CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Practice variation in sepsis management in the eICU database

Master's thesis in Computer science and engineering

Nils Nordmark and Patrick Royer

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2021

Master's thesis 2021

Practice variation in sepsis management in the eICU database

Nils Nordmark and Patrick Royer

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2021

Practice variation in sepsis management in the eICU database
Nils Nordmark and Patrick Royer

© Nils Nordmark and Patrick Royer, 2021.

Supervisor: Fredrik Johansson, Department of Computer Science and Engineering
Examiner: Richard Johansson, Department of Computer Science and Engineering

Master's Thesis 2021
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Description of the picture on the cover page (if applicable)

Typeset in LaTeX
Gothenburg, Sweden 2021

Practice variation in sepsis management in the eICU database
Nils Nordmark and Patrick Royer
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Sepsis is a life-threatening syndrome triggered by an infection. Despite international guidelines, sepsis management varies between sites. This unwanted practice variation may affect negatively the quality of care but enables theoretically retrospective studies for finding optimal treatment strategies. The goal of the present master thesis was to find a relevant way to model practice variation in the management of sepsis-induced circulatory failure.

Sepsis patients were retrieved from the eICU critical care database. Nine treatments and nine relevant covariates were selected from domain knowledge. Practice variation was successively investigated in four intensive care units and four hospitals respectively. Missing values were imputed using forward filling, linear interpolation and the Multiple Imputation by Chained Equations algorithm. For each analysis, two logistic regression models were successively trained and calibrated. The first model yielded propensity estimates for being treated in a particular site given covariates. The second model was trained on the subset of patients having a reasonable probability of being treated in all the sites and yielded propensity estimates for being treated with a particular treatment. Practice variation was first defined as the expected difference in propensity for treatment between two sites and then characterized for a given patient with given covariates as the distance between the likelihood to get a certain treatment in a certain site s and the expectation of the likelihood to get the same treatment over all the sites with the assumption that this patient had the same likelihood of being treated in s as he had in the actual data.

A pairwise comparisons of propensity for treatments between sites revealed variations up to 12.5%. At a patient level, practice variation distributions showed a similar positively skewed distribution for both analyses and revealed variations up to 5.8%.

We demonstrated the feasibility of modeling practice variation among distinct sites in the management of sepsis-induced circulatory failure using retrospective data. The significance of this variation should be further evaluated by investigating which treatment policies are associated with a better outcome.

Keywords: sepsis, calibration, practice variation, importance sampling, multiple imputation

Acknowledgements

We want to thank our supervisor Fredrik Johansson for his support during our project.

Nils Nordmark & Patrick Royer, Gothenburg, October 2021

Contents

List of Abbreviations

AI Artificial intelligence. 5, 15
APACHE Acute Physiology And Chronic Health Evaluation. 13

BAC balanced accuracy. i, xvii, 22, 24, 38, 41, 48, 51, 61, 62

ECE expected calibration error. 9, 22, 38, 41, 48, 51, 62
eICU eICU Collaborative Research Database. I, 3, 12–14, 18, 28, 29, 59

ICD International Classification of Diseases. 12, 13
ICU Intensive Care Unit. i, xv, xvii, XXXVII–XXXIX, 3, 5, 12, 13, 19, 26, 29–49, 59, 61–63

KNN K Nearest Neighbors. 7

LOCF Last Observation Carried Forward. 7, 60

MAR missing at random. 6, 7, 17, 19, 60
MCAR missing completely at random. 6, 7
MCE maximum calibration error. XXXVIII, 9, 10, 22, 24, 38, 41, 48, 51, 62
MICE Multiple Imputation by Chained Equations. 7, 8, 19, 21, 35, 60
MNAR missing not at random. 6

NOCB Next Observation Carried Backward. 7

PCA principal component analysis. 63
PV Practice variation. 25, 62, 63

RF Random forest. 15

SMOTE Synthetic Minority Over-sampling Technique. 61
SOFA Sequential Organ Failure Assessment. 12, 13, 31, 32
SQL Structured Query Language. I, 28

List of Figures

List of Tables

1

Introduction

Sepsis is a common syndrome defined as life-threatening organ dysfunction caused by a dysregulated host response to infection [1]. The cardiovascular system is the most frequently compromised organ during sepsis [2]. This syndrome has recently been recognized as a global health priority by the World Health Organization [3]. Indeed, 11 million sepsis-related deaths and about 49 million cases of sepsis were reported worldwide in 2017 [4]. Furthermore, a significant number of sepsis patients require Intensive Care Unit (ICU) admission to receive advanced and costly treatments. Thus, almost one third of ICU patients in high income countries have sepsis [5]. The current outbreak of COVID 19 highlighted sorely the lack of medical resources and the need for critical care optimization even in Western countries [6].

Critical care societies regularly make recommendations for the management of patients in sepsis [7]. For various reasons [8], these recommendations are not uniformly implemented which led to practice variation across ICUs [9, 10]. Unwarranted practice variation may be undesirable for guaranteeing uniform quality of care. Some patients may be exposed to real harm from not receiving the treatment they need or potential harm from receiving treatment that they do not need. However, this undesired practice variation [11] could give the opportunity to retrospectively study treatment effects. If patients in similar condition are treated differently across sites and doctors, there is hope of identifying practices that work particularly well for that condition.

The eICU Collaborative Research Database (eICU) [12] is a freely available multicenter ICU database for critical care research that contains high granularity data for 139,367 ICU patients admitted between 2014 and 2015 at 208 hospitals located throughout the United States of America. Notably, this database annotates admissions with hospital ID, enabling the study of practice variation across sites.

The goal of the present master thesis was to investigate practice variation in the management of sepsis-induced circulatory failure using patient data from the eICU Collaborative Research Database (eICU) database [12]. Studying practice variation involves addressing two questions: I) can "similar" patients be identified across different sites in the database, and II) can distinct treatment practices can be observed between these sites. In order to answer these two practical questions, we developed a method for learning site-specific treatment policies which consisted of two successive logistic regression models. The first one enabled the selection of comparable

patients with respect to relevant clinical features who where hospitalized in distinct sites. The second one provided an estimation of treatment variation across those sites in the cohort of similar patients.

2

Theory

Working with observational data often implies handling missing values. Different strategies are presented in this chapter especially multiple imputation methods that require dealing with multiple imputed datasets.

Quantifying practice variation between different sites involves estimating site-specific treatment policies. We defined below what is a policy, a way to evaluate it and the importance of a calibrated estimator to obtain an accurate policy evaluation.

2.1 Learning treatment policies from observational data

Randomized clinical trials are considered the gold standard for evaluating treatment effects since they minimize confounding by prospectively and randomly allocating treatments to patients [13]. However, carrying out such trials may be impossible due to ethical issues, limited time or insufficient funds [14]. The growing digitalization of the health-care system leads to an increase in electronic medical records which has raised new interest in retrospective observational studies [15], especially in the critical care setting [16]. There currently exist two significant openly-available critical care databases, eICU (multi-center) and MIMIC-III (single-center) [17].

An increasing number of works are based on these databases and the medical applications of AI. In the ICU, applications of AI have mainly concerned machine learning, which aims to generate knowledge from data. Machine learning typically includes three categories of techniques: supervised, unsupervised and reinforcement learning [18]. There have been many notable applications of supervised learning algorithms in sepsis. For example, automated algorithms that identify patients at-risk of having sepsis [19], a gradient tree boosting model for predicting sepsis and septic shock [20] and even simpler rule-based algorithms that could identify at-risk patients [21, 22]. In unsupervised learning, research mainly have been exploratory and hypothesis generating, but more recent progress utilizing clustering algorithms have made it possible to believe that such approaches will be in practical use in the near future [23, 24]. Reinforcement learning is arguably considered to be the most immature branch of machine learning w.r.t. technology readiness for intensive care applications [25]. Recently, a reinforcement learning model trained on MIMIC-

III claimed to learn optimal treatment strategies for sepsis in intensive care [26]. However, the reliability of the policy evaluation estimates raised concerns that such evaluation is ill-supported for some clinical decisions [27, 28].

A policy is defined as the probability of assigning treatment t to a subject i with covariates x_i and its quality is measured by the expected cumulative outcomes V. Evaluating the value V of a learned policy (target policy π) with data generated by another policy (behavior policy μ) is known as off-line policy evaluation [29]. Importance sampling is a classical method to correct the discrepancy between the distributions under the target policy and the behavior policy [30]. Indeed, the patient i with covariates x_i, treatment t_i and outcome Y_i is sampled from μ but importance sampling method reweights the data like if they were drawn from π. A global value $\hat{V}$ of the target policy could be estimated by taking the average over patients as follows (adapted from [29]):

$$\hat{V}(\pi) = \frac{1}{n} \sum_{i=1}^{n} \frac{\pi(x_i, T = t)}{\mu(x_i, T = t)} \times Y_i \tag{2.1}$$

2.2 Missing data

Missing data are prevalent in observational studies in healthcare as data are often routinely electronically collected data for clinical reasons, rather than research [31]. Missing data in observational studies is a pervasive challenge as it is nearly ubiquitous and its impact on inference can be substantial. In most studies the reporting of missing data and the underlying mechanisms for its missingness are often below par. For example, of 262 studies published in 2010 in 3 leading epidemiologic journals, 68% had insufficiently reported the amount of missing data to be quantified by reviewers and 46% had not clearly distinguished and classified the underlying mechanisms of the missing data [32].

The underlying mechanisms for missing data are commonly classified as missing completely at random (MCAR), missing at random (MAR), or missing not at random (MNAR) [33, 34, 31].

- MCAR: the missingness of a value is unrelated to the observed parameters and the unknown value itself. It occurs, for example, when the physician forgot to register a clinical observation or when a medical device broke down during a measurement. This scenario represents the strongest assumption and doesn't bias the analysis of data since the missing values have the same distribution as the available data. However, this assumption is often unrealistic in practice.

- MAR: the missingness of a value is related to the observed parameters but not to the missing value itself. This case arises when a measurement is not performed because it is not clinically indicated. For example, the patient has no sign of infection and no microbiological sample is taken.

- MNAR: the missingness of a value depends on the value itself. This pattern

is typically found in quality of life studies in intensive care survivors because disabled patients may not answer the survey.

Handling missing data is necessary as most statistical learning models do not work with datasets with missing values. Common methods to deal with this issue are:

- Deletion
  - (i) Subjects with missing values are deleted to perform a "complete case analysis".
- Imputation methods
  - (i) Mean/median/mode: the missing values are replaced with the mean, median or mode of the observed values of the variable.
  - (ii) K Nearest Neighbors (KNN): the missing values are replaced by an aggregate value (mean/median/mode) of the k most similar observations in the dataset.
  - (iii) Linear Regression - if a linear relation is assumed between variables, missing values can then be predicted by a linear regression model fitted with variables from the complete data.
  - (iv) Last Observation Carried Forward (LOCF) and Next Observation Carried Backward (NOCB): these methods are often used for time-series data especially when the period of time is short and the trend is monotonous without large fluctuation. Missing values are replaced either by the last observed value (LOCF) or by the first observation after the missing value (NOCB).
  - (v) Linear Interpolation: this is an alternative to LOCF and NOCB for time-series data as missing values are filled using both the observations before and after.
  - (vi) Multiple Imputation by Chained Equations (MICE) [35]: is a multiple imputation method that creates a specified number of imputed versions of the original dataset. For each dataset, missing values are iteratively imputed using a predictive model. For each iteration, each specified variable is imputed using the other variables and these iterations continue until convergence.

