_AWI_DF%>%
mutate(AOT80SBP_total_ratio=round(AOT80SBP_total_min/duration_Surgery_min,2),
      AOT80SBP_preCPB_ratio=round(AOT80SBP_preCPB_min/duration_preCPB_min,2),
      AOT80SBP_postCPB_ratio=round(AOT80SBP_postCPB_min/duration_postCPB_min,2)
    )
...

```

```
# DBP
```

```
## AUT DBP <= 30 mmHg length per period and ratios
```

```
`r`{r}
```

```
#AUT30DBP_total_min calculation
```

```
detach(package:plyr)
AUT30DBP<-VitalsignsF%>%
dplyr::select(Case_Name, DBP)%>%
group_by(Case_Name)%>%
filter(DBP <=30) %>%
summarise(AUT30DBP_total_min = n()*15/60)
AUT30DBP<-left_join(All_Cases, AUT30DBP)
AUT30DBP[is.na(AUT30DBP)]<-0
```

```
#AUT30DBP_total preCPB and postCPB min calculation
library(compare)
```

```
AUT30DBP_preCPB<-VitalsignsandCPBTimes %>% dplyr::group_by(Case_Name) %>%
dplyr::filter(preCPB==1 & DBP <=30) %>% dplyr::summarise(AUT30DBP_preCPB_min =
n()*15/60)
```

```
AUT30DBP_postCPB<-VitalsignsandCPBTimes %>% dplyr::group_by(Case_Name) %>%
dplyr::filter(postCPB==1 & DBP <=30) %>% dplyr::summarise(AUT30DBP_postCPB_min =
n()*15/60)
```

```
require(plyr)
AUT30DBP<-join_all(list(AUT30DBP,AUT30DBP_preCPB,AUT30DBP_postCPB), by ='Case_Name')
```

```
#now we add the ratios, for that we need duration info thus we merge our AUT dataset
with AllPeriodsDF
```

```
AllPeriodswith_AUT_AOT_and_AWI_DF<-
left_join(AllPeriodswith_AUT_AOT_and_AWI_DF,AUT30DBP)
```

```
AllPeriodswith_AUT_AOT_and_AWI_DF<-AllPeriodswith_AUT_AOT_and_AWI_DF%>%
mutate(AUT30DBP_total_ratio=round(AUT30DBP_total_min/duration_Surgery_min,2),
      AUT30DBP_preCPB_ratio=round(AUT30DBP_preCPB_min/duration_preCPB_min,2),
      AUT30DBP_postCPB_ratio=round(AUT30DBP_postCPB_min/duration_postCPB_min,2)
    )

```

```

...

## AOT DBP > 30 mmHg length per period and ratios

```{r}

#AOT30DBP_total_min calculation

detach(package:plyr)
AOT30DBP<-VitalsignsF%%
dplyr::select(Case_Name, DBP)%>%
group_by(Case_Name)%>%
filter(DBP >30) %>%
summarise(AOT30DBP_total_min = n()*15/60)
AOT30DBP<-left_join(All_Cases, AOT30DBP)
AOT30DBP[is.na(AOT30DBP)]<-0

#AOT30DBP_total preCPB and postCPB min calculation
library(compare)

AOT30DBP_preCPB<-VitalsignsandCPBTimes %>% dplyr::group_by(Case_Name) %>%
dplyr::filter(preCPB==1 & DBP >30) %>% dplyr::summarise(AOT30DBP_preCPB_min =
n()*15/60)

AOT30DBP_postCPB<-VitalsignsandCPBTimes %>% dplyr::group_by(Case_Name) %>%
dplyr::filter(postCPB==1 & DBP >30) %>% dplyr::summarise(AOT30DBP_postCPB_min =
n()*15/60)

require(plyr)
AOT30DBP<-join_all(list(AOT30DBP,AOT30DBP_preCPB,AOT30DBP_postCPB), by ='Case_Name')

#now we add the ratios, for that we need duration info thus we merge our AOT dataset
with AllPeriodsDF

AllPeriodswith_AUT_AOT_and_AWI_DF<-
left_join(AllPeriodswith_AUT_AOT_and_AWI_DF,AOT30DBP)

AllPeriodswith_AUT_AOT_and_AWI_DF<-AllPeriodswith_AUT_AOT_and_AWI_DF%%
mutate(AOT30DBP_total_ratio=round(AOT30DBP_total_min/duration_Surgery_min,2),
      AOT30DBP_preCPB_ratio=round(AOT30DBP_preCPB_min/duration_preCPB_min,2),
      AOT30DBP_postCPB_ratio=round(AOT30DBP_postCPB_min/duration_postCPB_min,2)
      )
...

# New AUC-related Variables

## MAP

### AUC MAP < 65 Variable Creation

AUC under 65 mmHg Dataset creation

```{r}

detach("package:plyr")

```

```

AUC65MAP<-VitalsignsF%>%
dplyr::select(Case_Name, MAP)%>%
filter(MAP<65) %>%
mutate(rectangle=(65-MAP)*0.25) %>%
group_by(Case_Name)%>%
summarise(AUC65MAP_total_mmHgmin = sum(rectangle)) %>%
arrange(Case_Name)

AUC65MAP_CPB<-VitalsignsandCPBTimes %>%
dplyr::group_by(Case_Name) %>%
dplyr::filter(CPB==1 & MAP <65) %>%
mutate(rectangle=(65-MAP)*0.25) %>%
group_by(Case_Name)%>%
summarise(AUC65MAP_CPB_total_mmHgmin = sum(rectangle)) %>%
arrange(Case_Name)

...

### AOC MAP > 65 Variable Creation

AOC over or equal 65 mmHg Dataset creation

```{r}

#detach("package:plyr")

AOC65MAP<-VitalsignsF%>%
dplyr::select(Case_Name, MAP)%>%
filter(MAP >=65) %>%
mutate(rectangle=(MAP-65)*0.25) %>%
group_by(Case_Name)%>%
summarise(AOC65MAP_total_mmHgmin = sum(rectangle)) %>%
arrange(Case_Name)
```

Combined TWA & AUC MAP <= 65 and All_AUC65MAP_variables Variable Creation

TWA-MAP under 65 mmHg dataset creation

```{r}
require(dplyr)
All_AUC65MAP_variables<-left_join(AllPeriodsDF,AUC65MAP)%>%
dplyr::select(Case_Name, duration_Surgery_min,AUC65MAP_total_mmHgmin)%>%
mutate(TWA65MAP_total_mmHg=round(AUC65MAP_total_mmHgmin/duration_Surgery_min,2)) %>%
dplyr::select(Case_Name,AUC65MAP_total_mmHgmin,TWA65MAP_total_mmHg)

All_AUC65MAP_variables<-left_join(All_AUC65MAP_variables,AUC65MAP_CPB)
#In case we want to add TWA:
# dplyr::select(Case_Name, duration_Surgery_min,AUC65MAP_CPB_total_mmHgmin)%>%
# mutate(TWA65MAP_total_mmHg=round(AUC65MAP_total_mmHgmin/duration_Surgery_min,2))
#>%
# select(Case_Name,AUC65MAP_total_mmHgmin,TWA65MAP_total_mmHg)

All_AUC65MAP_variables<-left_join(All_AUC65MAP_variables, AOC65MAP)
# those that didn't have values <65 or >=65 spent 0 minutes in the threshold.

```

```

All_AUC65MAP_variables[is.na(All_AUC65MAP_variables)]<-0
...

# REPLACING NA with 0

## Replacing 0 in CPB related calculated colums

What to do for those CPB-related variables that have no CPB times?
_We exclude those cases_

```{r eval=FALSE, include=FALSE}

#nested if else function creation

i <- function(if_stat, then) {
  if_stat <- lazyeval::expr_text(if_stat)
  then <- lazyeval::expr_text(then)
  sprintf("ifelse(%s, %s, ", if_stat, then)
}

e <- function(else_ret) {
  else_ret <- lazyeval::expr_text(else_ret)
  else_ret
}

ie <- function(...) {
  args <- list(...)

  for (i in 1:(length(args) - 1) ) {
    if (substr(args[[i]], 1, 6) != "ifelse") {
      stop("All but the last argument, need to be i functions.", call. = FALSE)
    }
  }
  if (substr(args[[length(args)]], 1, 6) == "ifelse"){
    stop("Last argument needs to be an e function.", call. = FALSE)
  }
  args$final <- paste(rep(')', length(args) - 1), collapse = '')
  eval_string <- do.call('paste', args)
  eval(parse(text = eval_string))
}

preCPBcols<-names(AllPeriodswith_AUT_AOT_and_AWI_DF[,grep('preCPB_ratio|preCPB_min',
names(AllPeriodswith_AUT_AOT_and_AWI_DF))])
for (n in 1:length(CPBcols)) {
  colnam<-CPBcols[n]
#print(AllPeriodswith_AUT_AOT_and_AWI_DF[,colnam] )
AllPeriodswith_AUT_AOT_and_AWI_DF[,colnam] <-
# i( is.na(AllPeriodswith_AUT_AOT_and_AWI_DF[,colnam]) , NA),
  ie(
    i( is.na(AllPeriodswith_AUT_AOT_and_AWI_DF$Sur_Begin_Time) , 1),
    i( is.na(AllPeriodswith_AUT_AOT_and_AWI_DF$CPB_Begin_Time) , 2),
    e(0)
  )
}

```

```

    )

i(if_stat = 3>4,2)

}

...

## checking durations with neg values

```{r eval=FALSE, include=FALSE}
durations<-names(FinalHemodynamicsDataset[,grep('_min',
names(FinalHemodynamicsDataset))])
for (i in 1:length(durations)) {
  colnam<-durations[i]
  if(sum(FinalHemodynamicsDataset[,colnam] <0, na.rm = TRUE)>0){
    cat(colnam)
    print(sum(FinalHemodynamicsDataset[,colnam] <0, na.rm = TRUE))
  }
}
...

```{r eval=FALSE, include=FALSE}
STS_filtered$Stroke[is.na(STS_filtered$AUT50MAP_total_min)]<-0
...

---
title: "Anesthesiologists Selection, Surgeons_all_yearsDF, Any MAE"
author: "Miguel Ángel Armengol de la Hoz"
date: "10 de agosto de 2017"
output:
  html_document:
    toc: true
    theme: united
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
...

```{r}
#load the data
library(dplyr)
#load("C:/BIG FILES/Phenotyping/HemodynamicProjectsDatasetFeb18.RData")
...

#Providers dataframes

##Cleaning Providers dataframe (ComboDF creation)

```{r}
ComboDF<- rbind(Combo08, Combo09,
Combo10,Combo11,Combo12,Combo13,Combo14,Combo15,Combo16)

```

```

#Correcting the mistake in duplicate physician

# code removed since it contained phi

ComboDF$Time<-NULL
ComboDF$ItemID<-NULL
ComboDF$C_Type<-NULL
ComboDF$C_Type_Label<-NULL
ComboDF$C_Association<-NULL
ComboDF$ComboText<-NULL
ComboDF$ESIG<-NULL
ComboDF$ESIGNAMETEXT<-NULL
ComboDF$ESIGDATEANDTIME<-NULL

...

## Selected Anesthesiologists DF

In this list we select Anesthesiologists performing overall >=30 surgeries from 2008-
2016

```{r}
#Selecting anesthesiologists
library(dplyr)
# code removed since it contained phi

unique(Anesthesiologists_selection)

Anesthesiologists_selection<-Anesthesiologists_selection %>%
mutate(Anesthesiologist=C_Item) %>%
dplyr::select(Case_Name,Anesthesiologist)

Anesthesiologists_selection<-unique(Anesthesiologists_selection)

...

*code removed since it contained phi*
---
title: "Vasopressors Columns Creation"
author: "Miguel Angel Armengol de la Hoz"
date: "10 de agosto de 2017"
output:
  html_document:
    toc: true
    theme: united
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```{r}
#load the data
library(dplyr)

```

```

#load("~/BIG FILES/Phenotyping/HemodynamicProjectsDatasetFeb18.RData")
```

#Using the new Drugs total table

DRUGTEXT

-Excluded:
  Calcium Chloride
  Calcium Gluconate
  Methylene Blue (Only 43 cases)

-Group A:
  Phenylephrine
  Ephedrine
  Norepinephrine
  Vasopressin
  Norepinephrine (800/50)

-Group B
  Milrinone
  Epinephrine

Load data and convert units

```{r}
library(readr)
library(readxl)
DrugTotals_forstudy <- read_csv("~/BIG FILES/Phenotyping/DrugTotals_forstudy.csv",
  col_types = readr::cols(DOS = col_date(format = "%m/%d/%Y"),
    Group = readr::col_factor(levels = c("A","B"))))

#DrugTotals_forstudy<-
#detach(package:plyr)
DrugTotals_forstudy<-DrugTotals_forstudy %>%
mutate(converted_dose_mg=if_else(UNITOFMEASURETEXT=="mcg",
round(DOSEAMOUNT/1000,2),round(DOSEAMOUNT,2)))

#remove vasopresing so we can interpretate the dosing
#DrugTotals_forstudy<-
DrugTotals_forstudy[!(DrugTotals_forstudy$DRUGTEXT=="Vasopressin"),]

#remove rows without drug info
DrugTotals_forstudy<-DrugTotals_forstudy[!is.na(DrugTotals_forstudy$DRUGTEXT),]
```

Drugs with formula

```

```
generate vasopressors = epinephrinetotal + norepinephrinetotal + (phenylephrinetotal / 10) + (dopaminetotal / weightkg / 2)
```

```
generate vasopressors = epinephrinetotal + norepinephrinetotal + (phenylephrinetotal / 10) + (milrinonetotal/5) + (vasopressin * 8.33)
```

(milrinone equivalence added by clinical criteria)

Ref: Mancl, E. E., & Muzevich, K. M. (2013). Tolerability and safety of enteral nutrition in critically ill patients receiving intravenous vasopressor therapy. Journal of parenteral and enteral nutrition, 37(5), 641-651.

```
` `{r}
```

```
drugsinformula<-subset(DrugTotals_forstudy, DrugTotals_forstudy$DRUGTEXT %in%  
c("Phenylephrine", "Epinephrine", "Norepinephrine", "Norepinephrine  
(800/50)", "Milrinone", "Vasopressin") )
```

```
drugsinformula$DRUGTEXT[drugsinformula$DRUGTEXT=='Norepinephrine (800/50)']<-  
"Norepinephrine"  
drugsinformula$MRN<-NULL  
drugsinformula$DOS<-NULL  
drugsinformula$Vasopressor<-NULL  
drugsinformula$DRUGITEMID<-NULL  
drugsinformula$EZINDEX<-NULL  
drugsinformula$CONTAINERTEXT<-NULL
```

```
drugsinformula<-drugsinformula %>%  
  mutate(converted_dose_mg=if_else(UNITOFMEASURETEXT=="mcg",  
    round(DOSEAMOUNT/1000,2),round(DOSEAMOUNT,2) )  
  )
```

```
drugsinformula$DOSEAMOUNT<-NULL  
drugsinformula$UNITOFMEASURETEXT<-NULL
```

```
library(dplyr)  
drugsinformula<-drugsinformula %>%  
dplyr::group_by(CASENUMBER, DRUGTEXT) %>%  
dplyr::summarise(total_indiv_drug=sum(converted_dose_mg))
```

```
library(data.table)
```

```
r_drugsinformula<-dcast(drugsinformula, CASENUMBER ~ DRUGTEXT,  
value.var='total_indiv_drug')  
r_drugsinformula[is.na(r_drugsinformula)]<-0
```

```
#generate vasopressors = epinephrinetotal + norepinephrinetotal + (phenylephrinetotal / 10) + milrinonetotal/5
```

```
generate_vasopressors <-  
function(epinephrinetotal,norepinephrinetotal,phenylephrinetotal,milrinonetotal,vasop  
ressintotal){  
  return(epinephrinetotal+norepinephrinetotal+(phenylephrinetotal/10)+(milrinonetotal/5  
) +  
    (vasopressintotal * 8.33)  
  )
```



```

}
r_drugsinformula<-r_drugsinformula%>%
  mutate(total_pressors_formula_mg=round(generate_vasopressors(Epinephrine,
Norepinephrine,Phenylephrine,Milrinone,Vasopressin) ,3))
colnames(r_drugsinformula)[1]<-'Case_Name'
```

#__NOT_USED____

Correcting NA

```{r}
# We are only interested on those Cases that have both Vitalsigns data and drug data
r_drugsinformula<-inner_join(All_Cases, r_drugsinformula)
r_drugsinformula[is.na(r_drugsinformula)]<-0

#The join returned 5504 thus there are 204 cases with no drug invormation. We are
making all those cases NA since we have no drug info about them.
```

#OLD Bolus Drugs

##Bolus Drugs quantities

```{r eval=FALSE, include=FALSE}
library(dplyr)

BolusDrugsDF <- rbind(BolusDrugs08, BolusDrugs09,
BolusDrugs10,BolusDrugs11,BolusDrugs12,BolusDrugs13,BolusDrugs14,BolusDrugs15,BolusDr
ugs16)

#After creating our DF, let's perform our filtering and table reshape

BolusDrugsDF<-subset(BolusDrugsDF, BolusDrugsDF$Label!= 'Calcium Chloride' &
BolusDrugsDF$Label != 'Calcium Gluconate')

#BolusDrugsDF$Time<-NULL
BolusDrugsDF$ItemID<-NULL
BolusDrugsDF$Route<-NULL
BolusDrugsDF$ESIG<-NULL
BolusDrugsDF$ESIGNAMETEXT<-NULL
BolusDrugsDF$ESIGDATEANDTIME<-NULL

#unique(BolusDrugsDF$Label)

BolusDrugsDF<-BolusDrugsDF %>% mutate(Phenylephrine= if_else(Label ==
'Phenylephrine',1,0))
BolusDrugsDF<-BolusDrugsDF %>% mutate(Epinephrine= if_else(Label ==
'Epinephrine',1,0))
BolusDrugsDF<-BolusDrugsDF %>% mutate('Norepinephrine (800/50) '= if_else(Label ==
'Norepinephrine (800/50)' ,1,0))
BolusDrugsDF<-BolusDrugsDF %>% mutate('Norepinephrine '= if_else(Label ==
'Norepinephrine' ,1,0))

```

```

BolusDrugsDF<-BolusDrugsDF %>% mutate('Vasopressin'= if_else(Label == 'Vasopressin'
,1,0))

library(reshape)

bolusdrugspercase<-reshape(BolusDrugsDF, timevar = 'Label' , idvar = 'Case_Name',
direction = 'wide')

#barplot(table(BolusDrugsDF$Units[BolusDrugsDF$Label=='Phenylephrine']))

...

##Bolus Drugs binary CPB location

```{r eval=FALSE, include=FALSE}
library(dplyr)

BolusDrugsDF_CPBloc<- rbind(BolusDrugs08, BolusDrugs09,
BolusDrugs10,BolusDrugs11,BolusDrugs12,BolusDrugs13,BolusDrugs14,BolusDrugs15,BolusDr
ugs16)

#After creating our DF, let's perform our filtering and table reshape

BolusDrugsDF_CPBloc<-subset(BolusDrugsDF_CPBloc, BolusDrugsDF_CPBloc$Label!='Calcium
Chloride' & BolusDrugsDF_CPBloc$Label != 'Calcium Gluconate')

#BolusDrugsDF_CPBloc$Time<-NULL
BolusDrugsDF_CPBloc$ItemID<-NULL
BolusDrugsDF_CPBloc$Route<-NULL
BolusDrugsDF_CPBloc$ESIG<-NULL
BolusDrugsDF_CPBloc$ESIGNAMETEXT<-NULL
BolusDrugsDF_CPBloc$ESIGDATEANDTIME<-NULL

#merge the datasets we need
BolusDrugsDF_CPBloc<-merge(BolusDrugsDF_CPBloc, rCPBEventsDF)

##Add the bypass begin and end for comparation

#preCPB
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate(Phenylephrine_preCPB =
if_else(Label == 'Phenylephrine' & Time < CPB_Begin_Time ,1,0))
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate(Epinephrine_preCPB =
if_else(Label == 'Epinephrine' & Time < CPB_Begin_Time ,1,0))
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate(Epinephrine_preCPB= if_else(Label
== 'Epinephrine'& Time < CPB_Begin_Time ,1,0))
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate('Norepinephrine(800/50)_preCPB'=
if_else(Label == 'Norepinephrine (800/50)' & Time < CPB_Begin_Time ,1,0))
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate('Norepinephrine_preCPB'=
if_else(Label == 'Norepinephrine' & Time < CPB_Begin_Time ,1,0))
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate(Vasopressin_preCPB= if_else(Label
== 'Vasopressin' & Time < CPB_Begin_Time ,1,0))

```

```

#CPB
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate(Phenylephrine_CPB = if_else(Label
== 'Phenylephrine' & Time >= CPB_Begin_Time & Time <= CPB_End_Time ,1,0))
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate(Epinephrine_CPB = if_else(Label
== 'Epinephrine' & Time >= CPB_Begin_Time & Time <= CPB_End_Time ,1,0))
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate(Epinephrine_CPB= if_else(Label ==
'Epinephrine'& Time >= CPB_Begin_Time & Time <= CPB_End_Time ,1,0))
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate('Norepinephrine(800/50)_CPB'=
if_else(Label == 'Norepinephrine (800/50)' & Time >= CPB_Begin_Time & Time <=
CPB_End_Time ,1,0))
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate('Norepinephrine_CPB'=
if_else(Label == 'Norepinephrine' & Time >= CPB_Begin_Time & Time <= CPB_End_Time
,1,0))
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate(Vasopressin_CPB= if_else(Label ==
'Vasopressin' & Time >= CPB_Begin_Time & Time <= CPB_End_Time ,1,0))

#postCPB
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate(Phenylephrine_postCPB =
if_else(Label == 'Phenylephrine' & Time > CPB_End_Time ,1,0))
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate(Epinephrine_postCPB =
if_else(Label == 'Epinephrine' & Time > CPB_End_Time ,1,0))
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate(Epinephrine_postCPB=
if_else(Label == 'Epinephrine'& Time > CPB_End_Time ,1,0))
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate('Norepinephrine(800/50)_postCPB'=
if_else(Label == 'Norepinephrine (800/50)' & Time > CPB_End_Time ,1,0))
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate('Norepinephrine_postCPB'=
if_else(Label == 'Norepinephrine' & Time > CPB_End_Time ,1,0))
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate(Vasopressin_postCPB=
if_else(Label == 'Vasopressin' & Time > CPB_End_Time ,1,0))

#total
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate(Phenylephrine_total =
if_else(Label == 'Phenylephrine' ,1,0))
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate(Epinephrine_total = if_else(Label
== 'Epinephrine' ,1,0))
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate(Epinephrine_total= if_else(Label
== 'Epinephrine',1,0))
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate('Norepinephrine(800/50)_total'=
if_else(Label == 'Norepinephrine (800/50)' ,1,0))
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate('Norepinephrine_total'=
if_else(Label == 'Norepinephrine' ,1,0))
BolusDrugsDF_CPBloc<-BolusDrugsDF_CPBloc %>% mutate(Vasopressin_total= if_else(Label
== 'Vasopressin' ,1,0))

#We summarize the information so we just know if there was or not an infusion of a
certain drug per CPB petiod per case

BolusDrugsDF_CPBloc_final<-subset(BolusDrugsDF_CPBloc, select = -c(2:8))

library(dplyr)

#preCPB

```

```

BolusDrugsDF_bloc_sum<-
setNames(aggregate(BolusDrugsDF_CPBloc_final$Phenylephrine_preCPB,
by=list(Case_Name=BolusDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
"Phenylephrine_bolus_preCPB"))
BolusDrugsDF_bloc_sum<-merge(BolusDrugsDF_bloc_sum,
setNames(aggregate(BolusDrugsDF_CPBloc_final$Epinephrine_preCPB,
by=list(Case_Name=BolusDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
"Epinephrine_bolus_preCPB"))))
BolusDrugsDF_bloc_sum<-merge(BolusDrugsDF_bloc_sum,
setNames(aggregate(BolusDrugsDF_CPBloc_final$`Norepinephrine(800/50)_preCPB`,
by=list(Case_Name=BolusDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Norepinephrine(800/50)_bolus_preCPB'))))
BolusDrugsDF_bloc_sum<-merge(BolusDrugsDF_bloc_sum,
setNames(aggregate(BolusDrugsDF_CPBloc_final$`Norepinephrine_preCPB`,
by=list(Case_Name=BolusDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Norepinephrine_bolus_preCPB'))))
BolusDrugsDF_bloc_sum<-merge(BolusDrugsDF_bloc_sum,
setNames(aggregate(BolusDrugsDF_CPBloc_final$Vasopressin_preCPB,
by=list(Case_Name=BolusDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Vasopressin_bolus_preCPB'))))

#CPB
BolusDrugsDF_bloc_sum<-merge(BolusDrugsDF_bloc_sum,
setNames(aggregate(BolusDrugsDF_CPBloc_final$Phenylephrine_CPB,
by=list(Case_Name=BolusDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
"Phenylephrine_bolus_CPB"))))
BolusDrugsDF_bloc_sum<-merge(BolusDrugsDF_bloc_sum,
setNames(aggregate(BolusDrugsDF_CPBloc_final$Epinephrine_CPB,
by=list(Case_Name=BolusDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
"Epinephrine_bolus_CPB"))))
BolusDrugsDF_bloc_sum<-merge(BolusDrugsDF_bloc_sum,
setNames(aggregate(BolusDrugsDF_CPBloc_final$`Norepinephrine(800/50)_CPB`,
by=list(Case_Name=BolusDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Norepinephrine(800/50)_bolus_CPB'))))
BolusDrugsDF_bloc_sum<-merge(BolusDrugsDF_bloc_sum,
setNames(aggregate(BolusDrugsDF_CPBloc_final$`Norepinephrine_CPB`,
by=list(Case_Name=BolusDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Norepinephrine_bolus_CPB'))))
BolusDrugsDF_bloc_sum<-merge(BolusDrugsDF_bloc_sum,
setNames(aggregate(BolusDrugsDF_CPBloc_final$Vasopressin_CPB,
by=list(Case_Name=BolusDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Vasopressin_bolus_CPB'))))

#postCPB
BolusDrugsDF_bloc_sum<-merge(BolusDrugsDF_bloc_sum,
setNames(aggregate(BolusDrugsDF_CPBloc_final$Phenylephrine_postCPB,
by=list(Case_Name=BolusDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
"Phenylephrine_bolus_postCPB"))))
BolusDrugsDF_bloc_sum<-merge(BolusDrugsDF_bloc_sum,
setNames(aggregate(BolusDrugsDF_CPBloc_final$Epinephrine_postCPB,
by=list(Case_Name=BolusDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
"Epinephrine_bolus_postCPB"))))

```

```

BolusDrugsDF_bloc_sum<-merge(BolusDrugsDF_bloc_sum,
setNames(aggregate(BolusDrugsDF_CPBloc_final$`Norepinephrine(800/50)_postCPB`,
by=list(Case_Name=BolusDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Norepinephrine(800/50)_bolus_postCPB'))))
BolusDrugsDF_bloc_sum<-merge(BolusDrugsDF_bloc_sum,
setNames(aggregate(BolusDrugsDF_CPBloc_final$`Norepinephrine_postCPB`,
by=list(Case_Name=BolusDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Norepinephrine_bolus_postCPB'))))
BolusDrugsDF_bloc_sum<-merge(BolusDrugsDF_bloc_sum,
setNames(aggregate(BolusDrugsDF_CPBloc_final$Vasopressin_postCPB,
by=list(Case_Name=BolusDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Vasopressin_bolus_postCPB'))))

#total
BolusDrugsDF_bloc_sum<-merge(BolusDrugsDF_bloc_sum,
setNames(aggregate(BolusDrugsDF_CPBloc_final$Phenylephrine_total,
by=list(Case_Name=BolusDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
"Phenylephrine_bolus_total"))))
BolusDrugsDF_bloc_sum<-merge(BolusDrugsDF_bloc_sum,
setNames(aggregate(BolusDrugsDF_CPBloc_final$Epinephrine_total,
by=list(Case_Name=BolusDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
"Epinephrine_bolus_total"))))
BolusDrugsDF_bloc_sum<-merge(BolusDrugsDF_bloc_sum,
setNames(aggregate(BolusDrugsDF_CPBloc_final$`Norepinephrine(800/50)_total`,
by=list(Case_Name=BolusDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Norepinephrine(800/50)_bolus_total'))))
BolusDrugsDF_bloc_sum<-merge(BolusDrugsDF_bloc_sum,
setNames(aggregate(BolusDrugsDF_CPBloc_final$`Norepinephrine_total`,
by=list(Case_Name=BolusDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Norepinephrine_bolus_total'))))
BolusDrugsDF_bloc_sum<-merge(BolusDrugsDF_bloc_sum,
setNames(aggregate(BolusDrugsDF_CPBloc_final$Vasopressin_total,
by=list(Case_Name=BolusDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Vasopressin_bolus_total'))))

#in the following code we get rid of the quantity and convert it to boolean
BolusDrugs.bin<-replace(BolusDrugsDF_bloc_sum, BolusDrugsDF_bloc_sum > 0, 1)
BolusDrugs.bin<-sapply(BolusDrugs.bin, as.factor)
BolusDrugs.bin<-as.data.frame(BolusDrugs.bin)

#we exclude those cases not present in our cohort (STS+AIMS join)
BolusDrugs.bin<-inner_join(, BolusDrugs.bin)

...

# OLD Infusion Drugs

##Infusion Drugs quantities

```{r eval=FALSE, include=FALSE}
library(dplyr)

```

```

InfusionDrugsDF <- rbind(InfusionDrugs08, InfusionDrugs09,
InfusionDrugs10, InfusionDrugs11, InfusionDrugs12, InfusionDrugs13, InfusionDrugs14, InfusionDrugs15, InfusionDrugs16)

#InfusionDrugsDF$Time<-NULL
InfusionDrugsDF$ItemID<-NULL
InfusionDrugsDF$Route<-NULL
InfusionDrugsDF$ESIG<-NULL
InfusionDrugsDF$ESIGNAMETEXT<-NULL
InfusionDrugsDF$ESIGDATEANDTIME<-NULL

#unique(InfusionDrugsDF$Label)

InfusionDrugsDF<-InfusionDrugsDF %>% mutate(Phenylephrine= if_else(Label ==
'Phenylephrine',1,0))
InfusionDrugsDF<-InfusionDrugsDF %>% mutate(Epinephrine= if_else(Label ==
'Epinephrine',1,0))
InfusionDrugsDF<-InfusionDrugsDF %>% mutate('Norepinephrine (800/50)'= if_else(Label
== 'Norepinephrine (800/50)' ,1,0))
InfusionDrugsDF<-InfusionDrugsDF %>% mutate('Norepinephrine'= if_else(Label ==
'Norepinephrine' ,1,0))
InfusionDrugsDF<-InfusionDrugsDF %>% mutate('Vasopressin'= if_else(Label ==
'Vasopressin' ,1,0))

library(reshape)

InfusionDrugspercase<-reshape(InfusionDrugsDF, timevar = 'Label' , idvar =
'Case_Name', direction = 'wide')

#barplot(table(InfusionDrugsDF$Units[InfusionDrugsDF$Label=='Phenylephrine']))
...

##Infusion Drugs binary CPB location

```{r eval=FALSE, include=FALSE}
library(dplyr)

InfusionDrugsDF_CPBloc<- rbind(InfusionDrugs08, InfusionDrugs09,
InfusionDrugs10, InfusionDrugs11, InfusionDrugs12, InfusionDrugs13, InfusionDrugs14, InfusionDrugs15, InfusionDrugs16)

#After creating our DF, let's perform our filtering and table reshape

InfusionDrugsDF_CPBloc<-subset(InfusionDrugsDF_CPBloc,
InfusionDrugsDF_CPBloc$Label!='Calcium Chloride' & InfusionDrugsDF_CPBloc$Label !=
'Calcium Gluconate')

#InfusionDrugsDF_CPBloc$Time<-NULL
InfusionDrugsDF_CPBloc$ItemID<-NULL
InfusionDrugsDF_CPBloc$Route<-NULL
InfusionDrugsDF_CPBloc$ESIG<-NULL

```

```

InfusionDrugsDF_CPBloc$ESIGNAMETEXT<-NULL
InfusionDrugsDF_CPBloc$ESIGDATEANDTIME<-NULL

#merge the datasets we need
InfusionDrugsDF_CPBloc<-merge(InfusionDrugsDF_CPBloc, rCPBEventsDF)

##Add the bypass begin and end for comparation

#preCPB
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>% mutate(Phenylephrine_preCPB =
if_else(Label == 'Phenylephrine' & Time < CPB_Begin_Time ,1,0))
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>% mutate(Epinephrine_preCPB =
if_else(Label == 'Epinephrine' & Time < CPB_Begin_Time ,1,0))
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>% mutate(Epinephrine_preCPB=
if_else(Label == 'Epinephrine'& Time < CPB_Begin_Time ,1,0))
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>%
mutate('Norepinephrine(800/50)_preCPB'= if_else(Label == 'Norepinephrine (800/50)' &
Time < CPB_Begin_Time ,1,0))
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>% mutate('Norepinephrine_preCPB'=
if_else(Label == 'Norepinephrine' & Time < CPB_Begin_Time ,1,0))
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>% mutate(Vasopressin_preCPB=
if_else(Label == 'Vasopressin' & Time < CPB_Begin_Time ,1,0))

#CPB
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>% mutate(Phenylephrine_CPB =
if_else(Label == 'Phenylephrine' & Time >= CPB_Begin_Time & Time <= CPB_End_Time
,1,0))
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>% mutate(Epinephrine_CPB =
if_else(Label == 'Epinephrine' & Time >= CPB_Begin_Time & Time <= CPB_End_Time ,1,0))
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>% mutate(Epinephrine_CPB=
if_else(Label == 'Epinephrine'& Time >= CPB_Begin_Time & Time <= CPB_End_Time ,1,0))
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>%
mutate('Norepinephrine(800/50)_CPB'= if_else(Label == 'Norepinephrine (800/50)' &
Time >= CPB_Begin_Time & Time <= CPB_End_Time ,1,0))
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>% mutate('Norepinephrine_CPB'=
if_else(Label == 'Norepinephrine' & Time >= CPB_Begin_Time & Time <= CPB_End_Time
,1,0))
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>% mutate(Vasopressin_CPB=
if_else(Label == 'Vasopressin' & Time >= CPB_Begin_Time & Time <= CPB_End_Time ,1,0))

#postCPB
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>% mutate(Phenylephrine_postCPB =
if_else(Label == 'Phenylephrine' & Time > CPB_End_Time ,1,0))
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>% mutate(Epinephrine_postCPB =
if_else(Label == 'Epinephrine' & Time > CPB_End_Time ,1,0))
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>% mutate(Epinephrine_postCPB=
if_else(Label == 'Epinephrine'& Time > CPB_End_Time ,1,0))
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>%
mutate('Norepinephrine(800/50)_postCPB'= if_else(Label == 'Norepinephrine (800/50)' &
Time > CPB_End_Time ,1,0))
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>% mutate('Norepinephrine_postCPB'=
if_else(Label == 'Norepinephrine' & Time > CPB_End_Time ,1,0))

```

```

InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>% mutate(Vasopressin_postCPB=
if_else(Label == 'Vasopressin' & Time > CPB_End_Time ,1,0))

#total
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>% mutate(Phenylephrine_total =
if_else(Label == 'Phenylephrine' ,1,0))
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>% mutate(Epinephrine_total =
if_else(Label == 'Epinephrine' ,1,0))
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>% mutate(Epinephrine_total=
if_else(Label == 'Epinephrine',1,0))
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>%
mutate('Norepinephrine(800/50)_total'= if_else(Label == 'Norepinephrine (800/50)'
,1,0))
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>% mutate('Norepinephrine_total'=
if_else(Label == 'Norepinephrine' ,1,0))
InfusionDrugsDF_CPBloc<-InfusionDrugsDF_CPBloc %>% mutate(Vasopressin_total=
if_else(Label == 'Vasopressin' ,1,0))

#We summarize the information so we just know if there was or not an infusion of a
certain drug per CPB petiod per case

InfusionDrugsDF_CPBloc_final<-subset(InfusionDrugsDF_CPBloc, select = -c(2:8))

library(dplyr)

#preCPB
InfusionDrugsDF_bloc_sum<-
setNames(aggregate(InfusionDrugsDF_CPBloc_final$Phenylephrine_preCPB,
by=list(Case_Name=InfusionDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
"Phenylephrine_Infusion_preCPB"))
InfusionDrugsDF_bloc_sum<-merge(InfusionDrugsDF_bloc_sum,
setNames(aggregate(InfusionDrugsDF_CPBloc_final$Epinephrine_preCPB,
by=list(Case_Name=InfusionDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
"Epinephrine_Infusion_preCPB")))
InfusionDrugsDF_bloc_sum<-merge(InfusionDrugsDF_bloc_sum,
setNames(aggregate(InfusionDrugsDF_CPBloc_final$`Norepinephrine(800/50)_preCPB`,
by=list(Case_Name=InfusionDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Norepinephrine(800/50)_Infusion_preCPB'))))
InfusionDrugsDF_bloc_sum<-merge(InfusionDrugsDF_bloc_sum,
setNames(aggregate(InfusionDrugsDF_CPBloc_final$`Norepinephrine_preCPB`,
by=list(Case_Name=InfusionDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Norepinephrine_Infusion_preCPB'))))
InfusionDrugsDF_bloc_sum<-merge(InfusionDrugsDF_bloc_sum,
setNames(aggregate(InfusionDrugsDF_CPBloc_final$Vasopressin_preCPB,
by=list(Case_Name=InfusionDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Vasopressin_Infusion_preCPB'))))

#CPB
InfusionDrugsDF_bloc_sum<-merge(InfusionDrugsDF_bloc_sum,
setNames(aggregate(InfusionDrugsDF_CPBloc_final$Phenylephrine_CPB,
by=list(Case_Name=InfusionDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
"Phenylephrine_Infusion_CPB")))
InfusionDrugsDF_bloc_sum<-merge(InfusionDrugsDF_bloc_sum,
setNames(aggregate(InfusionDrugsDF_CPBloc_final$Epinephrine_CPB,

```



```

by=list(Case_Name=InfusionDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
"Epinephrine_Infusion_CPB"))))
InfusionDrugsDF_bloc_sum<-merge(InfusionDrugsDF_bloc_sum,
setNames(aggregate(InfusionDrugsDF_CPBloc_final$`Norepinephrine(800/50)_CPB`,
by=list(Case_Name=InfusionDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Norepinephrine(800/50)_Infusion_CPB'))))
InfusionDrugsDF_bloc_sum<-merge(InfusionDrugsDF_bloc_sum,
setNames(aggregate(InfusionDrugsDF_CPBloc_final$`Norepinephrine_CPB`,
by=list(Case_Name=InfusionDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Norepinephrine_Infusion_CPB'))))
InfusionDrugsDF_bloc_sum<-merge(InfusionDrugsDF_bloc_sum,
setNames(aggregate(InfusionDrugsDF_CPBloc_final$Vasopressin_CPB,
by=list(Case_Name=InfusionDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Vasopressin_Infusion_CPB'))))

```

#postCPB

```

InfusionDrugsDF_bloc_sum<-merge(InfusionDrugsDF_bloc_sum,
setNames(aggregate(InfusionDrugsDF_CPBloc_final$Phenylephrine_postCPB,
by=list(Case_Name=InfusionDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
"Phenylephrine_Infusion_postCPB"))))
InfusionDrugsDF_bloc_sum<-merge(InfusionDrugsDF_bloc_sum,
setNames(aggregate(InfusionDrugsDF_CPBloc_final$Epinephrine_postCPB,
by=list(Case_Name=InfusionDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
"Epinephrine_Infusion_postCPB"))))
InfusionDrugsDF_bloc_sum<-merge(InfusionDrugsDF_bloc_sum,
setNames(aggregate(InfusionDrugsDF_CPBloc_final$`Norepinephrine(800/50)_postCPB`,
by=list(Case_Name=InfusionDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Norepinephrine(800/50)_Infusion_postCPB'))))
InfusionDrugsDF_bloc_sum<-merge(InfusionDrugsDF_bloc_sum,
setNames(aggregate(InfusionDrugsDF_CPBloc_final$`Norepinephrine_postCPB`,
by=list(Case_Name=InfusionDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Norepinephrine_Infusion_postCPB'))))
InfusionDrugsDF_bloc_sum<-merge(InfusionDrugsDF_bloc_sum,
setNames(aggregate(InfusionDrugsDF_CPBloc_final$Vasopressin_postCPB,
by=list(Case_Name=InfusionDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Vasopressin_Infusion_postCPB'))))

```

#total

```

InfusionDrugsDF_bloc_sum<-merge(InfusionDrugsDF_bloc_sum,
setNames(aggregate(InfusionDrugsDF_CPBloc_final$Phenylephrine_total,
by=list(Case_Name=InfusionDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
"Phenylephrine_Infusion_total"))))
InfusionDrugsDF_bloc_sum<-merge(InfusionDrugsDF_bloc_sum,
setNames(aggregate(InfusionDrugsDF_CPBloc_final$Epinephrine_total,
by=list(Case_Name=InfusionDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
"Epinephrine_Infusion_total"))))
InfusionDrugsDF_bloc_sum<-merge(InfusionDrugsDF_bloc_sum,
setNames(aggregate(InfusionDrugsDF_CPBloc_final$`Norepinephrine(800/50)_total`,
by=list(Case_Name=InfusionDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Norepinephrine(800/50)_Infusion_total'))))
InfusionDrugsDF_bloc_sum<-merge(InfusionDrugsDF_bloc_sum,
setNames(aggregate(InfusionDrugsDF_CPBloc_final$`Norepinephrine_total`,
by=list(Case_Name=InfusionDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Norepinephrine_Infusion_total'))))

```

```

InfusionDrugsDF_bloc_sum<-merge(InfusionDrugsDF_bloc_sum,
setNames(aggregate(InfusionDrugsDF_CPBloc_final$Vasopressin_total,
by=list(Case_Name=InfusionDrugsDF_CPBloc_final$Case_Name), FUN=sum), c("Case_Name",
'Vasopressin_Infusion_total'))))

#in the following code we get rid of the quantity and convert it to boolean
InfusionDrugs.bin<-replace(InfusionDrugsDF_bloc_sum, InfusionDrugsDF_bloc_sum > 0, 1)
InfusionDrugs.bin<-sapply(InfusionDrugs.bin, as.factor)

InfusionDrugs.bin<-as.data.frame(InfusionDrugs.bin)

...

```{r}
generate_vasopressors <- function(epinephrine_total, norepinephrine_total,
phenylephrine_total ){
generate_vasopressors = epinephrine_total + norepinephrine_total +
(phenylephrine_total/10)
return(generate_vasopressors)
}
...