The deletion method and the imputation with mean/median/mode usually produce unbiased estimates only for MCAR values whereas the other above methods may work for MAR values. However, among the latter, only MICE avoids standard errors to shrink because it generates n imputed datasets with different imputed missing values [35].

2.3 Multiple imputation and predictions

With MICE a predictive model consequently has to be performed on n imputed datasets. As such, MICE yields n imputation-specific regression coefficients. These n regression coefficients can be pooled according to Rubin's rules [36] to get pooled regression coefficients. There are two alternatives to obtain an individual prediction from such a predictive model with n imputed datasets:

1. A prediction is obtained from the pooled regression coefficients.
2. A pooled prediction is obtained from pooling the imputation-specific predictions of each imputed dataset.

The pros and cons of these two alternatives are not yet established [37]. Similarly, a measure of model performance can be obtained in two ways:

1. Pooled performance strategy: a pooled estimate of model performance is obtained from imputation-specific estimate of model performance.
2. Pooled prediction strategy: model performance is estimated from the pooled predictions.

Figure 2.1 which illustrates these two alternatives is adapted from [37].

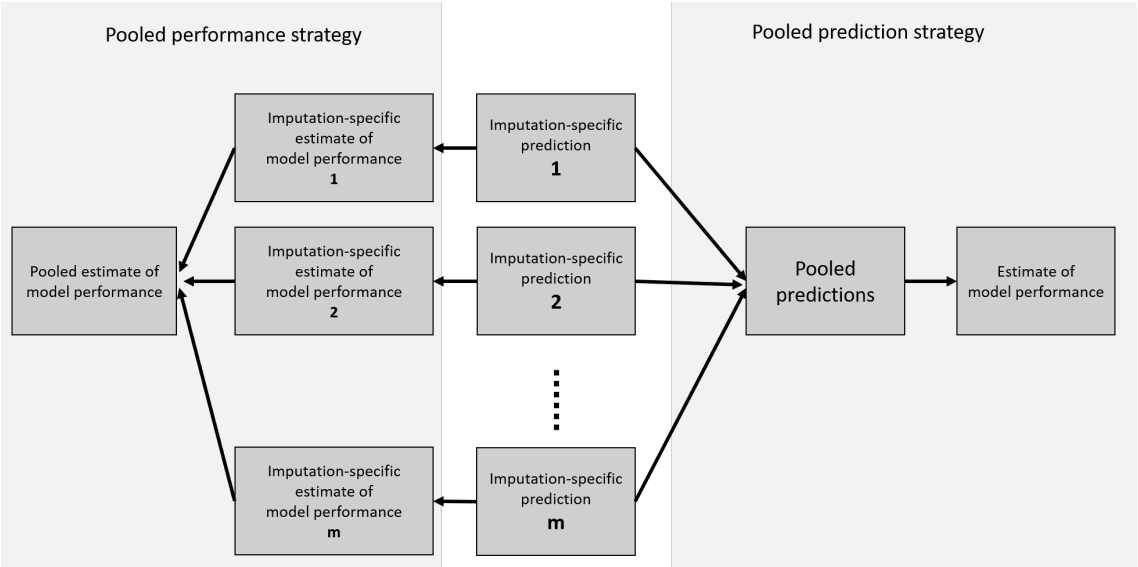


Figure 2.1: Model performance estimation with multiple imputed datasets (adapted from Wood AM et al.).

2.4 Model Calibration

In a medical database, the true behaviour policy that generated the data is generally unknown i.e we don't have access to the set of rules used by the clinicians to treat the patients. However, this behaviour policy can be estimated from the data by

training various type of models such as logistic regression, random forests or neural network. Thus, it is possible to estimate the probability of treatment given a set of relevant covariates. These estimates have to be calibrated i.e the probabilities of treatment under the estimated behaviour policy model should correspond to the true probabilities because the accuracy of the importance sampling-based off-policy policy evaluation described above depends on the quality of the calibration [38].

In a binary classification setting with the input $X\ in\ \mathcal{X}$ and the label $Y\ in\ \mathcal{Y} = \{1, 0\}$, let's define the classifier C with $C(X) = (\hat{Y}, \hat{P})$, where $\hat{Y}$ is a class prediction and $\hat{P}$ its confidence. C is perfectly calibrated if the following equation holds [39]:

$$P(\hat{Y} = Y \mid \hat{P} = p) = p, \ \ \forall p \in [0, 1] \tag{2.2}$$

2.4.1 Reliability diagrams

Model calibration can be illustrated by reliability diagrams that plot the expected sample accuracy as a function of confidence. In case of a perfectly calibrated model, the plot shows the identity function. To construct reliability diagrams, the predictions are first grouped into M bins. Then the accuracy and the average confidence are computed in each bin according to the following formulas:

$$accuracy(B_m) = \frac{1}{|B_\mathrm{m}|} \sum_{i \in B_m} 1(\hat{y}_i = y), \tag{2.3}$$

where B_m represents the bin m and $\hat{y}_i$ and y_i are the predicted and true class labels for sample i

$$confidence(B_m) = \frac{1}{|B_\mathrm{m}|} \sum_{i \in B_m} \hat{p}_i, \tag{2.4}$$

where $\hat{p}_i$ is the confidence for sample i.

The quality of model calibration can be then estimated by computing the expected calibration error (ECE) and the maximum calibration error (MCE) [40]. ECE computes a weighted average of the bins' accuracy/confidence difference as follows:

$$ECE = \sum_{m=1}^{M} \frac{|B_\mathrm{m}|}{n} |accuracy(B_\mathrm{m}) - confidence(B_\mathrm{m})| \tag{2.5}$$

MCE computes the worst-case deviation between confidence and accuracy as follows:

$$MCE = \max_{m \in \{1,\ldots,M\}} |accuracy(B_\mathrm{m}) - confidence(B_\mathrm{m})| \tag{2.6}$$

MCE may be preferred in mission-critical applications where reliable confidence measures are essential.

3

Methods

The eICU database contains medical information about 139 367 patients hospitalized in ICU in 208 distinct hospitals across the United States during 2014 and 2015. All these patients had various diagnosis but we focused on patients with sepsis at ICU admission. The registered medical information is heterogeneous and consists of demographic data, vital sign measurements, laboratory tests, diagnosis and treatment information, among other things. An ICU stay corresponds to a limited period of time where a patient receives critical care in a critical care unit. The collected clinical information during an ICU stay is documented in the database as time series data and the time step varied with the type of data. Each ICU stay is uniquely associated to a hospital ID and an ICU ID (a hospital may contain several ICUs). We successively analysed practice variation between ICUs and hospitals. We first started by creating a dataset which contained sepsis patients with relevant clinical variables and treatments coded as time series with a four-hour time step that we called period as shown in Table 3.1.

We then trained and calibrated two estimators. The first model estimated the propensity of being hospitalized in a given site given relevant clinical variables. The patients who had a sensible probability of being treated in every site were considered comparable and were therefore selected for the next step. The second model estimated the propensity of receiving a given treatment given clinical variables and site. Finally this last model was used to estimate the treatment policy in each site. The implementation details are described in the following subsections.

ICU stay	Period	Site	v_1	...	v_n
1	1	2	-	...	-
1	2	2	-	...	-
1	3	2	-	...	-
1	4	2	-	...	-
2	1	1	-	...	-
2	2	1	-	...	-
2	3	1	-	...	-
⋮	⋮	⋮	⋮	⋮	⋮
n	1	2	-	...	-
n	2	2	-	...	-
n	3	2	-	...	-
n	4	2	-	...	-
n	5	2	-	...	-

Table 3.1: Dataset as time series with four-hour time steps called period. The column ICU stay corresponded to unique patient ICU stay, The column site contained either ICU ID or hospital ID depending on which practice variation analysis was performed. Variables v_1, ... v_n corresponded to relevant clinical information.

3.1 Data extraction and preprocessing

3.1.1 Extraction of patients in sepsis

The identification of sepsis patients from electronic medical records can be performed using two methods [1]:

1. The explicit method is based on International Classification of Diseases (ICD) codes [41]. Patients with severe sepsis and septic shock are explicitly coded in eICU with ICD codes 995.92 or 785.52 and R65.20 or R65.2 respectively.

2. The implicit method consists in retrieving three criteria in a specified period:

   (a) order of body fluid culture

   (b) antibiotic prescription

   (c) organ failure represented by an increased Sequential Organ Failure Assessment (SOFA) score > 1 [42]

Concomitant orders for body fluid cultures and antibiotics set the time of suspected infection. The addition of a SOFA score > 1 defines sepsis.

Since body fluid cultures are rarely documented in the eICU (90% of ICU stays have no data) we chose the explicit method [43]. However, in order to reduce the risk of retrieving miscoded sepsis patients, we decided to combine the explicit method with

the other two available criteria of the implicit method, i.e., antibiotic prescription and SOFA score > 1. Since the suspected infection time could not be characterized, we decided to define a cohort of patients admitted to ICU for sepsis using the following inclusion criteria:

(i) At least one ICD-9 or ICD-10 for sepsis or septic shock during the ICU stay (coding procedure and clinical activities are not necessarily synchronous).

(ii) ICU stays with antibiotic prescription and SOFA score > 1 in a period ranging from one day before and one day after ICU admission.

(iii) ICU stays with an admission diagnosis of sepsis registered in the Acute Physiology And Chronic Health Evaluation (APACHE). APACHE is a severity-of-disease classification system which is applied within the first day of ICU admission [44].

In the eICU database, the data were recorded heterogeneously across hospitals. As a consequence, the absence of documentation for a specific treatment doesn't indicate that this treatment was not administered [45]. To prevent any systematic bias, hospitals with potential poor data completion were excluded. Since we focused on vasopressor and intravenous fluid therapies with a 4-hour time window, hospitals with less than 6 daily records on average for these treatments were excluded as previously described [26].

Finally, ICU stays with patients under 18 years, with a length of stay < 6 hours and with a documentation of care limitation such as "Comfort measures only", "No vasopressors/inotropes" and "No augmentation of care" were excluded. Due to ICU readmissions, one patient could be included several times for distinct ICU stays.

3.1.2 Extraction of sites

We aim at studying practice variation across hospitals and ICUs. One hospital may contain several ICUs. In order to get a sufficient number of patients per sites, we only kept the sepsis patients who belonged to the four most represented hospitals and ICUs.

3.1.3 Extraction of treatments

The clinical manifestations of sepsis are heterogeneous because the function of all the organs can be impaired. We decided to focus on the cardiovascular system which is most commonly affected. The treatment of acute circulatory failure relies on a combination of intravenous fluid and vasopressor infusion and aims at restoring adequate organ perfusion [46].