# OLD Units normalization

## Study data
```{r eval=FALSE, include=FALSE}
library(dplyr)
BolusDrugsDF %>%
  group_by(Label, Units) %>%
  summarise(n = n()) %>%
  arrange(Label,desc(n))

library(dplyr)
InfusionDrugsDF %>%
  group_by(Label, Dose_Units) %>%
  summarise(n = n()) %>%
  arrange(Label,desc(n))
...

# OLD Groups

## Calculating total doses in group A pressors per case

```{r eval=FALSE, include=FALSE}
library(plyr)
detach(package:plyr)
library(dplyr)
drugTotalspercase_and_drug<-subset(DrugTotals_forstudy, Group=='A') %>%
  group_by(CASENUMBER, DRUGTEXT) %>%

```

```

    mutate(total_drug = cumsum(converted_dose_mg))

drugs_groupA_percase<-drugTotalspercase_and_drug %>% group_by(CASENUMBER) %>%
dplyr::summarize(total_groupA = max(total_drug))

names(drugs_groupA_percase)[1]<-'Case_Name'
...

## Calculating total doses in group B pressors per case

```{r eval=FALSE, include=FALSE}
library(dplyr)
drugTotalspercase_and_drug<-subset(DrugTotals_forstudy, Group=='B') %>%
  group_by(CASENUMBER) %>%
  mutate(total_drug = cumsum(converted_dose_mg))

drugs_groupB_percase<-drugTotalspercase_and_drug %>% group_by(CASENUMBER) %>%
summarise(total_groupB = max(total_drug))

names(drugs_groupB_percase)[1]<-'Case_Name'
...

---
title: "Input Output, HCT and EF variables creation"
author: "Miguel Angel Armengol de la Hoz"
date: "10 de agosto de 2017"
output:
  html_document:
    toc: true
    theme: united
---

# Libraries load

```{r setup, include=FALSE}
library(dplyr)
library(reshape2)
library(summarytools)
library(readxl)
library(reshape)
...

# Data Load

```{r}
# IO_data_Blood <- read_excel("~/BIG FILES/Phenotyping/Hemodynamics_IO_Blood.xlsx")
# IO_data_Fluids <- read_excel("~/BIG FILES/Phenotyping/Hemodynamics_IO_Fluids.xlsx")
# IO_data_Urine_Output <- read_excel("~/BIG
FILES/Phenotyping/Hemodynamics_IO_Urine_Output.xlsx")
#
# First_HCT <- read_excel("~/BIG FILES/Phenotyping/First HCT.xlsx", col_types =
c("text", "numeric", "date", "numeric", "text", "numeric", "text", "numeric"))
# First_HCT_after_SurgTime <- read_excel("~/BIG FILES/Phenotyping/First HCT After C
Time.xlsx",

```

```

# col_types = c("text", "numeric", "date",
#               "text", "date", "date", "text", "numeric",
#               "text"))

#load("C:/BIG FILES/Phenotyping/HemodynamicProjectsDataset101218.RData")
EF <- read_excel("/Users/marmengol/MEGA/Boston/BIDMC-Harvard/Phenotyping/Dataset
Creation/EF.xlsx",
  sheet = "echos", col_types = c("blank",
  "date", "blank", "blank", "blank",
  "blank", "text", "blank", "blank",
  "blank", "numeric", "numeric"))
EF$Surgery_Date<-as.Date(EF$Surgery_Date)

EF<-inner_join(
  EF,
  CasesofInterestDF[,c("Case_Name", "Surgery_Date", "Patient_ID")]
  ,by=c('mrn'='Patient_ID', 'Surgery_Date'='Surgery_Date')
)
...

# Loading the new dataset

```{r message=FALSE, warning=FALSE}

#let's first select only those columns we need

IO_data_Blood_filtered<-IO_data_Blood %>%
dplyr::select(Case_Name, RECORDTIME, Blood,VALUE)

IO_data_Blood_filtered<-IO_data_Blood_filtered%>%
dplyr::rename(Time=RECORDTIME)

IO_data_Fluids_filtered<-IO_data_Fluids %>%
dplyr::select(Case_Name, RECORDTIME, Fluid,VALUE)

IO_data_Fluids_filtered<-IO_data_Fluids_filtered%>%
dplyr::rename(Time=RECORDTIME)

IO_data_Urine_Output_filtered<-IO_data_Urine_Output %>%
dplyr::select(Case_Name, RECORDTIME, VALUE)

IO_data_Urine_Output_filtered<-IO_data_Urine_Output_filtered%>%
dplyr::rename(Time=RECORDTIME)

IO_data_Urine_Output_filtered['Urine_Output']<- 'Urine_Output'

First_HCT<-First_HCT%>%
dplyr::select(Case_Name,result_name,result_value)

First_HCT_after_SurgTime<-First_HCT_after_SurgTime%>%
dplyr::select(Case_Name,result_name,result_value)
...

# Sumarizing datasets

```

```

## IO_data_Blood

```{r}
#All units are in ml

my.sum <- function(x) ifelse( !all(is.na(x)), sum(x, na.=T), NA)

IO_data_Blood_summarized<-IO_data_Blood_filtered%>%
dplyr::select(Case_Name,Blood,VALUE)%>%
  cast(Case_Name~Blood,fun.aggregate = my.sum)

IO_data_Blood_summarized<-IO_data_Blood_summarized%>%
  mutate(
    Blood=rowSums(.[c(8,11,12)], na.rm=TRUE)*
ifelse(rowSums(is.na(IO_data_Blood_summarized[,c(8,11,12)])) ==
ncol(IO_data_Blood_summarized[,c(8,11,12)]), NA, 1)
  )

IO_data_Blood_summarized<-IO_data_Blood_summarized%>%
dplyr::select(Case_Name,`Cell Saver`,Blood)

IO_data_Blood_summarized<-IO_data_Blood_summarized%>%
  dplyr::rename(Cell_Saver_ml=`Cell Saver`)%>%
  dplyr::rename(Blood_ml=Blood)

...

## IO_data_Fluids

```{r}
#All units are in ml

my.sum <- function(x) ifelse( !all(is.na(x)), sum(x, na.rm=T), NA)

IO_data_Fluids_summarized<-IO_data_Fluids_filtered%>%
dplyr::select(Case_Name,Fluid,VALUE)%>%
  cast(Case_Name~Fluid,fun.aggregate = my.sum)

IO_data_Fluids_summarized<-IO_data_Fluids_summarized%>%
  mutate(
    Colloids_ml=rowSums(.[c(3,8)], na.rm=TRUE)* ifelse(rowSums(is.na(.[c(3,8)])) ==
ncol(.[c(3,8)]), NA, 1)
    ,Crystalloids_ml =rowSums(.[c(2,4,5,6,7,9,10,11,12)],
na.rm=TRUE)*ifelse(rowSums(is.na(IO_data_Fluids_summarized[,c(2,4,5,6,7,9,10,11,12)]))
) == ncol(IO_data_Fluids_summarized[,c(2,4,5,6,7,9,10,11,12)]), NA, 1)
  )

IO_data_Fluids_summarized<-IO_data_Fluids_summarized%>%
dplyr::select(Case_Name,Colloids_ml,Crystalloids_ml)

...

## IO_data_Urine_Output

```

```

```{r}
#All units are in ml

my.sum <- function(x) ifelse( !all(is.na(x)), sum(x, na.rm=T), NA)

IO_data_Urine_Output_summarized<-IO_data_Urine_Output_filtered%>%
dplyr::select(Case_Name,Urine_Output,VALUE)%>%
  cast(Case_Name~Urine_Output,fun.aggregate = my.sum)

IO_data_Urine_Output_summarized<-
IO_data_Urine_Output_summarized%>%dplyr::rename(Urine_Output_ml=Urine_Output)
```

First_HCT

```{r}
my.sum <- function(x) ifelse( !all(is.na(x)), sum(x, na.rm=T), NA)

First_HCT_summarized<-First_HCT%>%cast(Case_Name~result_name,fun.aggregate = my.sum)
First_HCT_summarized<-First_HCT_summarized%>%dplyr::rename(First_HCT_pct=Hematocrit)
```

First_HCT_after_SurgTime

```{r}
my.sum <- function(x) ifelse( !all(is.na(x)), sum(x, na.rm=T), NA)

First_HCT_after_SurgTime_summarized<-
First_HCT_after_SurgTime%>%cast(Case_Name~result_name,fun.aggregate = my.sum)
First_HCT_after_SurgTime_summarized<-
First_HCT_after_SurgTime_summarized%>%dplyr::rename(First_HCT_after_SurgTime_pct=Hema
tocrit)
```

Delta HTC

```{r}
All_HCT<-left_join(First_HCT_summarized,First_HCT_after_SurgTime_summarized)
All_HCT['delta_HCT_pct']<-(All_HCT$First_HCT_after_SurgTime_pct -
All_HCT$First_HCT_pct)

```

Create dataset for joining with our cohort

```{r}
# IO_data<-left_join(All_Cases,IO_data_Blood_summarized)
# IO_data<-left_join(IO_data,IO_data_Fluids_summarized)
# IO_data<-left_join(IO_data,IO_data_Urine_Output_summarized)
# IO_data<-left_join(IO_data,First_HCT_summarized)
# IO_data<-left_join(IO_data,First_HCT_after_SurgTime_summarized)
IO_data<-left_join(All_Cases,All_HCT)
```

```

```

Mean Ejection Fraction

Creating necessary formulas

```{r}
# creating my functions
my.min <- function(x) ifelse( !all(is.na(x)), min(x, na.rm=T), NA)
my.max <- function(x) ifelse( !all(is.na(x)), max(x, na.rm=T), NA)
my.sum <- function(x) ifelse( !all(is.na(x)), sum(x, na.rm=T), NA)
my.mean<- function(x,y) ifelse( is.na(x) | is.na(y)
                                , sum(x,y,na.rm = T)
                                , (x+y)/2
                                )
...

## Calculating mean EF

```{r}
Ejection_Fraction<-EF %>%
  dplyr::select(Case_Name,lv_ef_min,lv_ef_max ) %>%
  dplyr::group_by(Case_Name) %>%
  dplyr::summarise(
    min_ef_among_all=my.min(lv_ef_min)
    ,max_ef_among_all=my.max(lv_ef_max)
  )

Ejection_Fraction<-Ejection_Fraction%>%
  rowwise() %>%
  dplyr::mutate(
    mean_ef = my.mean(min_ef_among_all,max_ef_among_all)
  )

Ejection_Fraction<-Ejection_Fraction%>%
  dplyr::select(Case_Name,mean_ef)
...
---
title: "FINAL_JOIN and de-identification"
author: "Miguel Angel Armengol de la Hoz"
output:
  html_document:
    toc: true
    theme: united
---

```{r}
#Load the enviroment
#load("C:/BIG FILES/Phenotyping/HemodynamicProjectsDatasetFeb18.RData")
...

# FINAL LEFT JOIN

## STS dataset with HPE columns dataset join

```

```

```{r}
library(dplyr)
FinalHemodynamicsDataset<-inner_join(CasesofInterestDF,
AllPeriodswith_AUT_AOT_and_AWI_DF)
FinalHemodynamicsDataset$Surgery_Date<-as.Date(FinalHemodynamicsDataset$Surgery_Date,
"%Y-%m-%d")
```

Joining with AUC-related variables

```{r}
FinalHemodynamicsDataset<-inner_join(FinalHemodynamicsDataset,All_AUC65MAP_variables)
```

STS and HPE dataset join with Anesthesiologists_all_yearsDF and
Surgeons_all_yearsDF

```{r}
FinalHemodynamicsDataset<-left_join(FinalHemodynamicsDataset,
Anesthesiologists_selection)
#FinalHemodynamicsDataset<-left_join(FinalHemodynamicsDataset, Surgeons_all_yearsDF)
```

Removing PHI

```{r}
#Remove Patient Identifier variables
FinalHemodynamicsDataset$Patient_ID <- NULL
FinalHemodynamicsDataset$First_Name <- NULL
FinalHemodynamicsDataset$Last_Name <- NULL
FinalHemodynamicsDataset$SS_Patient_ID <- NULL
```

De-Identifying Anesthesiologists

```{r}

library(plyr)

#Anesthesiologist
FinalHemodynamicsDataset$Anesthesiologist<-
as.character(FinalHemodynamicsDataset$Anesthesiologist)
FinalHemodynamicsDataset$Anesthesiologist<-
revalue(FinalHemodynamicsDataset$Anesthesiologist, c(
'20025'='A','20026'='B','20030'='C','20036'='D','20038'='E','20051'='F','20156'='G','
20359'='H','20556'='I','20935'='J','21084'='K','21105'='L','21379'='M','21507'='N','2
1619'='O','21839'='P'
))

# Victor suggested studying providers involved in at least 30 surgeries between 2008-
2016
```

```



```

De-identifying Surgeons

```{r eval=FALSE, include=FALSE}
#Surgeon
FinalHemodynamicsDataset$Surgeon<-as.character(FinalHemodynamicsDataset$Surgeon)
FinalHemodynamicsDataset$Surgeon<-revalue(FinalHemodynamicsDataset$Surgeon,
c('20306'="alpha", '20479' = 'beta', '20966' = 'gamma'))

#We chose only Surgeons performing procedures along the 7 years but in some cases
within these 7 years, none of these 3 Surgeonss are performing the procedure. in
these cases Surgeons = NA
```

Joining total pressors formula

```{r}
FinalHemodynamicsDataset<-inner_join(FinalHemodynamicsDataset,
r_drugsinformula[,c("Case_Name","total_pressors_formula_mg")])
```

<!-- #Mini dataset Export -->
<!-- ```{r} -->
<!-- minidataset_selected_columns<-
c("Case_Name","Surgery_Date","Gender","Age","Category","STSBAS","ChronicLungDisease",
"Death","Reop_Bleeding","Pulm_Pneumonia","Renal_Failure","Oth_Tamponade","Oth_AFib","
Perioperative_MI","HbA1C","PreOpMedBetaBlockers","PreOpMedACE_ARBInhibitors","PreOpMe
dSteroids","PreOpMedInotropes","PreOpMedAspirin","PreOpMedLipidLowering","STS_RiskAlg
orithm","STS_Pred_Deep_Stern","STS_Pred_ReOp","STS_Pred_PermCVA","STS_Pred_ProlongVen
t","STS_Pred_RenalFail","STS_Pred_MorbMort","STS_Pred_ShortLOS","STS_Pred_LongLOS","H
ypertension.bin","CHF.bin","Cerebrovascular_Disease.bin","PreviousMI.bin","Diabetes.b
in","Family_History_CAD.bin","Status","Any_MAE","Sur_Begin_Time","Sur_End_Time","CPB_
Begin_Time","CPB_End_Time","duration_preCPB_min","duration_CPB_min","duration_postCPB
_min","duration_Surgery_min","AUT50MAP_total_min","AUT50MAP_preCPB_min","AUT50MAP_CP
B_min","AUT50MAP_postCPB_min","AUT50MAP_total_ratio","AUT50MAP_preCPB_ratio","AUT50MA
P_CPB_ratio","AUT50MAP_postCPB_ratio","AWI51_60MAP_total_min","AWI51_60MAP_preCPB_min
","AWI51_60MAP_CPB_min","AWI51_60MAP_postCPB_min","AWI51_60MAP_total_ratio","AWI51_60M
AP_preCPB_ratio","AWI51_60MAP_CPB_ratio","AWI51_60MAP_postCPB_ratio","AWI61_65MAP_tot
al_min","AWI61_65MAP_preCPB_min","AWI61_65MAP_CPB_min","AWI61_65MAP_postCPB_min","AWI
61_65MAP_total_ratio","AWI61_65MAP_preCPB_ratio","AWI61_65MAP_CPB_ratio","AWI61_65MA
P_postCPB_ratio","AUT95SBP_total_min","AUT95SBP_preCPB_min","AUT95SBP_postCPB_min","AU
T95SBP_total_ratio","AUT95SBP_preCPB_ratio","AUT95SBP_postCPB_ratio","AUT30DBP_total_
min","AUT30DBP_preCPB_min","AUT30DBP_postCPB_min","AUT30DBP_total_ratio","AUT30DBP_pr
eCPB_ratio","AUT30DBP_postCPB_ratio","Anesthesiologist","Surgeon") -->

<!-- FinalHemodynamicsDatasetMINI<-FinalHemodynamicsDataset[,
names(FinalHemodynamicsDataset) %in% minidataset_selected_columns] -->

<!-- write.table(FinalHemodynamicsDatasetMINI, "FinalHemodynamicsDatasetMINI.txt",
sep="\t", row.names = FALSE) -->
<!-- ``` -->

Solving NA problems

```{r}

```

```

#
FinalHemodynamicsDataset$AUT50MAP_total_min[is.na(FinalHemodynamicsDataset$AUT50MAP_t
otal_min)]<-0
#
FinalHemodynamicsDataset$AUT65MAP_total_min[is.na(FinalHemodynamicsDataset$AUT65MAP_t
otal_min)]<-0
#
FinalHemodynamicsDataset$AWI51_60MAP_total_min[is.na(FinalHemodynamicsDataset$AWI51_6
0MAP_total_min)]<-0
#
FinalHemodynamicsDataset$AWI61_65MAP_total_min[is.na(FinalHemodynamicsDataset$AWI61_6
5MAP_total_min)]<-0
#
FinalHemodynamicsDataset$AOT65MAP_total_min[is.na(FinalHemodynamicsDataset$AOT65MAP_t
otal_min)]<-0
...

# Export

```{r}
#write.table(FinalHemodynamicsDataset, "FinalHemodynamicsDataset.txt", sep="\t",
row.names = FALSE)
```

---
title: "11_Quartile_Columns"
author: "Miguel Angel Armengol de la Hoz"
output:
  html_document:
    toc: true
    theme: united

```{r}
library(Hmisc)

#Load the enviroment
#load("C:/BIG FILES/Phenotyping/HemodynamicProjectsDatasetFeb18.RData")
...

# Quartiles per variable

```{r}
Q_total_pressors_formula_mg<-
summary(FinalHemodynamicsDataset$total_pressors_formula_mg)
Q_AUT50MAP_total_min<-summary(FinalHemodynamicsDataset$AUT50MAP_total_min)
Q_AWI51_60MAP_total_min<-summary(FinalHemodynamicsDataset$AWI51_60MAP_total_min)
Q_AWI61_65MAP_total_min<-summary(FinalHemodynamicsDataset$AWI61_65MAP_total_min)
Q_AOT65MAP_total_min<-summary(FinalHemodynamicsDataset$AOT65MAP_total_min)
Q_AUT65MAP_total_min<-summary(FinalHemodynamicsDataset$AUT65MAP_total_min)
Q_AUT80SBP_total_min<-summary(FinalHemodynamicsDataset$AUT80SBP_total_min)
Q_AOT80SBP_total_min<-summary(FinalHemodynamicsDataset$AOT80SBP_total_min)

```

```

Q_AUT50MAP_total_ratio<-summary(FinalHemodynamicsDataset$AUT50MAP_total_ratio)
Q_AUT65MAP_total_ratio<-summary(FinalHemodynamicsDataset$AUT65MAP_total_ratio)
Q_AUC65MAP_total_mmHgmin<-summary(FinalHemodynamicsDataset$AUC65MAP_total_mmHgmin)
Q_AOC65MAP_total_mmHgmin<-summary(FinalHemodynamicsDataset$AOC65MAP_total_mmHgmin)
Q_TWA65MAP_total_mmHg<-summary(FinalHemodynamicsDataset$TWA65MAP_total_mmHg)
```

Comparing range to clasify each case's column in a Q

```{r}
#nested if else function

i <- function(if_stat, then) {
  if_stat <- lazyeval::expr_text(if_stat)
  then <- lazyeval::expr_text(then)
  sprintf("ifelse(%s, %s, ", if_stat, then)
}

e <- function(else_ret) {
  else_ret <- lazyeval::expr_text(else_ret)
  else_ret
}

ie <- function(...) {
  args <- list(...)

  for (i in 1:(length(args) - 1) ) {
    if (substr(args[[i]], 1, 6) != "ifelse") {
      stop("All but the last argument, need to be i functions.", call. = FALSE)
    }
  }
  if (substr(args[[length(args)]], 1, 6) == "ifelse"){
    stop("Last argument needs to be an e function.", call. = FALSE)
  }
  args$final <- paste(rep(')', length(args) - 1), collapse = '')
  eval_string <- do.call('paste', args)
  eval(parse(text = eval_string))
}

FinalHemodynamicsDataset['Q_AUC65MAP_total_mmHgmin']<-NA

FinalHemodynamicsDataset$Q_AUC65MAP_total_mmHgmin <-
  ie(
    i(is.na(FinalHemodynamicsDataset$AUC65MAP_total_mmHgmin),NA),
    i(FinalHemodynamicsDataset$AUC65MAP_total_mmHgmin <=
Q_AUC65MAP_total_mmHgmin[2],1),
    i(FinalHemodynamicsDataset$AUC65MAP_total_mmHgmin > Q_AUC65MAP_total_mmHgmin[2] &
FinalHemodynamicsDataset$AUC65MAP_total_mmHgmin <= Q_AUC65MAP_total_mmHgmin[3],2),
    i(FinalHemodynamicsDataset$AUC65MAP_total_mmHgmin > Q_AUC65MAP_total_mmHgmin[3]
&FinalHemodynamicsDataset$AUC65MAP_total_mmHgmin <= Q_AUC65MAP_total_mmHgmin[5],3),
    i(FinalHemodynamicsDataset$AUC65MAP_total_mmHgmin >=
Q_AUC65MAP_total_mmHgmin[5],4),
    e(NA)
  )

```

```

FinalHemodynamicsDataset['Q_AOC65MAP_total_mmHgmin']<-NA

FinalHemodynamicsDataset$Q_AOC65MAP_total_mmHgmin <-
  ie(
    i(is.na(FinalHemodynamicsDataset$AOC65MAP_total_mmHgmin),NA),
    i(FinalHemodynamicsDataset$AOC65MAP_total_mmHgmin <=
Q_AOC65MAP_total_mmHgmin[2],1),
    i(FinalHemodynamicsDataset$AOC65MAP_total_mmHgmin > Q_AOC65MAP_total_mmHgmin[2] &
FinalHemodynamicsDataset$AOC65MAP_total_mmHgmin <= Q_AOC65MAP_total_mmHgmin[3],2),
    i(FinalHemodynamicsDataset$AOC65MAP_total_mmHgmin > Q_AOC65MAP_total_mmHgmin[3]
&FinalHemodynamicsDataset$AOC65MAP_total_mmHgmin <= Q_AOC65MAP_total_mmHgmin[5],3),
    i(FinalHemodynamicsDataset$AOC65MAP_total_mmHgmin >=
Q_AOC65MAP_total_mmHgmin[5],4),
    e(NA)
  )

FinalHemodynamicsDataset['Q_TWA65MAP_total_mmHg']<-NA

FinalHemodynamicsDataset$Q_TWA65MAP_total_mmHg <-
  ie(
    i(is.na(FinalHemodynamicsDataset$TWA65MAP_total_mmHg),NA),
    i(FinalHemodynamicsDataset$TWA65MAP_total_mmHg <= Q_TWA65MAP_total_mmHg[2],1),
    i(FinalHemodynamicsDataset$TWA65MAP_total_mmHg > Q_TWA65MAP_total_mmHg[2] &
FinalHemodynamicsDataset$TWA65MAP_total_mmHg <= Q_TWA65MAP_total_mmHg[3],2),
    i(FinalHemodynamicsDataset$TWA65MAP_total_mmHg > Q_TWA65MAP_total_mmHg[3]
&FinalHemodynamicsDataset$TWA65MAP_total_mmHg <= Q_TWA65MAP_total_mmHg[5],3),
    i(FinalHemodynamicsDataset$TWA65MAP_total_mmHg >= Q_TWA65MAP_total_mmHg[5],4),
    e(NA)
  )

FinalHemodynamicsDataset['Q_total_pressors_formula_mg']<-NA

FinalHemodynamicsDataset$Q_total_pressors_formula_mg <-
  ie(
    i(is.na(FinalHemodynamicsDataset$total_pressors_formula_mg),NA),
    i(FinalHemodynamicsDataset$total_pressors_formula_mg <=
Q_total_pressors_formula_mg[2],1),
    i(FinalHemodynamicsDataset$total_pressors_formula_mg >
Q_total_pressors_formula_mg[2] & FinalHemodynamicsDataset$total_pressors_formula_mg
<= Q_total_pressors_formula_mg[3],2),
    i(FinalHemodynamicsDataset$total_pressors_formula_mg >
Q_total_pressors_formula_mg[3] &FinalHemodynamicsDataset$total_pressors_formula_mg <=
Q_total_pressors_formula_mg[5],3),
    i(FinalHemodynamicsDataset$total_pressors_formula_mg >=
Q_total_pressors_formula_mg[5],4),
    e(NA)
  )

FinalHemodynamicsDataset['Q_AUT50MAP_total_min']<-NA

FinalHemodynamicsDataset$Q_AUT50MAP_total_min <-
  ie(
    i(is.na(FinalHemodynamicsDataset$AUT50MAP_total_min),NA),
    i(FinalHemodynamicsDataset$AUT50MAP_total_min <= Q_AUT50MAP_total_min[2],1),

```

```

        i(FinalHemodynamicsDataset$AUT50MAP_total_min > Q_AUT50MAP_total_min[2] &
FinalHemodynamicsDataset$AUT50MAP_total_min <= Q_AUT50MAP_total_min[3],2),
        i(FinalHemodynamicsDataset$AUT50MAP_total_min> Q_AUT50MAP_total_min[3]
&FinalHemodynamicsDataset$AUT50MAP_total_min <= Q_AUT50MAP_total_min[5],3),
        i(FinalHemodynamicsDataset$AUT50MAP_total_min >= Q_AUT50MAP_total_min[5],4),
        e(NA)
    )

FinalHemodynamicsDataset['Q_AUT50MAP_total_ratio']<-NA

FinalHemodynamicsDataset$Q_AUT50MAP_total_ratio <-
    ie(
        i(is.na(FinalHemodynamicsDataset$AUT50MAP_total_ratio),NA),
        i(FinalHemodynamicsDataset$AUT50MAP_total_ratio <= Q_AUT50MAP_total_ratio[2],1),
        i(FinalHemodynamicsDataset$AUT50MAP_total_ratio > Q_AUT50MAP_total_ratio[2] &
FinalHemodynamicsDataset$AUT50MAP_total_ratio <= Q_AUT50MAP_total_ratio[3],2),
        i(FinalHemodynamicsDataset$AUT50MAP_total_ratio> Q_AUT50MAP_total_ratio[3]
&FinalHemodynamicsDataset$AUT50MAP_total_ratio <= Q_AUT50MAP_total_ratio[5],3),
        i(FinalHemodynamicsDataset$AUT50MAP_total_ratio >= Q_AUT50MAP_total_ratio[5],4),
        e(NA)
    )

FinalHemodynamicsDataset['Q_AWI51_60MAP_total_min']<-NA

FinalHemodynamicsDataset$Q_AWI51_60MAP_total_min <-
    ie(
        i(is.na(FinalHemodynamicsDataset$AWI51_60MAP_total_min),NA),
        i(FinalHemodynamicsDataset$AWI51_60MAP_total_min <=
Q_AWI51_60MAP_total_min[2],1),
        i(FinalHemodynamicsDataset$AWI51_60MAP_total_min > Q_AWI51_60MAP_total_min[2] &
FinalHemodynamicsDataset$AWI51_60MAP_total_min <= Q_AWI51_60MAP_total_min[3],2),
        i(FinalHemodynamicsDataset$AWI51_60MAP_total_min> Q_AWI51_60MAP_total_min[3]
&FinalHemodynamicsDataset$AWI51_60MAP_total_min <= Q_AWI51_60MAP_total_min[5],3),
        i(FinalHemodynamicsDataset$AWI51_60MAP_total_min >=
Q_AWI51_60MAP_total_min[5],4),
        e(NA)
    )

FinalHemodynamicsDataset['Q_AWI61_65MAP_total_min']<-NA

FinalHemodynamicsDataset$Q_AWI61_65MAP_total_min <-
    ie(
        i(is.na(FinalHemodynamicsDataset$AWI61_65MAP_total_min),NA),
        i(FinalHemodynamicsDataset$AWI61_65MAP_total_min <=
Q_AWI61_65MAP_total_min[2],1),
        i(FinalHemodynamicsDataset$AWI61_65MAP_total_min > Q_AWI61_65MAP_total_min[2] &
FinalHemodynamicsDataset$AWI61_65MAP_total_min <= Q_AWI61_65MAP_total_min[3],2),
        i(FinalHemodynamicsDataset$AWI61_65MAP_total_min> Q_AWI61_65MAP_total_min[3]
&FinalHemodynamicsDataset$AWI61_65MAP_total_min <= Q_AWI61_65MAP_total_min[5],3),
        i(FinalHemodynamicsDataset$AWI61_65MAP_total_min >=
Q_AWI61_65MAP_total_min[5],4),
        e(NA)
    )

FinalHemodynamicsDataset['Q_AOT65MAP_total_min']<-NA

```

```

FinalHemodynamicsDataset$Q_AOT65MAP_total_min <-
  ie(
    i(is.na(FinalHemodynamicsDataset$AOT65MAP_total_min),NA),
    i(FinalHemodynamicsDataset$AOT65MAP_total_min <= Q_AOT65MAP_total_min[2],1),
    i(FinalHemodynamicsDataset$AOT65MAP_total_min > Q_AOT65MAP_total_min[2] &
FinalHemodynamicsDataset$AOT65MAP_total_min <= Q_AOT65MAP_total_min[3],2),
    i(FinalHemodynamicsDataset$AOT65MAP_total_min > Q_AOT65MAP_total_min[3]
&FinalHemodynamicsDataset$AOT65MAP_total_min <= Q_AOT65MAP_total_min[5],3),
    i(FinalHemodynamicsDataset$AOT65MAP_total_min >= Q_AOT65MAP_total_min[5],4),
    e(NA)
  )

FinalHemodynamicsDataset['Q_AUT65MAP_total_min']<-NA

FinalHemodynamicsDataset$Q_AUT65MAP_total_min <-
  ie(
    i(is.na(FinalHemodynamicsDataset$AUT65MAP_total_min),NA),
    i(FinalHemodynamicsDataset$AUT65MAP_total_min <= Q_AUT65MAP_total_min[2],1),
    i(FinalHemodynamicsDataset$AUT65MAP_total_min > Q_AUT65MAP_total_min[2] &
FinalHemodynamicsDataset$AUT65MAP_total_min <= Q_AUT65MAP_total_min[3],2),
    i(FinalHemodynamicsDataset$AUT65MAP_total_min > Q_AUT65MAP_total_min[3]
&FinalHemodynamicsDataset$AUT65MAP_total_min <= Q_AUT65MAP_total_min[5],3),
    i(FinalHemodynamicsDataset$AUT65MAP_total_min >= Q_AUT65MAP_total_min[5],4),
    e(NA)
  )

FinalHemodynamicsDataset['Q_AUT65MAP_total_ratio']<-NA

FinalHemodynamicsDataset$Q_AUT65MAP_total_ratio <-
  ie(
    i(is.na(FinalHemodynamicsDataset$AUT65MAP_total_ratio),NA),
    i(FinalHemodynamicsDataset$AUT65MAP_total_ratio <= Q_AUT65MAP_total_ratio[2],1),
    i(FinalHemodynamicsDataset$AUT65MAP_total_ratio > Q_AUT65MAP_total_ratio[2] &
FinalHemodynamicsDataset$AUT65MAP_total_ratio <= Q_AUT65MAP_total_ratio[3],2),
    i(FinalHemodynamicsDataset$AUT65MAP_total_ratio > Q_AUT65MAP_total_ratio[3]
&FinalHemodynamicsDataset$AUT65MAP_total_ratio <= Q_AUT65MAP_total_ratio[5],3),
    i(FinalHemodynamicsDataset$AUT65MAP_total_ratio >= Q_AUT65MAP_total_ratio[5],4),
    e(NA)
  )

FinalHemodynamicsDataset['Q_AUT80SBP_total_min']<-NA

FinalHemodynamicsDataset$Q_AUT80SBP_total_min <-
  ie(
    i(is.na(FinalHemodynamicsDataset$AUT80SBP_total_min),NA),
    i(FinalHemodynamicsDataset$AUT80SBP_total_min <= Q_AUT80SBP_total_min[2],1),
    i(FinalHemodynamicsDataset$AUT80SBP_total_min > Q_AUT80SBP_total_min[2] &
FinalHemodynamicsDataset$AUT80SBP_total_min <= Q_AUT80SBP_total_min[3],2),
    i(FinalHemodynamicsDataset$AUT80SBP_total_min > Q_AUT80SBP_total_min[3]
&FinalHemodynamicsDataset$AUT80SBP_total_min <= Q_AUT80SBP_total_min[5],3),
    i(FinalHemodynamicsDataset$AUT80SBP_total_min >= Q_AUT80SBP_total_min[5],4),
    e(NA)
  )

```

```

FinalHemodynamicsDataset['Q_AOT80SBP_total_min']<-NA

FinalHemodynamicsDataset$Q_AOT80SBP_total_min <-
  ie(
    i(is.na(FinalHemodynamicsDataset$AOT80SBP_total_min),NA),
    i(FinalHemodynamicsDataset$AOT80SBP_total_min <= Q_AOT80SBP_total_min[2],1),
    i(FinalHemodynamicsDataset$AOT80SBP_total_min > Q_AOT80SBP_total_min[2] &
FinalHemodynamicsDataset$AOT80SBP_total_min <= Q_AOT80SBP_total_min[3],2),
    i(FinalHemodynamicsDataset$AOT80SBP_total_min > Q_AOT80SBP_total_min[3]
&FinalHemodynamicsDataset$AOT80SBP_total_min <= Q_AOT80SBP_total_min[5],3),
    i(FinalHemodynamicsDataset$AOT80SBP_total_min >= Q_AOT80SBP_total_min[5],4),
    e(NA)
  )

...

# Tertiles
```{r}
FinalHemodynamicsDataset['T_AUT50MAP_total_min']<-
as.numeric(cut2(FinalHemodynamicsDataset$AUT50MAP_total_min, g=3))
FinalHemodynamicsDataset['T_AUT65MAP_total_min']<-
as.numeric(cut2(FinalHemodynamicsDataset$AUT65MAP_total_min, g=3))
FinalHemodynamicsDataset['T_AOT65MAP_total_min']<-
as.numeric(cut2(FinalHemodynamicsDataset$AOT65MAP_total_min, g=3))
FinalHemodynamicsDataset['T_total_pressors_formula_mg']<-
as.numeric(cut2(FinalHemodynamicsDataset$total_pressors_formula_mg, g=3))
FinalHemodynamicsDataset['T_STS_RiskAlgorithm_median_imp']<-
as.factor(as.numeric(cut2(FinalHemodynamicsDataset$STS_RiskAlgorithm_median_imp,
g=3)))

...

# Creating Deciles Columns
```{r}
library(StatMeasures)
#decileScores <-
FinalHemodynamicsDataset['D_AUT50MAP_total_min']<-
decile(FinalHemodynamicsDataset$AUT50MAP_total_min)
FinalHemodynamicsDataset['D_AWI51_60MAP_total_min']<-
decile(FinalHemodynamicsDataset$AWI51_60MAP_total_min)
FinalHemodynamicsDataset['D_AWI61_65MAP_total_min']<-
decile(FinalHemodynamicsDataset$AWI61_65MAP_total_min)
FinalHemodynamicsDataset['D_AOT65MAP_total_min']<-
decile(FinalHemodynamicsDataset$AOT65MAP_total_min)
FinalHemodynamicsDataset['D_total_pressors_formula_mg']<-
decile(FinalHemodynamicsDataset$total_pressors_formula_mg)

...

# Export dataset

```

```

```{r eval=FALSE, include=FALSE}
write.csv(FinalHemodynamicsDataset, 'FinalHemodynamicsDataset.csv')
```

Exploring DataSet

```{r eval=FALSE, include=FALSE}
library(summarytools)
view(dfSummary(FinalHemodynamicsDataset))
```

Cut off points

Pressors

```{r}
library(dplyr)

FinalHemodynamicsDataset %>%
  dplyr::group_by(Q_total_pressors_formula_mg)%>%
  dplyr::summarize(min = round(min(total_pressors_formula_mg),2),
                  max = round(max(total_pressors_formula_mg),2)
                  )
```

Blood Pressure

```{r}
library(dplyr)

FinalHemodynamicsDataset %>%
  dplyr::group_by(Q_AUC65MAP_total_mmHgmin)%>%
  dplyr::summarize(min = round(min(AUC65MAP_total_mmHgmin), 2),
                  max = round(max(AUC65MAP_total_mmHgmin), 2)
                  )
```

```{r}
library(dplyr)

FinalHemodynamicsDataset %>%
  dplyr::group_by(Q_AOC65MAP_total_mmHgmin)%>%
  dplyr::summarize(min = round(min(AOC65MAP_total_mmHgmin), 2),
                  max = round(max(AOC65MAP_total_mmHgmin), 2)
                  )
```

```{r}
library(dplyr)

```



```

FinalHemodynamicsDataset %>%
  dplyr::group_by(Q_TWA65MAP_total_mmHg)%>%
  dplyr::summarize(min = round(min(TWA65MAP_total_mmHg), 2),
    max = round(max(TWA65MAP_total_mmHg), 2)
  )
...

## STS Risk Score

```{r}
library(dplyr)

FinalHemodynamicsDataset %>%
  dplyr::group_by(T_STS_RiskAlgorithm_median_imp)%>%
  dplyr::summarize(min = round(min(STS_RiskAlgorithm_median_imp), 2),
    max = round(max(STS_RiskAlgorithm_median_imp), 2)
  )
...

# Input Output Data

```{r}
FinalHemodynamicsDataset<-left_join(FinalHemodynamicsDataset, IO_data)
...

# Ejection Fraction Data

```{r}
FinalHemodynamicsDataset<-left_join(FinalHemodynamicsDataset,Ejection_Fraction)
...

# Data Summary

```{r}
library(summarytools)
# select only desired columns to be reviewed
#view(dfSummary(FinalHemodynamicsDataset[,c('AUC65MAP_total_mmHgmin','TWA65MAP_total_
mmHg')]))
write.csv(FinalHemodynamicsDataset,'FinalHemodynamicsDataset.csv')
...

---
title: "Data Imputation"
author: "Miguel Angel Armengol de la Hoz"
date: "10 de agosto de 2017"
output:
  html_document:
    toc: true
    theme: united
---

# Libraries load

```{r setup, include=FALSE}

```

```

library(dplyr)
library(reshape2)
library(summarytools)
library(readxl)
library(reshape)
```

Imputing missing data

STS_risk

Mean and median imputation

We chose median imputation after having run the model with the best AUROC among all imputations.

```{r}

FinalHemodynamicsDataset['STS_RiskAlgorithm_mean_imp']<-if_else(
  is.na(FinalHemodynamicsDataset$STS_RiskAlgorithm)
  ,mean(FinalHemodynamicsDataset$STS_RiskAlgorithm,na.rm = T)
  ,FinalHemodynamicsDataset$STS_RiskAlgorithm
)

FinalHemodynamicsDataset['STS_RiskAlgorithm_median_imp']<-ifelse(
  is.na(FinalHemodynamicsDataset$STS_RiskAlgorithm)
  ,median(FinalHemodynamicsDataset$STS_RiskAlgorithm,na.rm = T)
  ,FinalHemodynamicsDataset$STS_RiskAlgorithm)
```

Multiple imputation

```{r eval=FALSE, include=FALSE}
#min and max taken from STS risk scores documentation
bds <- matrix(c(7, 0, 100), nrow = 1, ncol = 3)# we need a matrix of bounds(number of
column to bound, min, max)

relevant_data_imputation_matrix<-amelia(FinalHemodynamicsDataset, parallel =
"multicore",idvars = 'Case_Name',noms =
c('Gender','Category','Q_AUT50MAP_total_min','Q_AUT65MAP_total_min','Q_AOT65MAP_total
_min','Q_total_pressors_formula_mg')
      #, ords = #Ords is only for ingegers, not floats
      ,bounds = bds
    )

FinalHemodynamicsDataset<-FinalHemodynamicsDataset_imputation_matrix$imputations$imp5

names(FinalHemodynamicsDataset)[names(FinalHemodynamicsDataset)=="STS_RiskAlgorithm"]
<-"STS_RiskAlgorithm_multiple_imp"

summary(FinalHemodynamicsDataset$STS_RiskAlgorithm_multiple_imp)

#STS risk cannot be negative

```

```

# to avoid problems when using log
#there can not be 0 values in a log(variable) in a logreg
```

Mean EF

```{r}
FinalHemodynamicsDataset['mean_ef_median_imp']<-if_else(
is.na(FinalHemodynamicsDataset$mean_ef)
,median(FinalHemodynamicsDataset$mean_ef,na.rm = T)
,FinalHemodynamicsDataset$mean_ef
)
```

Delta HTC

```{r}
FinalHemodynamicsDataset['delta_HCT_pct_median_imp']<-if_else(
is.na(FinalHemodynamicsDataset$delta_HCT_pct)
,median(FinalHemodynamicsDataset$delta_HCT_pct,na.rm = T)
,FinalHemodynamicsDataset$delta_HCT_pct
)
```

Cross Clamp Time

```{r}
FinalHemodynamicsDataset['Cross_Clamp_Time_imp']<-if_else(
is.na(FinalHemodynamicsDataset$Cross_Clamp_Time)
,median(FinalHemodynamicsDataset$Cross_Clamp_Time,na.rm = T)
,as.double(FinalHemodynamicsDataset$Cross_Clamp_Time)
)
```

```

## Modeling

```

---
title: "Models creation"
author: "Miguel Ángel Armengol de la Hoz"
output:
  html_document:
    toc: true
    theme: united

Libraries

```{r}
library(readr)

```

```

library(dplyr)
library(Amelia)
library(ggplot2)
library(xlsx)
library(Hmisc)
library(ResourceSelection)
library(rms)
#sudo ln -f -s $(/usr/libexec/java_home)/jre/lib/server/libjvm.dylib /usr/local/lib
library(rJava)
library(Deducer)
require(MASS)
```

Creating dataset to fit logistic regression model

```{r}
#detach('package::dplyr')

relevant_data<-FinalHemodynamicsDataset%>%
dplyr::select( Case_Name
               ,Any_MAE
               ,Gender
               ,Age
               ,Category
               ,duration_Surgery_min
               ,T_STS_RiskAlgorithm_median_imp
               ,total_pressors_formula_mg
               ,AUT50MAP_total_min
               ,Q_AUT50MAP_total_min
               ,Q_total_pressors_formula_mg
               ,Q_AUT65MAP_total_min
               ,Q_AOT65MAP_total_min
               ,AUT65MAP_total_min
               ,AOT65MAP_total_min
               ,delta_HCT_pct_median_imp
               ,mean_ef_median_imp
               ,AUC65MAP_total_mmHgmin
               ,TWA65MAP_total_mmHg
               ,Q_AUC65MAP_total_mmHgmin
               ,Q_TWA65MAP_total_mmHg
               ,Q_AOC65MAP_total_mmHgmin
               ,Cross_Clamp_Time_imp
               ,duration_CPB_min
               ,AUC65MAP_CPB_total_mmHgmin
               )
```

Factorizing Variables

```{r}
# Other variables to be factorized
relevant_data$Gender<-as.factor(relevant_data$Gender)
relevant_data$Category<-as.factor(relevant_data$Category)
relevant_data$Any_MAE<-as.factor(relevant_data$Any_MAE)

```

```

# Factorizing Quarrrtiles
relevant_data$Q_AUT50MAP_total_min<-as.factor(relevant_data$Q_AUT50MAP_total_min)
relevant_data$Q_AUT65MAP_total_min<-as.factor(relevant_data$Q_AUT65MAP_total_min)
relevant_data$Q_AUT65MAP_total_min<-as.factor(relevant_data$Q_AUT65MAP_total_min)
relevant_data$Q_AOT65MAP_total_min<-as.factor(relevant_data$Q_AOT65MAP_total_min)
relevant_data$Q_total_pressors_formula_mg<-
as.factor(relevant_data$Q_total_pressors_formula_mg)
relevant_data$Q_AUC65MAP_total_mmHgmin<-
as.factor(relevant_data$Q_AUC65MAP_total_mmHgmin)
relevant_data$Q_AOC65MAP_total_mmHgmin<-
as.factor(relevant_data$Q_AOC65MAP_total_mmHgmin)
relevant_data$Q_TWA65MAP_total_mmHg<-as.factor(relevant_data$Q_TWA65MAP_total_mmHg)

# Factorizing Tertiles
relevant_data$T_STS_RiskAlgorithm_median_imp<-
as.factor(relevant_data$T_STS_RiskAlgorithm_median_imp)

#relevant_data$Q_AUT80SBP_total_min<-as.factor(relevant_data$Q_AUT80SBP_total_min)

# Factorizing tertiles
# relevant_data$T_AUT50MAP_total_min<-as.factor(relevant_data$T_AUT50MAP_total_min)
# relevant_data$T_AUT65MAP_total_min<-as.factor(relevant_data$T_AUT65MAP_total_min)
# relevant_data$T_AOT65MAP_total_min<-as.factor(relevant_data$T_AOT65MAP_total_min)
# relevant_data$T_total_pressors_formula_mg<-
# as.factor(relevant_data$T_total_pressors_formula_mg)

#relevant_data$Anesthesiologist<-as.factor(relevant_data$Anesthesiologist)

# creating T_STS_RiskAlgorithm_median_imp variable

...

# Multivariate log regression Tertiles of Pressors and MAP adjusted with offset and
OR

```{r}
# Selecting reference levels for the categorical variables
relevant_data$Gender <- relevel(relevant_data$Gender, ref = "Male")
relevant_data$Category <- relevel(relevant_data$Category, ref = "CABG")

mod_multiv<-glm(Any_MAE ~
  +Q_total_pressors_formula_mg
  +Q_AUC65MAP_total_mmHgmin
  +T_STS_RiskAlgorithm_median_imp
  +Gender
  +Age
  +Category
  +delta_HCT_pct_median_imp
  +mean_ef_median_imp
  #uncomment only for studying interaction (don't factorize variable Q
when studying interaction)
  +Q_AOC65MAP_total_mmHgmin*Q_total_pressors_formula_mg

```

```

        #+offset(log(duration_Surgery_min))
        +Cross_Clamp_Time_imp
        #+as.factor(as.numeric(cut2(Cross_Clamp_Time, g=4)))
        #+duration_CPB_min
        ,family='binomial'
        ,data = relevant_data
    )

mod_multiv

oddsratios_table<-exp(cbind(coef(mod_multiv), confint(mod_multiv)))
options(scipen=999)
oddsratios_table<-round(oddsratios_table,4)
#oddsratios_table['x']<-(1:10)

oddsratios_table<-as.data.frame(oddsratios_table)
colnames(oddsratios_table)[1]<-"OR"

oddsratios_table<-oddsratios_table[order(row.names(oddsratios_table)), ]
boxLabels<-rownames(oddsratios_table)
#oddsratios_table['yAxis']<- length(boxLabels):1

write.xlsx(as.data.frame(print(oddsratios_table)),
"/Users/marmengol/MEGA/Boston/BIDMC-Harvard/Phenotyping/Tables and
Figures/oddsratios_table.xlsx")
...

## Evaluating the model

### AUC

```{r}
ROC<-rocplot(mod_multiv)
cat('AUC: ')
ROC$plot_env$auc
```

Hosmer-Lemeshow goodness of fit test

If the p-value is too small, we don't have to include it, given the conservative
property of the test. If it's larger than 0.05, we include it in the manuscript to
make it more complete.

```{r}
hoslem.test(mod_multiv$y, fitted(mod_multiv),g=10)
```

Exploratory analysis

New model for STS risk ratio tertile 3

```{r}

```

```

# Selecting reference levels for the categorical variables
relevant_data$Gender <- relevel(relevant_data$Gender, ref = "Male")
relevant_data$Category <- relevel(relevant_data$Category, ref = "CABG")

relevant_data_STSrisk3<-subset(relevant_data, T_STS_RiskAlgorithm_median_imp==3)

mod_multiv<-glm(Any_MAE ~
  +Q_total_pressors_formula_mg
  +Q_AUC65MAP_total_mmHgmin
  +Gender
  +Age
  +Category
  +delta_HCT_pct_median_imp
  +mean_ef_median_imp
  #uncomment only for studying interaction (don't factorize variable Q
when studying interaction)
  #+Q_AOC65MAP_total_mmHgmin*Q_total_pressors_formula_mg
  #+offset(log(duration_Surgery_min))
  +Cross_Clamp_Time_imp
  #+as.factor(as.numeric(cut2(Cross_Clamp_Time, g=4)))
  #+duration_CPB_min
  ,family='binomial'
  ,data = relevant_data_STSrisk3
)

oddsratios_STST3_table<-exp(cbind(coef(mod_multiv), confint(mod_multiv)))
options(scipen=999)
oddsratios_STST3_table<-round(oddsratios_STST3_table,4)
#oddsratios_STST3_table['x']<-(1:10)

oddsratios_STST3_table<-as.data.frame(oddsratios_STST3_table)
colnames(oddsratios_STST3_table)[1]<-"OR"

oddsratios_STST3_table<-
oddsratios_STST3_table[order(row.names(oddsratios_STST3_table)), ]
boxLabels<-rownames(oddsratios_STST3_table)

write.xlsx(as.data.frame(print(oddsratios_STST3_table)),
"/Users/marmengol/MEGA/Boston/BIDMC-Harvard/Phenotyping/Tables and
Figures/oddsratios_STST3_table.xlsx")
...

# Studying the DataSet

``{r eval=FALSE, include=FALSE}
library(summarytools)
view(dfSummary(relevant_data))

test<-FinalHemodynamicsDataset %>%
  filter(AUT65MAP_total_ratio==0 & Any_MAE==1)
library(dplyr)

```

```

check<-inner_join(FinalHemodynamicsDataset,listofKeys)%>%
  filter(duration_Surgery_min>60) %>%
  dplyr::select(Case_Name,DOS,MRN,Sur_Begin_Time,Sur_End_Time,CPB_Begin_Time,
CPB_End_Time, duration_preCPB_min, duration_CPB_min, duration_Surgery_min)

write.csv(check[1:10,],'highsurgerylength2.csv')

key1<-as.vector(unlist(unname(sapply(TextList, `[`, 1))))
key2<-as.vector(unlist(unname(sapply(TextList, `[`, 2))))
key3<-as.vector(unlist(unname(sapply(TextList, `[`, 5))))
key3<-as.numeric(test3)

listofKeys<-cbind.data.frame(key1,key2,key3)
listofKeys<-listofKeys[complete.cases(listofKeys),]
names(listofKeys)<-c('Case_Name','DOS','MRN')

df<-as.data.frame(key3,key1)
df['test']<-NA
df<-rename(df, eee = key3)
colnames(df)["key3"] = 'eee'

colnames(df)["key3"]<- 'eee'

names(df)['key3']

colnames(df)["name"]
colnames(df)['name']
```

```

## Figures

```

---
title: "Quantiles"
author: "Miguel Ángel Armengol de la Hoz"
output:
  html_document:
    toc: true
    theme: united

Libs

```{r}
library(dplyr)
library(plotly)
#library(corrplot)
library(Hmisc)
#library(officer)
library(xlsx)
```

```



```

```{r setup, include=FALSE}
#load the data
#load("C:/BIG FILES/Phenotyping/HemodynamicProjectsDatasetFeb18.RData")
```

Plots

Wrangling the dataset for plotting

```{r eval=FALSE, include=FALSE}
###IMPORTANT detach plyr!!
#detach('package:plyr')

#Tertiles

T_AUT50MAP_total_min_pressors_MAE<-FinalHemodynamicsDataset %>%
dplyr::group_by(T_AUT50MAP_total_min, T_total_pressors_formula_mg)%>%
dplyr::summarise(total_patientswith_MAE=sum(Any_MAE)/518)
T_AUT50MAP_total_min_pressors_MAE<-
T_AUT50MAP_total_min_pressors_MAE[complete.cases(T_AUT50MAP_total_min_pressors_MAE),
]
library(reshape)
T_AUT50MAP_total_min_pressors_MAE_matrix<-cast(data=T_AUT50MAP_total_min_pressors_MAE
, T_AUT50MAP_total_min ~ T_total_pressors_formula_mg
,value = 'total_patientswith_MAE')
T_AUT50MAP_total_min_pressors_MAE_matrix<-
as.matrix(T_AUT50MAP_total_min_pressors_MAE_matrix)

T_AUT65MAP_total_min_pressors_MAE<-FinalHemodynamicsDataset %>%
dplyr::group_by(T_AUT65MAP_total_min, T_total_pressors_formula_mg) %>%
dplyr::summarise(total_patientswith_MAE=sum(Any_MAE)/518)
T_AUT65MAP_total_min_pressors_MAE<-
T_AUT65MAP_total_min_pressors_MAE[complete.cases(T_AUT65MAP_total_min_pressors_MAE),
]
library(reshape)
T_AUT65MAP_total_min_pressors_MAE_matrix<-cast(data=T_AUT65MAP_total_min_pressors_MAE
, T_AUT65MAP_total_min ~ T_total_pressors_formula_mg
,value = 'total_patientswith_MAE')
T_AUT65MAP_total_min_pressors_MAE_matrix<-
as.matrix(T_AUT65MAP_total_min_pressors_MAE_matrix)

T_AOT65MAP_total_min_pressors_MAE<-(FinalHemodynamicsDataset) %>%
dplyr::group_by(T_AOT65MAP_total_min, T_total_pressors_formula_mg) %>%
dplyr::summarise(total_patientswith_MAE=sum(Any_MAE)/518)

```

```

T_AOT65MAP_total_min_pressors_MAE<-
T_AOT65MAP_total_min_pressors_MAE[complete.cases(T_AOT65MAP_total_min_pressors_MAE),
]

#Quartiles

Q_AUT50MAP_total_min_pressors_MAE<-FinalHemodynamicsDataset %>%
dplyr::group_by(Q_AUT50MAP_total_min, Q_total_pressors_formula_mg)%>%
dplyr::summarise(total_patientswith_MAE=sum(Any_MAE)/518*100)
Q_AUT50MAP_total_min_pressors_MAE<-
Q_AUT50MAP_total_min_pressors_MAE[complete.cases(Q_AUT50MAP_total_min_pressors_MAE),
]
library(reshape)
Q_AUT50MAP_total_min_pressors_MAE_matrix<-cast(data=Q_AUT50MAP_total_min_pressors_MAE
, Q_AUT50MAP_total_min ~ Q_total_pressors_formula_mg
,value = 'total_patientswith_MAE')
Q_AUT50MAP_total_min_pressors_MAE_matrix<-
as.data.frame(Q_AUT50MAP_total_min_pressors_MAE_matrix)
colnames(Q_AUT50MAP_total_min_pressors_MAE_matrix)<-c('','total pressors Q1','total
pressors Q2','total pressors Q3','total pressors Q4')
rownames(Q_AUT50MAP_total_min_pressors_MAE_matrix)<-c('total AUT50 MAP Q1','total
AUT50 MAP Q2','total AUT50 MAP Q3','total AUT50 MAP Q4')
Q_AUT50MAP_total_min_pressors_MAE_matrix[,1]<-NULL
Q_AUT50MAP_total_min_pressors_MAE_matrix<-
round(Q_AUT50MAP_total_min_pressors_MAE_matrix,2)

Q_AUT65MAP_total_min_pressors_MAE<-FinalHemodynamicsDataset%>%
dplyr::group_by(Q_AUT65MAP_total_min, Q_total_pressors_formula_mg) %>%
dplyr::summarise(total_patientswith_MAE=sum(Any_MAE)/518*100)
Q_AUT65MAP_total_min_pressors_MAE<-
Q_AUT65MAP_total_min_pressors_MAE[complete.cases(Q_AUT65MAP_total_min_pressors_MAE),
]
library(reshape)
Q_AUT65MAP_total_min_pressors_MAE_matrix<-cast(data=Q_AUT65MAP_total_min_pressors_MAE
, Q_AUT65MAP_total_min ~ Q_total_pressors_formula_mg
,value = 'total_patientswith_MAE')
Q_AUT65MAP_total_min_pressors_MAE_matrix<-
as.data.frame(Q_AUT65MAP_total_min_pressors_MAE_matrix)
colnames(Q_AUT65MAP_total_min_pressors_MAE_matrix)<-c('','total pressors Q1','total
pressors Q2','total pressors Q3','total pressors Q4')
rownames(Q_AUT65MAP_total_min_pressors_MAE_matrix)<-c('total AUT65 MAP Q1','total
AUT65 MAP Q2','total AUT65 MAP Q3','total AUT65 MAP Q4')
Q_AUT65MAP_total_min_pressors_MAE_matrix[,1]<-NULL
Q_AUT65MAP_total_min_pressors_MAE_matrix<-
round(Q_AUT65MAP_total_min_pressors_MAE_matrix,2)

Q_AOT65MAP_total_min_pressors_MAE<-(FinalHemodynamicsDataset) %>%
dplyr::group_by(Q_AOT65MAP_total_min, Q_total_pressors_formula_mg) %>%
dplyr::summarise(total_patientswith_MAE=sum(Any_MAE)/518*100)
Q_AOT65MAP_total_min_pressors_MAE<-
Q_AOT65MAP_total_min_pressors_MAE[complete.cases(Q_AOT65MAP_total_min_pressors_MAE),
]
library(reshape)
Q_AOT65MAP_total_min_pressors_MAE_matrix<-cast(data=Q_AOT65MAP_total_min_pressors_MAE

```

```

      ,Q_AOT65MAP_total_min ~ Q_total_pressors_formula_mg
      ,value = 'total_patientswith_MAE')
Q_AOT65MAP_total_min_pressors_MAE_matrix<-
as.data.frame(Q_AOT65MAP_total_min_pressors_MAE_matrix)
colnames(Q_AOT65MAP_total_min_pressors_MAE_matrix)<-c('','total pressors Q1','total
pressors Q2','total pressors Q3','total pressors Q4')
rownames(Q_AOT65MAP_total_min_pressors_MAE_matrix)<-c('total AOT65 MAP Q1','total
AOT65 MAP Q2','total AOT65 MAP Q3','total AOT65 MAP Q4')
Q_AOT65MAP_total_min_pressors_MAE_matrix[,1]<-NULL
Q_AOT65MAP_total_min_pressors_MAE_matrix<-
round(Q_AOT65MAP_total_min_pressors_MAE_matrix,2)

#Quartiles of ratio

Q_AUT50MAP_total_ratio_pressors_MAE<-FinalHemodynamicsDataset %>%
dplyr::group_by(Q_AUT50MAP_total_ratio, Q_total_pressors_formula_mg)%>%
dplyr::summarise(total_patientswith_MAE=sum(Any_MAE)/518*100)
Q_AUT50MAP_total_ratio_pressors_MAE<-
Q_AUT50MAP_total_ratio_pressors_MAE[complete.cases(Q_AUT50MAP_total_ratio_pressors_MA
E), ]
library(reshape)
Q_AUT50MAP_total_ratio_pressors_MAE_matrix<-
cast(data=Q_AUT50MAP_total_ratio_pressors_MAE
      ,Q_AUT50MAP_total_ratio ~ Q_total_pressors_formula_mg
      ,value = 'total_patientswith_MAE')
Q_AUT50MAP_total_ratio_pressors_MAE_matrix<-
as.data.frame(Q_AUT50MAP_total_ratio_pressors_MAE_matrix)
colnames(Q_AUT50MAP_total_ratio_pressors_MAE_matrix)<-c('','total pressors Q1','total
pressors Q2','total pressors Q3','total pressors Q4')
rownames(Q_AUT50MAP_total_ratio_pressors_MAE_matrix)<-c('ratio AUT50 MAP Q1','ratio
AUT50 MAP Q2','ratio AUT50 MAP Q3','ratio AUT50 MAP Q4')
Q_AUT50MAP_total_ratio_pressors_MAE_matrix[,1]<-NULL
Q_AUT50MAP_total_ratio_pressors_MAE_matrix<-
round(Q_AUT50MAP_total_ratio_pressors_MAE_matrix,2)

Q_AUT65MAP_total_ratio_pressors_MAE<-FinalHemodynamicsDataset%>%
dplyr::group_by(Q_AUT65MAP_total_ratio, Q_total_pressors_formula_mg) %>%
dplyr::summarise(total_patientswith_MAE=sum(Any_MAE)/518*100)
Q_AUT65MAP_total_ratio_pressors_MAE<-
Q_AUT65MAP_total_ratio_pressors_MAE[complete.cases(Q_AUT65MAP_total_ratio_pressors_MA
E), ]
library(reshape)
Q_AUT65MAP_total_ratio_pressors_MAE_matrix<-
cast(data=Q_AUT65MAP_total_ratio_pressors_MAE
      ,Q_AUT65MAP_total_ratio ~ Q_total_pressors_formula_mg
      ,value = 'total_patientswith_MAE')
Q_AUT65MAP_total_ratio_pressors_MAE_matrix<-
as.data.frame(Q_AUT65MAP_total_ratio_pressors_MAE_matrix)
colnames(Q_AUT65MAP_total_ratio_pressors_MAE_matrix)<-c('','total pressors Q1','total
pressors Q2','total pressors Q3','total pressors Q4')
rownames(Q_AUT65MAP_total_ratio_pressors_MAE_matrix)<-c('ratio AUT65 MAP Q1','ratio
AUT65 MAP Q2','ratio AUT65 MAP Q3','ratio AUT65 MAP Q4')
Q_AUT65MAP_total_ratio_pressors_MAE_matrix[,1]<-NULL

```

```

Q_AUT65MAP_total_ratio_pressors_MAE_matrix<-
round(Q_AUT65MAP_total_ratio_pressors_MAE_matrix,2)

# AUC Quartiles

Q_AUC65MAP_total_mmHgmin_pressors_MAE<-FinalHemodynamicsDataset%>%
dplyr::group_by(Q_AUC65MAP_total_mmHgmin, Q_total_pressors_formula_mg) %>%
dplyr::summarise('MAE rate (%)'=sum(Any_MAE)/518*100)
Q_AUC65MAP_total_mmHgmin_pressors_MAE<-
Q_AUC65MAP_total_mmHgmin_pressors_MAE[complete.cases(Q_AUC65MAP_total_mmHgmin_pressor
s_MAE), ]
library(reshape)
Q_AUC65MAP_total_mmHgmin_pressors_MAE_matrix<-
cast(data=Q_AUC65MAP_total_mmHgmin_pressors_MAE
      ,Q_AUC65MAP_total_mmHgmin ~ Q_total_pressors_formula_mg
      ,value = 'total_patientswith_MAE')
Q_AUC65MAP_total_mmHgmin_pressors_MAE_matrix<-
as.data.frame(Q_AUC65MAP_total_mmHgmin_pressors_MAE_matrix)
colnames(Q_AUC65MAP_total_mmHgmin_pressors_MAE_matrix)<-c('', 'total pressors
Q1', 'total pressors Q2', 'total pressors Q3', 'total pressors Q4')
rownames(Q_AUC65MAP_total_mmHgmin_pressors_MAE_matrix)<-c('total AUC65 MAP
Q1', 'total AUC65 MAP Q2', 'total AUC65 MAP Q3', 'total AUC65 MAP Q4')
Q_AUC65MAP_total_mmHgmin_pressors_MAE_matrix[,1]<-NULL
Q_AUC65MAP_total_mmHgmin_pressors_MAE_matrix<-
round(Q_AUC65MAP_total_mmHgmin_pressors_MAE_matrix,2)

# TWA Quartiles

Q_TWA65MAP_total_mmHg_pressors_MAE<-FinalHemodynamicsDataset%>%
dplyr::group_by(Q_TWA65MAP_total_mmHg, Q_total_pressors_formula_mg) %>%
dplyr::summarise(total_patientswith_MAE=sum(Any_MAE)/518*100)
Q_TWA65MAP_total_mmHg_pressors_MAE<-
Q_TWA65MAP_total_mmHg_pressors_MAE[complete.cases(Q_TWA65MAP_total_mmHg_pressors_MAE)
, ]
library(reshape)
Q_TWA65MAP_total_mmHg_pressors_MAE_matrix<-
cast(data=Q_TWA65MAP_total_mmHg_pressors_MAE
      ,Q_TWA65MAP_total_mmHg ~ Q_total_pressors_formula_mg
      ,value = 'total_patientswith_MAE')
Q_TWA65MAP_total_mmHg_pressors_MAE_matrix<-
as.data.frame(Q_TWA65MAP_total_mmHg_pressors_MAE_matrix)
colnames(Q_TWA65MAP_total_mmHg_pressors_MAE_matrix)<-c('', 'total pressors Q1', 'total
pressors Q2', 'total pressors Q3', 'total pressors Q4')
rownames(Q_TWA65MAP_total_mmHg_pressors_MAE_matrix)<-c('total TWA MAP 65 Q1', 'total
TWA MAP 65 Q2', 'total TWA MAP 65 Q3', 'total TWA MAP 65 Q4')
Q_TWA65MAP_total_mmHg_pressors_MAE_matrix[,1]<-NULL
Q_TWA65MAP_total_mmHg_pressors_MAE_matrix<-
round(Q_TWA65MAP_total_mmHg_pressors_MAE_matrix,2)

...

# 3D Plots Data

```

```

## Exporting to excel

```{r}
write.xlsx(T_AUT50MAP_total_min_pressors_MAE_matrix,
"T_AUT50MAP_total_min_pressors_MAE_matrix.xlsx")
write.xlsx(T_AUT65MAP_total_min_pressors_MAE_matrix,
"T_AUT65MAP_total_min_pressors_MAE_matrix.xlsx")

write.xlsx(Q_AUT50MAP_total_min_pressors_MAE_matrix,
"Q_AUT50MAP_total_min_pressors_MAE_matrix.xlsx")
write.xlsx(Q_AUT65MAP_total_min_pressors_MAE_matrix,
"Q_AUT65MAP_total_min_pressors_MAE_matrix.xlsx")
write.xlsx(Q_AOT65MAP_total_min_pressors_MAE_matrix,
"Q_AOT65MAP_total_min_pressors_MAE_matrix.xlsx")

#quartiles ratio

write.xlsx(Q_AUT50MAP_total_ratio_pressors_MAE_matrix,
"Q_AUT50MAP_total_ratio_pressors_MAE_matrix.xlsx")
write.xlsx(Q_AUT65MAP_total_ratio_pressors_MAE_matrix,
"Q_AUT65MAP_total_ratio_pressors_MAE_matrix.xlsx")

# AUC
write.xlsx(Q_AUC65MAP_total_mmHgmin_pressors_MAE_matrix,
'Q_AUC65MAP_total_mmHgmin_pressors_MAE_matrix.xlsx')
# TWA
write.xlsx(Q_TWA65MAP_total_mmHg_pressors_MAE_matrix,
'Q_TWA65MAP_total_mmHg_pressors_MAE_matrix.xlsx')
```

3D Tertiles in R

```{r}
library('plot3D')

hist3D( main = paste("T AUT50MAP Plot\n Number of Cases =",
sum(T_AUT50MAP_total_min_pressors_MAE_matrix))
, x = c(1:3)
, y = c(1:3)
, z = T_AUT50MAP_total_min_pressors_MAE_matrix
, scale = F, expand = 0.01, bty = "b2", phi = 20, col = "#1e8bc3"
, border = "white", shade = 0.5, ltheta = 90
, space = 0.3 #space between bars
, ticktype = "detailed", d = 2
,resfac=3 #resolution
,xlab='T AUT50MAP total'
,ylab='T total pressors'
,zlab='patients with Any MAE'
,zlim=c(0,170))

```

```

hist3D( main = paste("T AUT65MAP Plot\n Number of Cases =",
sum(T_AUT65MAP_total_min_pressors_MAE_matrix))
, x = c(1:3)
, y = c(1:3)
, z = T_AUT65MAP_total_min_pressors_MAE_matrix
, scale = F, expand = 0.01, bty = "b2", phi = 20, col = "#1e8bc3"
, border = "white", shade = 0.5, ltheta = 90
, space = 0.3 #space between bars
, ticktype = "detailed", d = 2
, resfac=3 #resolution
, xlab='T AUT65MAP total'
, ylab='T total pressors'
, zlab='patients with Any MAE'
, zlim=c(0,170))

# Make the rgl version
# Animated version
# library("plot3Drgl")
# plotrgl()
```

3D Quartiles in R

```{r}
library('plot3D')

hist3D( main = paste("Q AUT50MAP Plot\n Number of Cases =",
sum(Q_AUT50MAP_total_min_pressors_MAE_matrix))
, x = c(1:4)
, y = c(1:4)
, z = Q_AUT50MAP_total_min_pressors_MAE_matrix
, scale = F, expand = 0.01, bty = "b2", phi = 20, col = "#1e8bc3"
, border = "white", shade = 0.5, ltheta = 90
, space = 0.3 #space between bars
, ticktype = "detailed", d = 2
, resfac=3 #resolution
, xlab='Q AUT50MAP total'
, ylab='Q total pressors'
, zlab='patients with Any MAE'
, zlim=c(0,170))

hist3D( main = paste("Q AUT65MAP Plot\n Number of Cases =",
sum(Q_AUT65MAP_total_min_pressors_MAE_matrix))
, x = c(1:4)
, y = c(1:4)
, z = Q_AUT65MAP_total_min_pressors_MAE_matrix
, scale = F, expand = 0.01, bty = "b2", phi = 20, col = "#1e8bc3"
, border = "white", shade = 0.5, ltheta = 90
, space = 0.3 #space between bars
, ticktype = "detailed", d = 2
, resfac=3 #resolution
, xlab='Q AUT65MAP total'
, ylab='Q total pressors'
, zlab='patients with Any MAE'
, zlim=c(0,170))

```

```

...

# 2D Plots

## Quartiles

## Q_AUT50MAP_total_min_pressors_MAE 2D test

### Q_AUT50MAP_total_min

```{r eval=FALSE, include=FALSE}
library(plotly)
library(dplyr)
library(scales)

p1<-Q_AUT50MAP_total_min_pressors_MAE %>%
  dplyr::filter(Q_total_pressors_formula_mg==1) %>%
  dplyr::group_by(Q_AUT50MAP_total_min) %>%
  dplyr::summarise(total_patientswith_MAE=sum(total_patientswith_MAE)) %>%
  plot_ly( x = ~Q_AUT50MAP_total_min
           , y= ~total_patientswith_MAE
           , type = 'bar'
           , marker = list(color = brewer_pal(palette="OrRd")(9)[5] )
           , name = 'Q1 pressors(mg)'
           ) %>%layout( yaxis = list(range = c(0, 170)) )

p2<-Q_AUT50MAP_total_min_pressors_MAE %>%
  dplyr::filter(Q_total_pressors_formula_mg==2) %>%
  dplyr::group_by(Q_AUT50MAP_total_min) %>%
  dplyr::summarise(total_patientswith_MAE=sum(total_patientswith_MAE)) %>%
  plot_ly( x = ~Q_AUT50MAP_total_min
           , y = ~total_patientswith_MAE
           , type = 'bar'
           , marker = list(color = brewer_pal(palette="OrRd")(9)[6] )
           , name = 'Q2 pressors(mg)'
           ) %>%layout( yaxis = list(range = c(0, 170)) )

p3<-Q_AUT50MAP_total_min_pressors_MAE %>%
  dplyr::filter(Q_total_pressors_formula_mg==3) %>%
  dplyr::group_by(Q_AUT50MAP_total_min) %>%
  dplyr::summarise(total_patientswith_MAE=sum(total_patientswith_MAE)) %>%
  plot_ly( x = ~Q_AUT50MAP_total_min
           , y = ~total_patientswith_MAE
           , type = 'bar'
           , marker = list(color = brewer_pal(palette="OrRd")(9)[7] )
           , name = 'Q3 pressors(mg)'
           ) %>%layout( yaxis = list(range = c(0, 170)) )

p4<-Q_AUT50MAP_total_min_pressors_MAE %>%
  dplyr::filter(Q_total_pressors_formula_mg==4) %>%
  dplyr::group_by(Q_AUT50MAP_total_min) %>%
  dplyr::summarise(total_patientswith_MAE=sum(total_patientswith_MAE)) %>%
  plot_ly( x = ~Q_AUT50MAP_total_min

```

```

    , y = ~total_patientswith_MAE
    , type = 'bar'
    , marker = list(color = brewer_pal(palette="OrRd")(9)[8] )
    , name = 'Q4 pressors(mg) '
    ) %>%layout( yaxis = list(range = c(0, 170)) )

library(plotly)
plotly::subplot(p1, p2, p3, p4,nrows = 2,titleY = T,titleX = T,shareX = T,shareY = T)
%>%
  layout(
    title = paste('Total_Patients with Any_MAE: ',
sum(Q_AUT50MAP_total_min_pressors_MAE$total_patientswith_MAE))
  )
...

### Q_total_pressors_formula_mg

```{r}
library(plotly)
library(dplyr)
library(scales)

p1<-Q_AUT50MAP_total_min_pressors_MAE %>%
  dplyr::filter(Q_AUT50MAP_total_min==1) %>%
  dplyr::group_by(Q_total_pressors_formula_mg) %>%
  dplyr::summarise(total_patientswith_MAE=sum(total_patientswith_MAE)) %>%
  plot_ly(
    x = ~Q_total_pressors_formula_mg
    , y= ~total_patientswith_MAE
    , type = 'bar'
    , marker = list(color = brewer_pal(palette="Blues")(9)[5] )
    , name = 'Q1 AUT50MAP (mins) '
    ) %>%layout( yaxis = list(range = c(0, 170)) )

p2<-Q_AUT50MAP_total_min_pressors_MAE %>%
  dplyr::filter(Q_AUT50MAP_total_min==2) %>%
  dplyr::group_by(Q_total_pressors_formula_mg) %>%
  dplyr::summarise(total_patientswith_MAE=sum(total_patientswith_MAE)) %>%
  plot_ly(
    x = ~Q_total_pressors_formula_mg
    , y = ~total_patientswith_MAE
    , type = 'bar'
    , marker = list(color = brewer_pal(palette="Blues")(9)[6] )
    , name = 'Q2 AUT50MAP (mins) '
    ) %>%layout( yaxis = list(range = c(0, 170)) )

p3<-Q_AUT50MAP_total_min_pressors_MAE %>%
  dplyr::filter(Q_AUT50MAP_total_min==3) %>%
  dplyr::group_by(Q_total_pressors_formula_mg) %>%
  dplyr::summarise(total_patientswith_MAE=sum(total_patientswith_MAE)) %>%
  plot_ly(
    x = ~Q_total_pressors_formula_mg
    , y = ~total_patientswith_MAE
    , type = 'bar'
    , marker = list(color = brewer_pal(palette="Blues")(9)[7] )
    , name = 'Q3 AUT50MAP (mins) '

```



```

    ) %>%layout( yaxis = list(range = c(0, 170)) )

p4<-Q_AUT50MAP_total_min_pressors_MAE %>%
  dplyr::filter(Q_AUT50MAP_total_min==4) %>%
  dplyr::group_by(Q_total_pressors_formula_mg) %>%
  dplyr::summarise(total_patientswith_MAE=sum(total_patientswith_MAE)) %>%
  plot_ly(
    x = ~Q_total_pressors_formula_mg
    , y = ~total_patientswith_MAE
    , type = 'bar'
    , marker = list(color = brewer_pal(palette="Blues")(9)[8] )
    , name = 'Q4 AUT50MAP (mins) '
  ) %>%layout( yaxis = list(range = c(0, 170)) )

library(plotly)
pfinal<-plotly::subplot(p1, p2, p3, p4,nrows = 2,titleY = T,titleX = T,shareX =
T,shareY = T) %>%
  layout(
    title = paste('Total Patients with Any_MAE: ',
sum(Q_AUT50MAP_total_min_pressors_MAE$total_patientswith_MAE))
  )

...

# Boxplot for all the bands MAP

```{r}
library(dplyr)
library(plotly)
library(RColorBrewer)
palette <- brewer.pal(9, "Set1")

# annotations
a <- list(
  text = "Mean Arterial Pressure Bands and time in minutes patients spent on that
band.",
  font = list(size = 16),
  xref = "paper",
  yref = "paper",
  yanchor = "bottom",
  xanchor = "center",
  align = "center",
  x = 0.5,
  y = 1,
  showarrow = FALSE
)

x <- list( title = "Total time spent in a given threshold (min)" )
p1 <- plot_ly(y =
FinalHemodynamicsDataset[FinalHemodynamicsDataset$Any_MAE==0,]$AUT65MAP_total_min,
type = "box",name = "<65 No MAE"
,marker = list(color = '#2ecc71'),opacity = 0.8, line = list(color =
'#27ae60')) %>%
  add_trace(y =
FinalHemodynamicsDataset[FinalHemodynamicsDataset$Any_MAE==1,]$AUT65MAP_total_min,nam
e = "<65 Any MAE"

```

```

        ,marker = list(color = '#e74c3c'),opacity = 0.8, line = list(color =
'#c0392b')) %>%

    add_trace(y =
FinalHemodynamicsDataset[FinalHemodynamicsDataset$Any_MAE==0,]$AOT65MAP_total_min,nam
e = ">65 No MAE"
        ,marker = list(color = '#2ecc71'),opacity = 0.8, line = list(color =
'#27ae60')) %>%
    add_trace(y =
FinalHemodynamicsDataset[FinalHemodynamicsDataset$Any_MAE==1,]$AOT65MAP_total_min,nam
e = ">65 Any MAE"
        ,marker = list(color = '#e74c3c'),opacity = 0.8, line = list(color =
'#c0392b')) %>%

layout(showlegend = F,xaxis = x
      #, yaxis = list(range=c(0,800))
      , annotations = a)

#htmlwidgets::saveWidget(sp1, "boxplots.html")
```

Boxplot for threshold <=65

```{r}
library(dplyr)
library(plotly)
library(RColorBrewer)
palette <- brewer.pal(9, "Set1")

# annotations
a <- list(
  text = "MAP < 65 mmHg duration per group",
  font = list(size = 16),
  xref = "paper",
  yref = "paper",
  yanchor = "bottom",
  xanchor = "center",
  align = "center",
  x = 0.5,
  y = 1,
  showarrow = FALSE
)

y <- list( title = "Total duration (min)" )
x <- list( title = "Groups" )
p1 <- plot_ly(y =
FinalHemodynamicsDataset[FinalHemodynamicsDataset$Any_MAE==0,]$AUT65MAP_total_min,
type = "box",name = "Absent MAE"
        ,marker = list(color = '#2ecc71'),opacity = 0.8, line = list(color =
'#27ae60')) %>%

```

```

    add_trace(y =
FinalHemodynamicsDataset[FinalHemodynamicsDataset$Any_MAE==1,]$AUT65MAP_total_min,name
e = "Present MAE"
    ,marker = list(color = '#e74c3c'),opacity = 0.8, line = list(color =
'#c0392b')) %>%

    add_trace(y = FinalHemodynamicsDataset$AUT65MAP_total_min,name = "All patients"
    ,marker = list(color = '#2980b9'),opacity = 0.8, line = list(color =
'#2980b9')) %>%

    layout(showlegend = F
    , xaxis = x
    , yaxis = y
    , annotations = a)

a <- list(
  text = "AUC MAP < 65 mmHg",
  font = list(size = 16),
  xref = "paper",
  yref = "paper",
  yanchor = "bottom",
  xanchor = "center",
  align = "center",
  x = 0.5,
  y = 1,
  showarrow = FALSE
)

y <- list( title = "Area under the curve (mmHg•min)" )
x <- list( title = "Groups" )
p2 <- plot_ly(y =
FinalHemodynamicsDataset[FinalHemodynamicsDataset$Any_MAE==0,]$AUC65MAP_total_mmHgmin
, type = "box",name = "Absent MAE"
    ,marker = list(color = '#2ecc71'),opacity = 0.8, line = list(color =
'#27ae60')) %>%
  add_trace(y =
FinalHemodynamicsDataset[FinalHemodynamicsDataset$Any_MAE==1,]$AUC65MAP_total_mmHgmin
,name = "Present MAE"
    ,marker = list(color = '#e74c3c'),opacity = 0.8, line = list(color =
'#c0392b')) %>%

  add_trace(y = FinalHemodynamicsDataset$AUC65MAP_total_mmHgmin,name = "All patients"
    ,marker = list(color = '#2980b9'),opacity = 0.8, line = list(color =
'#2980b9')) %>%

  layout(showlegend = F
    , xaxis = x
    , yaxis = y
    , annotations = a)

#htmlwidgets::saveWidget(sp1, "boxplots.html")
` ``

```

```

# Correlations

```

```

```{r}

supply(relevant_data_imputed, class)

cor_periods_min<-
cor(relevant_data_imputed[,c('Age','duration_Surgery_min','total_pressors_formula_mg'
,'STS_RiskAlgorithm_mean_imp')])

#non interactive circle corplot
cortable<-corrplot(cor_periods_min, method = "circle")

write.xlsx(as.data.frame(print(cortable)), "cortable.xlsx")

...

# Odd Ratio Plots

# General model

```{r}
#Generate plots

df <- oddsratios_table

labels<-c('Age', 'Category: Aortic Surgery', 'Category: CABG + Valve', 'Category:
Other', 'Category: Valve', 'Cross Clamp Time', 'delta HCT pct', 'Gender: Female',
'Mean EF', 'AUC MAP 65 mmHg Q2', 'AUC MAP 65 mmHg Q3', 'AUC MAP 65 mmHg Q4', 'total
pressors formula mg Q2', 'total pressors formula mg Q3', 'total pressors formula mg
Q4', 'STS Risk Algorithm Tertile 2', 'STS Risk Algorithm Tertile 3')
number_of_exposures<-length(labels)+1

df <- data.frame(yAxis = 2:number_of_exposures,
                 boxOdds = oddsratios_table$OR[2:number_of_exposures] ,
                 boxCILOW = oddsratios_table$`2.5 %`[2:number_of_exposures],
                 boxCIHigh = oddsratios_table$`97.5 %`[2:number_of_exposures]
)
#df <- df[order(df$boxOdds),]

p<-ggplot(df, aes(x = boxOdds
                 , y = labels
                 )) +
  geom_vline(aes(xintercept = 1), size = .1, linetype = "dashed") +
  geom_errorbarh(aes(xmax = boxCIHigh, xmin = boxCILOW)
                , size = .5
                , height = .2
                , color = "gray50") +
  geom_point(size = 2, color = "#2980b9") +
  #coord_trans(x = scales::exp_trans(1.01)) +
  #scale_y_continuous(breaks = c(-1:1), labels = c(-1:1)) +
  theme_bw()+
  theme(panel.grid.minor = element_blank()) +
  ylab("Exposure") +
  xlab("Odds ratio") +
  ggtitle("Impact on Any MAE")

```

```

p<-ggplotly(p%>%layout(hovermode = 'compare'))
...

# Exploratory Analysis

```{r}
#Generate plots

df <- oddsratios_table

labels2<-c('Age', 'Category: CABG + Valve', 'Category: Valve', 'delta HCT pct',
'Gender: Female', 'Mean EF', 'AUC MAP 65 mmHg Q2', 'AUC MAP 65 mmHg Q3', 'AUC MAP 65
mmHg Q4', 'total pressors formula mg Q2', 'total pressors formula mg Q3', 'total
pressors formula mg Q4')
number_of_exposures2<-length(labels2)+1

df <- data.frame(yAxis = 2:number_of_exposures2,
                boxOdds = oddsratios_table$OR[2:number_of_exposures2] ,
                boxCILOW = oddsratios_table$`2.5 %`[2:number_of_exposures2],
                boxCIHigh = oddsratios_table$`97.5 %`[2:number_of_exposures2]
)
#df <- df[order(df$boxOdds),]

p<-ggplot(df, aes(x = boxOdds
                , y = labels2
                )) +
  geom_vline(aes(xintercept = 1), size = .1, linetype = "dashed") +
  geom_errorbarh(aes(xmax = boxCIHigh, xmin = boxCILOW)
                , size = .5
                , height = .2
                , color = "gray50") +
  geom_point(size = 2, color = "#2980b9") +
  #coord_trans(x = scales::exp_trans(1.01)) +
  #scale_y_continuous(breaks = c(-1:1), labels = c(-1:1)) +
  theme_bw()+
  theme(panel.grid.minor = element_blank()) +
  ylab("Exposure") +
  xlab("Odds ratio") +
  ggtitle("Impact on Any MAE")
...

# Bubble plot

```{r}
colnames(Q_TWA65MAP_total_mmHg_pressors_MAE)[3]<-'MAE rate percentage'

p <- plot_ly(Q_TWA65MAP_total_mmHg_pressors_MAE
            , x = ~as.character(Q_TWA65MAP_total_mmHg)
            , y = ~as.character(Q_total_pressors_formula_mg)
            , text = ~round(`MAE rate percentage`,2)

```

```

    , textfont = list( size = 10
                      #, color = '#000000'
                      )
    , type = 'scatter'
    , mode = 'markers+text'
    , color = ~`MAE rate percentage`
    , size = ~`MAE rate percentage`
    , colors = c('#f1a9a087', '#bd1200')
    , sizes = c(10, 120)
    , showlegend = F
    , marker = list(
      #size = ~total_patientswith_MAE
      opacity = 1
    , sizemode = 'diameter')) %>%
  layout(
#title = 'Any MAE rate (%) for TWA-MAP under 65 mmHg
Number of cases = 518',
    xaxis = list(showgrid = T, title='vasopressors dose Q'),
    yaxis = list(showgrid = T, title='TWA-MAP < 65 mmHg Q'))

htmlwidgets::saveWidget(p, "bubbleplot.html")
```

```

## Tables

```

---
title: "Table1"
author: "Miguel Angel Armengol de la Hoz"
output:
  html_document:
    toc: true
    theme: united

```{r setup, include=FALSE}
#load the data
#load("C:/BIG FILES/Phenotyping/HemodynamicProjectsDatasetFeb18.RData")
```

Load libs
```{r}
library(tableone)
library(magrittr)
library(officer)
library(xlsx)
library(dplyr)
```

Including fluid data only for table1 (not for modeling)

```{r}
FinalHemodynamicsDataset_for_table1<-FinalHemodynamicsDataset

```

```

FinalHemodynamicsDataset_for_table1<-
left_join(FinalHemodynamicsDataset_for_table1,IO_data_Fluids_summarized)
```

Table 1

```{r}
listVars
listVarstab1<-
c(listVars,"STS_RiskAlgorithm_median_imp","delta_HCT_pct_median_imp","mean_ef_median_
imp","Colloids_ml","Crystalloids_ml","Cross_Clamp_Time","duration_CPB_min","AUT65MAP_
CPB_min")

catVars<-c("Gender",
  "Category",
  #Background diseases
  "Diabetes",
  "RFDyslipidemia",
  "Hypertension_final",
  "Smoking",
  "Congestive_Heart_Failure",
  "Previous_myocardial_infarction",
  "ChronicLungDisease",
  "Dialysis",
  #Preoperative medications
  "PreOpMedBetaBlockers",
  "PreOpMedACEI_or_ARBs",
  "PreOpMedInotropes",
  "PreOpMedSteroids",
  "PreOpMedAspirin",
  "PreOpMedLipidLowering") #AKA statins

table1 <- CreateTableOne(vars = listVarstab1, data =
FinalHemodynamicsDataset_for_table1, factorVars = catVars,testNormal=oneway.test
#strata is usefull in case we want several columns to
stratify by
)

table1strat <- CreateTableOne(vars = listVarstab1, data =
FinalHemodynamicsDataset_for_table1, factorVars = catVars,testNormal=oneway.test
,strata = c("Any_MAE")
#strata is usefull in case we want several columns to
stratify by
)
## nonnormal specifies variables to be shown as median [IQR]
#
# testApprox A function used to perform the large sample approximation based tests.
The
# default is chisq.test. This is not recommended when some of the cell have
# small counts like fewer than 5.

```

As an aside, the following code may help for your projects, as it improves the presentation of the tables above. You will still need to update the column and row names manually, but this should paste nicely into Word or LaTeX!