For each patient, the vasopressor infusion rate and the amount of received fluid are registered at irregular intervals in the eICU database.

Extraction of vasopressors The main vasopressors clinically used are: Norepinephrine, Epinephrine, Dopamine, Phenylephrine and Vasopressin.

The standard infusion rate is usually expressed in units/min for Vasopressin and in mcg/kg/min for Norepinephrine, Epinephrine, Dopamine and Phenylephrine.

Infusion rates are inconsistently registered in the eICU database and the following expressions could be found in the infusiondrug table: 'mcg/kg/min', 'mcg/kg/hr', 'ml/hr', 'mcg/min', 'mcg/hr', 'units/min' and 'none'. Infusion rates expressed in 'ml/hr' were converted in 'mcg/min' using concentrations inferred from the medication table.

Dopamine infusion rates were almost entirely expressed in 'ml/hr'. Since concentration information was missing, infusion rates could not be converted and patients who received Dopamine were excluded.

Then 'mcg/min' and 'mcg/hr' were converted into 'mcg/kg/min' using patient's weight. The weights were retrieved by taking an average from the tables: patient, intakeoutput and infusiondrug. When a patient had no registered weight in either of those tables, the patient's weight was imputed with the average weight of the cohort of sepsis patients. Finally, Vasopressin, Epinephrine and Phenylephrine were converted to Norepinephrine equivalent using previously described formulas [47] and as shown in Table 3.2.

Vasopressor	Unit	Norepinephrine equivalent infusion rate in mcg/kg/min
Norepinephrine	mcg/kg/min	infusion rate
Norepinephrine	mcg/kg/hr	infusion rate * 0.167
Norepinephrine	mcg/min	infusion rate / weight
Norepinephrine	ml/hr	infusion rate * 0.533 / weight
Epinephrine	mcg/kg/min	infusion rate
Epinephrine	ml/hr	infusion rate * 0.533 / weight
Phenylephrine	mcg/kg/min	infusion rate * 0.1
Phenylephrine	mcg/hr	infusion rate * 0.0017 / weight
Phenylephrine	ml/hr	infusion rate * 0.266 / weight
Vasopressin	units/min	infusion rate * 2.5
Vasopressin	ml/hr	infusion rate * 0.025

Table 3.2: Conversion to obtain Norepinephrine equivalent dose.

Extraction of fluids The amount of intravenous fluid received by each patient was obtained at different time intervals from the eICU intakeOutput table. Intravenous fluid could consist in crystalloids, colloids and blood products.

3.1.4 Extraction of covariates

We selected from domain knowledge the covariates that could influence the treatment strategy of the circulatory failure (Table 3.3).

Based on Random forest models, the Artificial intelligence (AI) Clinician study [26] found that these covariates had a significant importance for predicting both the clinicians' and the AI policy for circulation optimization.

Covariate	Measured in unit
Systolic Blood Pressure	mmHg
Mean Blood Pressure	mmHg
Diastolic Blood Pressure	mmHg
Shock Index (Heart Rate/Systolic Blood Pressure)	(min-1/mmHg)
Lactates	mmol/l
Base Excess	mmol/l
PaO2 / FiO2	mmHg
SaO2	%
Creatinine	mg/dl

Table 3.3: Selected covariates.

(i) Covariates that describe the circulation system:

Blood pressure is a continuous variable measured in mmHg. It varies during each heartbeat between a maximum and a minimum value, respectively systolic and diastolic blood pressure. The mean blood pressure is the average of blood pressure over a cardiac cycle and is considered as the tissue perfusion pressure in clinical practice.

Shock index is defined by the ratio of heart rate and systolic blood pressure. It can be used to rapidly assess the cardiovascular system. High values are correlated with shock.

(ii) Covariates that reflect the quality of global perfusion:

During sepsis, the circulatory system (basically the heart and the blood vessels) can be impaired and a state of shock occurs if the tissue oxygen demand exceeds the supply. Inadequate tissue perfusion leads to impaired cellular aerobic (i.e., with oxygen) respiration and lactate is produced. Lactate is a global biological marker of shock and blood lactate elevation is partly responsible for acidosis (blood acidification). The level of acidosis is automatically characterized by blood gas analysis that report a calculated base excess.

Base excess is the theoretical amount of acid needed to get back a neutral blood pH and its value decreases when acidosis increases.

(iii) Covariates that reflect the quality of local perfusion:

All the organs may suffer during sepsis but the kidneys are particularly vulnerable in case of low perfusion pressure. Creatinine is a blood marker of renal failure.

(iv) Covariates that inform on fluid tolerance:

Most oxygen in the blood is reversibly bound to hemoglobin.

SaO2 or arterial oxygen saturation corresponds to the percentage of hemoglobin binding sites occupied by oxygen.

PaO2 or partial pressure of oxygen in mmHg corresponds to the small amount of oxygen directly dissolved in the blood.

Sepsis-induced lung injury is characterized by increased alveolar–capillary permeability, pulmonary edema and reduced blood oxygen level. Administration of oxygen is required when PaO2 or SaO2 are critically low.

FiO2 or fraction of inspired oxygen represents the percentage of oxygen concentration in the air delivered to the patient. Fio2 can range from 0.21 to 1.

In case of severe sepsis-induced lung injury, fluid infusion may worsen pulmonary edema and oxygen diffusion.

All intensive care patients are continuously monitored for blood pressure, heart rate and SaO2 but the frequencies of measurements vary with the clinical context. Lactate, PaO2 and Creatinine are intermittently obtained from blood samples if clinically needed.

Preliminary covariates. The preliminary covariates were extracted to five datasets: vital periodic, vital aperiodic, laboratory, fluids and vasopressor. Observations that contained outliers were removed using domain knowledge as shown in Table 3.4.

For each preliminary covariate, a value was associated with an observation offset.

Values in the vital periodic table were registered every five minutes whereas values in the other four tables were not regularly documented. Indeed, the vital periodic table contained continuous monitoring whereas the other tables contained intermittent information. The frequency of this intermittent information depends on the medical context.

Blood pressure can be measured non-invasively or invasively depending on the medical context. The non-invasive procedure is often preferred when the clinical situation is stable whereas invasive measurements are privileged in case of hemodynamic instability. The non-invasive and the invasive blood pressure columns were merged into one variable for each component of blood pressure (systolic, mean and diastolic) respectively.

Covariate or Treatment	Outliers' limits
Vital Periodic	
Invasive Systolic Blood Pressure	[0 - 300]
Invasive Mean Blood Pressure	[0 - 200]
Invasive Diastolic Blood Pressure	[0 - 200]
Heart Rate	[0 - 250]
Sao2	[0 - 100]
Vital Aperiodic	
Non-Invasive Systolic Blood Pressure	[0 - 300]
Non-Invasive Mean Blood Pressure	[0 - 200]
Non-Invasive Diastolic Blood Pressure	[0 - 200]
Laboratory	
Creatinine	[0 - 25]
Lactate	[0 - 50]
FiO2	[0.2 - 1]
PaO2	[0 - 700]
Base Excess	[-50 - 50]
Fluids	
Fluids	[0 - 10 000]
Vasopressor	
Vasopressor	[0 - 10]

Table 3.4: Outliers' limits for preliminary covariates.

Definitive covariates. Definitive covariates were obtained by combining the preliminary ones as follows:

- Shock Index = Heart Rate / Systolic Blood Pressure
- P/F = PaO2 / FiO2

Finally, all the preliminary covariates (five datasets) were combined into one dataset.

Normalization. All the features were normalized with min max scaling (0-1) before continuing with imputation.

3.1.5 Imputation

We made the assumption that the variables were normally distributed and that most of the missing values were MAR. Indeed, if a patient had not a registered value at a particular point in time, then we considered that no measurement was clinically indicated. The imputation methods were adapted to the characteristics of the features.

Covariate imputation.

- Linear interpolation was used for the features based on continuous monitoring.

Missing values were imputed from the two closest available values if those were observed within a specific time-length.

- Forward filling was selected for 'intermittent' features.

The physician point of view was adopted to discriminate between the features. For example, in the clinical setting, medical actions are based on immediate data like monitoring and delayed data like blood samples. If for some reason the monitoring stops working for a short while, the physician will guess the invisible values based on the latest trend. On the other hand, the values of the latest blood samples may be considered 'unaltered' during a certain period depending on the context. Our decisions for covariate imputations are described in Table 3.5.

Covariate	Monitoring	Imputation method	Length (minutes)
Shock Index	Continuous	Linear Interpolation	2880
SaO2	Continuous	Linear Interpolation	2880
Systolic Blood Pressure	Continuous	Linear Interpolation	2880
Mean Blood Pressure	Continuous	Linear Interpolation	2880
Diastolic Blood Pressure	Continuous	Linear Interpolation	2880
Lactate	Intermittent	Forward Fill	2880
Creatinine	Intermittent	Forward Fill	2880
P/F	Intermittent	Forward Fill	2880
Base Excess	Intermittent	Forward Fill	2880

Table 3.5: Imputation method for each covariate.

Treatment imputation. We made the assumption that there was no missing values for vasopressors and fluids.

- Vasopressors:

  (i) Missing values before the first non-missing value and after the last non-missing value were replaced by 0 because the patient did not receive vasopressor

  (ii) The remaining missing values were imputed by forward filling since the dose was considered constant between two consecutive observation offsets.

- Fluids:

  (i) All the missing values were replaced by 0 because the registered values in eICU correspond to cumulative measurements up to the current observation offset.

Time range after ICU admission. The time range of our study was set to three days after ICU admission. As such, we divided the time series data in 18 periods of 4 hours by grouping the features as shown in the Table 3.6.

Covariate or treatment	Unit	Formula
Systolic Blood Pressure	mmHg	mean
Mean Blood Pressure	mmHg	mean
Diastolic Blood Pressure	mmHg	mean
Shock Index (Heart Rate/Systolic Blood Pressure)	(min-1/mmHg)	mean
Lactates	mmol/l	mean
Base Excess	mmol/l	mean
PaO2 / FiO2	mmHg	mean
SaO2	%	mean
Creatinine	mg/dl	mean
Vasopressor	mcg/kg/min	max
Fluids		
Input total	ml	cumsum
Input 4 hours	ml/4h	sum

Table 3.6: Covariates and treatments for each period.

Grouping the features into 4-hour time windows created missing values (MAR values) since the time steps with a registered value could vary between features.

Final dataset The remaining missing values were imputed performing MICE and yielded five imputed datasets without any missing values.

3.1.6 Defining the treatment variable

Vasopressors and fluids doses were discretized in three classes respectively. Class 0 represented the absence of vasopressor or fluid infusion. Class 1 and 2 corresponded to vasopressors and fluids doses below and above the median respectively. The final treatment variable T was obtained by nine combinations of vasopressors [0, low, high] and fluids [0, low, high] classes as shown in Table 3.7.

	Treatment								
	1	2	3	4	5	6	7	8	9
Vasopressor	0	0	0	low	low	low	high	high	high
Fluids	0	low	high	0	low	high	0	low	high

Table 3.7: Treatments 1-9 with their combination of vasopressor and fluid classes.

3.1.7 Train and test sets

Each final imputed dataset was split in training and test set, comprising 80% and 20% of patients, respectively. Moreover, the training and validation sets contained

the same patients across the imputed datasets.

3.2 Training estimators

We defined:

- X as the set of vectors containing the 9 covariates [systolic blood pressure, mean blood pressure, diastolic blood pressure, shock index, lactate, P/F, saO2, creatinine, base excess].
- $T_{(\text{vasopressor, fluid})}$ as the set of 9 treatments {(0, 0), (0, low), (0, high), (low, 0), (low, low), (low, high), (high, 0), (high, low), (high, high)} or {1, 2, 3, 4, 5, 6, 7, 8, 9}.
- S as the set of 4 sites {1, 2, 3, 4}.

Each patient i was associated to a site $s \in S$ and each couple (patient, period) was associated to one treatment $t \in T$ and to one covariates vector $x_{i,j} \in X$. The number $j \in \{1, ..., 18\}$ of $x_{i,j}$ depended on the length of stay of patient i.

Table 3.8 gives an example of the dataset before training.

Patient	Period	Site	Covariates	Treatment
1	1	2	$x_{1,1}$	3
1	2	2	$x_{1,2}$	2
1	3	2	$x_{1,3}$	2
1	4	2	$x_{1,4}$	5
2	1	1	$x_{2,1}$	1
2	2	1	$x_{2,2}$	2
2	3	1	$x_{2,3}$	1
⋮	⋮	⋮	⋮	⋮
n	1	2	$x_{n,1}$	4
n	2	2	$x_{n,2}$	5
n	3	2	$x_{n,3}$	7
n	4	2	$x_{n,4}$	7
n	5	2	$x_{n,5}$	5

Table 3.8: Dataset example before training.

3.2.1 Estimator of site propensity

First we defined a logistic regression model f to estimate the probability of belonging to site s given covariates $x_{i,j}$:

$$f(s \mid x_{i,j}) \approx P(S = s \mid X = x_{i,j}) \tag{3.1}$$

We then defined the set C that corresponded to the set of covariates $x_{i,j}$ that were likely to be observed in all sites s:

$$C = \{x_{i,j} : \forall s : f(s \mid x_{i,j}) > \epsilon\} \tag{3.2}$$

$$P = \{patient : retained\ periods > \delta\} \tag{3.3}$$

After training, each patient i has a varying number of periods with covariates $x_{i,j}$ that belonged to C as shown in Table 3.9.

Patient	Period	$f(S=1 \mid x_{i,j}) > \epsilon$	$f(S=2 \mid x_{i,j}) > \epsilon$	$f(S=3 \mid x_{i,j}) > \epsilon$	$f(S=4 \mid x_{i,j}) > \epsilon$	$x_{i,j}$ in C
1	1	True	True	True	True	True
1	2	False	True	True	True	False
1	3	False	False	True	True	False
1	4	True	True	True	True	True
2	1	True	False	False	True	False
2	2	True	True	True	True	True
2	3	True	True	True	False	False
⋮	⋮	⋮	⋮	⋮		
n	1	False	True	True	True	False
n	2	True	True	True	True	True
n	3	False	True	True	False	False
n	4	True	True	True	True	True
n	5	True	True	True	True	True

Table 3.9: Dataset example after training.

For further analysis, we selected only the patients who had more than δ% of their periods with covariates $x_{i,j}$ that belonged to C and we defined the set U which contained the covariates of this sub set of patients.

Sensible ϵ and δ thresholds were determined after plotting the number of patients possibly retained for training the second estimator.

3.2.2 Estimator of treatment propensity

The next step consisted in training a second logistic regression model g that gave an estimate $g_s(s,t)$ of the propensity for treatment t given covariate $x_{i,j}$ and site s:

$$\forall t \in T, s \in S, x_{i,j} \in U : g_s(t \mid x_{i,j}) \approx P(T = t \mid X = x_{i,j}, S = s)\ . \tag{3.4}$$

3.2.3 Training with multiply imputed datasets

The MICE algorithm yielded 5 imputed datasets. The pooled prediction strategy described above was chosen for practical reasons. Figure 3.1 illustrates the process. For each logistic regression model m with parameters p, five trainings were performed using the five imputed datasets and the predictions of each model $\mathrm{m_{p,\,i}}$ $i \in \{1, ..., 5\}$ were pooled.

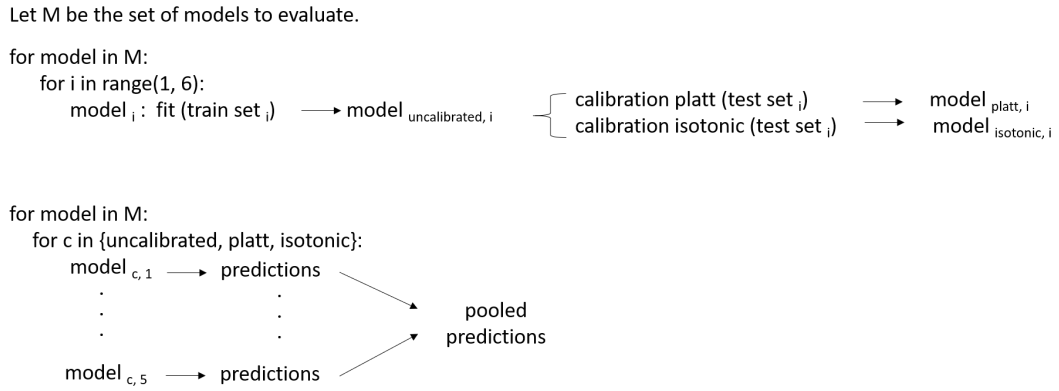


Figure 3.1: Pooled predictions strategy.

Logistic regression models were instantiated with the multinomial option and a lbfgs solver. 48 parameter combinations were obtained from different regularization strengths [0.00001, ... , 500] and penalties ['none', 'l2']. Finally, a dummy classifier with stratified strategy was also trained. The performance of each logistic regression model $\mathrm{m_p}$ $p \in \{1, ..., 48\}$ was assessed with balanced accuracy (BAC) using pooled predictions.

3.2.4 Calibration and model selection

Since practice variation analysis builds upon propensity estimates successively yielded by two classifiers, getting reliable estimates was crucial. Therefore, each model trained for f and g was calibrated using Platt scaling and Isotonic regression with the test sets as shown in Figure 3.1.

These calibrated models were also evaluated using BAC. Moreover, the quality of calibration was assessed for each calibrated model by computing the ECE and the MCE. Finally, reliability curves for the best estimators f and g were plotted. Reliability diagrams, ECE and MCE calculations were based on the following data manipulations:

First, for each couple (patient - period), the estimators f and g yielded propensity estimates for sites (or treatments) as shown in Table 3.10.

Patient	Period	Site	$\hat{y}_{s1}$	$\hat{y}_{s2}$	$\hat{y}_{s3}$	$\hat{y}_{s4}$
1	1	1	0.22	0.11	0.33	0.40
⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	18	3	0.59	0.050	0.20	0.25

Table 3.10: Examples of propensity estimates for sites yielded by estimator f.

Each probability was then associated with its accuracy as shown in Table 3.11.

Patient	Period	Site	s == 1	$\hat{y}_{s1}$	s == 2	$\hat{y}_{s2}$	s == 3	$\hat{y}_{s3}$	s == 4	$\hat{y}_{s4}$
1	1	1	True	0.22	False	0.11	False	0.33	False	0.40
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	18	3	False	0.59	False	0.05	True	0.20	False	0.25

Table 3.11: Examples of propensity estimates for sites yielded by estimator f associated to prediction accuracy.

Then all the probabilities were grouped as shown in Table 3.12

Accuracy	Confidence
True	0.22
False	0.11
False	0.33
False	0.40
⋮	⋮
False	0.59
False	0.05
True	0.20
False	0.25

Table 3.12: Propensity estimates for each site are grouped.

and sorted and placed into 10 bins as shown in Table 3.13.

Bin	Accuracy	Confidence
1	False	0.05
2	False	0.11
3	True, True, False	0.22, 0.20, 0.25
4	False	0.33
5	False	0.40
6	False	0.59
7	-	-
8	-	-
9	-	-
10	-	-

Table 3.13: Propensity estimates are sorted and place in bins.

Finally, an average accuracy and confidence were computed for each bin as shown in Table 3.14.

Bin	Accuracy	Confidence
1	0	0.05
2	0	0.11
3	0.66	0.22
4	0	0.33
5	0	0.40
6	0	0.59
7	-	-
8	-	-
9	-	-
10	-	-

Table 3.14: Average accuracy and confidence are computed for each bin.

Thus the reliability diagrams could be plot with the expected sample accuracy as a function of confidence.

As a compromise between class-prediction accuracy and prediction confidence, the calibrated model with lowest MCE in the subgroup of models having the 5% highest BAC was chosen for f and g respectively.

3.3 Practice variation

Once trained, the second classifier g_s was used to obtain the treatment policy, i.e. the propensity for each treatment $t \in T$ for each site s in S as shown in Table 3.15.

Patient	Period	Site	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9
1	1	1	-	-	-	-	-	-	-	-	-
1	2	1	-	-	-	-	-	-	-	-	-
1	3	1	-	-	-	-	-	-	-	-	-
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	4	1	-	-	-	-	-	-	-	-	-
n	5	1	-	-	-	-	-	-	-	-	-
1	1	2	-	-	-	-	-	-	-	-	-
1	2	2	-	-	-	-	-	-	-	-	-
1	3	2	-	-	-	-	-	-	-	-	-
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	4	2	-	-	-	-	-	-	-	-	-
n	5	2	-	-	-	-	-	-	-	-	-
1	1	3	-	-	-	-	-	-	-	-	-
1	2	3	-	-	-	-	-	-	-	-	-
1	3	3	-	-	-	-	-	-	-	-	-
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	4	3	-	-	-	-	-	-	-	-	-
n	5	3	-	-	-	-	-	-	-	-	-
1	1	4	-	-	-	-	-	-	-	-	-
1	2	4	-	-	-	-	-	-	-	-	-
1	3	4	-	-	-	-	-	-	-	-	-
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	4	4	-	-	-	-	-	-	-	-	-
n	5	4	-	-	-	-	-	-	-	-	-

Table 3.15: Treatment policy for each site.

We predicted the probabilities for treatment t given covariates $x_{i,j}$ for every patient i whose covariates $x_{i,j} \in U$ four times, each with a different input site $s \in S$.