```
write.xlsx(as.data.frame(print(table1)), "/Users/marmengol/MEGA/Boston/BIDMC-Harvard/Phenotyping/Tables and Figures/table1.xlsx")
write.xlsx(as.data.frame(print(table1strat)), "/Users/marmengol/MEGA/Boston/BIDMC-Harvard/Phenotyping/Tables and Figures/table1strat.xlsx")
```

```
# if(!("dplyr" %in% installed.packages()[,1])) {
#   install.packages("dplyr")
# }
# library(dplyr)
# test<-table1 %>% print(
#   printToggle      = FALSE,
#   showAllLevels     = TRUE,
#   cramVars          = "kon"
# ) %>%
# {data.frame(
#   variable_name      = gsub(" ", "&nbsp;", rownames(.), fixed = TRUE), .,
#   row.names          = NULL,
#   check.names        = FALSE,
#   stringsAsFactors = FALSE)} %>%
# knitr::kable()
```

```
```
```

```
# Table 2
```

```
```{r}
```

```
library(tableone)
library(magrittr)
```

```
listExposures<-
c('AOT30DBP_postCPB_min', 'AOT30DBP_preCPB_min', 'AOT30DBP_total_min', 'AOT65MAP_CPB_min',
  'AOT65MAP_postCPB_min', 'AOT65MAP_preCPB_min', 'AOT65MAP_total_min', 'AOT95SBP_postCPB_min',
  'AOT95SBP_preCPB_min', 'AOT95SBP_total_min', 'AUT30DBP_postCPB_min', 'AUT30DBP_preCPB_min',
  'AUT30DBP_total_min', 'AUT50MAP_CPB_min', 'AUT50MAP_postCPB_min', 'AUT50MAP_preCPB_min',
  'AUT50MAP_total_min', 'AUT65MAP_CPB_min', 'AUT65MAP_postCPB_min', 'AUT65MAP_preCPB_min',
  'AUT65MAP_total_min', 'AUT95SBP_postCPB_min', 'AUT95SBP_preCPB_min', 'AUT95SBP_total_min',
  'AWI51_60MAP_CPB_min', 'AWI51_60MAP_postCPB_min', 'AWI51_60MAP_preCPB_min', 'AWI51_60MAP_total_min',
  'AWI61_65MAP_CPB_min', 'AWI61_65MAP_postCPB_min', 'AWI61_65MAP_preCPB_min', 'AWI61_65MAP_total_min',
  'duration_CPB_min', 'duration_postCPB_min', 'duration_preCPB_min', 'duration_Surgery_min')
```

```
table2all <- CreateTableOne(vars = listExposures, data =
  FinalHemodynamicsDataset_for_table1, testNormal=oneway.test)
```

```
write.xlsx(as.data.frame(print(table2all)), "table2all.xlsx")
```



```

table2strat <- CreateTableOne(vars = listExposures, data =
FinalHemodynamicsDataset_for_table1, factorVars =
listOutstable2,testNormal=oneway.test
                        ,strata = c("Any_MAE") )

write.xlsx(as.data.frame(print(table2strat)), "table2strat.xlsx")

#strata is usefull in case we want several columns to stratify by
# nonnormal specifies variables to be shown as median [IQR]
# testApprox A function used to perform the large sample approximation based tests.
The
# default is chisq.test. This is not recommended when some of the cell have
# small counts like fewer than 5.

...

# Table 3

```{r}

library(tableone)
library(magrittr)

listOutstable2<-c('Any_MAE',listOuts)

table2all <- CreateTableOne(vars = listOutstable2, data = FinalHemodynamicsDataset,
factorVars = listOutstable2,testNormal=oneway.test
                        )

table2strat <- CreateTableOne(vars = listOutstable2, data = FinalHemodynamicsDataset,
factorVars = listOutstable2,testNormal=oneway.test
                        ,strata = c("Any_MAE"))

write.xlsx(as.data.frame(print(table2strat)), "table2strat.xlsx")

                        #strata is usefull in case we want several columns to
stratify by
## nonnormal specifies variables to be shown as median [IQR]
#
# testApprox A function used to perform the large sample approximation based tests.
The
# default is chisq.test. This is not recommended when some of the cell have
# small counts like fewer than 5.

# table2<-print(table2,estApprox=listVars, exact = "stage", quote =
TRUE,showAllLevels = TRUE)
#
# docx( ) %>%
#   addFlexTable(table2 %>%
#   FlexTable(header.cell.props = cellProperties( background.color = "#003366"),
#   header.text.props = textBold( color = "white" ),
#   add.rownames = TRUE ) %>%
#   setZebraStyle( odd = "#DDDDDD", even = "#FFFFFF" ) ) %>%
#   writeDoc(file = "table2.docx")

```

```
#  
# browseURL("table2.docx") #wordfile  
``
```

APPENDIX VIII: MACHINE LEARNING MODELS WITH PREOPERATIVE RISK FACTORS AND INTRAOPERATIVE HYPOTENSION PARAMETERS PREDICT MORTALITY FOLLOWING CARDIAC SURGERY

STS (Society of Thoracic Surgeons) VERSION 2.81 Definitions for postoperative outcomes

Complications

Indicate whether a postoperative event occurred during the hospitalization for surgery. This includes the entire postoperative period up to discharge, even if over 30 days.

Renal Failure

Indicate whether the patient had acute renal failure or worsening renal function resulting in one or both of the following

1. Increase in serum creatinine level 3.0 x greater than baseline
2. serum creatinine level ≥ 4 mg/dL
3. Acute rise must be at least 0.5 mg/dl
4. A new requirement for dialysis postoperatively.

Stroke

Indicate whether the patient has a postoperative stroke (i.e., any confirmed neurological deficit of abrupt onset caused by a disturbance in cerebral blood supply) that did not resolve within 24 hours.

Death

Operative Death includes: (1) all deaths, regardless of cause, occurring during the hospitalization in which the operation was performed, even if after 30 days (including patients transferred to other acute care facilities); and (2) all deaths, regardless of cause, occurring after discharge from the hospital, but before the end of the thirtieth postoperative day.

STS Dataset Notations

As a reference and in order to verify every variable in the STS dataset, we used the manuscript 'Increased Glycemic Variability in Patients with Elevated Preoperative HbA1C Predicts Adverse Outcomes Following Coronary Artery Bypass Grafting Surgery', this paper used the 2014 dataset and our current dataset contains data up to 2017.

Perioperative_MI was discontinued in July 2011 thus it cannot be found anywhere else since the lab data to calculate it has not been collected, thus, we excluded it from the study.

Infect_Sternum_Deep was overcoded before 2011, so after 2011, when it was redefined to PostOpDeepSternalMediastin there are no cases of Infect_Sternum_Deep, so we included any kind of infection in the study.

Neuro_Stroke_Permanent changed to PostOpNeuroStrokePermanent in the version dataset STS_2.81_July_2014_June_2017

Smoking is the most inconsistent variable throughout the database due to changing definitions version to version. It has been consistent since July 2014.

CHF changes in 2014 to HeartFailure2wks

But it is since 2013 when Percentages are multiplied by 2

The variable was better documented since 2013. They drastically undercoded this variable in the past.

In June 2014 PreviousMI changed to PriorMI

Hypertension changes to RFHypertension in 2014.

In the version dataset STS_2.73_July_2011_Jun_2014 PreOpMedACE_ARBIInhibitors changed to PreOpMedACEIARBI48hrs.

Extended description of ML models and their performance

We developed three sets of data from our database based on specific phases of CPB

1. CPB – only during CPB,
2. Outside CPB – before and after CPB [total surgery – CPB] and
3. Total Surgery – anesthesia start to anesthesia end including CPB [CPB + outside CPB]

We trained the following with each of the datasets using hyper-parameter tuning (automatically iterating over each defined combination of parameters per model until the best performance is achieved):

- Logistic Regression (LR)
- Random forests (RF)
- Neural Networks (NN)
- Support Vector Machines (SVM)
- Extreme gradient boosting (XGB)

All 5 models were trained with the three datasets that were developed based on the phase of surgery (15 models in total). Model number #10 was the one performing the best.

Table A2.1.: Different models trained for each surgery phase and method.

	LR	RF	NN	SVM	XGB
CPB	Model #1	Model #2	Model #3	Model #4	Model #5
outsideCPB	Model #6	Model #7	Model #8	Model #9	Model #10
Total Surgery	Model #11	Model #12	Model #13	Model #14	Model #15

Table A.2.2.: Ability of models in predicting mortality based on specific phases of surgery

Models	AUROC	PPV	Specificity	Sensitivity
<i>Outside CPB phase[†]</i>				
Logistic Regression	0.80 [0.69-0.88]	0.08 [0.04-0.11]	0.80 [0.77-0.82]	0.71 [0.57-0.88]
Random Forests	0.82 [0.73-0.90]	0.09 [0.05-0.14]	0.86 [0.83-0.88]	0.68 [0.48-0.84]
Neural Networks	0.78 [0.68-0.85]	0.11 [0.06-0.19]	0.92 [0.90-0.93]	0.36 [0.20-0.55]
Support Vector Machine	0.70 [0.59-0.79]	0.06 [0.02-0.10]	0.89 [0.87-0.91]	0.36 [0.20-0.55]
Extreme Gradient Boosting	0.87 [0.81-0.93]	0.10 [0.06-0.15]	0.86 [0.84-0.88]	0.75 [0.57-0.90]
<i>CPB phase</i>				
Logistic Regression	0.83 [0.72-0.90]	0.07 [0.04-0.10]	0.77 [0.76-0.80]	0.75 [0.57-0.89]
Random Forests	0.86 [0.76-0.92]	0.09 [0.05-0.13]	0.81 [0.79-0.83]	0.79 [0.61-0.92]
Neural Networks	0.72 [0.59-0.85]	0.09 [0.04-0.13]	0.89 [0.88-0.91]	0.43 [0.20-0.61]
Support Vector Machine	0.69 [0.57-0.80]	0.08 [0.04-0.11]	0.84 [0.82-0.86]	0.57 [0.44-0.71]

Extreme Gradient Boosting	0.80 [0.67-0.90]	0.09 [0.06-0.14]	0.83 [0.81-0.85]	0.79 [0.61-0.91]
Total surgery phase[‡]				
Logistic Regression	0.80 [0.70-0.88]	0.08 [0.04-0.12]	0.81 [0.79-0.83]	0.68 [0.51-0.84]
Random Forests	0.83 [0.74-0.92]	0.10 [0.06-0.14]	0.85 [0.83-0.88]	0.68 [0.52-0.89]
Neural Networks	0.70 [0.59-0.80]	0.10 [0.03-0.18]	0.96 [0.94-0.97]	0.21 [0.07-0.36]
Support Vector Machine	0.66 [0.52-0.79]	0.08 [0.04-0.12]	0.87 [0.86-0.89]	0.46 [0.30-0.65]
Extreme Gradient Boosting	0.85 [0.76-0.91]	0.10 [0.06-0.15]	0.86 [0.84-0.88]	0.68 [0.50-0.81]

AUROC, area under the receiver operating characteristics curve; CPB, cardiopulmonary bypass; [†] Outside CPB: duration before and after CPB (Total surgery duration – CPB duration); [‡] Total surgery: CPB + outside CPB

CPB, Outside CPB and Total Surgery AUROC

Figure A2.1.: AUROC for all different surgery phases and methods approached.

Table A2.2.: Parameters tuned and their ranges for each machine learning method.

	Hyperparameters
LR	'clf__C': [0.0001, 0.001, 0.01, 0.1, 1.0], 'clf__solver': ['newton-cg', 'lbfgs', 'sag', 'saga'], 'clf__warm_start': [True, False]
RF	'clf__n_estimators': [100, 150, 200, 250, 300, 350, 1000], 'clf__max_features': ['sqrt'], 'clf__max_depth': [2, 3, 4, 5], 'clf__bootstrap': [True, False], 'clf__criterion': ["gini"], 'clf__warm_start': [True, False]
NN	'clf__hidden_layer_sizes': [(10, 10, 10), (50, 100, 50), (100, 20)], 'clf__activation': ['tanh', 'relu'], 'clf__solver': ['sgd', 'adam'], 'clf__alpha': [0.0001, 0.05], 'clf__learning_rate': ['constant', 'adaptive']
SVM	'clf__C': [1, 10, 100, 1000], 'clf__gamma': [0.001, 0.0001], 'clf__kernel': ['rbf', 'linear']
XGB	'clf__max_depth': [1, 3, 100], 'clf__learning_rate': [0.1], 'clf__n_estimators': [100, 500, 1000], 'clf__silent': [True], 'clf__objective': ['binary:logistic'], 'clf__booster': ['gbtree'], 'clf__n_jobs': [1], 'clf__nthread': [None], 'clf__gamma': [0], 'clf__min_child_weight': [1], 'clf__max_delta_step': [0], 'clf__subsample': [1], 'clf__colsample_bytree': [1], 'clf__colsample_bylevel': [1], 'clf__reg_alpha': [0, 0.0001, 0.05, 0.5], 'clf__reg_lambda': [0, 1], 'clf__scale_pos_weight': [1, 53.22], (Controls the balance of positive and negative weights, useful for unbalanced classes. A typical value to consider: sum negative instances / sum positive instances) 'clf__base_score': [0.5], 'clf__random_state': [0], 'clf__seed': [None]

Code

Data exploration

```
title: "01_data_exploration"
author: "Miguel Armengol"
date: "2/20/2019"
output:
  html_document:
    toc: true
    theme: united
---
```

```
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
```

```
# Loading libraries
```

```
```{r message=FALSE, warning=FALSE}
library(dplyr)
library(feather)
library(knitr)
library(plotly)
library(sqldf)
library(zoo)
library(xts)
```
```

```
# Loading the data
```

```
```{r message=FALSE, warning=FALSE}
FinalHemodynamicsDataset<-readRDS("~/MEGA/Boston/BIDMC-Harvard/Phenotyping/Dataset
Creation/FinalHemodynamicsDataset.rds")
df_vitals<-read_feather("/Users/marmengol/BIG FILES/neural networks/vitals.feather")
```
```

```
## Studying the data
```

```
### Vitalsigns Level
```

```
#### Data counts
```

```
```{r}
#source_counts<-data.frame(table(df_vitals$Source_Name))
table(df_vitals$Source_Name)
```
```

Let's focus on:

| Src          | Freq    |
|--------------|---------|
| HeartRate    | 6892126 |
| ABPMean      | 6851284 |
| PAPMean      | 6166719 |
| PulseRate    | 5517302 |
| PulseRateII  | 5185806 |
| ABPDiastolic | 4992502 |

```
| ABPSystolic | 4992444 |
```

```
#### Data distribution
```

```
```{r}
tapply(df_vitals$Value[df_vitals$Source_Name %in% c( "Heart Rate"
, "ABP-Mean"
, "PAP-Mean"
, "Pulse Rate"
, "Pulse Rate II"
, "ABP-Diastolic"
, "ABP-Systolic"
)], df_vitals$Source_Name[df_vitals$Source_Name %in% c( "Heart Rate"
, "ABP-Mean"
, "PAP-Mean"
, "Pulse Rate"
, "Pulse Rate II"
, "ABP-Diastolic"
, "ABP-Systolic"
)], summary)
```
```

```
#### Data minimum and maximum sampling time (s)
```

```
```{r echo=TRUE, message=FALSE, warning=FALSE}
df_vitals_selected<-df_vitals%>%
  filter(Source_Name %in% c( "Heart Rate"
, "ABP-Mean"      #comment
, "PAP-Mean"
, "Pulse Rate"
, "Pulse Rate II"
, "ABP-Diastolic"
, "ABP-Systolic"
)
)

df_vitals_selected<-df_vitals_selected%>%
  arrange(Case_Name,Source_Name,Time)

heart_rate<-subset(df_vitals_selected, df_vitals_selected$Source_Name=='Heart Rate')
heart_rate<-arrange(heart_rate, Case_Name, Time)

abp_mean<-subset(df_vitals_selected, df_vitals_selected$Source_Name=='ABP-Mean')
abp_mean<-arrange(abp_mean, Case_Name, Time)

pap_mean<-subset(df_vitals_selected, df_vitals_selected$Source_Name=="PAP-Mean")
pap_mean<-arrange(pap_mean, Case_Name, Time)

pulse_rate<-subset(df_vitals_selected,df_vitals_selected$Source_Name=="Pulse Rate")
pulse_rate<-arrange(pulse_rate, Case_Name, Time)

pulse_rateII<-subset(df_vitals_selected,df_vitals_selected$Source_Name=="Pulse Rate II")
pulse_rateII<-arrange(pulse_rateII, Case_Name, Time)

abp_diastolic<-subset(df_vitals_selected,df_vitals_selected$Source_Name=="ABP-Diastolic")
abp_diastolic<-arrange(abp_diastolic, Case_Name, Time)
```

```

abp_systolic<-subset(df_vitals_selected, df_vitals_selected$Source_Name=="ABP-Systolic")
abp_systolic<-arrange(abp_systolic, Case_Name, Time)

...

*DON'T FORGET*
!!!!!!!!!!!!!!*Run file 01B_fixing_surgery times after this.(file not available on open repo
since it contains PHI)* !!!!!!!

```

Dataset creation full

```

---
title: "02_dataset_creation_full"
author: "Miguel Armengol"
date: "2/20/2019"
output:
  html_document:
    toc: true
    theme: united
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

Environment

```{r message=FALSE, warning=FALSE}
library(dplyr)
library(feather)
library(knitr)
library(plotly)
library(sqldf)
library(zoo)
library(data.table)
library(summarytools)
library(imputeTS)
library(pracma)
library(entropy)
```

abp_mean_full_surgery

Filtering out

Duration Surgery (min) calculation

First and End Times of VitalSigns, VitalSigns_Times creation

*Run 011_fixing_surgery_times.Rmd before going on (the 011 file won't be
included on github since it contains PHI)*

```{r}
library(dplyr)

```

```

abp_mean_First <- abp_mean %>%
group_by(Case_Name) %>%
dplyr::select(c(Case_Name,Time)) %>%
arrange(Time)%>%
filter(!duplicated(Case_Name))
colnames(abp_mean_First) <- c('Case_Name', 'abp_mean_First_Time')

abp_mean_Last <- abp_mean %>%
group_by(Case_Name) %>%
dplyr::select(c(Case_Name,Time)) %>%
arrange(desc(Time))%>%
filter(!duplicated(Case_Name))
colnames(abp_mean_Last) <- c('Case_Name', 'abp_mean_Last_Time')

abp_mean_Times<-merge(abp_mean_First, abp_mean_Last)
```

Duration times

```{r}
abp_mean_Times['duration_Surgery_min']<-
as.numeric(difftime(abp_mean_Times$`abp_mean_Last_Time`,abp_mean_Times$`abp_m
ean_First_Time`), units="mins")
```

Creating lag and diff columns

```{r message=FALSE, warning=FALSE}
abp_mean<-abp_mean%>%
  group_by(Case_Name)%>%
  mutate(
    time_lag=lag(Time)
    ,diff= abs(as.numeric(time_lag-Time))
  )
```

Plotting sampling time

```{r}
abp_mean%>%
group_by(Case_Name)%>%
filter(diff>5*60)%>%
  summarise(n_distinct(Case_Name))
```

The median surgery length is 283.5 minutes.
We have a total of 174 (3%) cases with missing sampling times > 5 minutes
(big 'holes' without data).
We have decided to exclude those cases.

Excluding cases with missing sampling gaps

We are excluding gaps > 5 minutes or surgery length < 60 minutes.

```

According to our previous work with this data-set, times-tamps indicating surgeries lasted less than 60 minutes might be an error from the monitor that is recording the vitals, thus we are excluding those cases.

```

```{r}
# Creating the not in function

'!in%' <- function(x,y)!('%in%'(x,y))

cases_to_exclude<-union_all(
  abp_mean%>%
  group_by(Case_Name)%>%
  filter(diff>5*60)%>%
  select(Case_Name)
,
  abp_mean_Times%>%
  filter(duration_Surgery_min <= 60)%>%
  select(Case_Name)
)

cases_to_exclude<-unique(cases_to_exclude$Case_Name)
print('Number of Initial cases')
a<-n_distinct(abp_mean$Case_Name)
a
abp_mean_filtered<-abp_mean%>%
  filter(Case_Name %!in% cases_to_exclude)
print('Number of Cases Excluded due to gaps > 5 minutes or surgery length <
60 minutes.')
b<-n_distinct(abp_mean_filtered$Case_Name)
a-b
print('Number of Final cases')
b

...

5979 - 186 (missing sampling gaps > 5 minutes or surgery length < 60 minutes)
= 5793

### Expanding data points

Expanding data points when several missing time-stamps are found

```{r}
  abp_mean_full_times<-as.data.frame(matrix(0,ncol = 1, nrow = 0))
  colnames(abp_mean_full_times)<-c('Time')

  for (i in 1:nrow(abp_mean_Times)) {
    correct_sequence<-
as.data.frame(seq(abp_mean_Times$abp_mean_First_Time[i],abp_mean_Times$abp_me
an_Last_Time[i], by = '15 secs'))
    correct_sequence['Case_Name']<-abp_mean_Times$Case_Name[i]
    abp_mean_full_times<-rbind(correct_sequence,abp_mean_full_times)
  }
  colnames(abp_mean_full_times)[1]<-'Time'

```

```

# excluding cases as previously defined, also from this dataframe
abp_mean_full_times<-abp_mean_full_times%>%
  filter(Case_Name %!in% cases_to_exclude)

abp_mean_expanded<-
left_join(abp_mean_full_times,abp_mean_filtered%>%select('Case_Name','Time','
Value'), by = c('Case_Name','Time'))
```

Excluding cases with invalid cpb, precpb, postcpb or outsidecpb data-
points

```{r}

# cases with valid cpb data
abp_mean_valid_cases1<-inner_join( abp_mean_expanded, AllPeriodsDF)%>%
  filter(Time>=CPB_Begin_Time & Time<=CPB_End_Time)%>%
  distinct(Case_Name)

# cases with valid outsidecpb data
abp_mean_valid_cases2<-inner_join( abp_mean_expanded, AllPeriodsDF)%>%
  filter(Time< CPB_Begin_Time | Time>CPB_End_Time)%>%
  distinct(Case_Name)

# cases with valid precpb data
abp_mean_valid_cases3<-inner_join( abp_mean_expanded, AllPeriodsDF)%>%
  filter(Time< CPB_Begin_Time)%>%
  distinct(Case_Name)

# cases with valid postcpb data
abp_mean_valid_cases4<-inner_join( abp_mean_expanded, AllPeriodsDF)%>%
  filter(Time > CPB_Begin_Time)%>%
  distinct(Case_Name)

valid_cases<-inner_join(abp_mean_valid_cases1,abp_mean_valid_cases2)
valid_cases<-inner_join(valid_cases,abp_mean_valid_cases3)
valid_cases<-inner_join(valid_cases,abp_mean_valid_cases4)
```

Selecting only valid cases from our dataset

```{r}
print('Number of Initial cases')
a<-n_distinct(abp_mean_expanded$Case_Name)
a

abp_mean_expanded<-inner_join(abp_mean_expanded , valid_cases)

print('Number of Cases Excluded due to Inadequate Perioperative Data')
b<-n_distinct(abp_mean_expanded$Case_Name)
a-b

print('Number of Final cases')
n_distinct(abp_mean_expanded$Case_Name)
```

```

### ### Artifact Removal

Remove artifacts using the following rules, in order:

- (a) SBP (Systolic Blood Pressure) greater than or equal to 300 or SBP less than or equal to 20 mmhg.
- (b) SBP less than or equal to DBP + 5 mmhg.
- (c) DBP (Diastolic Blood Pressure) less than or equal to 5 mmhg or DBP greater than or equal to 225 mmhg.

\_Ref: Salmasi, V., Maheshwari, K., Yang, D., Mascha, E. J., Singh, A., Sessler, D. I., & Kurz, A. (2017). Relationship between Intraoperative Hypotension, Defined by Either Reduction from Baseline or Absolute Thresholds, and Acute Kidney and Myocardial Injury after Noncardiac SurgeryA Retrospective Cohort Analysis. *Anesthesiology: The Journal of the American Society of Anesthesiologists*, 126(1), 47-65.\_

```
```{r}
#MAP artifact removal
abp_mean_expanded$Value[abp_mean_expanded$Value <= 0]<-NA
abp_mean_expanded$Value[abp_mean_expanded$Value >= 255]<-NA
```
```

### ### Imputing intra-surgery missing data

#### #### Median imputation

We used median imputation since sometimes the first values of a dataframe are all artifacts (so they were removed), therefore using LPV would give NA to all of them.

```
```{r}
impute.median <- function(x) replace(x, is.na(x), median(x, na.rm = TRUE))
abp_mean_expanded<-abp_mean_expanded%>%
  group_by(Case_Name)%>%
  mutate(
    Value_imputed = as.integer(impute.median(Value))
  )
```
```

LVP and linear interpolation methods can also be tried for imputation.

### ### Reshaping

```
```{r}
# I'm creating a counter column so we know how many datapoints we have per
case (we'll later use this for reshaping the dataframe)
abp_mean_expanded_count<-abp_mean_expanded %>%
  group_by(Case_Name) %>%
  mutate(
    ones = 1,
    counter = cumsum(ones)
  )
```
```

```
Reshaping the table so we have 1 row per patient and one column each data
point
abp_mean_resaped<-dcast(abp_mean_expanded_count, Case_Name ~ counter,
value.var = "Value_imputed")
```
```

Each observation (row) corresponds to one case, there are 3650 data points (variables=columns), 1 every 15 seconds.

This is because the longest case contains 3650 data points using ST 15 seconds.

Cases shorter than this one will contain NA from the last time-stamp of the surgery until data-point number 3650.

```
'data.frame': 5793 obs. of 3651 variables:
 $ Case_Name: int 1 2 3 4 ...
 $ 1 : int 66 63 23 216 197 64 72 73 78 39 ...
 $ 2 : int 66 63 121 69 93 64 72 73 78 98 ...
 $ 3 : int 66 63 193 69 91 64 72 73 78 125 ...
 $ 4 : int 66 63 103 67 91 64 72 73 78 91 ...
 $ 5 : int 66 63 103 70 90 64 72 73 78 79 ...
 $ 6 : int 66 63 103 73 90 64 72 73 78 77 ...
 $ 7 : int 66 63 101 70 87 64 5 73 73 81 ...
 $ 8 : int 66 63 108 69 83 64 85 73 70 82 ...
 $ 9 : int 66 63 108 67 86 64 100 73 72 85 ...
 $ 10 : int 66 63 106 67 85 64 101 73 75 85 ...
 $ 11 : int 66 63 111 68 84 64 104 73 76 81 ...
 $ 12 : int 66 63 122 69 90 56 100 73 78 82 ...
 $ 13 : int 66 63 129 78 99 60 97 73 76 81 ...
 $ 14 : int 115 63 112 78 97 78 96 73 78 80 ...
 $ 15 : int 86 63 99 70 93 77 94 198 76 82 ...
```

Imputing data with a special value.

Instead of removing the rows with NaN values when surgery finishes, we can replace all NaN values with a specific value that does not appear naturally in the input, such as -1.

The problem can be modeled as-is and we can encourage the model to learn that a specific value means "missing."

Ref: <https://machinelearningmastery.com/handle-missing-timesteps-sequence-prediction-problems-python/>

Replacing NA with -1

```
`r`
abp_mean_resaped[is.na(abp_mean_resaped)]<-(-1)
`r`
```

Entropy

We are not using entropy anymore.

```
`r eval=FALSE, include=FALSE`
abp_mean_resaped['entropy']<-
round(apply(abp_mean_resaped[2:ncol(abp_mean_resaped)], 1 , entropy ) , 2)
`r`
```


189 (3%) of the cases are outliers according to the boxplot upper fence.

In the end we are not removing data beyond outliers

Removing cases beyond outliers

In the end we are not removing data beyond outliers

```
```{r eval=FALSE, include=FALSE}
abp_mean_no_outliers<-abp_mean_reshaped[,1:1791] # we need to add 1 (1790+1)
since the first column is Case_Name
```
```

Imputing after surgery missing data

We are not imputing after surgery missing data anymore

Linear interpolation imputation works vertically, so we transpose the dataframe to have one column per case, instead of one row per case.

```
```{r eval=FALSE, include=FALSE}
abp_mean_no_outliers_t<-as.data.frame(t(abp_mean_no_outliers[,,-1]))
colnames(abp_mean_no_outliers_t) <- abp_mean_no_outliers[,1]
abp_mean_no_outliers_t_imputed<-na.interpolation(abp_mean_no_outliers_t,
option = "linear")

abp_mean_no_outliers_imputed<-
as.data.frame(t(abp_mean_no_outliers_t_imputed))
abp_mean_no_outliers_imputed['Case_Name']<-
rownames(abp_mean_no_outliers_imputed)
row.names(abp_mean_no_outliers_imputed)<-1:nrow(abp_mean_no_outliers_imputed)

# re-ordering columns, we want Case_Name to be the first column

desired_order<-1791
desired_order<-c(desired_order,1:1790)
abp_mean_no_outliers_imputed<-abp_mean_no_outliers_imputed[,desired_order]
```
```

Joining with other data-sets

```
```{r}
abp_mean_all_data<-inner_join(abp_mean_reshaped,FinalHemodynamicsDataset%>%
  dplyr::select( Case_Name
                ,Gender
                ,Age
                ,Category

                ,duration_Surgery_min
                ,duration_CPB_min
                ,duration_outsideCPB_min
                ,duration_preCPB_min
                ,duration_postCPB_min

                ,STS_RiskAlgorithm_multiple_imp
```

```

,STS_RiskAlgorithm_median_imp
,STS_RiskAlgorithm
,total_pressors_formula_mg

# categorical variables
,Q_total_pressors_formula_mg
,T_STS_RiskAlgorithm_median_imp
,T_STS_RiskAlgorithm_multiple_imp
,Q_AUC65MAP_total_mmHgmin
,Q_TWA65MAP_total_mmHg
,Q_AUC65MAP_CPB_mmHgmin
,Q_AUC65MAP_outsideCPB_mmHgmin
,Q_TWA65MAP_CPB_mmHg
,Q_TWA65MAP_outsideCPB_mmHg

# TWA variables
,TWA65MAP_total_mmHg
,TWA65MAP_CPB_mmHg
,TWA65MAP_outsideCPB_mmHg
,TWA65MAP_preCPB_mmHg
,TWA65MAP_postCPB_mmHg

,delta_HCT_pct_median_imp

,mean_ef_median_imp
,Cross_Clamp_Time_imp

,AUC65MAP_total_mmHgmin
,AUC65MAP_CPB_mmHgmin
,AUC65MAP_outsideCPB_mmHgmin
,AUC65MAP_preCPB_mmHgmin
,AUC65MAP_postCPB_mmHgmin
# Outcomes
,Any_MAE
,Renal_Failure
,Death
,Stroke
,RFDS
,RFD
,DS
,RFS

))

```

Removing columns where 100% is NA

```{r}
#abp_mean_all_data <-
abp_mean_all_data[,colSums(is.na(abp_mean_all_data))<nrow(abp_mean_all_data)]
```

```

```

Imputing missing values in mean_ef_median_imp

there were some missing values in mean_ef_median_imp, imputing
mean_ef_median_imp for 81 cases missing

```{r}
abp_mean_all_data$mean_ef_median_imp[is.na(abp_mean_all_data$mean_ef_median_i
mp)]<-round(mean(abp_mean_all_data$mean_ef_median_imp,na.rm = T),2)
```

Mapping variables

We don't need all the variables in our dataset to be numbers anymore.

```{r}
#abp_mean_all_data$Gender<-
plyr::revalue(abp_mean_all_data$Gender,c('Female'=1,'Male'=2))
#abp_mean_all_data$Category<-
plyr::revalue(abp_mean_all_data$Category,c('Aortic Surgery'=1,'CABG'=2,'CABG
+ Valve'=3,'Other'=3,'Valve'=4))
```

Generating report

```{r}
view(dfSummary(abp_mean_all_data))
```

Writing dataframe

```{r}
write.csv(abp_mean_all_data,'abp_mean_all_data.csv',row.names = F)
```

```

## train\_test\_df\_creation

```

---
title: "03_train_test_df_creation"
author: "Miguel Armengol"
date: "2/20/2019"
output: html_notebook

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

Loading libraries

```{r message=FALSE, warning=FALSE}
library(dplyr)
library(caret)
```

Creating de-identified id

```

```

```{r}

abp_mean_all_data['id']<- seq.int(nrow(abp_mean_all_data))
abp_mean_all_data_deid<-abp_mean_all_data
abp_mean_all_data_deid$Case_Name<-NULL
```

Splitting de-identified data into testing and training, balanced version.

We want the data to be sampled randomly but always the same way and we want to be
sure that train and test must be balanced.

```{r}
## set the seed to make our partition reproducible
set.seed(123)
# createDataPartition: "the random sampling is done within the levels of y when y is
a factor in an attempt to balance the class distributions within the splits."
## 75% of the sample size
train_idx <- createDataPartition(as.factor(abp_mean_all_data_deid$Death), times = 1,
p = 0.75, list=F)

train <- abp_mean_all_data_deid[train_idx, ]
test <- abp_mean_all_data_deid[-train_idx, ]

round(prop.table(table(train$Death)),2)
round(prop.table(table(test$Death)),2)
```

Exporting datasets

```{r}
write.csv(train,'../data/train.csv',row.names = F)
write.csv(test,'../data/test.csv',row.names = F)
```

Checking proportion of outcomes

```{r}
round(prop.table(table(train$Any_MAE)),2)
```

Splitting de-identified data into testing and training, unbalanced version.

The original dataset has a total of 5020 rows.

We are going to adjust how the 1s of the outcome (506) are divided across training
and test sets in the proportion of 30%, 70%.
The overall proportion of patients in training and test dataset is still 75%-25%
respectively.

ub in the following datasets stands for unbalanced.

```

```

```{r}
## CREATING THE TRAIN UNBALANCED DATASET

## set the seed to make our partition reproducible
set.seed(123)

# select random 30 % rows of the dataframe where Any_MAE ==1
train_ub<-abp_mean_all_data_deid%>%
filter(Any_MAE==1) %>%
sample_n(152) # 30% of 506 rows with Any_MAE = 1

#152 represents 4% of total number of rows (3765) in the training dataset which is
75% of the total dataset, so we still need 96 % of rows (3613) where Any_MAE == 0

train_ub<-rbind(train_ub,
  abp_mean_all_data_deid%>%
  filter(Any_MAE==0) %>%
  sample_n(3613)
)

#creating 'not in' function
'!in%' <- function(x,y)!('%in%'(x,y))

## CREATING THE TEST UNBALANCED DATASET

# select random 70 % rows of the dataframe where Any_MAE ==1 and that have not been
included in the training dataset
test_ub<-abp_mean_all_data_deid%>%
filter(Any_MAE==1 & id %!in% train_ub$id) %>%
sample_n(354) # 70% of 506 rows with Any_MAE = 1

#354 represents 28% of total number of rows (1255) in the test dataset which is 25%
of the total dataset, so we still need 72 % of rows (3613) where Any_MAE == 0 and
that have not been included in the training dataset

test_ub<-rbind(test_ub,
  abp_mean_all_data_deid%>%
  filter(Any_MAE==0 & id %!in% train_ub$id) %>%
  sample_n(901)
)
...

## Exporting datasets

```{r}
write.csv(train_ub,'train_ub.csv',row.names = F)
write.csv(test_ub,'test_ub.csv',row.names = F)
```

```

## dataset\_creation\_CPB

```
---
title: "01_data_exploration"
author: "Miguel Armengol"
date: "2/20/2019"
output:
  html_document:
    toc: true
    theme: united

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

Loading libraries

```{r message=FALSE, warning=FALSE}
library(dplyr)
library(feather)
library(knitr)
library(plotly)
library(sqldf)
library(zoo)
library(data.table)
library(summarytools)
library(imputeTS)
library(pracma)
library(entropy)
```

Importing CPB times data

Data was imported from the project "Nonlinear Association of Intraoperative Vasopressor-Inotrope Dose and Hypotension Duration with Adverse Outcomes following Cardiac Surgery" where these variables were generated, code on how these variables were generated is available on the project [repo](https://github.com/theonesp/vasopressor_dose_mae/).

- From our Vitalsign table, there are 390 Cases with no CPB Times Info.
- Meaning there are 5318 Cases with all the info we need
- We are not interested on Cases with NO CPB informations, thus we exclude this cases.

TIMEZONES NEED TO BE THE SAME ACROSS DATASETS AND COLUMNS

```{r}
AllPeriodsDF <- readRDS("~/MEGA/Boston/BIDMC-Harvard/bp_nn/AllPeriodsDF.rds")
# before working with the dataset we need to make sure we are using the same timezone across all our columns and datasets
sapply(AllPeriodsDF, function(x) attr(x,"tzone"))
desiredtimezone<-attr(abp_mean_expanded$Time,"tzone")
```

```
AllPeriodsDF$CPB_Begin_Time<-as.character(AllPeriodsDF$CPB_Begin_Time)
AllPeriodsDF$CPB_Begin_Time<-as.POSIXct(AllPeriodsDF$CPB_Begin_Time, format='%Y-%m-%d
%H:%M:%S', tz=desiredtimezone)
```

```
AllPeriodsDF$CPB_End_Time<-as.character(AllPeriodsDF$CPB_End_Time)
AllPeriodsDF$CPB_End_Time<-as.POSIXct(AllPeriodsDF$CPB_End_Time, format='%Y-%m-%d
%H:%M:%S', tz=desiredtimezone)
```
```

ingest vitasigns\_df!!!!!! the other timestamps are corrupted

```
# abp_mean_cpb
```

```
*We are going to include only those cases we defined match our inclusion criteria as
explained on '02_dataset_creation_full'*
```

We use abp_mean_expanded as created in 02_dataset_creation_full.Rmd so values have already been processed:

- Excluding gaps > 5 minutes or surgery length < 60 minutes.
- Expanding data points when several missing time-stamps are found.
- Artifact Removal.
- Imputing intra-surgery missing data: Median imputation.

```
# Subsetting the data only to include CPB time
```

```
```{r}
abp_mean_expanded_cpb<-inner_join( abp_mean_expanded, AllPeriodsDF)%>%
  filter(Time>=CPB_Begin_Time & Time<=CPB_End_Time)%>%
  select(Time, Case_Name, Value,Value_imputed)
```
```

```
### Reshaping
```

```
```{r}
# I'm creating a counter column so we know how many datapoints we have per case
(we'll later use this for reshaping the dataframe)
abp_mean_expanded_cpb_count<-abp_mean_expanded_cpb %>%
  group_by(Case_Name) %>%
  mutate(
    ones = 1,
    counter = cumsum(ones)
  )
```

```
# Reshaping the table so we have 1 row per patient and one column each data point
abp_mean_resaped_cpb<-dcast(abp_mean_expanded_cpb_count, Case_Name ~ counter,
value.var = "Value_imputed")
```
```

```
### Imputing data with a special value.
```

LVP and linear interpolation methods can also be tried for imputation.

Instead of removing the rows with NaN values, we can replace all NaN values with a specific value that does not appear naturally in the input, such as -1.

The problem can be modeled as-is and we can encourage the model to learn that a specific value means “missing.”

Ref: <https://machinelearningmastery.com/handle-missing-timesteps-sequence-prediction-problems-python/>

```
```{r}
abp_mean_reshaped_cpb[is.na(abp_mean_reshaped_cpb)]<-(-1)
```

Entropy

```{r}
abp_mean_reshaped_cpb['entropy']<-
round(apply(abp_mean_reshaped_cpb[2:ncol(abp_mean_reshaped_cpb)], 1 , entropy ) , 2)
```
```

Each observation (row) corresponds to one case, there are 3650 data points (variables=columns), 1 every 15 seconds.  
This is because the longest case contains 3650 data points using ST 15 seconds.  
Cases shorter than this one will contain NA from the last time-stamp of the surgery until data-point number 3650.

### Joining with other data-sets

```
```{r}
abp_mean_cpb_all_data<-merge(abp_mean_reshaped_cpb,
                             FinalHemodynamicsDataset%>%
                             dplyr::select( Case_Name
                                             ,Gender
                                             ,Age
                                             ,Category

                                             ,duration_Surgery_min
                                             ,duration_CPB_min
                                             ,duration_outsideCPB_min

                                             ,STS_RiskAlgorithm
                                             ,STS_RiskAlgorithm_multiple_imp
                                             ,STS_RiskAlgorithm_median_imp
                                             ,total_pressors_formula_mg

                                             # categorical variables
                                             ,Q_total_pressors_formula_mg
                                             ,T_STS_RiskAlgorithm_median_imp
                                             ,T_STS_RiskAlgorithm_multiple_imp
                                             ,Q_AUC65MAP_total_mmHgmin
                                             ,Q_TWA65MAP_total_mmHg
                                             ,Q_AUC65MAP_CPB_mmHgmin
                                             ,Q_AUC65MAP_outsideCPB_mmHgmin
                                             ,Q_TWA65MAP_CPB_mmHg
                                             ,Q_TWA65MAP_outsideCPB_mmHg

                                             # TWA variables
```



```

, TWA65MAP_total_mmHg
, TWA65MAP_CPB_mmHg
, TWA65MAP_outsideCPB_mmHg

, delta_HCT_pct_median_imp

, mean_ef_median_imp
, Cross_Clamp_Time_imp

, AUC65MAP_total_mmHgmin
, AUC65MAP_CPB_mmHgmin
, AUC65MAP_outsideCPB_mmHgmin

# Outcomes
, Any_MAE
, Renal_Failure
, Death
, Stroke
, RFDS
, RFD
, DS
, RFS

))

### Imputing missing values in mean_ef_median_imp

there were some missing values in mean_ef_median_imp, imputing mean_ef_median_imp for
81 cases missing

```{r}
abp_mean_cpb_all_data$mean_ef_median_imp[is.na(abp_mean_cpb_all_data$mean_ef_median_i
mp)] <- round(mean(abp_mean_cpb_all_data$mean_ef_median_imp, na.rm = T), 2)
```

Generating report

```{r eval=FALSE, include=FALSE}
view(dfSummary(abp_mean_cpb_all_data))
```

Exporting df

```{r eval=FALSE, include=FALSE}
write.csv(abp_mean_cpb_all_data, 'abp_mean_cpb_all_data.csv', row.names = F)
```

```

train\_test\_df\_creation\_CPB

```

---
title: "03_train_test_df_creation"
author: "Miguel Armengol"
date: "2/20/2019"
output: html_notebook

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

Loading libraries

```{r message=FALSE, warning=FALSE}
library(dplyr)
library(keras)
library(tfruns)
library(caret)
```

Creating de-identified id

```{r}
abp_mean_cpb_all_data['id']<- seq.int(nrow(abp_mean_cpb_all_data))
abp_mean_cpb_all_data_deid<-abp_mean_cpb_all_data
abp_mean_cpb_all_data_deid$Case_Name<-NULL
```

Splitting de-identified data into testing and training, balanced version.

We want the data to be sampled randomly but always the same way and we want
to be sure that train and test must be balanced.

```{r}
## set the seed to make our partition reproducible
set.seed(123)
# createDataPartition: "the random sampling is done within the levels of y
when y is a factor in an attempt to balance the class distributions within
the splits."
## 75% of the sample size
train_cpb_idx <-
createDataPartition(as.factor(abp_mean_cpb_all_data_deid$Death), times = 1, p
= 0.75, list=F)

train_cpb <- abp_mean_cpb_all_data_deid[train_cpb_idx, ]
test_cpb <- abp_mean_cpb_all_data_deid[-train_cpb_idx, ]

round(prop.table(table(train_cpb$Death)), 2)
round(prop.table(table(test_cpb$Death)), 2)
```

Exporting datasets

```{r}
write.csv(train_cpb, '../data/train_cpb.csv', row.names = F)
write.csv(test_cpb, '../data/test_cpb.csv', row.names = F)
```

```

## dataset\_creation\_outsideCPB

```
---
title: "01_data_exploration"
author: "Miguel Armengol"
date: "2/20/2019"
output:
  html_document:
    toc: true
    theme: united

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

Loading libraries

```{r message=FALSE, warning=FALSE}
library(dplyr)
library(feather)
library(knitr)
library(plotly)
library(sqldf)
library(zoo)
library(data.table)
library(summarytools)
library(imputeTS)
```

Importing CPB times data

Data was imported from the project "Nonlinear Association of Intraoperative Vasopressor-Inotrope Dose and Hypotension Duration with Adverse Outcomes following Cardiac Surgery" where these variables were generated, code on how these variables were generated is available on the project [repo] (https://github.com/theonesp/vasopressor_dose_mae/).

- From our Vitalsign table, there are 390 Cases with no CPB Times Info.
- Meaning there are 5318 Cases with all the info we need
- We are not interested on Cases with NO CPB informations, thus we exclude this cases.

TIMEZONES NEED TO BE THE SAME ACROSS DATASETS AND COLUMNS

```{r}
AllPeriodsDF <- readRDS("~/MEGA/Boston/BIDMC-Harvard/bp_nn/AllPeriodsDF.rds")
# before working with the dataset we need to make sure we are using the same
timezone across all our columns and datasets
sapply(AllPeriodsDF, function(x) attr(x,"tzone"))
desiredtimezone<-attr(abp_mean_expanded$Time,"tzone")

AllPeriodsDF$CPB_Begin_Time<-as.character(AllPeriodsDF$CPB_Begin_Time)
AllPeriodsDF$CPB_Begin_Time<-as.POSIXct(AllPeriodsDF$CPB_Begin_Time,
format='%Y-%m-%d %H:%M:%S', tz=desiredtimezone)
```

```

AllPeriodsDF$CPB_End_Time<-as.character(AllPeriodsDF$CPB_End_Time)
AllPeriodsDF$CPB_End_Time<-as.POSIXct(AllPeriodsDF$CPB_End_Time, format='%Y-
%m-%d %H:%M:%S', tz=desiredtimezone)
```

ingest vitasigns_df!!!!!!! the other timestamps are corrupted

abp_mean_outsidecpb

*We are going to include only those cases we defined match our inclusion
criteria as explained on '02_dataset_creation_full'*

We use abp_mean_expanded as created in 02_dataset_creation_full.Rmd so values
have already been processed:
- Excluding gaps > 5 minutes or surgery length < 60 minutes.
- Expanding data points when several missing time-stamps are found.
- Artifact Removal.
- Imputing intra-surgery missing data: Median imputation.

Subsetting the data only to include CPB time

```{r}
abp_mean_expanded_outsidecpb<-inner_join( abp_mean_expanded,
AllPeriodsDF)%>%
  filter(Time< CPB_Begin_Time | Time>CPB_End_Time)%>%
  select(Time, Case_Name, Value,Value_imputed)
```

LVP and linear interpolation methods can also be tried for imputation.

Reshaping

```{r}
# I'm creating a counter column so we know how many datapoints we have per
case (we'll later use this for reshaping the dataframe)
abp_mean_expanded_outsidecpb_count<-abp_mean_expanded_outsidecpb %>%
  group_by(Case_Name) %>%
  mutate(
    ones = 1,
    counter = cumsum(ones)
  )

# Reshaping the table so we have 1 row per patient and one column each data
point
abp_mean_resaped_outsidecpb<-dcast(abp_mean_expanded_outsidecpb_count,
Case_Name ~ counter, value.var = "Value_imputed")
```

Each observation (row) corresponds to one case, there are 3650 data points
(variables=columns), 1 every 15 seconds.
This is because the longest case contains 3650 data points using ST 15
seconds.
Cases shorter than this one will contain NA from the last time-stamp of the
surgery until data-point number 3650.

'data.frame': 5793 obs. of 3651 variables:

```

```

$ Case_Name: int 1 2 3 4 ...
$ 1   : int 66 63 23 216 197 64 72 73 78 39 ...
$ 2   : int 66 63 121 69 93 64 72 73 78 98 ...
$ 3   : int 66 63 193 69 91 64 72 73 78 125 ...
$ 4   : int 66 63 103 67 91 64 72 73 78 91 ...
$ 5   : int 66 63 103 70 90 64 72 73 78 79 ...
$ 6   : int 66 63 103 73 90 64 72 73 78 77 ...
$ 7   : int 66 63 101 70 87 64 5 73 73 81 ...
$ 8   : int 66 63 108 69 83 64 85 73 70 82 ...
$ 9   : int 66 63 108 67 86 64 100 73 72 85 ...
$ 10   : int 66 63 106 67 85 64 101 73 75 85 ...
$ 11   : int 66 63 111 68 84 64 104 73 76 81 ...
$ 12   : int 66 63 122 69 90 56 100 73 78 82 ...
$ 13   : int 66 63 129 78 99 60 97 73 76 81 ...
$ 14   : int 115 63 112 78 97 78 96 73 78 80 ...
$ 15   : int 86 63 99 70 93 77 94 198 76 82 ...

```

### ### Replacing NAs

Instead of removing the rows with NaN values, we can replace all NaN values with a specific value that does not appear naturally in the input, such as -1.

The problem can be modeled as-is and we can encourage the model to learn that a specific value means "missing."

```

```{r}
abp_mean_reshaped_outsidecpb[is.na(abp_mean_reshaped_outsidecpb)]<-(-1)
```

```

### ### Entropy

```

```{r}
abp_mean_reshaped_outsidecpb['entropy']<-
round(apply(abp_mean_reshaped_outsidecpb[2:ncol(abp_mean_reshaped_outsidecpb)
], 1 , entropy ) , 2)
```

```

Ref: <https://machinelearningmastery.com/handle-missing-timesteps-sequence-prediction-problems-python/>

### ### Joining with other data-sets

```

```{r}
abp_mean_outsidecpb_all_data<-merge(abp_mean_reshaped_outsidecpb,
                                     FinalHemodynamicsDataset%>%
                                     dplyr::select( Case_Name
                                                    ,Gender
                                                    ,Age
                                                    ,Category
                                                    ,duration_Surgery_min
                                                    ,duration_CPB_min
                                                    ,duration_outsideCPB_min
                                                    ,STS_RiskAlgorithm
                                                    ,STS_RiskAlgorithm_multiple_imp

```

```

,STS_RiskAlgorithm_median_imp
,total_pressors_formula_mg

# categorical variables
,Q_total_pressors_formula_mg
,T_STS_RiskAlgorithm_median_imp
,T_STS_RiskAlgorithm_multiple_imp
,Q_AUC65MAP_total_mmHgmin
,Q_TWA65MAP_total_mmHg
,Q_AUC65MAP_CPB_mmHgmin
,Q_AUC65MAP_outsideCPB_mmHgmin
,Q_TWA65MAP_CPB_mmHg
,Q_TWA65MAP_outsideCPB_mmHg

# TWA variables
,TWA65MAP_total_mmHg
,TWA65MAP_CPB_mmHg
,TWA65MAP_outsideCPB_mmHg

,delta_HCT_pct_median_imp

,mean_ef_median_imp
,Cross_Clamp_Time_imp

,AUC65MAP_total_mmHgmin
,AUC65MAP_CPB_mmHgmin
,AUC65MAP_outsideCPB_mmHgmin

# Outcomes
,Any_MAE
,Renal_Failure
,Death
,Stroke
,RFDS
,RFD
,DS
,RFS

))

...

*Why cases number is not matching???*

### Removing columns where 100% is NA

```{r}
#abp_mean_outsidecpb_all_data <-
abp_mean_outsidecpb_all_data[,colSums(is.na(abp_mean_outsidecpb_all_data))<nr
ow(abp_mean_outsidecpb_all_data)]
```

Imputing missing values in mean_ef_median_imp

```

```

there were some missing values in mean_ef_median_imp, imputing
mean_ef_median_imp for 81 cases missing

```{r}
abp_mean_outsidecpb_all_data$mean_ef_median_imp[is.na(abp_mean_outsidecpb_all
_data$mean_ef_median_imp)]<-
round(mean(abp_mean_outsidecpb_all_data$mean_ef_median_imp,na.rm = T),2)
```

Mapping variables

We don't need all the variables in our dataset to be numbers anymore

```{r}
#abp_mean_all_data$Gender<-
plyr::revalue(abp_mean_all_data$Gender,c('Female'=1,'Male'=2))
#abp_mean_all_data$Category<-
plyr::revalue(abp_mean_all_data$Category,c('Aortic Surgery'=1,'CABG'=2,'CABG
+ Valve'=3,'Other'=3,'Valve'=4))
```

Generating report

```{r}
view(dfSummary(abp_mean_outsidecpb_all_data))
```

Exporting df

```{r}
#write.csv(abp_mean_outsidecpb_all_data,'abp_mean_outsidecpb_all_data.csv',ro
w.names = F)
```

```

## train\_test\_df\_creation\_outsideCPB

```

---
title: "03_train_test_df_creation"
author: "Miguel Armengol"
date: "2/20/2019"
output: html_notebook

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

Loading libraries

```{r message=FALSE, warning=FALSE}
library(dplyr)
library(keras)
library(tfruns)
library(caret)
```

```

```

Creating de-identified id

```{r}
abp_mean_outsidecpb_all_data['id']<-
seq.int(nrow(abp_mean_outsidecpb_all_data))
abp_mean_outsidecpb_all_data_deid<-abp_mean_outsidecpb_all_data
abp_mean_outsidecpb_all_data_deid$Case_Name<-NULL
```

Splitting de-identified data into testing and training, balanced version.

We want the data to be sampled randomly but always the same way and we want
to be sure that train and test must be balanced.

```{r}
## set the seed to make our partition reproducible
set.seed(123)
# createDataPartition: "the random sampling is done within the levels of y
when y is a factor in an attempt to balance the class distributions within
the splits."
## 75% of the sample size
train_outsidecpb_idx <-
createDataPartition(as.factor(abp_mean_outsidecpb_all_data_deid$Death), times
= 1, p = 0.75, list=F)

train_outsidecpb <- abp_mean_outsidecpb_all_data_deid[train_outsidecpb_idx, ]
test_outsidecpb <- abp_mean_outsidecpb_all_data_deid[-train_outsidecpb_idx, ]

round(prop.table(table(train_outsidecpb$Death)),2)
round(prop.table(table(test_outsidecpb$Death)),2)
```

Exporting datasets

```{r}
write.csv(train_outsidecpb,'../data/train_outsidecpb.csv',row.names = F)
write.csv(test_outsidecpb,'../data/test_outsidecpb.csv',row.names = F)
```

```

## STS Risk Performance

### Environment

In [19]:

```

import numpy as np
import pandas as pd
from numpy import median, percentile
from numpy.random import seed, randint

from sklearn import metrics
#from scipy.integrate import cumtrapz
from sklearn.metrics import (roc_auc_score, cohen_kappa_score, confusion_matrix, precision_score,
                             accuracy_score, average_precision_score, f1_score,
                             precision_recall_curve, roc_curve) #, auc, recall_score)
import matplotlib.pyplot as pyplot
from inspect import signature

```



```

import joblib
from sklearn.calibration import (CalibratedClassifierCV,
                                _CalibratedClassifier, calibration_curve)
from sklearn.pipeline import Pipeline, FeatureUnion
from sklearn.model_selection import RandomizedSearchCV
from sklearn.preprocessing import FunctionTransformer, MinMaxScaler
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import (make_scorer, roc_auc_score) #, cohen_kappa_score, confusion_matrix
, precision_score,
#
#
#
import dill
import sys
from xgboost import XGBClassifier

```

STS Threshold analysis  
Defining functions  
Specificity and Sensitivity

In [20]:

```

def spec(y_true,y_pred):
    TN, FP, FN, TP = confusion_matrix(y_true,y_pred).ravel()
    return TN/(TN+FP)

```

```

def sens(y_true,y_pred):
    TN, FP, FN, TP = confusion_matrix(y_true,y_pred).ravel()
    return TP/(TP+FN)

```

Optimal AUC threshold

In [21]:

```

def optimal_threshold_auc(target, predicted):

    fpr, tpr, threshold = roc_curve(target, predicted)
    i = np.arange(len(tpr))
    roc = pd.DataFrame({'tf' : pd.Series(tpr-(1-fpr), index=i), 'threshold' : pd.Series(thresh
old, index=i)})
    roc_t = roc.iloc[(roc.tf-0).abs().argsort()[:1]]
    return list(roc_t['threshold'])

```

AUROC

In [22]:

```

def auROC(y, ypred,title):
    fig = pyplot.figure(figsize=(8,5))
    ax = fig.add_subplot(111)
    fpr, tpr, thresholds = roc_curve(y, ypred)
    auc_score = roc_auc_score(y, ypred)

    pyplot.plot([0, 1], [0, 1], linestyle='--')
    pyplot.plot(fpr, tpr, marker='.',color='c')
    pyplot.xlabel('False positive rate')
    pyplot.ylabel('Recall')
    pyplot.ylim([0.0, 1.05])
    pyplot.xlim([0.0, 1.0])
    pyplot.title(title+' '+ 'AUROC={0:0.2f}'.format(auc_score))
    for item in ([ax.title, ax.xaxis.label, ax.yaxis.label] +
                  ax.get_xticklabels() + ax.get_yticklabels()):
        item.set_fontsize(20)

```

## AUPR

In [23]:

```
def aupr(y, ypred, title):
    fig = pyplot.figure(figsize=(8,5))
    ax = fig.add_subplot(111)

    Baseline = sum(y)/len(y)
    average_precision = metrics.average_precision_score(y, ypred)

    precision, recall, _ = precision_recall_curve(y, ypred)
    auc = metrics.auc(recall, precision)
    step_kwargs = ({'step': 'post'})
    if 'step' in signature(pyplot.fill_between).parameters
    else {}
    pyplot.step(recall, precision, color='c', where='post')
    pyplot.fill_between(recall, precision, color='c', **step_kwargs)
    pyplot.xlabel('Recall')
    pyplot.ylabel('Precision')
    pyplot.ylim([0.0, 1.05])
    pyplot.xlim([0.0, 1.0])
    pyplot.title(title+' '+ 'AUPRC={0:0.2f}'.format(auc))
    # pyplot.plot(Baseline, 'b-', label='Baseline')
    # pyplot.legend(bbox_to_anchor=(0.7, 0.9), loc=2, borderaxespad=0., fontsize = 16)
    for item in ([ax.title, ax.xaxis.label, ax.yaxis.label] +
                  ax.get_xticklabels() + ax.get_yticklabels()):
        item.set_fontsize(20)

def metric_aupr(y_true, y_pred):
    precision, recall, _ = precision_recall_curve(y_true, y_pred)
    auc = metrics.auc(recall, precision)
    return auc
```

## Bootstrapping function

In [24]:

```
def metric_SMR(y_true, y_pred):
    SMR = sum(y_pred)/sum(y_true)
    return SMR

def get_CI_boot(y_true, y_pred, metric, boot):
    # bootstrap confidence intervals
    # seed the random number generator
    seed(1)
    i = 0
    # generate dataset
    dataset = y_pred
    real = y_true
    # bootstrap
    scores = list()
    while i < boot:
        # bootstrap sample
        indices = randint(0, len(y_pred) - 1, len(y_pred))
        sample = dataset[indices]
        real = y_true[indices]
        if len(np.unique(y_true[indices])) < 2:
            # We need at least one positive and one negative sample for ROC AUC
            # to be defined: reject the sample
            continue
        # calculate and store statistic
    else:
```

```

        statistic = metric(real,sample)
        scores.append(statistic)
        i += 1

    # calculate 95% confidence intervals (100 - alpha)
    alpha = 5.0
    # calculate lower percentile (e.g. 2.5)
    lower_p = alpha / 2.0
    # retrieve observation at lower percentile
    lower = max(0.0, percentile(scores, lower_p))
    # calculate upper percentile (e.g. 97.5)
    upper_p = (100 - alpha) + (alpha / 2.0)
    # retrieve observation at upper percentile
    if metric == metric_SMR:
        upper = percentile(scores, upper_p)
    else:
        upper = min(1.0, percentile(scores, upper_p))
    delta = upper-lower
    mediana = median(scores)
    return mediana,(lower,upper),delta

```

Importing the STS_threshold train and test datasets

In [25]:

```

# Import
train_ststheshold = pd.read_csv("../../data/train.csv", delimiter=";", low_memory=False) # ../
../ go two folders back from current location

test_ststheshold = pd.read_csv("../../data/test.csv", delimiter=";", low_memory=False)

train_ststheshold = train_ststheshold.drop(columns = cols)
test_ststheshold = test_ststheshold.drop(columns = cols)

train_ststheshold = pd.DataFrame.dropna(train_ststheshold)
test_ststheshold = pd.DataFrame.dropna(test_ststheshold)

# Defining the ststheshold_five to consider a patient is likely to die

train_ststheshold_five = pd.DataFrame.dropna(train_ststheshold)
test_ststheshold_five = pd.DataFrame.dropna(test_ststheshold)

def f(row):
    if row['STS_RiskAlgorithm'] < 0.05 :
        val = 0
    else:
        val = 1
    return val

train_ststheshold_five['STS_threshold_pred'] = train_ststheshold_five.apply(f, axis=1)
test_ststheshold_five['STS_threshold_pred'] = test_ststheshold_five.apply(f, axis=1)

## Split data into x y training and x y test sets

y_train_ststheshold_five = train_ststheshold_five['Death']
X_train_ststheshold_five = train_ststheshold_five.drop(columns = ['Death'])

y_test_ststheshold_five = test_ststheshold_five['Death']
X_test_ststheshold_five = test_ststheshold_five.drop(columns = ['Death'])

# ALL

```

```

X_num_ststhreshold_five = X_train_ststhreshold_five

get_numeric_data_ststhreshold_five = FunctionTransformer(lambda x: x[X_num_ststhreshold_five.columns], validate=False)

get_categorical_ststhreshold_five = FunctionTransformer(lambda x: x[[]], validate=False)

scaler_ststhreshold_five = MinMaxScaler()

num_pipe_ststhreshold_five = Pipeline([
    ('select_num', get_numeric_data_ststhreshold_five),
    ('scale', scaler_ststhreshold_five)
])

cat_pipe_ststhreshold_five = Pipeline([
    ('select_cat', get_categorical_ststhreshold_five)
])

## Predictions in train and test

y_train_pred_ststhreshold_five = train_ststhreshold_five['STS_threshold_pred']
y_test_pred_ststhreshold_five = test_ststhreshold_five['STS_threshold_pred']
y_test_pred_ststhreshold_five = np.array(y_test_pred_ststhreshold_five)

# Defining the ststhreshold_eight to consider a patient is likely to die

train_ststhreshold_eight = pd.DataFrame.dropna(train_ststhreshold)
test_ststhreshold_eight = pd.DataFrame.dropna(test_ststhreshold)

def f(row):
    if row['STS_RiskAlgorithm'] < 0.08 :
        val = 0
    else:
        val = 1
    return val

train_ststhreshold_eight['STS_threshold_pred'] = train_ststhreshold_eight.apply(f, axis=1)
test_ststhreshold_eight['STS_threshold_pred'] = test_ststhreshold_eight.apply(f, axis=1)

## Split data into x y training and x y test sets

y_train_ststhreshold_eight = train_ststhreshold_eight['Death']
X_train_ststhreshold_eight = train_ststhreshold_eight.drop(columns = ['Death'])

y_test_ststhreshold_eight = test_ststhreshold_eight['Death']
X_test_ststhreshold_eight = test_ststhreshold_eight.drop(columns = ['Death'])

# All

X_num_ststhreshold_eight = X_train_ststhreshold_eight

get_numeric_data_ststhreshold_eight = FunctionTransformer(lambda x: x[X_num_ststhreshold_eight.columns], validate=False)

get_categorical_ststhreshold_eight = FunctionTransformer(lambda x: x[[]], validate=False)

scaler_ststhreshold_eight = MinMaxScaler()

num_pipe_ststhreshold_eight = Pipeline([
    ('select_num', get_numeric_data_ststhreshold_eight),
    ('scale', scaler_ststhreshold_eight)
])

```

```

    ])

cat_pipe_ststhreshold_eight = Pipeline([
    ('select_cat', get_categorical_ststhreshold_eight)
])

## Predictions in train and test

y_train_pred_ststhreshold_eight = train_ststhreshold_eight['STS_threshold_pred']
y_test_pred_ststhreshold_eight = test_ststhreshold_eight['STS_threshold_pred']
y_test_pred_ststhreshold_eight = np.array(y_test_pred_ststhreshold_eight)

# Defining the ststhreshold_eleven to consider a patient is likely to die

train_ststhreshold_eleven = pd.DataFrame.dropna(train_ststhreshold)
test_ststhreshold_eleven = pd.DataFrame.dropna(test_ststhreshold)

def f(row):
    if row['STS_RiskAlgorithm'] < 0.11 :
        val = 0
    else:
        val = 1
    return val

train_ststhreshold_eleven['STS_threshold_pred'] = train_ststhreshold_eleven.apply(f, axis=1)
test_ststhreshold_eleven['STS_threshold_pred'] = test_ststhreshold_eleven.apply(f, axis=1)

## Split data into x y training and x y test sets

y_train_ststhreshold_eleven = train_ststhreshold_eleven['Death']
X_train_ststhreshold_eleven = train_ststhreshold_eleven.drop(columns = ['Death'])

y_test_ststhreshold_eleven = test_ststhreshold_eleven['Death']
X_test_ststhreshold_eleven = test_ststhreshold_eleven.drop(columns = ['Death'])

# All

X_num_ststhreshold_eleven = X_train_ststhreshold_eleven

get_numeric_data_ststhreshold_eleven = FunctionTransformer(lambda x: x[X_num_ststhreshold_eleven.columns], validate=False)

get_categorical_ststhreshold_eleven = FunctionTransformer(lambda x: x[[]], validate=False)

scaler_ststhreshold_eleven = MinMaxScaler()

num_pipe_ststhreshold_eleven = Pipeline([
    ('select_num', get_numeric_data_ststhreshold_eleven),
    ('scale', scaler_ststhreshold_eleven)
])

cat_pipe_ststhreshold_eleven = Pipeline([
    ('select_cat', get_categorical_ststhreshold_eleven)
])

## Predictions in train and test

y_train_pred_ststhreshold_eleven = train_ststhreshold_eleven['STS_threshold_pred']
y_test_pred_ststhreshold_eleven = test_ststhreshold_eleven['STS_threshold_pred']
y_test_pred_ststhreshold_eleven = np.array(y_test_pred_ststhreshold_eleven)

```

In [26]:

```
print(train_ststhreshold.shape[0]+test_ststhreshold.shape[0])
4107
```

In [27]:

```
print(list(train_ststhreshold.columns))
['STS_RiskAlgorithm', 'Death']
```

In [28]:

```
## Optimal AUC threshold
```

In [29]:

```
def optimal_threshold_auc(target, predicted):
    fpr, tpr, threshold = roc_curve(target, predicted)
    i = np.arange(len(tpr))
    roc = pd.DataFrame({'tf' : pd.Series(tpr-(1-fpr), index=i), 'threshold' : pd.Series(thresh
old, index=i)})
    roc_t = roc.iloc[(roc.tf-0).abs().argsort()[:1]]
    return list(roc_t['threshold'])
```

AUROC STS thresold dataset vs preCPB dataset

In [ ]:

```
# model1 (random forest) with all preCPB data
#model1 = bestmodels[1] # select the best performing model here
#y_train_pred_model1 = model1.predict_proba(X_train)[:,:1]
#y_test_pred_model1 = model1.predict_proba(X_test)[:,:1]
#fpr_model1, tpr_model1, thresholds_model1 = roc_curve(y_test, y_test_pred_model1)
#auc_score_model1 = roc_auc_score(y_test, y_test_pred_model1)

# predictions using ststhreshold_five data
y_train_pred_ststhreshold_five = train_ststhreshold_five['STS_threshold_pred']
y_test_pred_ststhreshold_five = test_ststhreshold_five['STS_threshold_pred']
y_test_pred_ststhreshold_five = np.array(y_test_pred_ststhreshold_five)
fpr_ststhreshold_five, tpr_ststhreshold_five, thresholds = roc_curve(y_test_ststhreshold_five,
    y_test_pred_ststhreshold_five)
auc_score_ststhreshold_five = roc_auc_score(y_test_ststhreshold_five, y_test_pred_ststhreshold
_five)

my_dpi=100
pyplot.figure(figsize=(1200/my_dpi,1000/my_dpi), dpi=my_dpi)
pyplot.plot([0, 1], [0, 1], linestyle='--', color='#bdc3c7')

# AUROC postCPB model1
#pyplot.plot(fpr_model1, tpr_model1, marker='',color='#3498db',label='pre CPB RF AUC:'+ '{0
:0.2f}'.format(auc_score_model1))

## AUROC STS Risk Score
#AUC ststhreshold_five
pyplot.plot(fpr_ststhreshold_five, tpr_ststhreshold_five, marker='',color='#c0392b',label='STS
    Risk Score AUC:'+ '{0:0.2f}'.format(auc_score_ststhreshold_five))

pyplot.legend(loc='lower right')
pyplot.xlabel('Specificity')
pyplot.ylabel('Sensitivity')
pyplot.ylim([0.0, 1.05])
pyplot.xlim([0.0, 1.0])
```

In [ ]:

```
bestmodels[1]
```

Insights STS model

In [ ]:

```

for i,model in enumerate(bestmodels):
    modelname = all_models_names[i]
    print(modelname)
    y_train_pred = model.predict_proba(X_train)[:,1]
    y_test_pred = model.predict_proba(X_test)[:,1]
    # Threshold in train
    threshold = optimal_threshold_auc(y_train, y_train_pred)

    # performance_all = [roc_auc_score,average_precision_score, accuracy_score, spec,
    #                     sens, precision_score, f1_score, cohen_kappa_score,metric_SMR,metric_aupr]

    #
    # performance_names_all = np.array(['roc_auc_score', 'average_precision_score'
    #                                   , ' accuracy_score', ' spec', 'sens', ' precision_score'
    #                                   , ' f1_score', ' cohen_kappa_score', 'metric_SMR', 'metric_aupr
    # ])
    #
    performance = [roc_auc_score,average_precision_score, spec, sens]

    performance_names = np.array(['roc_auc_score', 'average_precision_score', 'spec', 'sens'])

    conf = []
    delt = []
    met = []
    med = []
    te = []
    performance_insights = []

    y_test = np.array(y_test)

    for i,p in enumerate(performance):

        if (p == roc_auc_score or p == average_precision_score or p == metric_aupr):
            mediana,extremes,delta = get_CI_boot(y_test,y_test_pred,p,boot=100) #atencao aqui
ao boot
            met.append(p)
            conf.append(extremes)
            delt.append(delta)
            med.append(mediana)
            te.append(p(y_test,y_test_pred))
            performance_insights.append(performance_names[i]+' '+str(round(p(y_test,y_test_pre
d), 2)) + ' CI [' + str(round(extremes[0], 2)) + '-'+ str(round(extremes[1], 2)) +']')
        else:
            y_pred_threshold = (y_test_pred >= threshold).astype(np.int)
            mediana,extremes,delta = get_CI_boot(y_test,y_pred_threshold,p,boot=100)
            met.append(p)
            conf.append(extremes)
            delt.append(delta)
            med.append(mediana)
            te.append(p(y_test,y_pred_threshold))
            performance_insights.append(performance_names[i]+' '+str(round(p(y_test,y_pred_thr
eshold), 2)) + ' CI [' + str(round(extremes[0], 2)) + '-'+ str(round(extremes[1], 2)) +']')

    performance_insights = str(performance_insights)
    performance_insights = performance_insights.replace("'", "")
    print(performance_insights)

```

AUPR STS threshold dataset vs preCPB dataset

STS threshold Insights

In [41]:

```
# predictions using ststhreshold_five data
y_train_pred_ststhreshold_five = train_ststhreshold_five['STS_threshold_pred']
y_test_pred_ststhreshold_five = test_ststhreshold_five['STS_threshold_pred']
y_test_pred_ststhreshold_five = np.array(y_test_pred_ststhreshold_five)
fpr_ststhreshold_five, tpr_ststhreshold_five, thresholds = roc_curve(y_test_ststhreshold_five,
    y_test_pred_ststhreshold_five)
auc_score_ststhreshold_five = roc_auc_score(y_test_ststhreshold_five, y_test_pred_ststhreshold
_five)

# Threshold in train
threshold = optimal_threshold_auc(y_train_ststhreshold_five, y_train_pred_ststhreshold_five)

performance = [roc_auc_score, average_precision_score, accuracy_score, spec,
    sens, precision_score, f1_score, cohen_kappa_score, metric_SMR, metric_aupr]

performance_names = np.array(['ROC AUC', 'Average Precision', 'Accuracy'
    , 'Specificity', 'Sensitivity', 'Precision'
    , 'F1', 'Cohen Kappa', 'SMR', 'AUPROC'])

conf = []
delt = []
met = []
med = []
te = []
performance_insights = []

y_test_ststhreshold_five = np.array(y_test_ststhreshold_five)

for i,p in enumerate(performance):

    if (p == roc_auc_score or p == average_precision_score or p == metric_aupr):
        mediana, extremes, delta = get_CI_boot(y_test_ststhreshold_five, y_test_pred_ststhreshold
_five, p, boot=100) #atencao aqui ao boot
        met.append(p)
        conf.append(extremes)
        delt.append(delta)
        med.append(mediana)
        te.append(p(y_test_ststhreshold_five, y_test_pred_ststhreshold_five))
        performance_insights.append(
            performance_names[i]+
            ' '+str(round(p(y_test_ststhreshold_five, y_test_pred_ststhreshold_five), 2))
            +' CI [' + str(round(extremes[0], 2)) + '-' + str(round(extremes[1], 2)) + ']'
        )
    else:
        y_pred_threshold = (y_test_pred_ststhreshold_five >= threshold).astype(np.int)
        mediana, extremes, delta = get_CI_boot(y_test_ststhreshold_five, y_pred_threshold, p, boot=
100)

        met.append(p)
        conf.append(extremes)
        delt.append(delta)
        med.append(mediana)
        te.append(p(y_test_ststhreshold_five, y_pred_threshold))
        performance_insights.append(
            performance_names[i]+
```



```

        '+' + str(round(p(y_test_ststhreshold_five, y_pred_threshold), 2))
        + ' CI [' + str(round(extremes[0], 2)) + '-' + str(round(extremes[1], 2)) + ']'
    )

performance_insights = str(performance_insights)
performance_insights = performance_insights.replace("'", "")
#print(model)
print(performance_insights)
[ROC AUC 0.72 CI [0.61-0.82], Average Precision 0.05 CI [0.02-0.09], Accuracy 0.88 CI [0.86-0.
9], Specificity 0.89 CI [0.87-0.9], Sensitivity 0.56 CI [0.33-0.74], Precision 0.08 CI [0.04-0
.12], F1 0.14 CI [0.07-0.2], Cohen Kappa 0.11 CI [0.05-0.17], SMR 6.83 CI [4.88-11.99], AUPROC
0.32 CI [0.2-0.44]]

```

Calibration

In [ ]:

```

for i, model in enumerate(bestmodels):
    # Add title and axis names
    pyplot.title('Calibration plot (reliability curve)')
    pyplot.xlabel('Mean predicted value')
    pyplot.ylabel('Fraction of positives')

    model.fit(X_train, y_train)
    #y_pred = (model.predict_proba(X_test)[: ,1] >= threshold).astype(np.int)
    prob_pos = model.predict_proba(X_test)[: ,1] #).astype(np.int)
    fraction_of_positives, mean_predicted_value = \
        calibration_curve(y_test, prob_pos, n_bins=10, normalize=True)
    pyplot.plot(mean_predicted_value, fraction_of_positives, 'o-c', label="model_"+all_models_n
ames[i])

    pyplot.title('Calibration plot (reliability curve)')
    pyplot.xlabel('Mean predicted value')
    pyplot.ylabel('Fraction of positives')

    model_isotonic = _CalibratedClassifier(model, method='isotonic')
    model_isotonic.fit(X_train, y_train)
    calibrated = model_isotonic
    # predict probabilities
    probs = calibrated.predict_proba(X_test)[: , 1]
    # reliability diagram
    fop, mpv = calibration_curve(y_test, probs, n_bins=10, normalize=True)
    # plot calibrated reliability
    pyplot.plot(mpv, fop, marker='.', label="Isotonic")

    model_isotonic = _CalibratedClassifier(model, method='sigmoid')
    model_isotonic.fit(X_train, y_train)
    calibrated = model_isotonic
    # predict probabilities
    probs = calibrated.predict_proba(X_test)[: , 1]
    # reliability diagram
    fop, mpv = calibration_curve(y_test, probs, n_bins=10, normalize=True)
    pyplot.plot([0, 1], [0, 1], 'k--', label='Perfect calibration')
    # plot calibrated reliability
    pyplot.plot(mpv, fop, marker='.', label="sigmoid")

    pyplot.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)

```

Voting classifier

In [ ]:

```
# voting classifier 2 modalities
# all models
# ('lr', lr), ('rf', rf), ('nn', nn), ('svm', svm), ('xgb', xgb)
clf_soft = VotingClassifier(estimators=[
    ('lr', lr), ('rf', rf)], voting='soft')

clf_hard = VotingClassifier(estimators=[
    ('lr', lr), ('rf', rf)], voting='hard')

clf = clf_soft.fit(X_train, y_train)
```

Features Importance

Random forest uses gini importance or mean decrease in impurity (MDI) to calculate the importance of each feature. Gini importance is also known as the total decrease in node impurity. This is how much the model fit or accuracy decreases when you drop a variable. The larger the decrease, the more significant the variable is. Here, the mean decrease is a significant parameter for variable selection. The Gini index can describe the overall explanatory power of the variables.

In [38]:

```
cols_names = list(X_train)

# LR or LinearSVC
#coef = clf.best_estimator_.steps[1][1].coef_
#coef = clf.steps[1][1].coef_

# RF
coef = bestmodels[1].steps[1][1].feature_importances_
coef = np.matrix(coef)
feature_importance = abs(coef)
#intercept = clf.best_estimator_.steps[1][1].intercept_

feature_importance = 100.0 * (feature_importance / feature_importance.max())
sorted_idx = np.argsort(np.transpose(feature_importance))
pos = np.arange(sorted_idx.shape[0]) + .5

#np.column_stack([cols_names,list(feature_importance)])

#pd.dataframe.feature_importance

feature_importance_table = pd.DataFrame(feature_importance, columns=cols_names).T # transposing
feature_importance_table.columns = ['Importance']

feature_importance_table.to_csv(path_or_buf='feature_importance_table.csv')

print(feature_importance_table)

#featfig = pyplot.figure(figsize=(15, 15))
#featax = featfig.add_subplot(1, 1, 1)
#featax.barh(pos, feature_importance[sorted_idx], align='center')
#featax.set_yticks(pos)
#featax.set_yticklabels(np.array(cols_names)[sorted_idx], fontsize=18)
#featax.set_xlabel('Relative Feature Importance', fontsize=18)
#
#
#pyplot.tight_layout()
#pyplot.show()
```

training_models_cpb

Environment

In [2]:

```
import numpy as np
import pandas as pd
from numpy import median, percentile
from numpy.random import seed, randint

from sklearn import metrics
#from scipy.integrate import cumtrapz
from sklearn.metrics import (roc_auc_score, cohen_kappa_score, confusion_matrix, precision_score,
                             accuracy_score, average_precision_score, f1_score,
                             precision_recall_curve, roc_curve) #, auc, recall_score)

import matplotlib.pyplot as pyplot
from inspect import signature
import joblib
from sklearn.calibration import (CalibratedClassifierCV,
                                 _CalibratedClassifier, calibration_curve)

from sklearn.pipeline import Pipeline, FeatureUnion
from sklearn.model_selection import RandomizedSearchCV
from sklearn.preprocessing import FunctionTransformer, MinMaxScaler
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import (make_scorer, roc_auc_score) #, cohen_kappa_score, confusion_matrix, precision_score,
#
#                             accuracy_score, auc, average_precision_score, f1_score,
#                             precision_recall_curve, recall_score, roc_curve)

import dill
import sys
from xgboost import XGBClassifier

IMPORT DATA
```

In [3]:

```
train = pd.read_csv("../data/train_cpb.csv", delimiter=",", low_memory=False) # ..
../ go two folders back from current location

test = pd.read_csv("../data/test_cpb.csv", delimiter=",", low_memory=False)

cols = ['entropy', 'duration_Surgery_min', 'duration_outsideCPB_min',
        'STS_RiskAlgorithm_median_imp', 'STS_RiskAlgorithm',
        'Q_total_pressors_formula_mg',
        'T_STS_RiskAlgorithm_median_imp', 'T_STS_RiskAlgorithm_multiple_imp',
        'Q_AUC65MAP_total_mmHgmin', 'Q_TWA65MAP_total_mmHg',
        'Q_AUC65MAP_CPB_mmHgmin', 'Q_AUC65MAP_CPB_mmHgmin',
        'Q_TWA65MAP_CPB_mmHg', 'Q_TWA65MAP_CPB_mmHg',
        'Q_AUC65MAP_outsideCPB_mmHgmin', 'Q_TWA65MAP_outsideCPB_mmHg',
        'TWA65MAP_CPB_mmHg',
```

```

        'TWA65MAP_total_mmHg', 'TWA65MAP_outsideCPB_mmHg',
        'AUC65MAP_total_mmHgmin', 'AUC65MAP_outsideCPB_mmHgmin'
    ,
        'Any_MAE', 'Stroke', 'RFDS', 'RFD', 'DS', 'RFS', 'id']

train = train.drop(columns = cols)
test = test.drop(columns = cols)

# Converting categorical variables to numeric

train.Gender.loc[train.Gender == 'Male'] = 0
train.Gender.loc[train.Gender == 'Female'] = 1

train.Category.loc[train.Category == 'Valve'] = 0
train.Category.loc[train.Category == 'CABG'] = 1
train.Category.loc[train.Category == 'CABG + Valve'] = 2
train.Category.loc[train.Category == 'Aortic Surgery'] = 3
train.Category.loc[train.Category == 'Other'] = 4

test.Gender.loc[test.Gender == 'Male'] = 0
test.Gender.loc[test.Gender == 'Female'] = 1

test.Category.loc[test.Category == 'Valve'] = 0
test.Category.loc[test.Category == 'CABG'] = 1
test.Category.loc[test.Category == 'CABG + Valve'] = 2
test.Category.loc[test.Category == 'Aortic Surgery'] = 3
test.Category.loc[test.Category == 'Other'] = 4

# since we are not competing again STS risk score now I did not exclude patients with
no STS risk score but I impute the missing values.

# Removing rows based on data availability
#train = train[train['STS_RiskAlgorithm'].notna()]
#test = test[test['STS_RiskAlgorithm'].notna()]

C:\Users\miguel\AppData\Local\Continuum\anaconda3\lib\site-packages\pandas\core\index
ing.py:205: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
    self._setitem_with_indexer(indexer, value)

In [4]:
print(train.shape[0]+test.shape[0])
Split data into x y training and x y test sets

In [5]:
y_train = train['Death']
X_train = train.drop(columns = ['Death'])

y_test = test['Death']
X_test = test.drop(columns = ['Death'])

# ALL

```

```
X_num = X_train.drop(columns = ['Gender', 'Category', 'Renal_Failure'])

get_numeric_data = FunctionTransformer(lambda x: x[X_num.columns], validate=False)

get_categorical = FunctionTransformer(lambda x: x[['Gender', 'Category', 'Renal_Failure']], validate=False)

scaler = MinMaxScaler()
```

Modeling

In [6]:

```
# Logistic Regression
lr = LogisticRegression(random_state = 42, tol=1**-4, verbose=1) #, solver='lbfgs') #, class_weight='balanced')

# Random forests
rf = RandomForestClassifier(random_state = 42
                           #, n_jobs = -1 # set The number of jobs to run in parallel
                           # -1 means using all processors
                           )

# Neural Network
nn = MLPClassifier(random_state = 42, max_iter=500)

# SVM
svm = SVC(random_state = 42, probability=True) # Whether to enable probability estimates. This must be enabled prior to calling fit, and will slow down that method.

# XGB
xgb = XGBClassifier(random_state = 42)

all_models_list = [lr,
                  rf,
                  nn,
                  svm,
                  xgb]

all_models_names = ['lr',
                  'rf',
                  'nn',
                  'svm',
                  'xgb']

#all_models_list = [rf]

#all_models_names = ['rf']
```

Pipelines

These are the parameters we are going to pass to each model in the hyper-parameter tuning.