3.3.1 Among sites

Practice variation (PV) between sites could first be illustrated as the expected difference in propensity of treatment t, for two sites s_1, s_2:

$$PV_{\text{s1, s2}} = \mathbb{E}_X \left[g_{s_1}(t \mid x_{i,j}) - g_{s_2}(t \mid x_{i,j}) \right] \tag{3.5}$$

3.3.2 Among patients

To investigate practice variation at a patient level, we averaged the covariates $x_{i,j}$ over each patient ICU stay as follows:

$$x_i \leftarrow \frac{1}{n}\sum_{j=1}^{n} x_{i,j} \tag{3.6}$$

In equation 3.5 n represents the total number of periods for patient i. After this step, each patient i had only one covariates vector x_i and the expected propensity for treatment t across sites was defined as:

$$\mathbb{E}_{s|x_i}[g_s(t \mid x_i)] = \sum_{k=1}^{4} f(s_k \mid x_i) \times g_{s_k}(t \mid x_i) \tag{3.7}$$

For each patient i, we obtained 9 expected propensities, one for each treatment. Practice variation for patient i with covariates x_i in site $s = 1$ for treatment $t = 1$ could then be estimated as:

$$PV_i(S = 1, T = 1) = |g_{s_1}(t_1 \mid x_i) - \mathbb{E}_{s|x_i}[g_s(t = 1 \mid x_i)]| \tag{3.8}$$

Then, a global practice variation metric across sites and treatments for patient i could be computed as follows:

$$\mathbb{E}_{(s,t|x_i)}[PV_i(S = s, T = t)] = \sum_{s=1}^{S}\sum_{t=1}^{T} PV_i(S = s, T = t) \times P(S = s, T = t \mid X = x_i) \tag{3.9}$$

with:

$$\begin{aligned} P(S = s, T = t \mid X = x_i) &= P(T = t \mid S = s, X = x_i) \times P(S = s \mid X = x_i) \\ &= g_s(t \mid x_i) \times f(s \mid x_i) \end{aligned} \tag{3.10}$$

Then, we analysed the patients who corresponded to the 1% highest practice variation. We performed a principal component analysis (PCA) to reduce the feature space to only two components and we situated these extreme patients among all the others in a plot.

Finally, we investigated the correlation between practice variation and covariates at a patient level.

3.3.3 Importance sampling

Evaluating the value of a learned policy (target policy) with data generated by another policy (behavior policy) is known as off-line policy evaluation [29]. For example, we would like to know if the learned policy g_s conducted in site $S = 1$ would lead to lower hospital mortality than the behavior policy. Importance sampling is a classical method to correct the discrepancy between the target π and the behavior μ distributions.

In our case, the outcome mortality Y_i for patient i would be weighted by the ratio of its likelihood of occurring under the two distributions and a global value $\hat{V}$ of the target policy could be estimated by taking the average over patients as follows:

$$\hat{V}(\pi) = \frac{1}{n} \sum_{i=1}^{n} \frac{\pi(x_i, T = t)}{\mu(x_i, T = t)} \times Y_i \tag{3.11}$$

π and μ can be expressed as follows:

$$\begin{aligned} \pi(X = x_i, T = t) &= P(X = x_i, T = t) \\ &= P(X = x_i) \times P(T = t \mid X = x_i) \\ &= P(X = x_i) \times g_s(t|x_i) \end{aligned} \tag{3.12}$$

$$\begin{aligned} \mu(X = x_i, T = t) &= \mathbb{E}_{s|x_i}[g_s(t, x_i)] \\ &= \sum_{k=1}^{4} f(s_k \mid x_i) \times g_{s_k}(t, x_i) \\ &= P(X = x_i) \times \sum_{k=1}^{4} f(s_k \mid x_i) \times g_{s_k}(t \mid x_i) \\ &= P(X = x_i) \times \mathbb{E}_{s|x_i}[g_s(t \mid x_i)] \end{aligned} \tag{3.13}$$

The value $\hat{V}$ of the target policy can be rewritten as follows:

$$\hat{V}(\pi) = \frac{1}{n} \sum_{i=1}^{n} \frac{g_s(t|x_i)}{\mathbb{E}_{s|x_i}[g_s(t \mid x_i)]} \times Y_i \tag{3.14}$$

An alternative way to illustrate practice variation among sites was to plot the distributions of the ratio π over μ for each target policy and each treatment. For a given treatment and a given policy, a ratio greater than one for most of the patients could indicate that this policy recommended this treatment and vice versa.

3.4 Software used

The cohort of sepsis patients and the preliminary covariates were obtained by using Structured Query Language (SQL) to query the PostgreSQL eICU database. The queries for obtaining the SOFA score and the data on creatinine, lactate, FiO2 and base excess were adapted from van den Boom et al. [48] and Tom Pollard et al. [45].

The code for this work was written in Python (version 3.7.3) using the Jupyter Notebook interface.

4

Results

4.1 Cohort of sepsis patients and sites selection

2820 ICU stays corresponding to 2668 unique sepsis patients were retrieved from the eICU database. Figure 4.1 shows the flowchart of selecting the cohort of patients in sepsis.

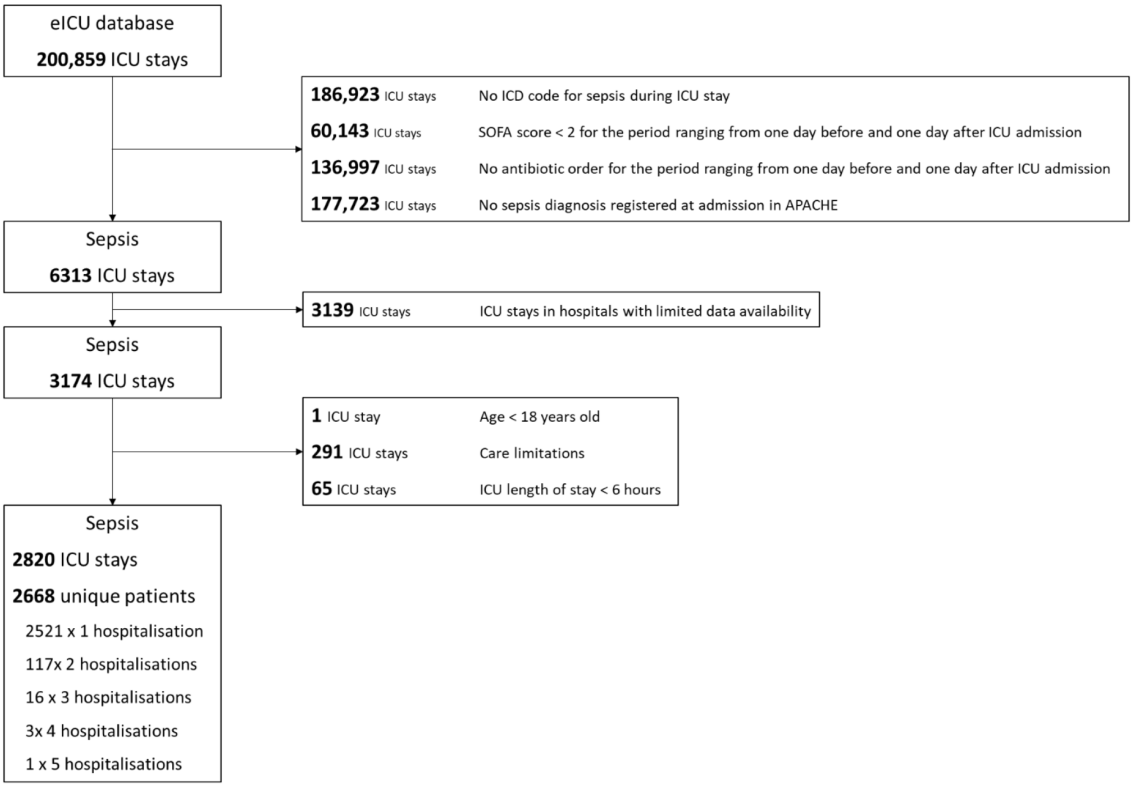
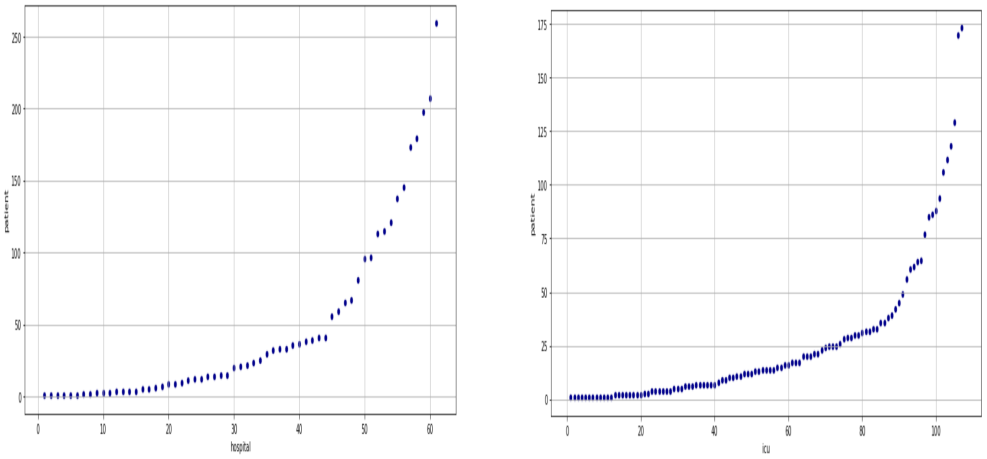


Figure 4.1: Flowchart of selecting the cohort of patients in sepsis.

In order to get a sufficient number of ICU stays per site, we selected the most represented hospitals and ICUs. The distributions of ICU stays with sepsis over hospitals (n = 61) and ICUs (n = 107) are represented in the Figure 4.2.

We selected the five most represented sites of each category which yielded 1016 and 702 ICU stays for the hospitals and the ICUs respectively. However, due to missing



(a) Distribution of ICU stays with sepsis over hospitals.

(b) Distribution of ICU stays with sepsis over ICUs

Figure 4.2: Distribution of stays in hospitals and ICUs.

vasopressor concentrations, one hospital from the five most represented hospitals and one ICU from the five most represented ICUs had to be excluded. This led to a subset of four hospitals (837 ICU stays) and a subset of four ICUs (573 ICU stays) as displayed in Figure 4.3.

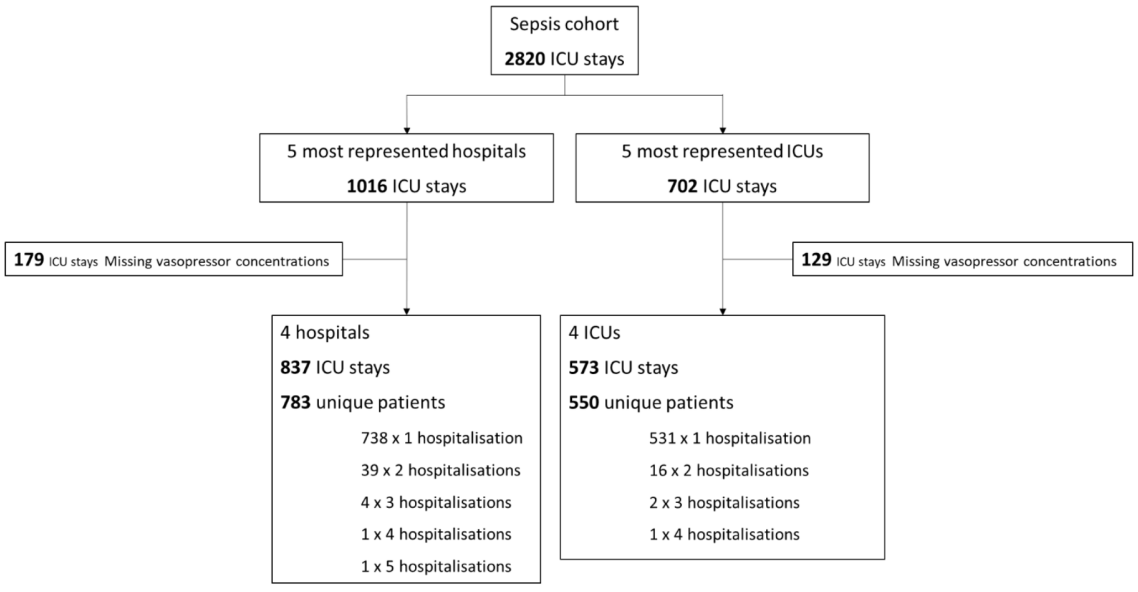


Figure 4.3: Flowchart of site selection.