In [7]:

```
num_pipe = Pipeline([
```

```

    ('select_num', get_numeric_data),
    ('scale', scaler)
])

cat_pipe = Pipeline([
    ('select_cat', get_categorical)
])

# Logistic Regression
lr_params = { 'clf__C': [0.0001, 0.001, 0.01, 0.1, 1.0]
              , 'clf__solver': ['newton-cg', 'lbfgs', 'sag', 'saga']
              , 'clf__warm_start': [True, False]
            }

# Random forests
rf_params = { 'clf__n_estimators': [1000, 1500],
              'clf__max_features': ['sqrt'],
              'clf__max_depth': [2, 3, 4, 5],
              'clf__bootstrap': [True, False],
              'clf__criterion': ["gini"],
              'clf__warm_start': [True, False]
              #, 'clf__oob_score': [True, False]
            }

# Neural Network
nn_params = { 'clf__hidden_layer_sizes': [(10, 10, 10), (50, 100, 50), (100, 20)],
              'clf__activation': ['tanh', 'relu'],
              'clf__solver': ['sgd', 'adam'],
              'clf__alpha': [0.0001, 0.05],
              'clf__learning_rate': ['constant', 'adaptive']
            }

# SVM
svm_params = { 'clf__C': [1, 10, 100, 1000],
               'clf__gamma': [0.001, 0.0001],
               'clf__kernel': ['rbf', 'linear']
            }

# XGB
xgb_params = { 'clf__max_depth': [1, 3, 100],
               'clf__learning_rate': [0.1],
               'clf__n_estimators': [100, 500, 1000],
               'clf__silent': [True],
               'clf__objective': ['binary:logistic'],
               'clf__booster': ['gbtree'],
               'clf__n_jobs': [1],
               'clf__nthread': [None],
               'clf__gamma': [0],
               'clf__min_child_weight': [1],
               'clf__max_delta_step': [0],
               'clf__subsample': [1],
               'clf__colsample_bytree': [1],
               'clf__colsample_bylevel': [1],
               'clf__reg_alpha': [0, 0.0001, 0.05, 0.5],
               'clf__reg_lambda': [0, 1],
               'clf__scale_pos_weight': [1, 53.22], # Controls the balance of positive
               and negative weights, useful for unbalanced classes. A typical value to consider: su
               mnegative instances / sumpositive instances
            }

```

```

        'clf__base_score': [0.5],
        'clf__random_state': [0],
        'clf__seed': [None]}

all_params_list = [lr_params,
                   rf_params,
                   nn_params,
                   svm_params,
                   xgb_params
                   ]

#all_params_list = [rf_params
#
#                   ]

Scoring
In [8]:
bestmodels = [] # we create an empty list to store the best performing version of each model

if len(all_models_list) == len(all_params_list):
    for i,model in enumerate(all_models_list):
        print(type(all_models_list[i]))
        full_pipeline = Pipeline([('feat_union',
                                   FeatureUnion(transformer_list=[('cat_pipeline', cat_pipe),
                                                                    ('num_pipeline', num_pipe)])),
                                  ('clf', model)])

        scoring = make_scorer(roc_auc_score, needs_threshold = True)

        random_search = RandomizedSearchCV(full_pipeline, param_distributions=all_params_list[i], n_iter=10, cv=10, refit = True, scoring=scoring, n_jobs=-1, verbose=10, random_state = 42)

        sys.setrecursionlimit(10000)
        random_search.fit(X_train, y_train)

        # Best model in training
        bestmodels.append(random_search.best_estimator_)

else:
    print('Make sure the amount of models in all_models_list matches the amount of parameters in all_params_list')

<class 'sklearn.linear_model.logistic.LogisticRegression'>
Fitting 10 folds for each of 10 candidates, totalling 100 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 16 concurrent workers.
[Parallel(n_jobs=-1)]: Done   9 tasks      | elapsed:    5.5s
[Parallel(n_jobs=-1)]: Done  18 tasks      | elapsed:    9.5s
[Parallel(n_jobs=-1)]: Done  29 tasks      | elapsed:   12.7s
[Parallel(n_jobs=-1)]: Done  40 tasks      | elapsed:   15.0s
[Parallel(n_jobs=-1)]: Done  53 tasks      | elapsed:   16.5s
[Parallel(n_jobs=-1)]: Done  66 tasks      | elapsed:   17.8s
[Parallel(n_jobs=-1)]: Done  80 out of 100 | elapsed:   21.0s remaining:    5.2s
[Parallel(n_jobs=-1)]: Done  91 out of 100 | elapsed:   22.3s remaining:    2.1s
[Parallel(n_jobs=-1)]: Done 100 out of 100 | elapsed:   22.6s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.

```

```

[Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 0.3s finished
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 16 concurrent workers.
<class 'sklearn.ensemble forest.RandomForestClassifier'>
Fitting 10 folds for each of 10 candidates, totalling 100 fits
[Parallel(n_jobs=-1)]: Done 9 tasks | elapsed: 16.6s
[Parallel(n_jobs=-1)]: Done 18 tasks | elapsed: 25.8s
[Parallel(n_jobs=-1)]: Done 29 tasks | elapsed: 32.6s
[Parallel(n_jobs=-1)]: Done 40 tasks | elapsed: 38.9s
[Parallel(n_jobs=-1)]: Done 53 tasks | elapsed: 47.3s
[Parallel(n_jobs=-1)]: Done 66 tasks | elapsed: 1.1min
[Parallel(n_jobs=-1)]: Done 80 out of 100 | elapsed: 1.3min remaining: 19.1s
[Parallel(n_jobs=-1)]: Done 91 out of 100 | elapsed: 1.4min remaining: 7.9s
[Parallel(n_jobs=-1)]: Done 100 out of 100 | elapsed: 1.4min finished
<class 'sklearn.neural_network.multilayer_perceptron.MLPClassifier'>
Fitting 10 folds for each of 10 candidates, totalling 100 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 16 concurrent workers.
[Parallel(n_jobs=-1)]: Done 9 tasks | elapsed: 1.1min
[Parallel(n_jobs=-1)]: Done 18 tasks | elapsed: 1.4min
[Parallel(n_jobs=-1)]: Done 29 tasks | elapsed: 1.8min
[Parallel(n_jobs=-1)]: Done 40 tasks | elapsed: 2.3min
[Parallel(n_jobs=-1)]: Done 53 tasks | elapsed: 3.2min
[Parallel(n_jobs=-1)]: Done 66 tasks | elapsed: 3.6min
[Parallel(n_jobs=-1)]: Done 80 out of 100 | elapsed: 5.1min remaining: 1.3min
[Parallel(n_jobs=-1)]: Done 91 out of 100 | elapsed: 5.7min remaining: 33.7s
[Parallel(n_jobs=-1)]: Done 100 out of 100 | elapsed: 6.1min finished
<class 'sklearn.svm.classes.SVC'>
Fitting 10 folds for each of 10 candidates, totalling 100 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 16 concurrent workers.
[Parallel(n_jobs=-1)]: Done 9 tasks | elapsed: 1.2min
[Parallel(n_jobs=-1)]: Done 18 tasks | elapsed: 2.2min
[Parallel(n_jobs=-1)]: Done 29 tasks | elapsed: 2.8min
[Parallel(n_jobs=-1)]: Done 40 tasks | elapsed: 3.8min
[Parallel(n_jobs=-1)]: Done 53 tasks | elapsed: 5.1min
[Parallel(n_jobs=-1)]: Done 66 tasks | elapsed: 6.0min
[Parallel(n_jobs=-1)]: Done 80 out of 100 | elapsed: 7.2min remaining: 1.8min
[Parallel(n_jobs=-1)]: Done 91 out of 100 | elapsed: 8.1min remaining: 48.2s
[Parallel(n_jobs=-1)]: Done 100 out of 100 | elapsed: 8.3min finished
<class 'xgboost.sklearn.XGBClassifier'>
Fitting 10 folds for each of 10 candidates, totalling 100 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 16 concurrent workers.
C:\Users\miguel\AppData\Local\Continuum\anaconda3\lib\site-packages\joblib\externals\
loky\process_executor.py:706: UserWarning: A worker stopped while some jobs were give
n to the executor. This can be caused by a too short worker timeout or by a memory le
ak.
"timeout or by a memory leak.", UserWarning
[Parallel(n_jobs=-1)]: Done 9 tasks | elapsed: 4.8min
[Parallel(n_jobs=-1)]: Done 18 tasks | elapsed: 5.5min
[Parallel(n_jobs=-1)]: Done 29 tasks | elapsed: 10.5min
[Parallel(n_jobs=-1)]: Done 40 tasks | elapsed: 11.8min
[Parallel(n_jobs=-1)]: Done 53 tasks | elapsed: 13.8min
[Parallel(n_jobs=-1)]: Done 66 tasks | elapsed: 19.8min
[Parallel(n_jobs=-1)]: Done 80 out of 100 | elapsed: 24.5min remaining: 6.1min
[Parallel(n_jobs=-1)]: Done 91 out of 100 | elapsed: 27.3min remaining: 2.7min
[Parallel(n_jobs=-1)]: Done 100 out of 100 | elapsed: 28.1min finished

```


Defining functions

Specificity and Sensitivity

In [9]:

```
def spec(y_true,y_pred):
    TN, FP, FN, TP = confusion_matrix(y_true,y_pred).ravel()
    return TN/(TN+FP)

def sens(y_true,y_pred):
    TN, FP, FN, TP = confusion_matrix(y_true,y_pred).ravel()
    return TP/(TP+FN)
```

Optimal AUC threshold

In [10]:

```
def optimal_threshold_auc(target, predicted):

    fpr, tpr, threshold = roc_curve(target, predicted)
    i = np.arange(len(tpr))
    roc = pd.DataFrame({'tf' : pd.Series(tpr-(1-fpr), index=i), 'threshold' : pd.Series(threshold, index=i)})
    roc_t = roc.iloc[(roc.tf-0).abs().argsort()[:1]]
    return list(roc_t['threshold'])
```

AUROC

In [11]:

```
def auroc(y, ypred,title):
    fig = pyplot.figure(figsize=(8,5))
    ax = fig.add_subplot(111)
    fpr, tpr, thresholds = roc_curve(y, ypred)
    auc_score = roc_auc_score(y, ypred)

    pyplot.plot([0, 1], [0, 1], linestyle='--')
    pyplot.plot(fpr, tpr, marker='.',color='c')
    pyplot.xlabel('False positive rate')
    pyplot.ylabel('Recall')
    pyplot.ylim([0.0, 1.05])
    pyplot.xlim([0.0, 1.0])
    pyplot.title(title+' '+ 'AUROC={0:0.2f}'.format(auc_score))
    for item in ([ax.title, ax.xaxis.label, ax.yaxis.label] +
                  ax.get_xticklabels() + ax.get_yticklabels()):
        item.set_fontsize(20)
```

AUPR

In [12]:

```
def aupr(y, ypred,title):
    fig = pyplot.figure(figsize=(8,5))
    ax = fig.add_subplot(111)

    Baseline = sum(y)/len(y)
    average_precision = metrics.average_precision_score(y, ypred)

    precision, recall, _ = precision_recall_curve(y, ypred)
    auc = metrics.auc(recall, precision)
    step_kwargs = ({'step': 'post'})
    if 'step' in signature(pyplot.fill_between).parameters:
        else {}
```

```

pyplot.step(recall, precision, color='c', where='post')
pyplot.fill_between(recall, precision, color='c', **step_kwargs)
pyplot.xlabel('Recall')
pyplot.ylabel('Precision')
pyplot.ylim([0.0, 1.05])
pyplot.xlim([0.0, 1.0])
pyplot.title(title+' '+ 'AUPRC={0:0.2f}'.format(auc))
# pyplot.plot(Baseline, 'b-', label='Baseline')
# pyplot.legend(bbox_to_anchor=(0.7, 0.9), loc=2, borderaxespad=0., fontsize = 16)
for item in ([ax.title, ax.xaxis.label, ax.yaxis.label] +
             ax.get_xticklabels() + ax.get_yticklabels()):
    item.set_fontsize(20)

def metric_aupr(y_true, y_pred):
    precision, recall, _ = precision_recall_curve(y_true, y_pred)
    auc = metrics.auc(recall, precision)
    return auc

```

Bootstrapping function

In [13]:

```

def metric_SMR(y_true, y_pred):
    SMR = sum(y_pred)/sum(y_true)
    return SMR

def get_CI_boot(y_true,y_pred,metric,boot):
    # bootstrap confidence intervals
    # seed the random number generator
    seed(1)
    i = 0
    # generate dataset
    dataset = y_pred
    real = y_true
    # bootstrap
    scores = list()
    while i < boot:
        # bootstrap sample
        indices = randint(0, len(y_pred) - 1, len(y_pred))
        sample = dataset[indices]
        real = y_true[indices]
        if len(np.unique(y_true[indices])) < 2:
            # We need at least one positive and one negative sample for ROC AUC
            # to be defined: reject the sample
            continue
        # calculate and store statistic
        else:
            statistic = metric(real,sample)
            scores.append(statistic)
            i += 1

    # calculate 95% confidence intervals (100 - alpha)
    alpha = 5.0
    # calculate lower percentile (e.g. 2.5)
    lower_p = alpha / 2.0
    # retrieve observation at lower percentile

```

```

lower = max(0.0, percentile(scores, lower_p))
# calculate upper percentile (e.g. 97.5)
upper_p = (100 - alpha) + (alpha / 2.0)
# retrieve observation at upper percentile
if metric == metric_SMR:
    upper = percentile(scores, upper_p)
else:
    upper = min(1.0, percentile(scores, upper_p))
delta = upper-lower
mediana = median(scores)
return mediana,(lower,upper),delta

```

Predictions in train and test Insights, AUPR and AUROC curves

In [20]:

```

for i,model in enumerate(bestmodels):
    modelname = all_models_names[i]
    print(modelname)
    y_train_pred = model.predict_proba(X_train)[: ,1]
    y_test_pred = model.predict_proba(X_test)[: ,1]
    auROC(y_test, y_test_pred,all_models_names[i])
    aupr(y_test, y_test_pred,all_models_names[i])
    # Threshold in train
    threshold = optimal_threshold_auc(y_train, y_train_pred)

    # performance_all = [roc_auc_score,average_precision_score, accuracy_score, spec,
    #                     sens, precision_score, f1_score, cohen_kappa_score,metric_SMR,metric_aupr]
    #
    # performance_names_all = np.array(['roc_auc_score','average_precision_score'
    #                                   , ' accuracy_score', ' spec', 'sens', ' precision_score'
    #                                   , ' f1_score', ' cohen_kappa_score', 'metric_SMR', 'metric_aupr'])
    #
    performance = [roc_auc_score,precision_score, spec, sens]

    performance_names = np.array(['roc_auc_score','precision_score','spec','sens'])

    conf = []
    delt = []
    met = []
    med = []
    te = []
    performance_insights = []

    y_test = np.array(y_test)

    for i,p in enumerate(performance):

        if (p == roc_auc_score or p == average_precision_score or p == metric_aupr):
            mediana,extremes,delta = get_CI_boot(y_test,y_test_pred,p,boot=100) #atencao aqui ao boot
            met.append(p)

```

```

        conf.append(extremes)
        delt.append(delta)
        med.append(media)
        te.append(p(y_test,y_test_pred))
        performance_insights.append(performance_names[i]+' '+str(round(p(y_test,y
_test_pred), 2)) + ' CI [' + str(round(extremes[0], 2)) + '-' + str(round(extremes[1],
2)) + '])')
    else:
        y_pred_threshold = (y_test_pred >= threshold).astype(np.int)
        media,extremes,delta = get_CI_boot(y_test,y_pred_threshold,p,boot=100)
        met.append(p)
        conf.append(extremes)
        delt.append(delta)
        med.append(media)
        te.append(p(y_test,y_pred_threshold))
        performance_insights.append(performance_names[i]+' '+str(round(p(y_test,y
_pred_threshold), 2)) + ' CI [' + str(round(extremes[0], 2)) + '-' + str(round(extremes
[1], 2)) + '])')

    performance_insights = str(performance_insights)
    performance_insights = performance_insights.replace("'", "")
    print(performance_insights)

lr
[roc_auc_score 0.83 CI [0.72-0.9], precision_score 0.07 CI [0.04-0.1], spec 0.77 CI [
0.76-0.8], sens 0.75 CI [0.57-0.89]]
rf
[roc_auc_score 0.86 CI [0.76-0.92], precision_score 0.09 CI [0.05-0.13], spec 0.81 CI
[0.79-0.83], sens 0.79 CI [0.61-0.92]]
nn
[roc_auc_score 0.72 CI [0.59-0.85], precision_score 0.09 CI [0.04-0.13], spec 0.89 CI
[0.88-0.91], sens 0.43 CI [0.2-0.61]]
svm
[roc_auc_score 0.69 CI [0.57-0.8], precision_score 0.08 CI [0.04-0.11], spec 0.84 CI
[0.82-0.86], sens 0.57 CI [0.44-0.71]]
xgb
[roc_auc_score 0.8 CI [0.67-0.9], precision_score 0.09 CI [0.06-0.14], spec 0.83 CI [
0.81-0.85], sens 0.79 CI [0.61-0.91]]

```

Calibration

In [ ]:

```

for i,model in enumerate(bestmodels):
    # Add title and axis names
    pyplot.title('Calibration plot (reliability curve)')
    pyplot.xlabel('Mean predicted value')
    pyplot.ylabel('Fraction of positives')

    model.fit(X_train, y_train)
    #y_pred = (model.predict_proba(X_test)[:,-1] >= threshold).astype(np.int)
    prob_pos = model.predict_proba(X_test)[:,-1] #).astype(np.int)
    fraction_of_positives, mean_predicted_value = \
    calibration_curve(y_test, prob_pos, n_bins=10, normalize=True)
    pyplot.plot(mean_predicted_value, fraction_of_positives, 'o-c',label="model_"+all
_models_names[i])

```

```

pyplot.title('Calibration plot (reliability curve)')
pyplot.xlabel('Mean predicted value')
pyplot.ylabel('Fraction of positives')

model_isotonic = _CalibratedClassifier(model, method='isotonic')
model_isotonic.fit(X_train, y_train)
calibrated = model_isotonic
# predict probabilities
probs = calibrated.predict_proba(X_test)[:, 1]
# reliability diagram
fop, mpv = calibration_curve(y_test, probs, n_bins=10, normalize=True)
# plot calibrated reliability
pyplot.plot(mpv, fop, marker='.',label="Isotonic")

model_isotonic = _CalibratedClassifier(model, method='sigmoid')
model_isotonic.fit(X_train, y_train)
calibrated = model_isotonic
# predict probabilities
probs = calibrated.predict_proba(X_test)[:, 1]
# reliability diagram
fop, mpv = calibration_curve(y_test, probs, n_bins=10, normalize=True)
pyplot.plot([0, 1], [0, 1], 'k--',label='Perfect calibration')
# plot calibrated reliability
pyplot.plot(mpv, fop, marker='.',label="sigmoid")

pyplot.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)

```

Features Importance

In [ ]:

```

importances = clf.feature_importances_
std = np.std([tree.feature_importances_ for tree in clf.estimators_],
              axis=0)
indices = np.argsort(importances)[::-1]

# Print the feature ranking
print("Feature ranking:")

for f in range(X_train.shape[1]):
    print("%d. feature %d (%f)" % (f + 1, indices[f], importances[indices[f]]))

# Plot the feature importances of the forest
plt.figure()
plt.title("Feature importances")
plt.bar(range(X.shape[1]), importances[indices],
        color="r", yerr=std[indices], align="center")
plt.xticks(range(X.shape[1]), indices)
plt.xlim([-1, X.shape[1]])
plt.show()

```

Save results

In [18]:

```
db_name = 'OR'
```

```

# XGB
model_name = 'XGB'

features_name = 'cpb'

# Best model
filename = '{}_{}_{}_model.sav'.format(model_name, features_name, db_name)

with open(filename, 'wb') as pickle_file:
    dill.dump(random_search.best_estimator_, pickle_file)
    print('model saved')

# CV model
filename = '{}_{}_{}_model_CV.sav'.format(model_name, features_name, db_name)

with open(filename, 'wb') as pickle_file:
    dill.dump(random_search, pickle_file)
    print('model saved')

```

training_models_full

Environment

In [1]:

```
import numpy as np
import pandas as pd
from numpy import median, percentile
from numpy.random import seed, randint

from sklearn import metrics
#from scipy.integrate import cumtrapz
from sklearn.metrics import (roc_auc_score, cohen_kappa_score, confusion_matrix, precision_score,
                             accuracy_score, average_precision_score, f1_score,
                             precision_recall_curve, roc_curve) #, auc, recall_score)

import matplotlib.pyplot as pyplot
from inspect import signature
import joblib
from sklearn.calibration import (CalibratedClassifierCV,
                                 _CalibratedClassifier, calibration_curve)
from sklearn.pipeline import Pipeline, FeatureUnion
from sklearn.model_selection import RandomizedSearchCV
from sklearn.preprocessing import FunctionTransformer, MinMaxScaler
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import (make_scorer, roc_auc_score) #, cohen_kappa_score, confusion_matrix, precision_score,
#
#                             accuracy_score, auc, average_precision_score, f1_score,
#                             precision_recall_curve, recall_score, roc_curve)

import dill
import sys
from xgboost import XGBClassifier
```

Import Data

In [2]:

```
train = pd.read_csv("../data/train.csv", delimiter=",", low_memory=False) # ../data/
# go two folders back from current location

test = pd.read_csv("../data/test.csv", delimiter=",", low_memory=False)
cols = ['entropy', 'duration_CPB_min',
        'duration_outsideCPB_min', 'duration_preCPB_min', 'duration_postCPB_min',
        'STS_RiskAlgorithm_median_imp',
        'STS_RiskAlgorithm',
        'Q_total_pressors_formula_mg',
        'T_STS_RiskAlgorithm_median_imp',
        'T_STS_RiskAlgorithm_multiple_imp',
        'Q_AUC65MAP_outsideCPB_mmHgmin',
        'Q_TWA65MAP_outsideCPB_mmHg',
        'Q_AUC65MAP_total_mmHgmin',
```

```

        'Q_TWA65MAP_total_mmHg',
        'Q_AUC65MAP_CPB_mmHgmin',
        'Q_TWA65MAP_CPB_mmHg', 'TWA65MAP_outsideCPB_mmHg', 'TWA65MAP_CPB_mmHg
    ',
        'TWA65MAP_preCPB_mmHg', 'TWA65MAP_postCPB_mmHg', 'TWA65MAP_total_mmH
g',
        'AUC65MAP_preCPB_mmHgmin', 'AUC65MAP_postCPB_mmHgmin',
        'AUC65MAP_CPB_mmHgmin',
        'AUC65MAP_outsideCPB_mmHgmin',
        'Any_MAE', 'Stroke', 'RFDS', 'RFD', 'DS', 'RFS', 'id']

```

```

train = train.drop(columns = cols)
test = test.drop(columns = cols)

```

 # Converting categorical variables to numeric

```

train.Gender.loc[train.Gender == 'Male'] = 0
train.Gender.loc[train.Gender == 'Female'] = 1

```

```

train.Category.loc[train.Category == 'Valve'] = 0
train.Category.loc[train.Category == 'CABG'] = 1
train.Category.loc[train.Category == 'CABG + Valve'] = 2
train.Category.loc[train.Category == 'Aortic Surgery'] = 3
train.Category.loc[train.Category == 'Other'] = 4

```

```

test.Gender.loc[test.Gender == 'Male'] = 0
test.Gender.loc[test.Gender == 'Female'] = 1

```

```

test.Category.loc[test.Category == 'Valve'] = 0
test.Category.loc[test.Category == 'CABG'] = 1
test.Category.loc[test.Category == 'CABG + Valve'] = 2
test.Category.loc[test.Category == 'Aortic Surgery'] = 3
test.Category.loc[test.Category == 'Other'] = 4

```

 # since we are not competing again STS risk score now I did not exclude patients with no STS risk score but I impute the missing values.

 # Removing rows based on data availability

```
#train = train[train['STS_RiskAlgorithm'].notna()]
```

```
#test = test[test['STS_RiskAlgorithm'].notna()]
```

C:\Users\miguel\AppData\Local\Continuum\anaconda3\lib\site-packages\pandas\core\indexing.py:205: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
self._setitem_with_indexer(indexer, value)
```

In [3]:

```

print(train.shape[0]+test.shape[0])
print(list(train.columns))
5015

```

Split data into x y training and x y test sets

In [4]:


```

y_train = train['Death']
X_train = train.drop(columns = ['Death'])

y_test = test['Death']
X_test = test.drop(columns = ['Death'])

# ALL

X_num = X_train.drop(columns = ['Gender', 'Category', 'Renal_Failure'])

get_numeric_data = FunctionTransformer(lambda x: x[X_num.columns], validate=False)

get_categorical = FunctionTransformer(lambda x: x[['Gender', 'Category', 'Renal_Failure']], validate=False)

scaler = MinMaxScaler()

```

Modeling

In [5]:

```

# Logistic Regression
lr = LogisticRegression(random_state = 42, tol=1**-4, verbose=1) #, solver='lbfgs') #, class_weight='balanced')

# Random forests
rf = RandomForestClassifier(random_state = 42
                           #, n_jobs = -1 # set The number of jobs to run in parallel
                           -1 means using all processors
                           )

# Neural Network

nn = MLPClassifier(random_state = 42, max_iter=500)

# SVM

svm = SVC(random_state = 42, probability=True) # Whether to enable probability estimates. This must be enabled prior to calling fit, and will slow down that method.

# XGB

xgb = XGBClassifier(random_state = 42)

all_models_list = [lr,
                   rf,
                   nn,
                   svm,
                   xgb]

all_models_names = ['lr',
                    'rf',
                    'nn',
                    'svm',
                    'xgb']

#all_models_list = [rf]

```

```
#all_models_names = ['rf']
```

Pipelines

These are the parameters we are going to pass to each model in the hyper-parameter tuning.

In [6]:

```
num_pipe = Pipeline([
    ('select_num', get_numeric_data),
    ('scale', scaler)
])

cat_pipe = Pipeline([
    ('select_cat', get_categorical)
])

# Logistic Regression
lr_params = {'clf__C': [0.0001, 0.001, 0.01, 0.1, 1.0],
             'clf__solver': ['newton-cg', 'lbfgs', 'sag', 'saga'],
             'clf__warm_start': [True, False]}

# Random forests
rf_params = {'clf__n_estimators': [1000, 1500],
             'clf__max_features': ['sqrt'],
             'clf__max_depth': [2, 3, 4, 5],
             'clf__bootstrap': [True, False],
             'clf__criterion': ["gini"],
             'clf__warm_start': [True, False],
             #, 'clf__oob_score': [True, False]
             }

# Neural Network
nn_params = {'clf__hidden_layer_sizes': [(10, 10, 10), (50, 100, 50), (100, 20)],
             'clf__activation': ['tanh', 'relu'],
             'clf__solver': ['sgd', 'adam'],
             'clf__alpha': [0.0001, 0.05],
             'clf__learning_rate': ['constant', 'adaptive']}

# SVM
svm_params = {'clf__C': [1, 10, 100, 1000],
             'clf__gamma': [0.001, 0.0001],
             'clf__kernel': ['rbf', 'linear']}

# XGB
xgb_params = {'clf__max_depth': [1, 3, 100],
             'clf__learning_rate': [0.1],
             'clf__n_estimators': [100, 500, 1000],
             'clf__silent': [True],
             'clf__objective': ['binary:logistic'],
             'clf__booster': ['gbtree'],
             'clf__n_jobs': [1],
             'clf__nthread': [None],
             'clf__gamma': [0],
             'clf__min_child_weight': [1],
             'clf__max_delta_step': [0],
```

```

        'clf__subsample': [1],
        'clf__colsample_bytree': [1],
        'clf__colsample_bylevel': [1],
        'clf__reg_alpha': [0,0.0001, 0.05, 0.5],
        'clf__reg_lambda': [0,1],
        'clf__scale_pos_weight': [1,53.22], # Controls the balance of positive
and negative weights, useful for unbalanced classes. A typical value to consider: su
mnegative instances / sumpositive instances
        'clf__base_score': [0.5],
        'clf__random_state': [0],
        'clf__seed': [None]}

```

```

all_params_list = [lr_params,
                    rf_params,
                    nn_params,
                    svm_params,
                    xgb_params
                    ]

```

```

#all_params_list = [rf_params
#

```

Scoring

In [7]:

```

bestmodels = [] # we create an empty list to store the best performing version of eac
h model

```

```

if len(all_models_list) == len(all_params_list):
    for i,model in enumerate(all_models_list):
        print(type(all_models_list[i]))
        full_pipeline = Pipeline([('feat_union',
                                   FeatureUnion(transformer_list=[('cat_pipeline', ca
t_pipe),('num_pipeline', num_pipe) ])),
                                   ('clf', model)])

```

```

        scoring = make_scorer(roc_auc_score, needs_threshold = True)

```

```

        random_search = RandomizedSearchCV(full_pipeline, param_distributions=all_par
ams_list[i], n_iter=10, cv=10, refit = True, scoring=scoring, n_jobs=-1, verbose=10,
random_state = 42)

```

```

        sys.setrecursionlimit(10000)
        random_search.fit(X_train, y_train)

```

```

        # Best model in training
        bestmodels.append(random_search.best_estimator_)

```

else:

```

    print('Make sure the amount of models in all_models_list matches the amount of pa
rams in all_params_list')

```

```

<class 'sklearn.linear_model.logistic.LogisticRegression'>

```

Fitting 10 folds for each of 10 candidates, totalling 100 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 16 concurrent workers.

[Parallel(n_jobs=-1)]: Done 9 tasks | elapsed: 10.9s

[Parallel(n_jobs=-1)]: Done 18 tasks | elapsed: 18.5s

```

[Parallel(n_jobs=-1)]: Done 29 tasks      | elapsed: 26.7s
[Parallel(n_jobs=-1)]: Done 40 tasks      | elapsed: 30.5s
[Parallel(n_jobs=-1)]: Done 53 tasks      | elapsed: 33.2s
[Parallel(n_jobs=-1)]: Done 66 tasks      | elapsed: 35.7s
[Parallel(n_jobs=-1)]: Done 80 out of 100 | elapsed: 42.5s remaining: 10.6s
[Parallel(n_jobs=-1)]: Done 91 out of 100 | elapsed: 44.8s remaining: 4.3s
[Parallel(n_jobs=-1)]: Done 100 out of 100 | elapsed: 45.3s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 0.4s finished
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 16 concurrent workers.
<class 'sklearn.ensemble.random_forest.RandomForestClassifier'>
Fitting 10 folds for each of 10 candidates, totalling 100 fits
[Parallel(n_jobs=-1)]: Done 9 tasks      | elapsed: 32.2s
[Parallel(n_jobs=-1)]: Done 18 tasks     | elapsed: 49.5s
[Parallel(n_jobs=-1)]: Done 29 tasks     | elapsed: 1.0min
[Parallel(n_jobs=-1)]: Done 40 tasks     | elapsed: 1.2min
[Parallel(n_jobs=-1)]: Done 53 tasks     | elapsed: 1.5min
[Parallel(n_jobs=-1)]: Done 66 tasks     | elapsed: 2.2min
[Parallel(n_jobs=-1)]: Done 80 out of 100 | elapsed: 2.6min remaining: 38.8s
[Parallel(n_jobs=-1)]: Done 91 out of 100 | elapsed: 2.7min remaining: 16.2s
[Parallel(n_jobs=-1)]: Done 100 out of 100 | elapsed: 2.8min finished
<class 'sklearn.neural_network.mlp.MLPClassifier'>
Fitting 10 folds for each of 10 candidates, totalling 100 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 16 concurrent workers.
[Parallel(n_jobs=-1)]: Done 9 tasks      | elapsed: 2.0min
[Parallel(n_jobs=-1)]: Done 18 tasks     | elapsed: 3.1min
[Parallel(n_jobs=-1)]: Done 29 tasks     | elapsed: 3.9min
[Parallel(n_jobs=-1)]: Done 40 tasks     | elapsed: 5.6min
[Parallel(n_jobs=-1)]: Done 53 tasks     | elapsed: 7.0min
[Parallel(n_jobs=-1)]: Done 66 tasks     | elapsed: 8.2min
[Parallel(n_jobs=-1)]: Done 80 out of 100 | elapsed: 10.8min remaining: 2.7min
[Parallel(n_jobs=-1)]: Done 91 out of 100 | elapsed: 12.0min remaining: 1.2min
[Parallel(n_jobs=-1)]: Done 100 out of 100 | elapsed: 12.4min finished
<class 'sklearn.svm.classes.SVC'>
Fitting 10 folds for each of 10 candidates, totalling 100 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 16 concurrent workers.
[Parallel(n_jobs=-1)]: Done 9 tasks      | elapsed: 2.3min
[Parallel(n_jobs=-1)]: Done 18 tasks     | elapsed: 4.9min
[Parallel(n_jobs=-1)]: Done 29 tasks     | elapsed: 5.6min
[Parallel(n_jobs=-1)]: Done 40 tasks     | elapsed: 7.8min
[Parallel(n_jobs=-1)]: Done 53 tasks     | elapsed: 10.6min
[Parallel(n_jobs=-1)]: Done 66 tasks     | elapsed: 12.4min
[Parallel(n_jobs=-1)]: Done 80 out of 100 | elapsed: 13.8min remaining: 3.4min
[Parallel(n_jobs=-1)]: Done 91 out of 100 | elapsed: 15.3min remaining: 1.5min
[Parallel(n_jobs=-1)]: Done 100 out of 100 | elapsed: 15.5min finished
<class 'xgboost.sklearn.XGBClassifier'>
Fitting 10 folds for each of 10 candidates, totalling 100 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 16 concurrent workers.
C:\Users\miguel\AppData\Local\Continuum\anaconda3\lib\site-packages\joblib\externals\
loky\process_executor.py:706: UserWarning: A worker stopped while some jobs were give
n to the executor. This can be caused by a too short worker timeout or by a memory le
ak.
"timeout or by a memory leak.", UserWarning
[Parallel(n_jobs=-1)]: Done 9 tasks      | elapsed: 7.5min
[Parallel(n_jobs=-1)]: Done 18 tasks     | elapsed: 8.8min

```

```
[Parallel(n_jobs=-1)]: Done 29 tasks      | elapsed: 16.1min
[Parallel(n_jobs=-1)]: Done 40 tasks      | elapsed: 17.7min
[Parallel(n_jobs=-1)]: Done 53 tasks      | elapsed: 19.9min
[Parallel(n_jobs=-1)]: Done 66 tasks      | elapsed: 28.3min
[Parallel(n_jobs=-1)]: Done 80 out of 100 | elapsed: 35.2min remaining: 8.8min
[Parallel(n_jobs=-1)]: Done 91 out of 100 | elapsed: 39.7min remaining: 3.9min
[Parallel(n_jobs=-1)]: Done 100 out of 100 | elapsed: 40.9min finished
```

Defining functions

Specificity and Sensitivity

In [8]:

```
def spec(y_true,y_pred):
    TN, FP, FN, TP = confusion_matrix(y_true,y_pred).ravel()
    return TN/(TN+FP)

def sens(y_true,y_pred):
    TN, FP, FN, TP = confusion_matrix(y_true,y_pred).ravel()
    return TP/(TP+FN)
```

Optimal AUC threshold

In [9]:

```
def optimal_threshold_auc(target, predicted):

    fpr, tpr, threshold = roc_curve(target, predicted)
    i = np.arange(len(tpr))
    roc = pd.DataFrame({'tf' : pd.Series(tpr-(1-fpr), index=i), 'threshold' : pd.Series(threshold, index=i)})
    roc_t = roc.iloc[(roc.tf-0).abs().argsort()[:1]]
    return list(roc_t['threshold'])
```

AUROC

In [10]:

```
def auroc(y, ypred,title):
    fig = pyplot.figure(figsize=(8,5))
    ax = fig.add_subplot(111)
    fpr, tpr, thresholds = roc_curve(y, ypred)
    auc_score = roc_auc_score(y, ypred)

    pyplot.plot([0, 1], [0, 1], linestyle='--')
    pyplot.plot(fpr, tpr, marker='.',color='c')
    pyplot.xlabel('False positive rate')
    pyplot.ylabel('Recall')
    pyplot.ylim([0.0, 1.05])
    pyplot.xlim([0.0, 1.0])
    pyplot.title(title+' '+ 'AUROC={0:0.2f}'.format(auc_score))
    for item in ([ax.title, ax.xaxis.label, ax.yaxis.label] +
                  ax.get_xticklabels() + ax.get_yticklabels()):
        item.set_fontsize(20)
```

AUPR

In [11]:

```
def aupr(y, ypred,title):
    fig = pyplot.figure(figsize=(8,5))
    ax = fig.add_subplot(111)
```

```

Baseline = sum(y)/len(y)
average_precision = metrics.average_precision_score(y, ypred)

precision, recall, _ = precision_recall_curve(y, ypred)
auc = metrics.auc(recall, precision)
step_kwargs = ({'step': 'post'})
if 'step' in signature(pyplot.fill_between).parameters
else {}
pyplot.step(recall, precision, color='c', where='post')
pyplot.fill_between(recall, precision, color='c', **step_kwargs)
pyplot.xlabel('Recall')
pyplot.ylabel('Precision')
pyplot.ylim([0.0, 1.05])
pyplot.xlim([0.0, 1.0])
pyplot.title(title+' '+ 'AUPRC={0:0.2f}'.format(auc))
# pyplot.plot(Baseline, 'b-', label='Baseline')
# pyplot.legend(bbox_to_anchor=(0.7, 0.9), loc=2, borderaxespad=0., fontsize = 16)
for item in ([ax.title, ax.xaxis.label, ax.yaxis.label] +
             ax.get_xticklabels() + ax.get_yticklabels()):
    item.set_fontsize(20)

def metric_aupr(y_true, y_pred):
    precision, recall, _ = precision_recall_curve(y_true, y_pred)
    auc = metrics.auc(recall, precision)
    return auc

```

Bootstrapping function

In [12]:

```

def metric_SMR(y_true, y_pred):
    SMR = sum(y_pred)/sum(y_true)
    return SMR

def get_CI_boot(y_true,y_pred,metric,boot):
    # bootstrap confidence intervals
    # seed the random number generator
    seed(1)
    i = 0
    # generate dataset
    dataset = y_pred
    real = y_true
    # bootstrap
    scores = list()
    while i < boot:
        # bootstrap sample
        indices = randint(0, len(y_pred) - 1, len(y_pred))
        sample = dataset[indices]
        real = y_true[indices]
        if len(np.unique(y_true[indices])) < 2:
            # We need at least one positive and one negative sample for ROC AUC
            # to be defined: reject the sample
            continue
        # calculate and store statistic
    else:

```

```

        statistic = metric(real,sample)
        scores.append(statistic)
        i += 1

# calculate 95% confidence intervals (100 - alpha)
alpha = 5.0
# calculate lower percentile (e.g. 2.5)
lower_p = alpha / 2.0
# retrieve observation at lower percentile
lower = max(0.0, percentile(scores, lower_p))
# calculate upper percentile (e.g. 97.5)
upper_p = (100 - alpha) + (alpha / 2.0)
# retrieve observation at upper percentile
if metric == metric_SMR:
    upper = percentile(scores, upper_p)
else:
    upper = min(1.0, percentile(scores, upper_p))
delta = upper-lower
mediana = median(scores)
return mediana,(lower,upper),delta

```

Predictions in train and test
Insights, AUPR and AUROC curves

In [15]:

```

for i,model in enumerate(bestmodels):
    modelname = all_models_names[i]
    print(modelname)
    y_train_pred = model.predict_proba(X_train)[: ,1]
    y_test_pred = model.predict_proba(X_test)[: ,1]
    auroc(y_test, y_test_pred,all_models_names[i])
    aupr(y_test, y_test_pred,all_models_names[i])
    # Threshold in train
    threshold = optimal_threshold_auc(y_train, y_train_pred)

# performance_all = [roc_auc_score,average_precision_score, accuracy_score, spec,
#                    sens, precision_score, f1_score, cohen_kappa_score,metric_SMR,metric_aupr]
#
# performance_names_all = np.array(['roc_auc_score','average_precision_score',
#                                   'accuracy_score','spec','sens','precision_score',
#                                   'f1_score','cohen_kappa_score','metric_SMR','metric_aupr'])
#
performance = [roc_auc_score,precision_score, spec, sens]

performance_names = np.array(['roc_auc_score','precision_score','spec','sens'])

conf = []
delt = []
met = []
med = []
te = []

```

```

performance_insights = []

y_test = np.array(y_test)

for i,p in enumerate(performance):

    if (p == roc_auc_score or p == average_precision_score or p == metric_aupr):
        mediana,extremes,delta = get_CI_boot(y_test,y_test_pred,p,boot=100) #aten
cao aqui ao boot
        met.append(p)
        conf.append(extremes)
        delt.append(delta)
        med.append(mediana)
        te.append(p(y_test,y_test_pred))
        performance_insights.append(performance_names[i]+' '+str(round(p(y_test,y
_test_pred), 2)) + ' CI [' + str(round(extremes[0], 2)) + '-' + str(round(extremes[1],
2)) + '])')
    else:
        y_pred_threshold = (y_test_pred >= threshold).astype(np.int)
        mediana,extremes,delta = get_CI_boot(y_test,y_pred_threshold,p,boot=100)
        met.append(p)
        conf.append(extremes)
        delt.append(delta)
        med.append(mediana)
        te.append(p(y_test,y_pred_threshold))
        performance_insights.append(performance_names[i]+' '+str(round(p(y_test,y
_pred_threshold), 2)) + ' CI [' + str(round(extremes[0], 2)) + '-' + str(round(extremes
[1], 2)) + '])')

performance_insights = str(performance_insights)
performance_insights = performance_insights.replace("'", "")
print(performance_insights)

lr
[roc_auc_score 0.8 CI [0.7-0.88], precision_score 0.08 CI [0.04-0.12], spec 0.81 CI [
0.79-0.83], sens 0.68 CI [0.51-0.84]]
rf
[roc_auc_score 0.83 CI [0.74-0.92], precision_score 0.1 CI [0.06-0.14], spec 0.85 CI
[0.83-0.88], sens 0.68 CI [0.52-0.89]]
nn
[roc_auc_score 0.7 CI [0.59-0.8], precision_score 0.1 CI [0.03-0.18], spec 0.96 CI [0
.94-0.97], sens 0.21 CI [0.07-0.36]]
svm
[roc_auc_score 0.66 CI [0.52-0.79], precision_score 0.08 CI [0.04-0.12], spec 0.87 CI
[0.86-0.89], sens 0.46 CI [0.3-0.65]]
xgb
[roc_auc_score 0.85 CI [0.76-0.91], precision_score 0.1 CI [0.06-0.15], spec 0.86 CI
[0.84-0.88], sens 0.68 CI [0.5-0.81]]

```

Calibration

In [19]:

```

for i,model in enumerate(bestmodels):
    # Add title and axis names
    pyplot.title('Calibration plot (reliability curve)')

```



```

pyplot.xlabel('Mean predicted value')
pyplot.ylabel('Fraction of positives')

model.fit(X_train, y_train)
#y_pred = (model.predict_proba(X_test)[: ,1] >= threshold).astype(np.int)
prob_pos = model.predict_proba(X_test)[: ,1] #).astype(np.int)
fraction_of_positives, mean_predicted_value = \
    calibration_curve(y_test, prob_pos, n_bins=10, normalize=True)
pyplot.plot(mean_predicted_value, fraction_of_positives, 'o-c',label="model_"+all
_models_names[i])

pyplot.title('Calibration plot (reliability curve)')
pyplot.xlabel('Mean predicted value')
pyplot.ylabel('Fraction of positives')

model_isotonic = _CalibratedClassifier(model, method='isotonic')
model_isotonic.fit(X_train, y_train)
calibrated = model_isotonic
# predict probabilities
probs = calibrated.predict_proba(X_test)[: , 1]
# reliability diagram
fop, mpv = calibration_curve(y_test, probs, n_bins=10, normalize=True)
# plot calibrated reliability
pyplot.plot(mpv, fop, marker='.',label="Isotonic")

model_isotonic = _CalibratedClassifier(model, method='sigmoid')
model_isotonic.fit(X_train, y_train)
calibrated = model_isotonic
# predict probabilities
probs = calibrated.predict_proba(X_test)[: , 1]
# reliability diagram
fop, mpv = calibration_curve(y_test, probs, n_bins=10, normalize=True)
pyplot.plot([0, 1], [0, 1], 'k--',label='Perfect calibration')
# plot calibrated reliability
pyplot.plot(mpv, fop, marker='.',label="sigmoid")

pyplot.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)

```

Save results

In [14]:

```

db_name = 'OR'

# XGB
model_name = 'XGB'

features_name = 'full'

# Best model
filename = '{}_{}_{}_model.sav'.format(model_name, features_name, db_name)

with open(filename, 'wb') as pickle_file:
    dill.dump(random_search.best_estimator_, pickle_file)

```

```

    print('model saved')

# CV model
filename = '{}_{}_{}_model_CV.sav'.format(model_name, features_name, db_name)

with open(filename, 'wb') as pickle_file:
    dill.dump(random_search, pickle_file)
    print('model saved')

```

training_models_outsidecpb

Environment

In [92]:

```

import numpy as np
import pandas as pd
from numpy import median, percentile
from numpy.random import seed, randint

from sklearn import metrics
#from scipy.integrate import cumtrapz
from sklearn.metrics import (roc_auc_score, cohen_kappa_score, confusion_matrix, precision_score,
                             accuracy_score, average_precision_score, f1_score,
                             precision_recall_curve, roc_curve) #, auc, recall_score)

import matplotlib.pyplot as pyplot
from inspect import signature
import joblib
from sklearn.calibration import (CalibratedClassifierCV,
                                 _CalibratedClassifier, calibration_curve)
from sklearn.pipeline import Pipeline, FeatureUnion
from sklearn.model_selection import RandomizedSearchCV
from sklearn.preprocessing import FunctionTransformer, MinMaxScaler
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import (make_scorer, roc_auc_score) #, cohen_kappa_score, confusion_matrix,
# precision_score,
# accuracy_score, auc, average_precision_score, f1_score,
# precision_recall_curve, recall_score, roc_curve)
import dill
import sys
from xgboost import XGBClassifier

```

Import Data

In [29]:

```

train = pd.read_csv("../data/train_outsidecpb.csv", delimiter=",", low_memory=False) # ../.
./ go two folders back from current location

test = pd.read_csv("../data/test_outsidecpb.csv", delimiter=",", low_memory=False)

cols = ['entropy', 'duration_Surgery_min', 'duration_CPB_min',
        'STS_RiskAlgorithm_median_imp', 'STS_RiskAlgorithm',
        'Q_total_pressors_formula_mg', 'Cross_Clamp_Time_imp', # we don't
        have crossclam outside cpb (REMEMBER THIS WHEN RUNNING preCPB AND postCPB)

```

```

mp',
        'T_STS_RiskAlgorithm_median_imp', 'T_STS_RiskAlgorithm_multiple_i
        'Q_AUC65MAP_total_mmHgmin', 'Q_TWA65MAP_total_mmHg',
        'Q_AUC65MAP_CPB_mmHgmin', 'Q_AUC65MAP_outsideCPB_mmHgmin',
        'Q_TWA65MAP_outsideCPB_mmHg', 'Q_TWA65MAP_CPB_mmHg',
        'TWA65MAP_total_mmHg', 'TWA65MAP_CPB_mmHg', 'TWA65MAP_outsideCPB_m
mHg',
        'AUC65MAP_total_mmHgmin', 'AUC65MAP_CPB_mmHgmin',
        'Any_MAE', 'Stroke', 'RFDS', 'RFD', 'DS', 'RFS', 'id']

```

```

train = train.drop(columns = cols)
test = test.drop(columns = cols)

```

 # Converting categorical variables to numeric

```

train.Gender.loc[train.Gender == 'Male'] = 0
train.Gender.loc[train.Gender == 'Female'] = 1

train.Category.loc[train.Category == 'Valve'] = 0
train.Category.loc[train.Category == 'CABG'] = 1
train.Category.loc[train.Category == 'CABG + Valve'] = 2
train.Category.loc[train.Category == 'Aortic Surgery'] = 3
train.Category.loc[train.Category == 'Other'] = 4

```

```

test.Gender.loc[test.Gender == 'Male'] = 0
test.Gender.loc[test.Gender == 'Female'] = 1

```

```

test.Category.loc[test.Category == 'Valve'] = 0
test.Category.loc[test.Category == 'CABG'] = 1
test.Category.loc[test.Category == 'CABG + Valve'] = 2
test.Category.loc[test.Category == 'Aortic Surgery'] = 3
test.Category.loc[test.Category == 'Other'] = 4

```

 # since we are not competing again STS risk score now I did not exclude patients with no STS risk score but I impute the missing values.

 # Removing rows based on data availability

```
#train = train[train['STS_RiskAlgorithm'].notna()]
```

```
#test = test[test['STS_RiskAlgorithm'].notna()]
```

C:\Users\miguel\AppData\Local\Continuum\anaconda3\lib\site-packages\pandas\core\indexing.py:20

5: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
self._setitem_with_indexer(indexer, value)
```

In [30]:

Split data into x y training and x y test sets

In [31]:

```

y_train = train['Death']
X_train = train.drop(columns = ['Death'])

```

```

y_test = test['Death']
X_test = test.drop(columns = ['Death'])

```

 # ALL

```
X_num = X_train.drop(columns = ['Gender', 'Category', 'Renal_Failure'])
```

```

get_numeric_data = FunctionTransformer(lambda x: x[X_num.columns], validate=False)

get_categorical = FunctionTransformer(lambda x: x[['Gender', 'Category', 'Renal_Failure']], validate=False)

scaler = MinMaxScaler()

```

Modeling

In [32]:

```

# Logistic Regression
lr = LogisticRegression(random_state = 42, tol=1**-4, verbose=1) #, solver='lbfgs') #, class_weight='balanced')

# Random forests
rf = RandomForestClassifier(random_state = 42
                           #, n_jobs=-1 # set The number of jobs to run in parallel -1 means using all processors
                           )

# Neural Network

nn = MLPClassifier(random_state = 42, max_iter=500)

# SVM

svm = SVC(random_state = 42, probability=True) # Whether to enable probability estimates. This must be enabled prior to calling fit, and will slow down that method.

# XGB

xgb = XGBClassifier(random_state = 42)

all_models_list = [lr,
                   rf,
                   nn,
                   svm,
                   xgb]

all_models_names = ['lr',
                   'rf',
                   'nn',
                   'svm',
                   'xgb']

#all_models_list = [rf]

#all_models_names = ['rf']

```

Pipelines

These are the parameters we are going to pass to each model in the hyper-parameter tuning.

In [33]:

```

num_pipe = Pipeline([
    ('select_num', get_numeric_data),
    ('scale', scaler)
])

cat_pipe = Pipeline([
    ('select_cat', get_categorical)
])

```

```

# Logistic Regression
lr_params = {'clf__C': [0.0001, 0.001, 0.01, 0.1, 1.0],
             , 'clf__solver': ['newton-cg', 'lbfgs', 'sag', 'saga'],
             , 'clf__warm_start': [True, False]}

# Random forests
rf_params = {'clf__n_estimators': [1000, 1500],
             , 'clf__max_features': ['sqrt'],
             , 'clf__max_depth': [2, 3, 4, 5],
             , 'clf__bootstrap': [True, False],
             , 'clf__criterion': ['gini'],
             , 'clf__warm_start': [True, False]
             #, 'clf__oob_score': [True, False]
             }

# Neural Network
nn_params = {'clf__hidden_layer_sizes': [(10, 10, 10), (50, 100, 50), (100, 20)],
             , 'clf__activation': ['tanh', 'relu'],
             , 'clf__solver': ['sgd', 'adam'],
             , 'clf__alpha': [0.0001, 0.05],
             , 'clf__learning_rate': ['constant', 'adaptive']}

# SVM
svm_params = {'clf__C': [1, 10, 100, 1000],
             , 'clf__gamma': [0.001, 0.0001],
             , 'clf__kernel': ['rbf', 'linear']}

# XGB
xgb_params = {'clf__max_depth': [1, 3, 100],
             , 'clf__learning_rate': [0.1],
             , 'clf__n_estimators': [100, 500, 1000],
             , 'clf__silent': [True],
             , 'clf__objective': ['binary:logistic'],
             , 'clf__booster': ['gbtree'],
             , 'clf__n_jobs': [1],
             , 'clf__nthread': [None],
             , 'clf__gamma': [0],
             , 'clf__min_child_weight': [1],
             , 'clf__max_delta_step': [0],
             , 'clf__subsample': [1],
             , 'clf__colsample_bytree': [1],
             , 'clf__colsample_bylevel': [1],
             , 'clf__reg_alpha': [0, 0.0001, 0.05, 0.5],
             , 'clf__reg_lambda': [0, 1],
             , 'clf__scale_pos_weight': [1, 53.22], # Controls the balance of positive and negative weights, useful for unbalanced classes. A typical value to consider: sum(negative instances) / sum(positive instances)
             , 'clf__base_score': [0.5],
             , 'clf__random_state': [0],
             , 'clf__seed': [None]}

all_params_list = [lr_params,
                  rf_params,
                  nn_params,
                  svm_params,
                  xgb_params
                  ]

#all_params_list = [rf_params
#

```

Scoring

In [34]:

```
bestmodels = [] # we create an empty list to store the best performing version of each model

if len(all_models_list) == len(all_params_list):
    for i,model in enumerate(all_models_list):
        print(type(all_models_list[i]))
        full_pipeline = Pipeline([('feat_union',
                                   FeatureUnion(transformer_list=[('cat_pipeline', cat_pipe),
                                   ('num_pipeline', num_pipe) ])),
                                   ('clf', model)])

        scoring = make_scorer(roc_auc_score, needs_threshold = True)

        random_search = RandomizedSearchCV(full_pipeline, param_distributions=all_params_list[
i], n_iter=10, cv=10, refit = True, scoring=scoring, n_jobs=-1, verbose=10, random_state = 42)

        sys.setrecursionlimit(10000)
        random_search.fit(X_train, y_train)

        # Best model in training
        bestmodels.append(random_search.best_estimator_)

else:
    print('Make sure the amount of models in all_models_list matches the amount of params in a
ll_params_list')
<class 'sklearn.linear_model.logistic.LogisticRegression'>
Fitting 10 folds for each of 10 candidates, totalling 100 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 16 concurrent workers.
[Parallel(n_jobs=-1)]: Done   9 tasks      | elapsed:   6.7s
[Parallel(n_jobs=-1)]: Done  18 tasks      | elapsed:  10.4s
[Parallel(n_jobs=-1)]: Done  29 tasks      | elapsed:  14.3s
[Parallel(n_jobs=-1)]: Done  40 tasks      | elapsed:  16.9s
[Parallel(n_jobs=-1)]: Done  53 tasks      | elapsed:  17.8s
[Parallel(n_jobs=-1)]: Done  66 tasks      | elapsed:  19.4s
[Parallel(n_jobs=-1)]: Done  80 out of 100 | elapsed:  23.2s remaining:   5.7s
[Parallel(n_jobs=-1)]: Done  91 out of 100 | elapsed:  24.5s remaining:   2.3s
[Parallel(n_jobs=-1)]: Done 100 out of 100 | elapsed:  24.9s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:   0.2s finished
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 16 concurrent workers.
<class 'sklearn.ensemble forest.RandomForestClassifier'>
Fitting 10 folds for each of 10 candidates, totalling 100 fits
[Parallel(n_jobs=-1)]: Done   9 tasks      | elapsed:  24.1s
[Parallel(n_jobs=-1)]: Done  18 tasks      | elapsed:  38.5s
[Parallel(n_jobs=-1)]: Done  29 tasks      | elapsed:  48.0s
[Parallel(n_jobs=-1)]: Done  40 tasks      | elapsed:  55.6s
[Parallel(n_jobs=-1)]: Done  53 tasks      | elapsed:  1.1min
[Parallel(n_jobs=-1)]: Done  66 tasks      | elapsed:  1.6min
[Parallel(n_jobs=-1)]: Done  80 out of 100 | elapsed:  1.9min remaining:  28.1s
[Parallel(n_jobs=-1)]: Done  91 out of 100 | elapsed:  2.0min remaining:  11.9s
[Parallel(n_jobs=-1)]: Done 100 out of 100 | elapsed:  2.0min finished
<class 'sklearn.neural_network.multilayer_perceptron.MLPClassifier'>
Fitting 10 folds for each of 10 candidates, totalling 100 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 16 concurrent workers.
[Parallel(n_jobs=-1)]: Done   9 tasks      | elapsed:  1.3min
[Parallel(n_jobs=-1)]: Done  18 tasks      | elapsed:  1.7min
[Parallel(n_jobs=-1)]: Done  29 tasks      | elapsed:  2.3min
[Parallel(n_jobs=-1)]: Done  40 tasks      | elapsed:  3.2min
[Parallel(n_jobs=-1)]: Done  53 tasks      | elapsed:  4.1min
```

```

[Parallel(n_jobs=-1)]: Done 66 tasks      | elapsed: 4.5min
[Parallel(n_jobs=-1)]: Done 80 out of 100 | elapsed: 5.5min remaining: 1.4min
[Parallel(n_jobs=-1)]: Done 91 out of 100 | elapsed: 6.4min remaining: 37.8s
[Parallel(n_jobs=-1)]: Done 100 out of 100 | elapsed: 6.7min finished
<class 'sklearn.svm.classes.SVC'>
Fitting 10 folds for each of 10 candidates, totalling 100 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 16 concurrent workers.
[Parallel(n_jobs=-1)]: Done 9 tasks      | elapsed: 1.6min
[Parallel(n_jobs=-1)]: Done 18 tasks     | elapsed: 3.5min
[Parallel(n_jobs=-1)]: Done 29 tasks     | elapsed: 4.3min
[Parallel(n_jobs=-1)]: Done 40 tasks     | elapsed: 7.0min
[Parallel(n_jobs=-1)]: Done 53 tasks     | elapsed: 9.9min
[Parallel(n_jobs=-1)]: Done 66 tasks     | elapsed: 11.5min
[Parallel(n_jobs=-1)]: Done 80 out of 100 | elapsed: 12.4min remaining: 3.1min
[Parallel(n_jobs=-1)]: Done 91 out of 100 | elapsed: 13.4min remaining: 1.3min
[Parallel(n_jobs=-1)]: Done 100 out of 100 | elapsed: 13.7min finished
<class 'xgboost.sklearn.XGBClassifier'>
Fitting 10 folds for each of 10 candidates, totalling 100 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 16 concurrent workers.
C:\Users\miguel\AppData\Local\Continuum\anaconda3\lib\site-packages\joblib\externals\loky\process_executor.py:706: UserWarning: A worker stopped while some jobs were given to the executor.
This can be caused by a too short worker timeout or by a memory leak.
  "timeout or by a memory leak.", UserWarning
[Parallel(n_jobs=-1)]: Done 9 tasks      | elapsed: 5.3min
[Parallel(n_jobs=-1)]: Done 18 tasks     | elapsed: 6.2min
[Parallel(n_jobs=-1)]: Done 29 tasks     | elapsed: 44.6min
[Parallel(n_jobs=-1)]: Done 40 tasks     | elapsed: 45.7min
[Parallel(n_jobs=-1)]: Done 53 tasks     | elapsed: 47.2min
[Parallel(n_jobs=-1)]: Done 66 tasks     | elapsed: 53.0min
[Parallel(n_jobs=-1)]: Done 80 out of 100 | elapsed: 57.9min remaining: 14.5min
[Parallel(n_jobs=-1)]: Done 91 out of 100 | elapsed: 61.2min remaining: 6.1min
[Parallel(n_jobs=-1)]: Done 100 out of 100 | elapsed: 62.0min finished

```

Defining functions

Specificity and Sensitivity

In [43]:

```

def spec(y_true,y_pred):
    TN, FP, FN, TP = confusion_matrix(y_true,y_pred).ravel()
    return TN/(TN+FP)

def sens(y_true,y_pred):
    TN, FP, FN, TP = confusion_matrix(y_true,y_pred).ravel()
    return TP/(TP+FN)

```

Optimal AUC threshold

In [44]:

```

def optimal_threshold_auc(target, predicted):

    fpr, tpr, threshold = roc_curve(target, predicted)
    i = np.arange(len(tpr))
    roc = pd.DataFrame({'tf' : pd.Series(tpr-(1-fpr), index=i), 'threshold' : pd.Series(threshold, index=i)})
    roc_t = roc.iloc[(roc.tf-0).abs().argsort()[:1]]
    return list(roc_t['threshold'])

```

AUROC

In [45]:

```

def auROC(y, ypred,title):
    fig = pyplot.figure(figsize=(8,5))

```

```

ax = fig.add_subplot(111)
fpr, tpr, thresholds = roc_curve(y, ypred)
auc_score = roc_auc_score(y, ypred)

pyplot.plot([0, 1], [0, 1], linestyle='--')
pyplot.plot(fpr, tpr, marker='.', color='c')
pyplot.xlabel('False positive rate')
pyplot.ylabel('Recall')
pyplot.ylim([0.0, 1.05])
pyplot.xlim([0.0, 1.0])
pyplot.title(title+' '+ 'AUROC={0:0.2f}'.format(auc_score))
for item in ([ax.title, ax.xaxis.label, ax.yaxis.label] +
             ax.get_xticklabels() + ax.get_yticklabels()):
    item.set_fontsize(20)

```

AUPR

In [46]:

```

def aupr(y, ypred, title):
    fig = pyplot.figure(figsize=(8,5))
    ax = fig.add_subplot(111)

    Baseline = sum(y)/len(y)
    average_precision = metrics.average_precision_score(y, ypred)

    precision, recall, _ = precision_recall_curve(y, ypred)
    auc = metrics.auc(recall, precision)
    step_kwargs = ({'step': 'post'})
    if 'step' in signature(pyplot.fill_between).parameters
    else {}
    pyplot.step(recall, precision, color='c', where='post')
    pyplot.fill_between(recall, precision, color='c', **step_kwargs)
    pyplot.xlabel('Recall')
    pyplot.ylabel('Precision')
    pyplot.ylim([0.0, 1.05])
    pyplot.xlim([0.0, 1.0])
    pyplot.title(title+' '+ 'AUPRC={0:0.2f}'.format(auc))
#     pyplot.plot(Baseline, 'b-', label='Baseline')
#     pyplot.legend(bbox_to_anchor=(0.7, 0.9), loc=2, borderaxespad=0., fontsize = 16)
    for item in ([ax.title, ax.xaxis.label, ax.yaxis.label] +
                 ax.get_xticklabels() + ax.get_yticklabels()):
        item.set_fontsize(20)

def metric_aupr(y_true, y_pred):
    precision, recall, _ = precision_recall_curve(y_true, y_pred)
    auc = metrics.auc(recall, precision)
    return auc

```

Bootstrapping function

In [47]:

```

def metric_SMR(y_true, y_pred):
    SMR = sum(y_pred)/sum(y_true)
    return SMR

def get_CI_boot(y_true, y_pred, metric, boot):
    # bootstrap confidence intervals
    # seed the random number generator
    seed(1)
    i = 0
    # generate dataset

```



```

dataset = y_pred
real = y_true
# bootstrap
scores = list()
while i < boot:
    # bootstrap sample
    indices = randint(0, len(y_pred) - 1, len(y_pred))
    sample = dataset[indices]
    real = y_true[indices]
    if len(np.unique(y_true[indices])) < 2:
        # We need at least one positive and one negative sample for ROC AUC
        # to be defined: reject the sample
        continue
    # calculate and store statistic
    else:
        statistic = metric(real,sample)
        scores.append(statistic)
        i += 1

# calculate 95% confidence intervals (100 - alpha)
alpha = 5.0
# calculate lower percentile (e.g. 2.5)
lower_p = alpha / 2.0
# retrieve observation at lower percentile
lower = max(0.0, percentile(scores, lower_p))
# calculate upper percentile (e.g. 97.5)
upper_p = (100 - alpha) + (alpha / 2.0)
# retrieve observation at upper percentile
if metric == metric_SMR:
    upper = percentile(scores, upper_p)
else:
    upper = min(1.0, percentile(scores, upper_p))
delta = upper-lower
mediana = median(scores)
return mediana,(lower,upper),delta

```

Predictions in train and test Insights, AUPR and AUROC curves

In [116]:

```

for i,model in enumerate(bestmodels):
    modelname = all_models_names[i]
    print(modelname)
    y_train_pred = model.predict_proba(X_train)[: ,1]
    y_test_pred = model.predict_proba(X_test)[: ,1]
    auroc(y_test, y_test_pred,all_models_names[i])
    aupr(y_test, y_test_pred,all_models_names[i])
    # Threshold in train
    threshold = optimal_threshold_auc(y_train, y_train_pred)

# performance_all = [roc_auc_score,average_precision_score, accuracy_score, spec,
#                     sens, precision_score, f1_score, cohen_kappa_score,metric_SMR,metric_aupr]

#
# performance_names_all = np.array(['roc_auc_score','average_precision_score'
#                                   , ' accuracy_score', ' spec', 'sens', ' precision_score'
#                                   , ' f1_score', ' cohen_kappa_score', 'metric_SMR', 'metric_aupr'
'])
#
performance = [roc_auc_score,precision_score, spec, sens]

```

```

performance_names = np.array(['roc_auc_score', 'precision_score', 'spec', 'sens'])

conf = []
delt = []
met = []
med = []
te = []
performance_insights = []

y_test = np.array(y_test)

for i,p in enumerate(performance):

    if (p == roc_auc_score or p == average_precision_score or p == metric_aupr):
        mediana,extremes,delta = get_CI_boot(y_test,y_test_pred,p,boot=100) #atencao aqui
    ao boot
        met.append(p)
        conf.append(extremes)
        delt.append(delta)
        med.append(mediana)
        te.append(p(y_test,y_test_pred))
        performance_insights.append(performance_names[i]+' '+str(round(p(y_test,y_test_pre
d), 2)) + ' CI [' + str(round(extremes[0], 2)) + '-' + str(round(extremes[1], 2)) + '])')
    else:
        y_pred_threshold = (y_test_pred >= threshold).astype(np.int)
        mediana,extremes,delta = get_CI_boot(y_test,y_pred_threshold,p,boot=100)
        met.append(p)
        conf.append(extremes)
        delt.append(delta)
        med.append(mediana)
        te.append(p(y_test,y_pred_threshold))
        performance_insights.append(performance_names[i]+' '+str(round(p(y_test,y_pred_thr
eshold), 2)) + ' CI [' + str(round(extremes[0], 2)) + '-' + str(round(extremes[1], 2)) + '])')

performance_insights = str(performance_insights)
performance_insights = performance_insights.replace("'", "")
print(performance_insights)

lr
[roc_auc_score 0.79 CI [0.69-0.87], precision_score 0.08 CI [0.04-0.11], spec 0.81 CI [0.78-0.
83], sens 0.68 CI [0.49-0.86]]
rf
[roc_auc_score 0.83 CI [0.75-0.91], precision_score 0.09 CI [0.05-0.14], spec 0.84 CI [0.81-0.
86], sens 0.71 CI [0.54-0.89]]
nn
[roc_auc_score 0.76 CI [0.67-0.86], precision_score 0.11 CI [0.06-0.19], spec 0.94 CI [0.92-0.
95], sens 0.36 CI [0.2-0.56]]
svm
[roc_auc_score 0.69 CI [0.59-0.79], precision_score 0.06 CI [0.02-0.1], spec 0.9 CI [0.88-0.91
], sens 0.29 CI [0.15-0.43]]
xgb
[roc_auc_score 0.87 CI [0.81-0.93], precision_score 0.1 CI [0.06-0.15], spec 0.85 CI [0.83-0.8
7], sens 0.71 CI [0.52-0.85]]

```

Features Importance

In [59]:

```

cols_names = list(X_train)

# RF
coef = bestmodels[4].steps[1][1].feature_importances_
coef = np.matrix(coef)
feature_importance = abs(coef)
#intercept = clf.best_estimator_.steps[1][1].intercept_

feature_importance = 100.0 * (feature_importance / feature_importance.max())
sorted_idx = np.argsort(np.transpose(feature_importance))
pos = np.arange(sorted_idx.shape[0]) + .5

#np.column_stack([cols_names,list(feature_importance)])

#pd.dataframe.feature_importance

feature_importance_table = pd.DataFrame(feature_importance, columns=cols_names).T # transposing
feature_importance_table.columns = ['Importance']

feature_importance_table.to_csv(path_or_buf='feature_importance_table.csv')

print(feature_importance_table)

#featfig = pyplot.figure(figsize=(15, 15))
#featax = featfig.add_subplot(1, 1, 1)
#featax.barh(pos, feature_importance[sorted_idx], align='center')
#featax.set_yticks(pos)
#featax.set_yticklabels(np.array(cols_names)[sorted_idx], fontsize=18)
#featax.set_xlabel('Relative Feature Importance', fontsize=18)
#
#
#pyplot.tight_layout()
#pyplot.show()

```

	Importance
1	0.000000
2	0.000000
3	100.000000
4	0.000000
5	0.000000
...	...
total_pressors_formula_mg	94.811130
delta_HCT_pct_median_imp	86.445445
mean_ef_median_imp	0.000000
AUC65MAP_outsideCPB_mmHgmin	0.000000
Renal_Failure	47.871733

[2439 rows x 1 columns]

AUROC figures all preCPB ML models

In [111]:

```

# model0 prediction
model0 = bestmodels[0] # select the best performing model here
y_train_pred_model0 = model0.predict_proba(X_train)[:,-1]
y_test_pred_model0 = model0.predict_proba(X_test)[:,-1]
fpr_model0, tpr_model0, thresholds_model0 = roc_curve(y_test, y_test_pred_model0)
auc_score_model0 = roc_auc_score(y_test, y_test_pred_model0)

# model1 prediction
model1 = bestmodels[1] # select the best performing model here

```

```

y_train_pred_model1 = model1.predict_proba(X_train)[: ,1]
y_test_pred_model1 = model1.predict_proba(X_test)[: ,1]
fpr_model1, tpr_model1, thresholds_model1 = roc_curve(y_test, y_test_pred_model1)
auc_score_model1 = roc_auc_score(y_test, y_test_pred_model1)

# model2 prediction
model2 = bestmodels[2] # select the best performing model here
y_train_pred_model2 = model2.predict_proba(X_train)[: ,1]
y_test_pred_model2 = model2.predict_proba(X_test)[: ,1]
fpr_model2, tpr_model2, thresholds_model2 = roc_curve(y_test, y_test_pred_model2)
auc_score_model2 = roc_auc_score(y_test, y_test_pred_model2)

# model3 prediction
model3 = bestmodels[3] # select the best performing model here
y_train_pred_model3 = model3.predict_proba(X_train)[: ,1]
y_test_pred_model3 = model3.predict_proba(X_test)[: ,1]
fpr_model3, tpr_model3, thresholds_model3 = roc_curve(y_test, y_test_pred_model3)
auc_score_model3 = roc_auc_score(y_test, y_test_pred_model3)

# model4 prediction
model4 = bestmodels[4] # select the best performing model here
y_train_pred_model4 = model4.predict_proba(X_train)[: ,1]
y_test_pred_model4 = model4.predict_proba(X_test)[: ,1]
fpr_model4, tpr_model4, thresholds_model4 = roc_curve(y_test, y_test_pred_model4)
auc_score_model4 = roc_auc_score(y_test, y_test_pred_model4)

my_dpi=100
pyplot.figure(figsize=(700/my_dpi,500/my_dpi), dpi=my_dpi)
pyplot.plot([0, 1], [0, 1], linestyle='--', color='#bdc3c7')

#We place the best performing one the first

# AUROC model4
pyplot.plot(fpr_model4, tpr_model4, marker='',color='#2980b9',label='eXtreme Gradient Boosting
AUROC: '+' '{0:0.2f}'.format(auc_score_model4))

# AUROC model0
pyplot.plot(fpr_model0, tpr_model0, marker='',color='#FFC312',label='Logistic Regression AUROC
: '+' '{0:0.2f}'.format(auc_score_model1))

# AUROC model1
pyplot.plot(fpr_model1, tpr_model1, marker='',color='#EE5A24',label='Random Forests AUROC: '+'
' '{0:0.2f}'.format(auc_score_model1))

# AUROC model2
pyplot.plot(fpr_model2, tpr_model2, marker='',color='#009432',label='Neural Networks AUROC: '+'
' '{0:0.2f}'.format(auc_score_model2))

# AUROC model3
pyplot.plot(fpr_model3, tpr_model3, marker='',color='#9980FA',label='Support Vector Machine AU
ROC: '+' '{0:0.2f}'.format(auc_score_model3))

pyplot.legend(loc='lower right')
pyplot.xlabel('1 - Specificity')
pyplot.ylabel('Sensitivity')
pyplot.ylim([0.0, 1.05])
pyplot.xlim([0.0, 1.0])

```

Out[111]:

(0.0, 1.0)

Calibration

In [117]:

```
##%% Calibration FOR BEST MODEL: XGBoost #####

# best model here
clf = model

# Calibrate model
y_test_hat = clf.predict_proba(X_test)[: ,1]
clf2 = LogisticRegression(random_state = 44, max_iter=10**3,tol=1**-4,solver='lbfgs')

clf_isotonic = CalibratedClassifierCV(clf2, cv = 2, method='isotonic')
clf_isotonic.fit(y_test_hat.reshape(-1, 1), y_test)
prob_pos_test = clf_isotonic.predict_proba(y_test_hat.reshape(-1, 1))[: ,1] # predict
fraction_of_positives, mean_predicted_value = \
fop, mpv = calibration_curve(y_test, prob_pos_test, n_bins=10, normalize=True)

pyplot.plot(mpv, fop, 'o-',label="Isotonic calibration")

prob_pos = clf.predict_proba(X_test)[: ,1] #).astype(np.int)
fraction_of_positives, mean_predicted_value = \
calibration_curve(y_test, prob_pos, n_bins=10, normalize=True)
pyplot.plot([0, 1], [0, 1], 'k--',label='Perfect calibration')
pyplot.plot(mean_predicted_value, fraction_of_positives, 'o-c',label="XGBoost")

# Add title and axis names
pyplot.title('Calibration plot (reliability curve)')
pyplot.xlabel('Mean predicted value')
pyplot.ylabel('Fraction of positives')

pyplot.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)

# new y_test_pred
y_test_pred = prob_pos_test

# perform bootstrapping with new y_test_pred #####

performance = [roc_auc_score,precision_score, spec, sens]
performance_names = np.array(['roc_auc_score','precision_score','spec','sens'])

conf = []
delt = []
met = []
med = []
te = []
performance_insights = []

y_test = np.array(y_test)

for i,p in enumerate(performance):

    if (p == roc_auc_score):
        mediana,extremes,delta = get_CI_boot(y_test,y_test_pred,p,boot=100) #atencao aqui ao b
oot
        met.append(p)
        conf.append(extremes)
```

```

        delt.append(delta)
        med.append(media)
        te.append(p(y_test,y_test_pred))
        performance_insights.append(performance_names[i]+' '+str(round(p(y_test,y_test_pred),
2)) + ' CI [' + str(round(extremes[0], 2)) + '-' + str(round(extremes[1], 2)) + '])')
    else:
        y_pred_threshold = (y_test_pred >= threshold).astype(np.int)
        media,extremes,delta = get_CI_boot(y_test,y_pred_threshold,p,boot=100)
        met.append(p)
        conf.append(extremes)
        delt.append(delta)
        med.append(media)
        te.append(p(y_test,y_pred_threshold))
        performance_insights.append(performance_names[i]+' '+str(round(p(y_test,y_pred_thresho
ld), 2)) + ' CI [' + str(round(extremes[0], 2)) + '-' + str(round(extremes[1], 2)) + '])')

performance_insights = str(performance_insights)
performance_insights = performance_insights.replace("'", "")
print(performance_insights)
[roc_auc_score 0.88 CI [0.83-0.94], precision_score 0.1 CI [0.06-0.15], spec 0.85 CI [0.83-0.8
7], sens 0.75 CI [0.57-0.9]]

```

In [104]:

```
roc_auc_score
```

Out[104]:

```
<function sklearn.metrics.ranking.roc_auc_score(y_true, y_score, average='macro', sample_weight=None, max_fpr=None)>
```

Calibrated AUROC figure

In [115]:

```

# model0 prediction
#model0 = bestmodels[0] # select the best performing model here
#y_train_pred_model0 = model0.predict_proba(X_train)[:,-1]
#y_test_pred_model0 = model0.predict_proba(X_test)[:,-1]

fpr_model0, tpr_model0, thresholds_model0 = roc_curve(y_test, y_test_pred)
auc_score_model0 = roc_auc_score(y_test, y_test_pred)

my_dpi=100
pyplot.figure(figsize=(700/my_dpi,500/my_dpi), dpi=my_dpi)
pyplot.plot([0, 1], [0, 1], linestyle='--', color='#bdc3c7')

# AUROC model0
pyplot.plot(fpr_model0, tpr_model0, marker='',color='#0984e3',label='Calibrated eXtreme Gradient Boosting AUROC:'+ '+' + '{0:0.2f}'.format(auc_score_model0))

# AUROC model4 # this plot was already generated up in the code
pyplot.plot(fpr_model4, tpr_model4, marker='',color='#d63031',label='eXtreme Gradient Boosting AUROC:'+ '+' + '{0:0.2f}'.format(auc_score_model4))

pyplot.legend(loc='lower right')
pyplot.xlabel('1 - Specificity')
pyplot.ylabel('Sensitivity')
pyplot.ylim([0.0, 1.05])
pyplot.xlim([0.0, 1.0])

```

Out[115]:

(0.0, 1.0)

In [ ]:

Save results

In [76]:

```
db_name = 'OR'

# XGB
model_name = 'XGB'

features_name = 'outside_cpb'

# Best model
filename = '{}_{}_{}_model.sav'.format(model_name, features_name, db_name)

with open(filename, 'wb') as pickle_file:
    dill.dump(random_search.best_estimator_, pickle_file)
    print('model saved')

# CV model
filename = '{}_{}_{}_model_CV.sav'.format(model_name, features_name, db_name)

with open(filename, 'wb') as pickle_file:
    dill.dump(random_search, pickle_file)
    print('model saved')
```

APPENDIX IX ASSOCIATION OF CHLORIDE ION AND SODIUM-CHLORIDE DIFFERENCE WITH ACUTE KIDNEY INJURY AND MORTALITY IN CRITICALLY ILL PATIENTS

Tables

Supplemental Table 1. Multivariable logistic regression analysis for development of acute kidney injury

	Unweighted				Inverse Probability Weighted			
	OR	Lower 95% CI	Upper 95% CI	p-value	OR	Lower 95% CI	Upper 95% CI	p-value
mSID reference (32-34 mEq/L)								
<=26 (mEq/L)	1.33	1.05	1.69	0.02	1.38	1.08	1.75	0.01
26-28 (mEq/L)	1.11	0.91	1.36	0.31	1.13	0.92	1.39	0.25
28-30 (mEq/L)	1.09	0.93	1.28	0.27	1.11	0.94	1.3	0.22
30-32 (mEq/L)	1.07	0.93	1.22	0.37	1.06	0.92	1.21	0.45
34-36 (mEq/L)	1.17	1.02	1.35	0.02	1.17	1.01	1.34	0.03
36-38 (mEq/L)	1.14	0.97	1.33	0.11	1.12	0.96	1.32	0.16
38-40 (mEq/L)	1.40	1.16	1.68	<0.001	1.41	1.17	1.7	<0.001
40-42 (mEq/L)	1.46	1.18	1.81	0.001	1.46	1.17	1.82	0.001
>42 (mEq/L)	1.56	1.24	1.94	<0.001	1.52	1.21	1.92	<0.001
[Cl⁻] reference (104-106 mEq/L)								
<=94 (mEq/L)	1.24	1.02	1.51	0.03	1.26	1.03	1.55	0.03
94-96 (mEq/L)	1.18	0.94	1.49	0.16	1.19	0.94	1.51	0.15
96-98 (mEq/L)	1.19	0.97	1.45	0.09	1.18	0.97	1.45	0.11
98-100 (mEq/L)	1.2	1.01	1.44	0.04	1.19	0.99	1.42	0.06
100-102 (mEq/L)	1.07	0.90	1.26	0.43	1.07	0.90	1.27	0.44

102-104 (mEq/L)	1.01	0.86	1.19	0.88	1.01	0.86	1.19	0.90
106-108 (mEq/L)	1.04	0.89	1.22	0.62	1.03	0.88	1.22	0.69
108-110 (mEq/L)	1.04	0.87	1.24	0.66	1.02	0.85	1.22	0.83
110-112 (mEq/L)	0.92	0.74	1.13	0.42	0.9	0.73	1.12	0.35
112-114 (mEq/L)	1.24	0.99	1.56	0.06	1.19	0.95	1.5	0.13
114-116 (mEq/L)	1.17	0.87	1.56	0.29	1.18	0.87	1.59	0.29
>116 (mEq/L)	1.16	0.90	1.51	0.25	1.16	0.89	1.51	0.27
Age (*10 years)	1.12	1.09	1.16	<0.001				
Male	1.05	0.94	1.16	0.42				
Weight (*10 kg)	1.06	1.04	1.07	<0.001				
Hight (*10 cm)	0.95	0.90	1.00	0.051				
Ethnicity reference (African American)								
Asian	1.21	0.80	1.84	0.36				
Caucasian	0.75	0.66	0.86	<0.001				
Hispanic	0.81	0.66	0.98	0.03				
Other/Unknown	0.74	0.60	0.91	0.005				
CHF	1.22	1.10	1.36	<0.001				
PVD	1.09	0.91	1.31	0.36				
Hypertension	1.09	1.00	1.19	0.048				
COPD	1.00	0.91	1.11	0.93				

Diabetes	0.97	0.86	1.09	0.62				
Malignancy	0.92	0.45	1.88	0.81				
Heart rate (*10 bpm)	1.11	1.09	1.13	<0.001				
Systolic blood pressure (*10 mmHg)	1.02	1.01	1.03	0.008				
Temperature (°C)	0.87	0.83	0.91	<0.001				
WBC (*10³/L)	1.02	1.01	1.03	<0.001				
Platelets (*10⁴/L)	0.95	0.97	0.99	<0.001				
Hematocrit (%)	1.06	1.02	1.09	0.001				
Hemoglobin (g/dl)	0.86	0.78	0.94	0.001				
Albumin (mg/dl)	0.71	0.65	0.77	<0.001				
Bilirubin (mg/dl)	1.10	1.07	1.12	<0.001				
BUN (*10 mg/dl)	0.92	0.89	0.95	<0.001				
Glucose (*20 mg/dl)	1.00	0.99	1.01	0.80				
Creatinine (mg/dl)	1.93	1.79	2.09	<0.001				
Potassium (mEq/L)	1.07	1.01	1.13	0.02				
Uric acid (mg/dl)	0.92	0.83	1.02	0.11				

Abbreviations: OR, odds ratio; CI, confidence interval; [Cl⁻], serum chloride ion concentration; CHF, congestive heart disease; PVD, peripheral vascular disease; COPD, chronic obstructive pulmonary disease; WBC, white cell count; BUN, blood urea nitrogen

Supplemental Table 2. Multivariable logistic regression analysis for intensive care unit mortality

	Unweighted				Inverse Probability Weighted			
	OR	Lower 95% CI	Upper 95% CI	p-value	OR	Lower 95% CI	Upper 95% CI	p-value
mSID reference (32-34 mEq/L)								
<=26 (mEq/L)	1.05	0.79	1.40	0.74	1.38	1.08	1.75	0.01
26-28 (mEq/L)	0.93	0.72	1.20	0.57	1.13	0.92	1.39	0.25
28-30 (mEq/L)	0.76	0.61	0.94	0.01	1.11	0.94	1.30	0.22
30-32 (mEq/L)	0.98	0.82	1.18	0.84	1.06	0.92	1.21	0.45
34-36 (mEq/L)	1.03	0.85	1.24	0.79	1.17	1.01	1.34	0.03
36-38 (mEq/L)	1.30	1.06	1.60	0.01	1.12	0.96	1.32	0.16
38-40 (mEq/L)	1.22	0.95	1.56	0.13	1.41	1.17	1.70	<0.001
40-42 (mEq/L)	1.66	1.26	2.19	<0.001	1.46	1.17	1.82	0.001
>42 (mEq/L)	1.22	0.89	1.66	0.21	1.52	1.21	1.92	<0.001
[Cl⁻] reference (104-106 mEq/L)								
<=94 (mEq/L)	1.09	0.83	1.43	0.52	1.26	1.03	1.55	0.03
94-96 (mEq/L)	1.11	0.81	1.53	0.52	1.19	0.94	1.51	0.15
96-98 (mEq/L)	1.27	0.98	1.65	0.07	1.18	0.97	1.45	0.11
98-100 (mEq/L)	1.04	0.81	1.33	0.77	1.19	0.99	1.42	0.06
100-102 (mEq/L)	1.09	0.87	1.36	0.47	1.07	0.90	1.27	0.44
102-104 (mEq/L)	0.98	0.79	1.22	0.87	1.01	0.86	1.19	0.90
106-108 (mEq/L)	0.88	0.70	1.10	0.25	1.03	0.88	1.22	0.69
108-110 (mEq/L)	1.08	0.85	1.36	0.53	1.02	0.85	1.22	0.83
110-112 (mEq/L)	1.17	0.90	1.51	0.25	0.9	0.73	1.12	0.35
112-114 (mEq/L)	1.45	1.09	1.93	0.01	1.19	0.95	1.50	0.13

114-116 (mEq/L)	1.31	0.92	1.86	0.13	1.18	0.87	1.59	0.29
>116 (mEq/L)	1.65	1.22	2.23	0.001	1.16	0.89	1.51	0.27
Age (*10 years)	1.16	1.12	1.21	<0.001				
Male	1.00	0.87	1.15	0.97				
Weight (*10 kg)	0.98	0.96	1.01	0.17				
Hight (*10 cm)	0.98	0.92	1.05	0.62				
Ethnicity reference (African American)								
Asian	0.98	0.47	2.04	0.95				
Caucasian	1.45	1.17	1.79	0.001				
Hispanic	0.95	0.7	1.29	0.77				
Other/Unknown	1.42	1.05	1.93	0.02				
CHF	0.96	0.83	1.11	0.58				
PVD	1.19	0.94	1.52	0.15				
Hypertension	0.91	0.82	1.02	0.12				
COPD	1.01	0.88	1.15	0.93				
Diabetes	0.69	0.58	0.82	<0.001				
Malignancy	1.73	0.81	3.7	0.16				
Heart rate (bpm)	1.02	1.01	1.02	<0.001				
Systolic blood pressure (*10 mmHg)	0.94	0.92	0.96	<0.001				
Temperature (°C)	0.64	0.61	0.68	<0.001				
WBC (*10³/L)	1.02	1.01	1.03	<0.001				
Platelets (*10⁴/L)	0.91	0.94	0.97	<0.001				

Hematocrit (%)	1.19	1.14	1.24	<0.001				
Hemoglobin (g/dl)	0.62	0.55	0.70	<0.001				
Albumin (mg/dl)	0.49	0.44	0.55	<0.001				
Bilirubin (mg/dl)	1.07	1.05	1.10	<0.001				
BUN (*10 mg/dl)	0.94	0.90	0.97	0.001				
Glucose (*10 mg/dl)	1.02	1.01	1.03	0.003				
Creatinine (mg/dl)	1.82	1.65	2.00	<0.001				
Potassium (mEq/L)	1.14	1.06	1.22	<0.001				
Uric acid (mg/dl)	0.98	0.87	1.10	0.71				

Abbreviations: OR, odds ratio; CI, confidence interval; mSID, main strong ion difference; [Cl⁻], serum chloride ion concentration; CHF, congestive heart disease; PVD, peripheral vascular disease; COPD, chronic obstructive pulmonary disease; WBC, white cell count; BUN, blood urea nitrogen

Supplemental Table 3. Multivariable logistic regression analysis for hospital mortality

	OR	Lower 95% CI	Upper 95% CI	p-value
mSID reference (32-34 mEq/L)				
<=26 (mEq/L)	0.95	0.74	1.22	0.69
26-28 (mEq/L)	0.92	0.74	1.14	0.43
28-30 (mEq/L)	0.77	0.64	0.92	0.004
30-32 (mEq/L)	0.91	0.78	1.06	0.25
34-36 (mEq/L)	1.09	0.93	1.27	0.27
36-38 (mEq/L)	1.33	1.12	1.57	0.001

38-40 (mEq/L)	1.30	1.05	1.6	0.01
40-42 (mEq/L)	1.52	1.20	1.93	<0.001
>42 (mEq/L)	1.47	1.15	1.89	0.002
[Cl⁻] reference (104-106 mEq/L)				
<=94 (mEq/L)	1.27	1.02	1.58	0.04
94-96 (mEq/L)	1.34	1.04	1.74	0.03
96-98 (mEq/L)	1.26	1.01	1.58	0.04
98-100 (mEq/L)	1.15	0.93	1.41	0.19
100-102 (mEq/L)	1.16	0.96	1.4	0.11
102-104 (mEq/L)	1.03	0.86	1.24	0.71
106-108 (mEq/L)	0.94	0.78	1.13	0.49
108-110 (mEq/L)	1.13	0.93	1.38	0.21
110-112 (mEq/L)	1.26	1.02	1.57	0.04
112-114 (mEq/L)	1.41	1.10	1.8	0.006
114-116 (mEq/L)	1.26	0.93	1.71	0.14
>116 (mEq/L)	1.61	1.24	2.09	<0.001

Abbreviations: OR, odds ratio; CI, confidence interval; mSID, main strong ion difference; [Cl⁻], serum chloride ion concentration

Supplemental Table 4. Subgroup analyses for development of A) acute kidney injury and B) mortality in the intensive care unit, dividing into the cohort into four subgroups based on mSID and [Cl⁻] and identifying the β -coefficient with mSID and [Cl⁻] as continuous variables.