Table 4.1 and 4.2 contains descriptive statistics for the four hospitals and the four ICUs respectively.

Hospital	1	2	3	4
Stays (n)	173	207	198	259
SOFA score	7 (5-10)	7 (4-9)	7 (4-9)	7 (5-10)
Age (years)	70 (60-79)	60 (49-71)	67 (57-78)	68 (57-78)
Weight (kg)	79 (66-94)	78 (65-102)	75 (64-98)	75 (63-94)
Length of stay (min)	52 (26-121)	71 (34-161)	58 (25-129)	88 (46-164)
ICU types (n)	1	3	3	6
Stays ICU type* (%)				
Med-Surg ICU	100	82	57	47
Cardiac	0	0	0	36
Surg	0	0	0	14
CCU-CTICU	0	0	0	3
Neuro	0	10	0	0
CSICU	0	8	43	0
Beds (n)	250 - 499	$\geq$ 500	Unknown	$\geq$ 500
Teaching status	No	Yes	No	Yes
Region	West	West	Unknown	Northeast
Male/Female (%)	50/50	50/50	53/47	54/46
Ethnicity (%)				
Caucasian	87	64	85	87
African-American	2	8	5	4
Native-American	4	13	0	0
Hispanic	1	6	1	0
Asian	0	0	3	1
Other-Unknown	6	9	6	8
Unit Admission (%)				
Emergency	73	59	72	65
Floor	21	22	22	20
SDU	2	1	2	1
Other Hospital	2	14	3	10
Other ICU	1	1	1	2
Direct	1	3	1	1
OR-recovery	0	0	0	1
Sepsis Type (%)				
Pulmonary	42	42	44	27
UTI-Renal	25	16	19	16
GI	15	21	14	13
Soft Tissue	9	11	12	3
Gyneco	0	1	0	0
Other-Unknown	9	8	11	41
ICU Mortality (%)	12	13	11	16
Hospital Mortality (%)	17	17	14	20
Vasopressor (%)	69	59	71	62

Table 4.1: Descriptive statistics for the four hospitals.

ICU	1	2	3	4
Stays (n)	112	118	173	170
SOFA score	7 (5-10)	6 (4-9)	7 (5-10)	7 (4-9)
Age (years)	62 (51-74)	72 (64-81)	70 (60-79)	57 (46-68)
Weight (kg)	78 (65-94)	79 (66-97)	79 (66-94)	77 (65-103)
Length of stay (min)	64 (27-107)	67 (35-139)	52 (26-121)	64 (31-156)
ICU type	Med-Surg ICU	MICU	Med-Surg ICU	Med-Surg ICU
Beds (n)	≥ 500	250 - 499	250 - 499	≥ 500
Teaching status	No	No	No	Yes
Region	West	West	West	West
Male/Female (%)	58.9/41.1	43.2/56.8	49.7/50.3	50/50
Ethnicity (%)				
Caucasian	79	86	87	60
African-American	4	3	2	9
Native-American	3	0	4	15
Hispanic	2	2	1	5
Asian	1	1	0	1
Other-Unknown	11	8	6	10
Unit Admission (%)				
Emergency	68	80	73	63
Floor	24	14	21	21
SDU	3	1	2	0
Other Hospital	3	1	2	13
Other ICU	2	3	1	1
Direct	0	1	1	2
Sepsis Type (%)				
Pulmonary	42	33	42	42
UTI-Renal	27	31	25	19
GI	10	20	15	22
Soft Tissue	12	5	9	8
Gyneco	1	0	0	2
Other-Unknown	9	10	9	6
ICU Mortality (%)	12	9	12	13
Hospital Mortality (%)	15	10	17	16
Vasopressor (%)	68	61	69	55

Table 4.2: Descriptive statistics for the four ICUs.

4.2 Covariates and treatments

The patients retrieved from our selection of ICUs and hospitals had a lot of missing covariate and treatment values. Table 4.3 and Figure 4.4 shows for each covariate and treatment: the number of patients with registered values and missing values before imputation.

Covariate or treatment	Patients with value	Values	Missing values	Proportion of missing values
Systolic blood pressure	1067	373758	427539	53.36
Mean blood pressure	1067	375082	426215	53.19
Diastolic blood pressure	1067	373563	427734	53.38
Shock Index	976	287520	513777	64.12
Sao2	902	600790	200507	25.02
Lactate	1067	2448	798849	99.69
Creatinine	1052	4030	797267	99.50
P/F	548	1965	799332	99.75
Base Excess	706	2733	798564	99.66
Input Fluid	1066	51459	749838	93.58
Vasopressor	668	23544	777753	97.06

Table 4.3: Covariates and treatments - missingness before imputation.

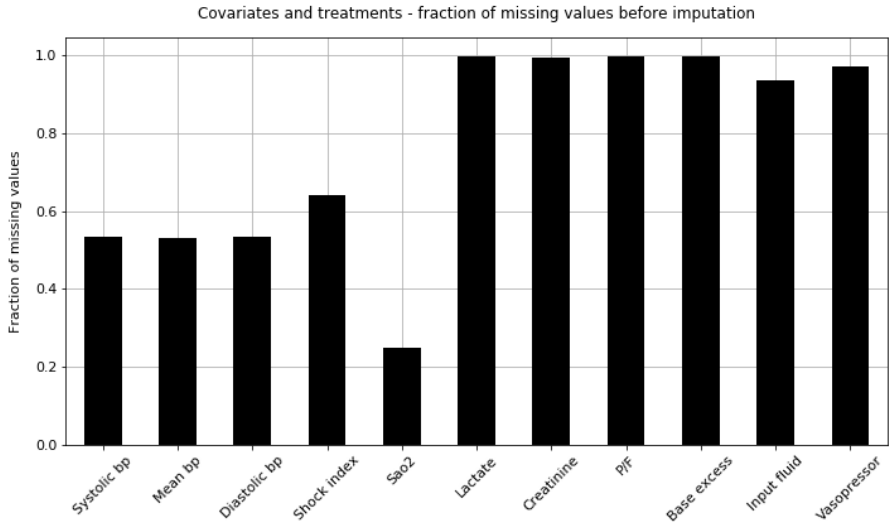


Figure 4.4: Covariates and treatments - fraction of missing values before imputation.

'Continuous' covariates like blood pressures, shock index and saO2 showed less missing values than 'intermittent' covariates collected from blood samples (lactate, creatinine, P/F and base excess), between around 0.2 and 0.65 versus over 0.95 respectively.

Covariates and treatments - missingness after linear interpolation and forward filling. Linear interpolation and forward filling were applied as described in 3.5. Table 4.4 and Figure 4.5 shows the missingness for each covariate and treatment before and after linear interpolation and forward filling.

Covariate or treatment	Values		Missing values		Proportions of missing values	
	before	after	before	after	before	after
Systolic blood pressure	373758	799952	427539	2318	53.36	0.29
Mean blood pressure	375082	799959	426215	2311	53.19	0.29
Diastolic blood pressure	373563	799952	427734	2318	53.38	0.29
Shock Index	287520	689789	513777	112481	64.12	14.02
Sao2	600790	798351	200507	3919	25.02	0.49
Lactate	2448	569914	798849	232356	99.69	28.96
Creatinine	4030	695793	797267	106477	99.50	13.27
P/F	1965	368235	799332	434035	99.75	54.10
Base Excess	2733	483284	798564	318986	99.66	39.76
Input Fluid	51459	802270	749838	0	93.58	0.00
Vasopressor	23544	802270	777753	0	97.06	0.00

Table 4.4: Covariates and treatments - missingness after linear interpolation and forward filling.

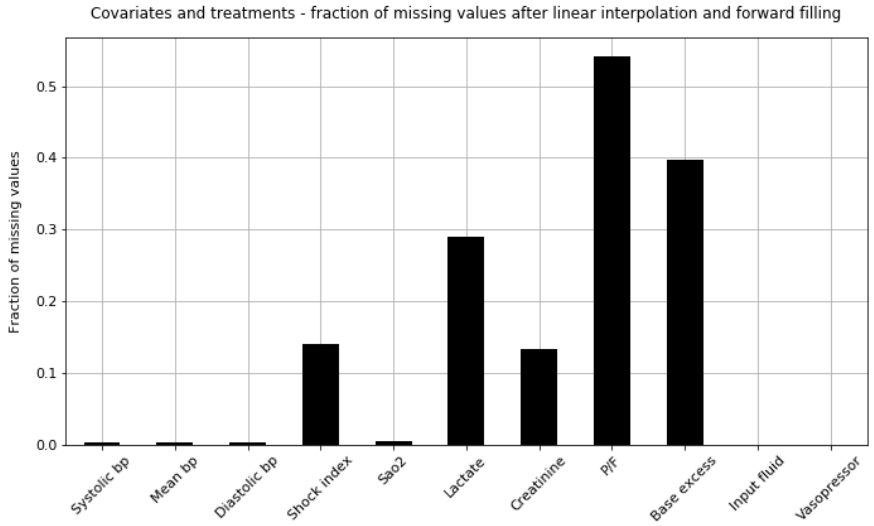


Figure 4.5: Covariates and treatments - fraction of missing values after linear interpolation and forward filling.

Covariates - missingness after grouping. Then each patient had their covariates grouped with a four-hour time window as describe in 3.6. This created a maximum up to 18 time steps depending on the ICU length of stay. This merging of covariates into 18 periods created missing values as covariates were recorded at different rates or only when medically needed. The missingness after grouping is shown in Table 4.5 and illustrated in Figure 4.6.

Covariate or treatment	Values	Missing values	Proportion of missing values
Systolic blood pressure	15056	378	2.45
Mean blood pressure	15056	378	2.45
Diastolic blood pressure	15056	378	2.45
Shock Index	13261	2173	14.08
Sao2	15034	400	2.59
Lactate	11171	4263	27.62
Creatinine	13706	1728	11.20
P/F	7003	8431	54.63
Base Excess	9207	6227	40.35

Table 4.5: Covariates - missingness after grouping (4-hour time windows).

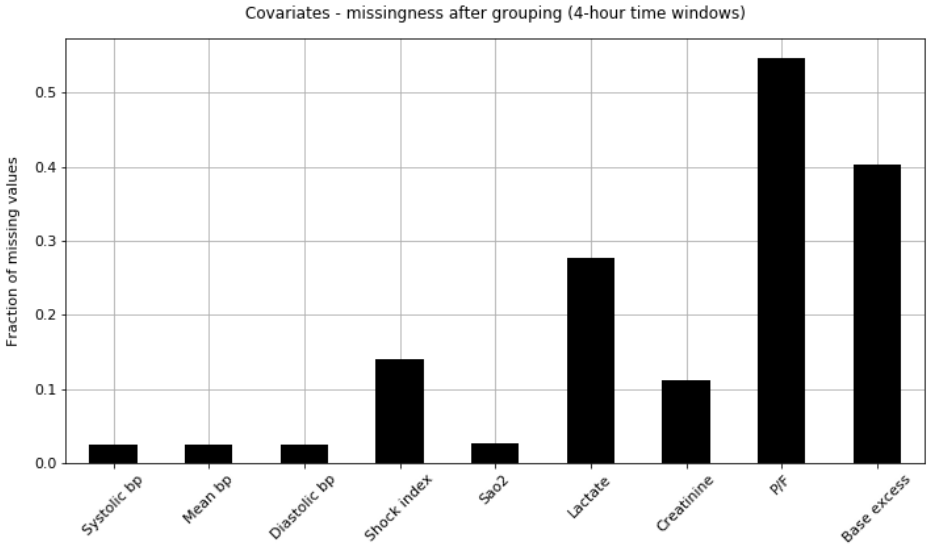


Figure 4.6: Covariates - missingness after grouping (4-hour time windows).

The remaining missing values were imputed performing MICE which yielded five imputed datasets with no missing values. The five imputed datasets had 15344 rows with 9 columns for covariates and 1 column for treatment. The number of patient stays and periods per hospitals and ICUs are shown in Table 4.6. Hospital 1, 2, 3 and 4 consisted of 1, 3, 3 and 6 ICUs respectively.

Datasets (1-5)	Stays	Periods
ICUs	573	7926
ICU 1	112	1569
ICU 2	118	1674
ICU 3	173	2280
ICU 4	170	2403
Hospitals	837	12191
Hospital 1	173	2280
Hospital 2	207	2979
Hospital 3	198	2704
Hospital 4	259	4228
Total	1067	15434

Table 4.6: Datasets - imputed.

Covariates distributions. Figure 4.7 and 4.8 display the distribution of covariates after imputations and normalization for sepsis patients in the four hospitals and the four ICUs respectively. Covariates values have been averaged across the five datasets only for the purpose of visualization.

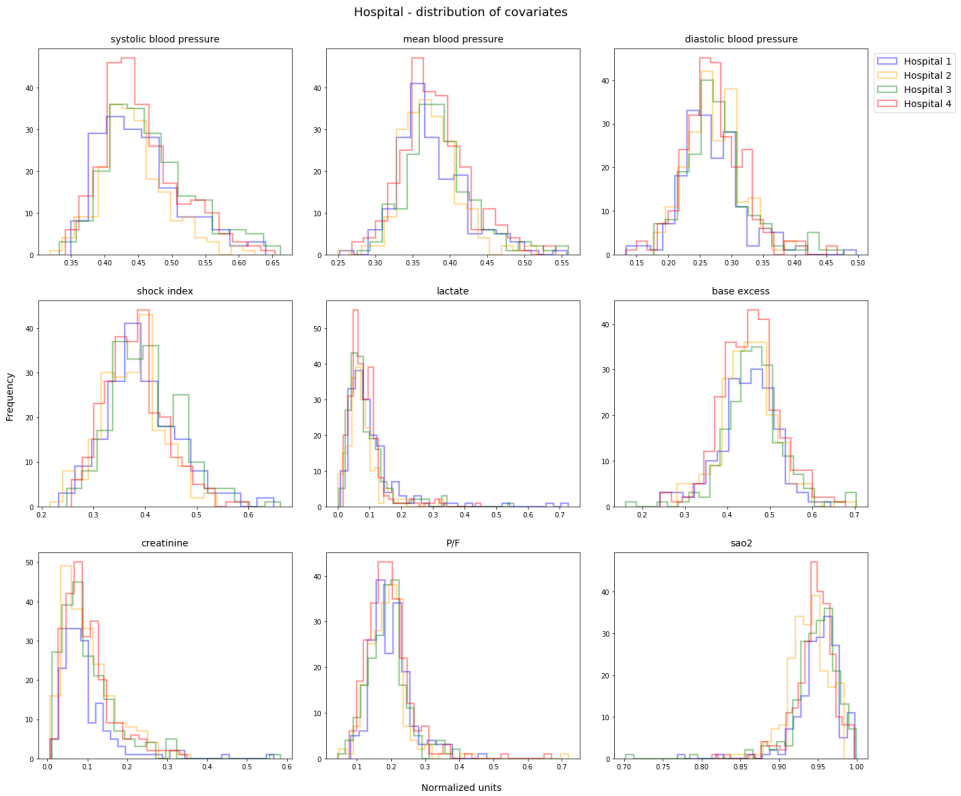


Figure 4.7: Hospital - distribution of covariates after imputations and normalization.

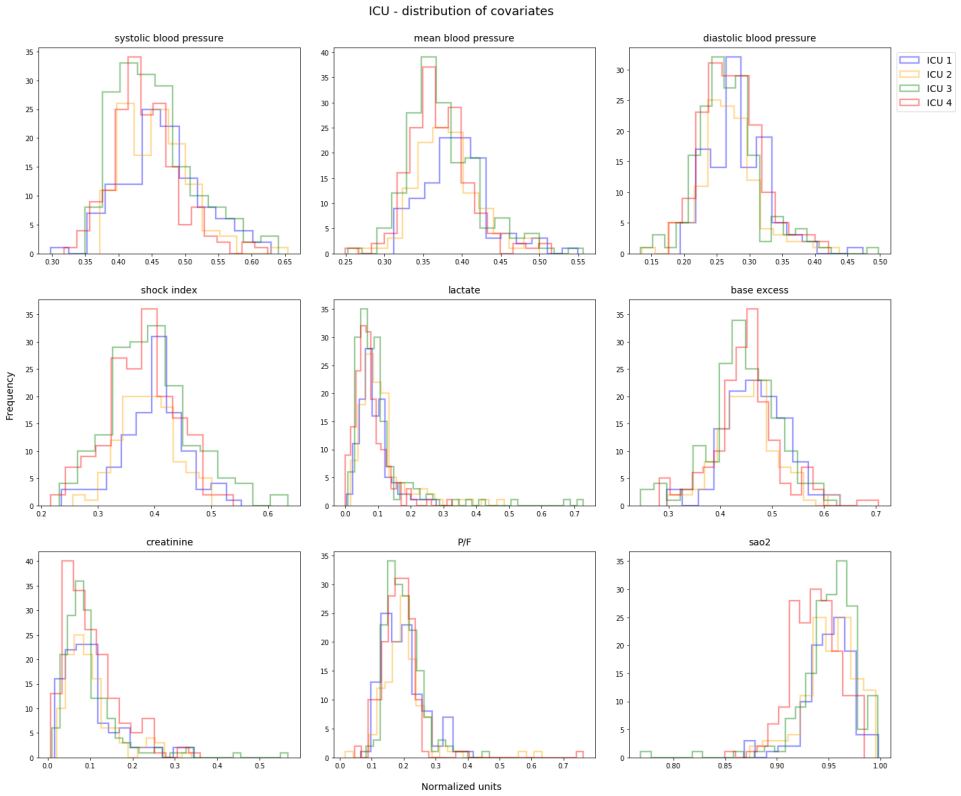


Figure 4.8: ICU - distribution of covariates after imputations and normalization.

The analysis of practice variation was then independently conducted on the subsets of sepsis patients coming from the four largest ICUs and the fours largest hospitals respectively.

4.3 ICU results

4.3.1 Estimator of site propensity

Training and calibration. The extracted and imputed datasets was split into training and testing sets with a split ratio of 80/20 without overlapping stays. The train set contained 458 patient stays with 6330 periods and the test set contained 115 stays with 1596 periods. These figures are identical for each imputed dataset and the complete figures with an overview of the train and test split on an ICU level can be found in Appendix B.1.

48 logistic regression estimators with distinct parameters were trained to predict ICUs given covariates for each imputed dataset. 16 models were not calibrated, 16 models were calibrated using Platt calibration and 16 models were calibrated using isotonic calibration. Among all these models, three dummy classifiers with stratified strategy were trained (uncalibrated, Platt and isotonic calibrated respectively). For each estimator, the performance prediction and the calibration quality were evaluated with BAC, ECE and MCE respectively after pooling the predictions from each imputed dataset as described above. The log loss was also retrieved. The dummy classifiers yielded a BAC of 0.25 (four classes) and served as baseline performance.

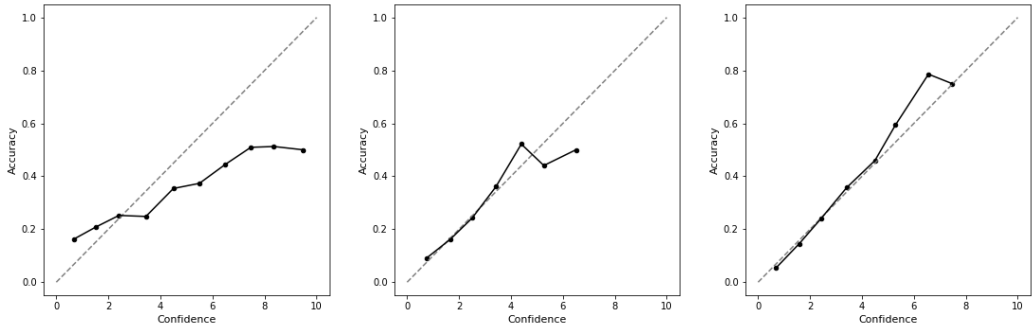
Table 4.7 displays the results of the best calibrated estimators regarding BAC (0.95 quantile) with information of their respective parameter settings and scores. The results showed that estimators calibrated with isotonic regression outperformed calibration with Platt scaling across all metrics on almost all parameter settings (except for some C values below 1).

No.	Estimator class	Solver	Penalty	C	Calibration	BAC	ECE	MCE	Log loss
8	logistic regression	lbfgs	l2	150	isotonic	0.341	0.039	0.12976	1.292
7	logistic regression	lbfgs	l2	160	isotonic	0.341	0.035	0.12979	1.292
6	logistic regression	lbfgs	l2	200	isotonic	0.341	0.038	0.12970	1.292
5	logistic regression	lbfgs	l2	240	isotonic	0.341	0.041	0.12999	1.292
4	logistic regression	lbfgs	l2	130	isotonic	0.341	0.037	0.12985	1.292
3	logistic regression	lbfgs	l2	170	isotonic	0.342	0.037	0.12990	1.292
2	logistic regression	lbfgs	none	none	isotonic	0.342	0.037	0.13001	1.292
1	logistic regression	lbfgs	l2	100	isotonic	0.342	0.039	0.12971	1.292

Table 4.7: ICU - best candidates for the estimator of site propensity of site propensity regarding BAC (0.95 quantile).