A.

Subgroup	mSID≤34 mEq/L and [Cl ⁻]≤106 mEq/L		mSID≤34 mEq/L and [Cl ⁻]>106 mEq/L		mSID>34 mEq/L and [Cl ⁻]≤106 mEq/L		mSID>34 mEq/L and [Cl ⁻]>106 mEq/L	
	β (lower 95% CI-upper 95% CI)	p-value	β (lower 95% CI-upper 95% CI)	p-value	β (lower 95% CI-upper 95% CI)	p-value	β (lower 95% CI-upper 95% CI)	p-value
mSID (mEq/L)	0.99 [0.95-1.02]	0.46	0.95 [0.92-0.97]	<0.001	1.03 [1.01-1.05]	0.008	1.10 [1.02-1.19]	0.01
[Cl⁻] (mEq/L)	0.98 [0.96-1.00]	0.11	1.01 [0.99-1.03]	0.27	0.98 [0.97-1.00]	0.01	0.98 [0.93-1.02]	0.32

B.

Subgroup	mSID≤34 mEq/L and [Cl ⁻]≤106 mEq/L		mSID≤34 mEq/L and [Cl ⁻]>106 mEq/L		mSID>34 mEq/L and [Cl ⁻]≤106 mEq/L		mSID>34 mEq/L and [Cl ⁻]>106 mEq/L	
	β (lower 95% CI-upper 95% CI)	p-value	β (lower 95% CI-upper 95% CI)	p-value	β (lower 95% CI-upper 95% CI)	p-value	β (lower 95% CI-upper 95% CI)	p-value
mSID (mEq/L)	0.99 [0.94-1.03]	0.53	0.98 [0.95-1.02]	0.36	1.02 [0.99-1.05]	0.12	1.15 [1.06-1.25]	0.001
[Cl⁻] (mEq/L)	0.98 [0.95-1.00]	0.10	1.04 [1.01-1.06]	0.002	0.99 [0.97-1.01]	0.46	1.09 [1.04-1.14]	0.0005

Abbreviations: mSID, main strong ion difference; $[\text{Cl}^-]$, serum chloride ion concentration; CI, confidence interval.

Figures

Supplemental Figure 1. Individual main strong ion difference plotted against serum bicarbonate ($r = 0.54$, 95% confidence interval 0.53 to 0.55, $p < 0.001$)

Abbreviations: mSID, main strong ion difference

Supplemental Figure 2. Individual serum chloride ion concentration plotted against main strong ion difference ($r = -0.62$, 95% confidence interval -0.63 to -0.61 , $p < 0.001$)

Abbreviations: mSID, main strong ion difference; $[Cl]^-$, serum chloride ion concentration

Code

Dataset creation and analysis (R+SQL)

```
---
title: "01_dataset_creation"
author: "Miguel Armengol"
date: "2/20/2019"
output: html_document
---

# Environment

```{r message=FALSE, warning=FALSE}
library(bigrquery)
library(summarytools)
```

```

library(readr)
library(stringr)
library(sqldf)
library(dplyr)
library(readxl)
```

Set up BigQuery related functions

This chunks also creates the run_query and get_sql function.

```{r setup, include=FALSE}

project_id <- "hst-953-2018"
options(httr_oauth_cache=FALSE)
run_query <- function(query){
  data <- query_exec(query, project=project_id,
use_legacy_sql=FALSE,max_pages = Inf)
  return(data)
}

# function for reading sql files
getSQL <- function(filepath){
  con = file(filepath, "r")
  sql.string <- ""

  while (TRUE){
    line <- readLines(con, n = 1)

    if ( length(line) == 0 ){
      break
    }

    line <- gsub("\\t", " ", line)

    if(grepl("--",line) == TRUE){
      line <- paste(sub("--","/*",line),"*/")
    }

    sql.string <- paste(sql.string, line)
  }

  close(con)
  return(sql.string)
}
```

Loading queries and extracting the data

Loads all queries from the sql files in the extraction folder and runs them into RBigQuey to extract the data.

```{r}
#folders contained in sql folder
#cat(list.files('sql'), sep = '\n')

```

```

# AKI related variables
first_creat<-run_query(getSQL("sql/aki/first_creat.sql"))
before_adm_creat<-run_query(getSQL("sql/aki/before_adm_creat.sql"))
adm_creat<-run_query(getSQL("sql/aki/adm_creat.sql"))
chronicAKI<-run_query(getSQL("sql/aki/chronicAKI.sql"))
first_rrt<-run_query(getSQL("sql/aki/first_rrt.sql"))
peakcreat48h<-run_query(getSQL("sql/aki/peakcreat48h.sql"))
peakcreat7days<-run_query(getSQL("sql/aki/peakcreat7days.sql"))

dialysis<-run_query(getSQL("sql/aki/dialysis.sql"))
dialysis_first7days<-run_query(getSQL("sql/aki/dialysis_first7days.sql"))

# rest of the variables
bg_24_48<-run_query(getSQL("sql/bg_24_48.sql"))
bg_first<-run_query(getSQL("sql/bg_first.sql"))
demographics<-run_query(getSQL("sql/demographics.sql"))
lab_first<-run_query(getSQL("sql/lab_first.sql"))
lab_24<-run_query(getSQL("sql/lab_24.sql"))
lab_48<-run_query(getSQL("sql/lab_48.sql"))
outcomes<-run_query(getSQL("sql/outcomes.sql"))
pasthistory<-run_query(getSQL("sql/pasthistory.sql"))
sid_first<-run_query(getSQL("sql/sid_first.sql"))
sid_24_48<-run_query(getSQL("sql/sid_24_48.sql"))
sid_twa_24<-run_query(getSQL("sql/sid_twa_24.sql"))
sid_twa_48<-run_query(getSQL("sql/sid_twa_48.sql"))
temp_day1<-run_query(getSQL("sql/temp_day1.sql"))
temp_day2<-run_query(getSQL("sql/temp_day2.sql"))
urine_output_24_48<-run_query(getSQL("sql/urine_output_24_48.sql"))
vent_bin_first_2days<-run_query(getSQL("sql/vent_bin_first_2days.sql"))
vitals_day1<-run_query(getSQL("sql/vitals_day1.sql"))
vitals_day2<-run_query(getSQL("sql/vitals_day2.sql"))

...

## AKI Dataset creation

```{r}
#We need to exclude patients who already had AKI before ICU admin)
AKIdevelopment<-subset(baseline_creat, !(baseline_creat$patientunitstayid
%in% chronicAKI$patientunitstayid) )
AKIdevelopment<-left_join(AKIdevelopment, peakcreat7days)
AKIdevelopment<-left_join(AKIdevelopment, peakcreat48h)
AKIdevelopment$position<-NULL
```

Windsorization

Clean data by means of winsorization, i.e., by shrinking outlying
observations to the border of the main part of the data.
The creatinine values have been winsorize replacing the extreme observations
using 95% and 5% percentiles.

```{r}
AKIdevelopmentW<-AKIdevelopment

```

```

## Winsorization: Replace the extreme observations using 95% and 5%
percentiles
winsorize_x = function(x, cut = 0.05){
  cut_point_top <- quantile(x, 1 - cut, na.rm = T)
  cut_point_bottom <- quantile(x, cut, na.rm = T)
  i = which(x >= cut_point_top)
  x[i] = cut_point_top
  j = which(x <= cut_point_bottom)
  x[j] = cut_point_bottom
  return(x)
}

AKIdevelopmentW$creat1<-winsorize_x(AKIdevelopmentW$creat1)
AKIdevelopmentW$peakcreat7d<-winsorize_x(AKIdevelopmentW$peakcreat7d)
AKIdevelopmentW$peakcreat48h<-winsorize_x(AKIdevelopmentW$peakcreat48h)

#Now we turn all empty values into 0

AKIdevelopmentW$creat1[is.na(AKIdevelopmentW$creat1)]<-0
AKIdevelopmentW$peakcreat7d[is.na(AKIdevelopmentW$peakcreat7d)]<-0
AKIdevelopmentW$peakcreat48h[is.na(AKIdevelopmentW$peakcreat48h)]<-0

...

###List of patients with AKI and offset

```{r}

AKIdevelopmentW['cr7undercreat1']<-
AKIdevelopmentW$peakcreat7d/AKIdevelopmentW$creat1
AKIdevelopmentW['cr48hdundercreat1']<-AKIdevelopmentW$peakcreat48h-
AKIdevelopmentW$creat1
AKIdevelopmentW['AKIoffset']<-NA
AKIdevelopmentW['AKIstage']<-NA

#nested if else function

i <- function(if_stat, then) {
  if_stat <- lazyeval::expr_text(if_stat)
  then <- lazyeval::expr_text(then)
  sprintf("ifelse(%s, %s, ", if_stat, then)
}

e <- function(else_ret) {
  else_ret <- lazyeval::expr_text(else_ret)
  else_ret
}

ie <- function(...) {
  args <- list(...)

  for (i in 1:(length(args) - 1) ) {
    if (substr(args[[i]], 1, 6) != "ifelse") {
      stop("All but the last argument, need to be i functions.", call. =
FALSE)
    }
  }
}

```

```

}
if (substr(args[[length(args)]], 1, 6) == "ifelse"){
  stop("Last argument needs to be an e function.", call. = FALSE)
}
args$final <- paste(rep(')', length(args) - 1), collapse = '')
eval_string <- do.call('paste', args)
eval(parse(text = eval_string))
}

AKIdevelopmentW$AKIstage <-
  ie(
    i(AKIdevelopmentW$cr7undercreat1>=3, 3),
    i(AKIdevelopmentW$cr7undercreat1>=2, 2),
    i(AKIdevelopmentW$cr7undercreat1>=1.5, 1),
    i(AKIdevelopmentW$cr48hdundercreat1>=0.3, 1),
    e(0)
  )

AKIdevelopmentW$AKIoffset <-
  ie(
    i(AKIdevelopmentW$cr7undercreat1>=3, AKIdevelopmentW$peakcreat7d_offset),
    i(AKIdevelopmentW$cr7undercreat1>=2, AKIdevelopmentW$peakcreat7d_offset),

i(AKIdevelopmentW$cr7undercreat1>=1.5, AKIdevelopmentW$peakcreat7d_offset),
    i(AKIdevelopmentW$cr48hdundercreat1>=0.3,
AKIdevelopmentW$peakcreat48h_offset),
    e(0)
  )

library(dplyr)
AKIlist<-AKIdevelopmentW %>%
  select(patientunitstayid, AKIstage
         # , AKIoffset The nephrologist suggested not to use this variable
since we can not infer when AKI was developed from it. Removing it to avoid
confusion.
         )
...

### Final List of patients with AKI and offset (with RRT)

Offset is not reliable so we exclude it.

```{r}
# Dr Danziger said the offset is not reliable for telling when the patient
exactly developed AKI so he advised not to use it

AKIlist_final<-left_join(AKIlist, first_rrt)

AKIlist_final$AKIstage <-
  ie(
    i(!is.na(AKIlist_final$first_rrtoffset), 3),
    e(AKIlist_final$AKIstage)
  )

# The nephrologist suggested not to use this variable since we can not infer
when AKI was developed from it. Removing it to avoid confusion.

```

```

# AKIlist_final$AKIOffset <-
#   ie(
#     i(!is.na(AKIlist_final$first_rrtoffset),
AKIlist_final$first_rrtoffset),
#     e(AKIlist_final$AKIOffset)
#   )

AKIlist_final$first_rrtoffset<-NULL
AKIlist_final[is.na(AKIlist_final)]<-0
AKIlist_final$rrt_bin<-NULL
```

New Variables

```{r}
admission_vars<-run_query(getSQL("sql/admission_vars.sql"))
mean_first_day<-run_query(getSQL("sql/mean_first_day.sql"))
total_na_cl_volume_day1<-run_query(getSQL("sql/total_na_cl_volume_day1.sql"))
intake_day1_missing_explain<-
run_query(getSQL("sql/intake_day1_missing_explain.sql"))
reliable_fluids_data<-run_query(getSQL("sql/all_reliable.sql"))
```

Merging all datasets

We are using a left join to join them

```{r}
# Exclude non reliable data ICU-year
sid_aki_study<-inner_join(demographics,reliable_fluids_data)

# left join = all.x=TRUE
sid_aki_study<-Reduce(function(...) merge(..., all.x=TRUE), list(
  sid_aki_study

, bg_first
, bg_24_48

, lab_first
, lab_24
, lab_48

, outcomes
, pasthistory
, sid_first
, sid_24_48
, sid_twa_24
, sid_twa_48
, temp_day1
, temp_day2
, urine_output_24_48

, vent_bin_first_2days

, vitals_day1
, vitals_day2

```

```

# aki vars
, first_creat
, before_adm_creat
, adm_creat
, chronicAKI
, first_rrt
, peakcreat48h
, peakcreat7days
, AKIlist_final

# new variables requested
, admission_vars
, mean_first_day
, total_na_cl_volume_day1
, intake_day1_missing_explain
, dialysis_first7days

))

#cat(names(sid_aki_study),sep = '\n')
` ``

```

According to the attached file, we have 12,551 patients with fluid data after inclusion and exclusion criteria were applied.

If fluid data=NA, appears_in_intakeoutput=0, receives_intake_at somepoint=0, then I think those are missing value (n=680).

If fluid data=NA, appears_in_intakeoutput=1; receives_intake_at somepoint=0, then I guess those patients took no fluid intake (n=4,733). ← That could be too many...

If fluid data=NA, appears_in_intakeoutput=1, receives_intake_at somepoint=1, I do not know if those are missing value or from our mistake (n=13,568). The number of patients are too many, so we should check further. In other words, we could miss important fluids in this table.
https://docs.google.com/spreadsheets/d/17BtBk7zkSggUxGriTbJtJt0VrFKpnBQxLVyywR_iHKM/edit#gid=100723980

Please look at those 13,568 patients and check if there are some variables that have large amount of cellvaluenumeric, let's say, more than 500 ml. That could be important fluid that we missed. If you find that kind of data, please send them to me.

Best,

```

# Dataset report

`` `{r}
view(dfSummary(sid_aki_study))
view(dfSummary())
` ``

```

```

# Export dataset

```{r}
write.csv(sid_aki_study, 'sid_aki_study.csv', row.names = F)
```

---
title: "02"
output: html_document

Environment

```{r}
library(dplyr)
library(Amelia)
```

Creating new variables

```{r, results='hide'}
#For outcomes/predictors with substantial missig data, IPW should be used?
#Exclude CKD, rrt before admision
df_1<-sid_aki_study%>%
  filter(creat1<=3)%>%
  mutate(aki=case_when(.$AKIstage==0~0, .$AKIstage%in%c(1,2,3)~1))%>%
  mutate(gender=case_when(.$gender=="Female"~0,
    .$gender=="Male"~1,
    TRUE~as.numeric(NA)),
    pre_rrt=case_when(.$rrt_bin==1&first_rrtoffset<=0~1,
    TRUE~0))%>%filter(pre_rrt==0)

df_1$ethnicity[df_1$ethnicity == "" |
  df_1$ethnicity == "Native American" |
  df_1$ethnicity == "Other/Unknow"] <- "Other/Unknown"

df_1$surgery<-NA
df_1$surgery[df_1$hospitalAdmitSource == "Operating Room" |
  df_1$hospitalAdmitSource == "PACU"] <- 1
df_1$surgery[df_1$hospitalAdmitSource == "Acute Care/Floor" |
  df_1$hospitalAdmitSource == "Chest Pain Center" |
  df_1$hospitalAdmitSource == "Direct Admit" |
  df_1$hospitalAdmitSource == "Emergency Department" |
  df_1$hospitalAdmitSource == "Floor" |
  df_1$hospitalAdmitSource == "ICU" |
  df_1$hospitalAdmitSource == "ICU to SDU" |
  df_1$hospitalAdmitSource == "Observation" |
  df_1$hospitalAdmitSource == "Other" |
  df_1$hospitalAdmitSource == "Other Hospital" |
  df_1$hospitalAdmitSource == "Other ICU" |
  df_1$hospitalAdmitSource == "Recovery Room" |
  df_1$hospitalAdmitSource == "Step-Down Unit (SDU)"] <- 0

#Categorizing Na, Cl, and SID
df_2<-df_1%>%mutate(cl_adm_cat=case_when(.$chloride_first<98 ~ 1,

```



```

                                .$chloride_first>=98 & .$chloride_first<=110 ~
0,
                                .$chloride_first>110 ~ 2,
                                TRUE ~ as.numeric(NA)),
na_adm_cat=case_when(.$sodium_first>145 ~ 2,
                                .$sodium_first<=145 & .$sodium_first>=135 ~
0,
                                .$sodium_first<135 ~ 1,
                                TRUE ~ as.numeric(NA)),
sid_adm_cat=case_when(.$SID_first>37 ~ 2,
                                .$SID_first<=37 & .$SID_first>=31 ~ 0,
                                .$SID_first<31 ~ 1,
                                TRUE ~ as.numeric(NA)),

# 24 hours variables start here
cl_24_min_cat=case_when(.$chloride_min_24<98 ~ 1,
                                .$chloride_min_24>=98 & .$chloride_min_24<=110
~ 0,
                                .$chloride_min_24>110 ~ 2,
                                TRUE ~ as.numeric(NA)),
cl_24_max_cat=case_when(.$chloride_max_24<98 ~ 1,
                                .$chloride_max_24>=98 & .$chloride_max_24<=110
~ 0,
                                .$chloride_max_24>110 ~ 2,
                                TRUE ~ as.numeric(NA)),
na_24_min_cat=case_when(.$sodium_min_24>145 ~ 2,
                                .$sodium_min_24<=145 & .$sodium_min_24>=135
~ 0,
                                .$sodium_min_24<135 ~ 1,
                                TRUE ~ as.numeric(NA)),
na_24_max_cat=case_when(.$sodium_max_24>145 ~ 2,
                                .$sodium_max_24<=145 & .$sodium_max_24>=135
~ 0,
                                .$sodium_max_24<135 ~ 1,
                                TRUE ~ as.numeric(NA)),
sid_24_min_cat=case_when(.$SID_min_24>37 ~ 2,
                                .$SID_min_24<=37 & .$SID_min_24>=31 ~ 0,
                                .$SID_min_24<31 ~ 1,
                                TRUE ~ as.numeric(NA)),
sid_24_max_cat=case_when(.$SID_max_24>37 ~ 2,
                                .$SID_max_24<=37 & .$SID_max_24>=31 ~ 0,
                                .$SID_max_24<31 ~ 1,
                                TRUE ~ as.numeric(NA)),
cl_24_mean_cat=case_when(.$chloride_twa_24<98 ~ 1,
                                .$chloride_twa_24>=98 & .$chloride_twa_24<=110
~ 0,
                                .$chloride_twa_24>110 ~ 2,
                                TRUE ~ as.numeric(NA)),
cl_24_mean_catd=case_when(.$chloride_twa_24>116 ~ 13,

.$chloride_twa_24>114&.$chloride_twa_24<=116 ~ 12,
.$chloride_twa_24>112&.$chloride_twa_24<=114 ~ 11,
.$chloride_twa_24>110&.$chloride_twa_24<=112 ~ 10,
.$chloride_twa_24>108&.$chloride_twa_24<=110 ~ 9,

```

```

.$chloride_twa_24>106&.$chloride_twa_24<=108 ~ 8,
.$chloride_twa_24>104&.$chloride_twa_24<=106 ~ 7,
.$chloride_twa_24>102&.$chloride_twa_24<=104 ~ 6,
.$chloride_twa_24>100&.$chloride_twa_24<=102 ~ 5,
.$chloride_twa_24>98&.$chloride_twa_24<=100~ 4,
.$chloride_twa_24>96&.$chloride_twa_24<=98
~ 3,
.$chloride_twa_24>94&.$chloride_twa_24<=96
~ 2,
.$chloride_twa_24<=94~1,
TRUE ~ as.numeric(NA)),
na_24_mean_cat=case_when(.$sodium_twa_24>145 ~ 2,
.$sodium_twa_24<=145 & .$sodium_twa_24>=135
~ 0,
.$sodium_twa_24<135 ~ 1,
TRUE ~ as.numeric(NA)),
na_24_mean_catd=case_when(.$sodium_twa_24>146 ~ 11,
.$sodium_twa_24>144&.$sodium_twa_24<=146 ~
10,
.$sodium_twa_24>142&.$sodium_twa_24<=144 ~
9,
.$sodium_twa_24>140&.$sodium_twa_24<=142 ~
8,
.$sodium_twa_24>138&.$sodium_twa_24<=140 ~
7,
.$sodium_twa_24>136&.$sodium_twa_24<=138 ~
6,
.$sodium_twa_24>134&.$sodium_twa_24<=136 ~
5,
.$sodium_twa_24>132&.$sodium_twa_24<=134 ~
4,
.$sodium_twa_24>130&.$sodium_twa_24<=132 ~
3,
.$chloride_twa_24<=130~1,
TRUE ~ as.numeric(NA)),
sid_24_mean_cat=case_when(.$SID_twa_24>37 ~ 2,
.$SID_twa_24<=37 & .$SID_twa_24>=31 ~ 0,
.$SID_twa_24<31 ~ 1,
TRUE ~ as.numeric(NA)),
sid_24_mean_catd=case_when(.$SID_twa_24>44 ~ 11,
.$SID_twa_24>42 & .$SID_twa_24<=44 ~ 10,
.$SID_twa_24>40 & .$SID_twa_24<=42 ~ 9,
.$SID_twa_24>38 & .$SID_twa_24<=40 ~ 8,
.$SID_twa_24>36 & .$SID_twa_24<=38 ~ 7,
.$SID_twa_24>34 & .$SID_twa_24<=36 ~ 6,
.$SID_twa_24>32 & .$SID_twa_24<=34 ~ 5,
.$SID_twa_24>30 & .$SID_twa_24<=32 ~ 4,
.$SID_twa_24>28 & .$SID_twa_24<=30 ~ 3,
.$SID_twa_24>26 & .$SID_twa_24<=28 ~ 2,
.$SID_twa_24<=26 ~ 1,
TRUE ~ as.numeric(NA)),

```

```

cl_24_mean_catq=cut(chloride_twa_24,breaks=10,labels=c("1","2","3","4","5","6",
"7","8","9","10")),
  delta_cl_d1=chloride_twa_24-chloride_first,
  delta_na_d1=sodium_twa_24-sodium_first,
  delta_sid_d1=SID_twa_24-SID_first,

# 48 hours variables start here
  cl_48_min_cat=case_when(.$chloride_min_48<98 ~ 1,
    ~ 0,
    .$chloride_min_48>=98 & .$chloride_min_48<=110
    ~ 0,
    .$chloride_min_48>110 ~ 2,
    TRUE ~ as.numeric(NA)),
  cl_48_max_cat=case_when(.$chloride_max_48<98 ~ 1,
    ~ 0,
    .$chloride_max_48>=98 & .$chloride_max_48<=110
    ~ 0,
    .$chloride_max_48>110 ~ 2,
    TRUE ~ as.numeric(NA)),
  na_48_min_cat=case_when(.$sodium_min_48>145 ~ 2,
    ~ 0,
    .$sodium_min_48<=145 & .$sodium_min_48>=135
    ~ 0,
    .$sodium_min_48<135 ~ 1,
    TRUE ~ as.numeric(NA)),
  na_48_max_cat=case_when(.$sodium_max_48>145 ~ 2,
    ~ 0,
    .$sodium_max_48<=145 & .$sodium_max_48>=135
    ~ 0,
    .$sodium_max_48<135 ~ 1,
    TRUE ~ as.numeric(NA)),
  sid_48_min_cat=case_when(.$SID_min_48>37 ~ 2,
    ~ 0,
    .$SID_min_48<=37 & .$SID_min_48>=31 ~ 0,
    .$SID_min_48<3 ~ 1,
    TRUE ~ as.numeric(NA)),
  sid_48_max_cat=case_when(.$SID_max_48>37 ~ 2,
    ~ 0,
    .$SID_max_48<=37 & .$SID_max_48>=31 ~ 0,
    .$SID_max_48<31 ~ 1,
    TRUE ~ as.numeric(NA)),
  cl_48_mean_cat=case_when(.$chloride_twa_48<98 ~ 1,
    ~ 0,
    .$chloride_twa_48>=98 & .$chloride_twa_48<=110
    ~ 0,
    .$chloride_twa_48>110 ~ 2,
    TRUE ~ as.numeric(NA)),
  cl_48_mean_catd=case_when(.$chloride_twa_48>116 ~ 13,
    ~ 0,
    .$chloride_twa_48>114&.$chloride_twa_48<=116 ~ 12,
    ~ 0,
    .$chloride_twa_48>112&.$chloride_twa_48<=114 ~ 11,
    ~ 0,
    .$chloride_twa_48>110&.$chloride_twa_48<=112 ~ 10,
    ~ 0,
    .$chloride_twa_48>108&.$chloride_twa_48<=110 ~ 9,
    ~ 0,
    .$chloride_twa_48>106&.$chloride_twa_48<=108 ~ 8,
    ~ 0,
    .$chloride_twa_48>104&.$chloride_twa_48<=106 ~ 7,
    ~ 0,
    .$chloride_twa_48>102&.$chloride_twa_48<=104 ~ 6,

```

```

.$chloride_twa_48>100&.$chloride_twa_48<=102 ~ 5,

.$chloride_twa_48>98&.$chloride_twa_48<=100~ 4,
      .$chloride_twa_48>96&.$chloride_twa_48<=98
~ 3,
      .$chloride_twa_48>94&.$chloride_twa_48<=96
~ 2,
      .$chloride_twa_48<=94~1,
      TRUE ~ as.numeric(NA)),
na_48_mean_cat=case_when(.$sodium_twa_48>145 ~ 2,
      .$sodium_twa_48<=145 & .$sodium_twa_48>=135
~ 0,
      .$sodium_twa_48<135 ~ 1,
      TRUE ~ as.numeric(NA)),
na_48_mean_catd=case_when(.$sodium_twa_48>146 ~ 11,
      .$sodium_twa_48>144&.$sodium_twa_48<=146 ~
10,
      .$sodium_twa_48>142&.$sodium_twa_48<=144 ~
9,
      .$sodium_twa_48>140&.$sodium_twa_48<=142 ~
8,
      .$sodium_twa_48>138&.$sodium_twa_48<=140 ~
7,
      .$sodium_twa_48>136&.$sodium_twa_48<=138 ~
6,
      .$sodium_twa_48>134&.$sodium_twa_48<=136 ~
5,
      .$sodium_twa_48>132&.$sodium_twa_48<=134 ~
4,
      .$sodium_twa_48>130&.$sodium_twa_48<=132 ~
3,
      .$chloride_twa_48<=130~1,
      TRUE ~ as.numeric(NA)),
sid_48_mean_cat=case_when(.$SID_twa_48>37 ~ 2,
      .$SID_twa_48<=37 & .$SID_twa_48>=31 ~ 0,
      .$SID_twa_48<31 ~ 1,
      TRUE ~ as.numeric(NA)),
sid_48_mean_catd=case_when(.$SID_twa_48>44 ~ 11,
      .$SID_twa_48>42 & .$SID_twa_48<=44 ~ 10,
      .$SID_twa_48>40 & .$SID_twa_48<=42 ~ 9,
      .$SID_twa_48>38 & .$SID_twa_48<=40 ~ 8,
      .$SID_twa_48>36 & .$SID_twa_48<=38 ~ 7,
      .$SID_twa_48>34 & .$SID_twa_48<=36 ~ 6,
      .$SID_twa_48>32 & .$SID_twa_48<=34 ~ 5,
      .$SID_twa_48>30 & .$SID_twa_48<=32 ~ 4,
      .$SID_twa_48>28 & .$SID_twa_48<=30 ~ 3,
      .$SID_twa_48>26 & .$SID_twa_48<=28 ~ 2,
      .$SID_twa_48<=26 ~ 1,
      TRUE ~ as.numeric(NA)),

cl_48_mean_catq=cut(chloride_twa_48,breaks=10,labels=c("1","2","3","4","5","6",
"7","8","9","10")),
delta_cl_d2=chloride_twa_48-chloride_first,
delta_na_d2=sodium_twa_48-sodium_first,
delta_sid_d2=SID_twa_48-SID_first

```

```

)%>%

mutate_at(c("gender", "pasthistory_chf", "pasthistory_pvd", "pasthistory_hyperte
nsion", "pasthistory_copd", "pasthistory_diabetes", "cl_adm_cat", "na_adm_cat", "s
id_adm_cat"

, "cl_24_min_cat", "cl_24_max_cat", "na_24_min_cat", "na_24_max_cat", "sid_24_min_
cat", "sid_24_max_cat", "ethnicity", "cl_24_mean_cat", "na_24_mean_cat", "sid_24_m
ean_cat", "surgery", "sid_24_mean_catd", "cl_24_mean_catd", "na_24_mean_catd"

, "cl_48_min_cat", "cl_48_max_cat", "na_48_min_cat", "na_48_max_cat", "sid_48_min_
cat", "sid_48_max_cat", "cl_48_mean_cat", "na_48_mean_cat", "sid_48_mean_cat", "su
rgery", "sid_48_mean_catd", "cl_48_mean_catd", "na_48_mean_catd"
), as.factor)

#Focusing on medical patients
df_me<-df_2%>%filter(surgery==0)
```

Dataset report

```{r}
view(dfSummary(df_me))
```

Selecting variables

Main predictors: (you may want to use 1 OR 2 in a model)
1. Rough categories: cl_24_mean_cat, na_24_mean_cat, sid_24_mean_cat
2. Detailed categories: cl_24_mean_catd, na_24_mean_catd, sid_24_mean_catd

Patients' baseline characteristics: (we may want to use all of the variables)
age_fixed, gender, ethnicity, pasthistory_chf, pasthistory_pvd, pasthistory_hyperte
nsion, pasthistory_copd, pasthistory_diabetes, pasthistory_cld, pasthistory_mali
gnancy, creatl, potassium_adm, bicarbonate_adm, ua_adm, heartrate_adm,
systolic_adm, temperature_adm

Variables for the first 24 Horus: (we may want to use all of the variables)
I have used the mean
glucose_mean_day1, potassium_mean_day1, bicarbonate_mean_day1,
albumin_mean_day1, hemoglobin_mean_day1, wbc_mean_day1, platelets_mean_day1,
heartrate_mean_day1, systolic_mean_day1, temperature_mean_day1

```{r}
## Excluding not desired variables
df_me_selected<-df_me%>%select( -one_of (c('apacheadmissiondx'
, 'unitadmitsource', 'BMI_group', 'unitType', 'hospitalAdmitSource'
, 'hospLOS_prior_ICUadm_days', 'unitlos_days', 'hosplos_days', 'ventdays_days'
, 'hospitalmortality', 'icumortality', 'urineoutput_total_24'
, 'urineoutput_total_48', 'creatloffset', 'first_rrtoffset', 'rrt_bin'
, 'peakcreat48h_offset', 'peakcreat48h', 'peakcreat7d', 'peakcreat7d_offset'
, 'peakcreat7d_to_discharge_offsetgap', 'cl_adm_cat', 'na_adm_cat'
, 'sid_adm_cat', 'cl_24_min_cat', 'cl_24_max_cat', 'na_24_min_cat'

```

```

, 'na_24_max_cat' , 'sid_24_min_cat' , 'sid_24_max_cat' , 'cl_24_mean_cat'
, 'cl_24_mean_catd' , 'na_24_mean_cat' , 'na_24_mean_catd' , 'sid_24_mean_cat'
, 'sid_24_mean_catd' , 'cl_24_mean_catq' , 'delta_cl_d1' , 'delta_na_d1'
, 'delta_sid_d1', 'AKIstage')) )
```

Imputation on selected variables

Mean Imputation

```{r}
df_me_selected_imputed<-df_me_selected

for(i in 1:ncol(df_me_selected_imputed)){
  df_me_selected_imputed[is.na(df_me_selected_imputed[,i]), i] <-
mean(df_me_selected_imputed[,i], na.rm = TRUE)
}
```

Recovering non-imputable columns

We are not imputing variables that:

A) *Almost sure they are MNAR*
fio2_first, peep_first, peep_min_48,

B) *Could be MNAR*
pao2_first, paco2_first, ph_first, aniongap_first, basedeficit_first,
baseexcess_first,

fio2_min_24, pao2_min_24, pH_min_24, aniongap_min_24, basedeficit_min_24,
baseexcess_min_24, peep_min24, fio2_max_24, pao2_max_24, paco2_max_24,
pH_max_24, aniongap_max_24, basedeficit_max_24, baseexcess_max_24,
peep_max24,

fio2_min_48, pao2_min_48, paco2_min_48, pH_min_48, aniongap_min_48,
basedeficit_min_48, baseexcess_min_48, fio2_max_48, pao2_max_48,
paco2_max_48, pH_max_48, aniongap_max_48, basedeficit_max_48,
baseexcess_max_48, peep_max_48

We are also not imputing those variables with more than 70% missing values.

Studying variables missing values

```{r}
missing_values<-as.data.frame(sapply(df_me_selected, function(x)
round(sum(is.na(x))/length(x),2)))
names(missing_values)<- 'percentage_missing'

toomany_missing<-
rownames(subset(missing_values,missing_values$percentage_missing>0.70))
```

Final replacement

We are not imputing values listed above, so we replace the imputed columns
with the original ones

```

```

```{r}
keep.orniginal<-select(df_me_selected,c(
# MNAR
fio2_first, peep_first, peep_min_48,
pao2_first, paco2_first, pH_first, aniongap_first, basedeficit_first,
baseexcess_first,

fio2_min_24, pao2_min_24, pH_min_24, aniongap_min_24, basedeficit_min_24,
baseexcess_min_24, peep_min24, fio2_max_24, pao2_max_24, paco2_max_24,
pH_max_24, aniongap_max_24, basedeficit_max_24, baseexcess_max_24,
peep_max24,

fio2_min_48, pao2_min_48, paco2_min_48, pH_min_48, aniongap_min_48,
basedeficit_min_48, baseexcess_min_48, fio2_max_48, pao2_max_48,
paco2_max_48, pH_max_48, aniongap_max_48, basedeficit_max_48,
baseexcess_max_48, peep_max_48,
# too many missing values
fio2_first,pao2_first,paco2_first,pH_first,basedeficit_first,baseexcess_first
,peep_first,fio2_min_24,basedeficit_min_24,peep_min24,fio2_min_48,basedeficit
_min_48,peep_min_48,fio2_max_24,basedeficit_max_24,peep_max24,fio2_max_48,bas
edeficit_max_48,peep_max_48,albumin_first,bilirubin_first,wbc_first,platelets
_first,ua_adm,

# we don't want to impute categorical variables
aki,

# concentration variables (MAYBE WE WANT TO IMPUTE THEM IN THE FUTURE)

total_na_meq_day1, total_cl_meq_day1, total_volume_ml_day1

)
)

# Final replacement

df_me_selected_imputed[,names(keep.orniginal)] <- keep.orniginal
```

Multiple imputation

The following code has not been finished yet:

```{r eval=FALSE, include=FALSE}
#min and max possible values need to be set per variable.
bds <- matrix(c(7, 0, 100), nrow = 1, ncol = 3)# we need a matrix of
bounds(number of column to bound, min, max)
relevant_data_imputation_matrix<-amelia(df_me_selected, parallel =
"multicore",idvars = 'Case_Name',noms =
c('Gender','Category','Q_AUT50MAP_total_min','Q_AUT65MAP_total_min','Q_AOT65M
AP_total_min','Q_total_pressors_formula_mg')
#, ords = #Ords is only for ingegers, not
floats
, bounds = bds
)
FinalHemodynamicsDataset<-df_me_selected$imputations$imp5

```

```

names(df_me_selected)[names(df_me_selected)=="STS_RiskAlgorithm"]<-
"STS_RiskAlgorithm_multiple_imp"
summary(df_me_selected$STS_RiskAlgorithm_multiple_imp)
#STS risk cannot be negative
# to avoid problems when using log
#there can not be 0 values in a log(variable) in a logreg
```

Dataset report

```{r}
view(dfSummary(df_me_selected_imputed))
```

Data export

```{r}
write.csv(df_me_selected_imputed, 'df_me_selected_imputed.csv', row.names = F)
```

---
title: "03_train_test_df_creation"
author: "Miguel Armengol"
date: "2/20/2019"
output: html_notebook

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

Loading libraries

```{r message=FALSE, warning=FALSE}
library(dplyr)
library(caret)
```

Creating de-identified id

```{r}
df_me_selected_imputed<-
df_me_selected_imputed%>%filter(!is.na(aki))%>%select(-one_of (c('AKIstage')
))

df_me_selected_imputed['id']<- seq.int(nrow(df_me_selected_imputed))
```

Splitting de-identified data into testing and training, balanced version.

We want the data to be sampled randomly but always the same way and we want
to be sure that train and test must be balanced.

```{r}
## set the seed to make our partition reproducible
set.seed(123)

```



```

# createDataPartition: "the random sampling is done within the levels of y
when y is a factor in an attempt to balance the class distributions within
the splits."
## 75% of the sample size
train_idx <- createDataPartition(as.factor(df_me_selected_imputed$aki), times
= 1, p = 0.75, list=F)
train <- df_me_selected_imputed[train_idx, ]
test <- df_me_selected_imputed[-train_idx, ]
round(prop.table(table(train$aki)),2)
round(prop.table(table(test$aki)),2)
```

Exporting datasets

```{r}
write.csv(train, 'train.csv', row.names = F)
write.csv(test, 'test.csv', row.names = F)
```

```