Model number 6 with parameters [solver: lbgfs, penalty: l2, c: 200] and isotonic calibration had the lowest MCE (0.12970) in the subgroup of models having the 5% highest BAC and was therefore chosen as the estimator of site propensity. It had also a BAC of 0.341.

Figure 4.9 represents the reliability diagram of model number 6 before calibration and after Platt and isotonic calibration.



(a) No calibration **(b)** Platt calibration **(c)** Isotonic calibration

Figure 4.9: ICU - Reliability diagrams for the logistic regression model with parameters [solver: lbgfs, penalty: l2, c: 200] before and after calibration. The model with isotonic calibration was selected as the estimator of site propensity.

Extracting comparable patients using the estimator of site propensity. Figure 4.10 represents the number of retained patients as a function of epsilon and delta. We visually estimated that an epsilon of 0.10 and a delta of 0.9 were sensible trade-offs for retaining a larger subset of stays (retention of around 75% of the patient ICU stays). Table 4.8 shows the number of patient ICU stays and periods retained per site for the next step.

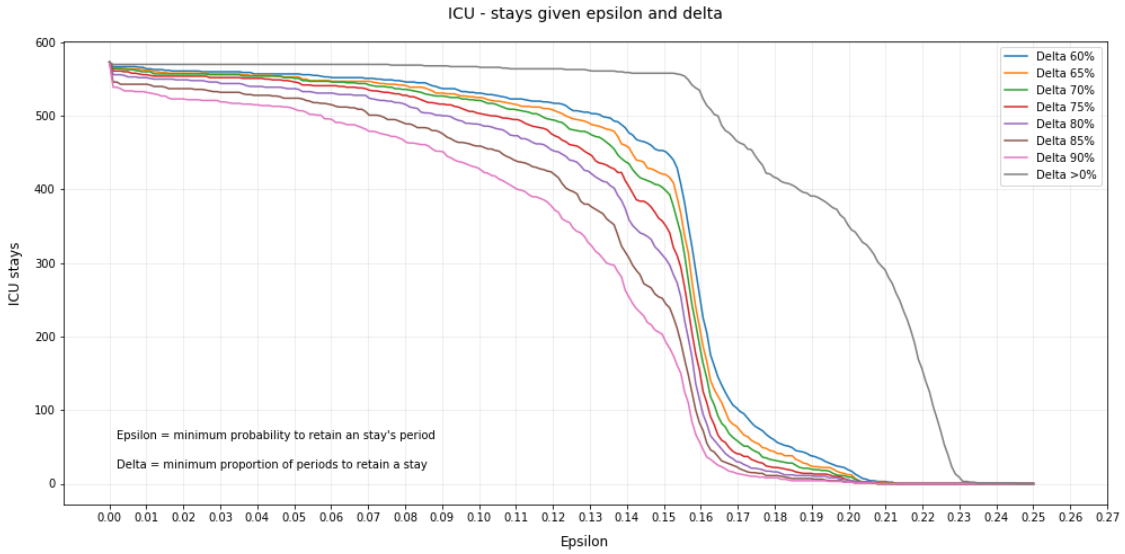


Figure 4.10: Number of patient ICU stays given epsilon and delta.

Site	Stays	Stays retained	Periods	Periods retained
ICU 1	112	77	1569	1053
ICU 2	118	68	1674	944
ICU 3	173	130	2280	1652
ICU 4	170	152	2403	2135
	573	427	7926	5784

Table 4.8: Subset of patient ICU stays retained after the first training.

4.3.2 The treatment variable

A treatment variable (1-9) was then computed for each period in the subset of patients retained. Figure 4.11 displays the marginal distribution of all nine treatment combinations obtained. The figure shows that treatments were clearly imbalanced as treatment 2 and 3 had more than one thousand occurrences, while treatment 4 and 7 only a few dozen ones. This indicates that fluid 0 combined with doses of vasopressor was rarely observed. These figures are identical for each imputed dataset and the complete figures with an overview of the marginal distribution of treatments on an ICU level can be found in Appendix B.3.

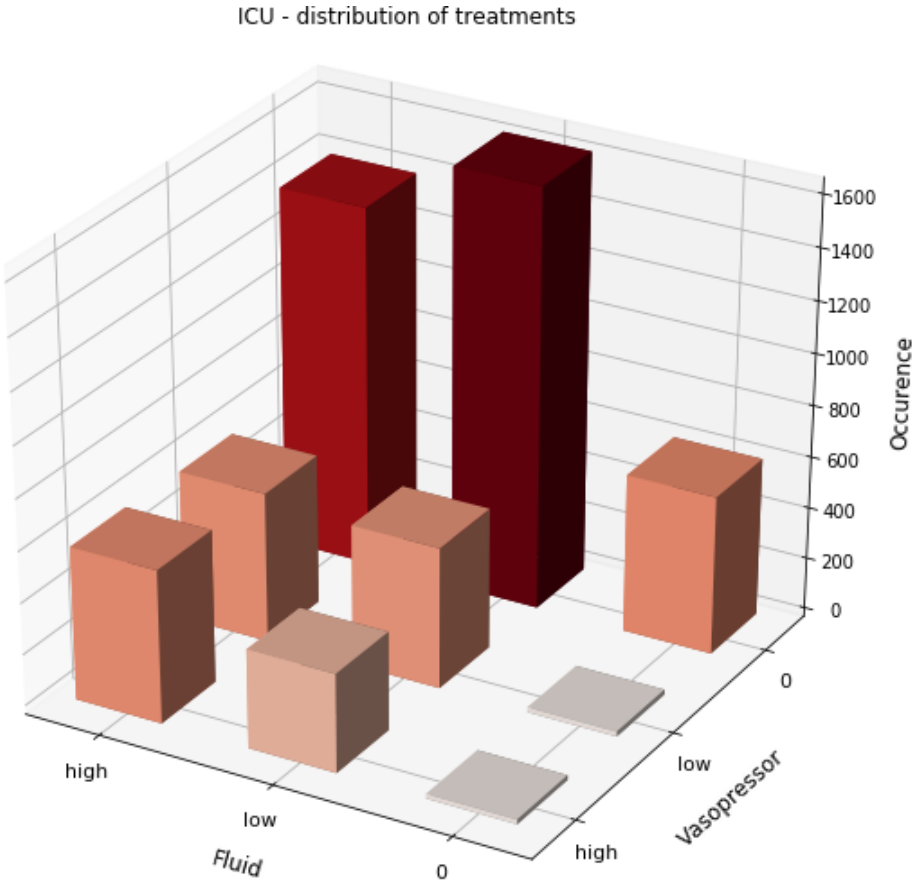


Figure 4.11: ICU - marginal distribution of treatments.

4.3.3 Estimator of treatment propensity

Training and calibration. The estimator of treatment propensity was trained to predict treatments given covariates and sites. The sites were one-hot encoded and each imputed dataset therefore contained 13 features (9 covariates + 4 sites).

As in the first step, the imputed datasets were split into 80% train and 20% test data, without overlapping stays. The train set contained 341 patient stays with 4584 periods and the test set contained 86 stays with 1200 periods. These figures are identical for each imputed dataset and the complete figures with an overview of the train and test split on an ICU level can be found in Appendix B.5.

In the same way as for the first step, we trained 48 logistic regression estimators with distinct parameters for each imputed dataset. 16 models were not calibrated, 16 models were calibrating using Platt calibration and 16 models were calibrated using isotonic calibration. Among all these models, three dummy classifiers with stratified strategy were trained (uncalibrated, Platt and isotonic calibrated respectively). For each estimator, the performance prediction and the calibration quality were evaluated with BAC, ECE and MCE respectively after pooling the predictions from each imputed dataset as described above. The log loss was also retrieved. The dummy classifiers yielded a BAC of 0.11 (nine classes) and served as baseline performance.

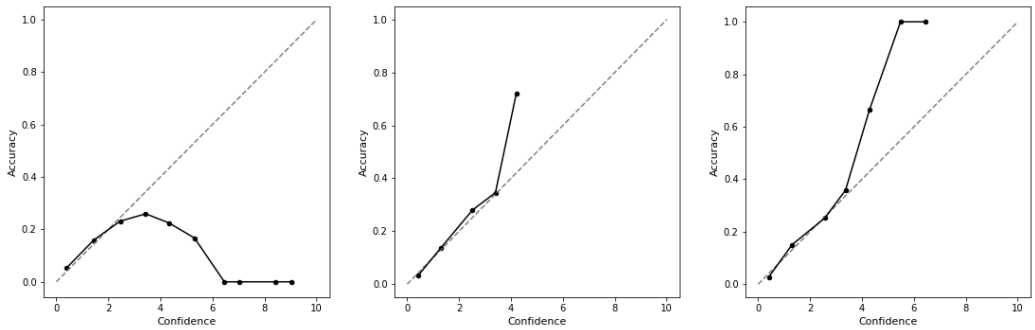
Table 4.9 displays the results of the best calibrated estimators regarding BAC (0.95 quantile) with information of their respective parameter settings and scores. Once again, the results showed that estimators calibrated with isotonic regression outperformed calibration with Platt scaling across all metrics on almost all parameter settings (except for some C values below 1).

No.	Estimator class	Solver	Penalty	C	Calibration	BAC	ECE	MCE	Log loss
7	logistic regression	lbfgs	l2	20	isotonic	0.139	0.143	0.483697	1.609
6	logistic regression	lbfgs	l2	70	isotonic	0.139	0.140	0.497638	1.610
5	logistic regression	lbfgs	l2	40	isotonic	0.140	0.156	0.451662	1.610
4	logistic regression	lbfgs	l2	30	isotonic	0.140	0.146	0.457316	1.609
3	logistic regression	lbfgs	l2	50	isotonic	0.140	0.147	0.496853	1.610
2	logistic regression	lbfgs	l2	15	isotonic	0.140	0.146	0.482657	1.608
1	logistic regression	lbfgs	l2	230	isotonic	0.140	0.155	0.493008	1.613

Table 4.9: ICU - best candidates for the estimator of treatment propensity regarding BAC (0.95 quantile).

Model number 5 with parameters [solver: lbgfs, penalty: l2, c: 40] and isotonic calibration had the lowest MCE in the subgroup of models having the 5% highest BAC and was therefore chosen as the estimator of treatment propensity. It had also a BAC of 0.140.

Figure 4.12 represents the reliability diagram of model number 5 before calibration and after Platt and isotonic calibration.



(a) No calibration **(b)** Platt calibration **(c)** Isotonic calibration

Figure 4.12: ICU - reliability diagrams for the logistic regression model with parameters [solver: lbgfs, penalty: l2, c: 40] before and after calibration. The model with isotonic calibration was selected as the estimator of treatment propensity.

Treatment policies. By sequentially modifying the one-hot encoding for sites in the imputed datasets, the estimator of treatment propensity could yield the propensity for treatments at each time steps for each site.

In Figure 4.13 the propensities for each treatment and each site were averaged over time and these aggregated propensities roughly matched the treatment distribution in the datasets.

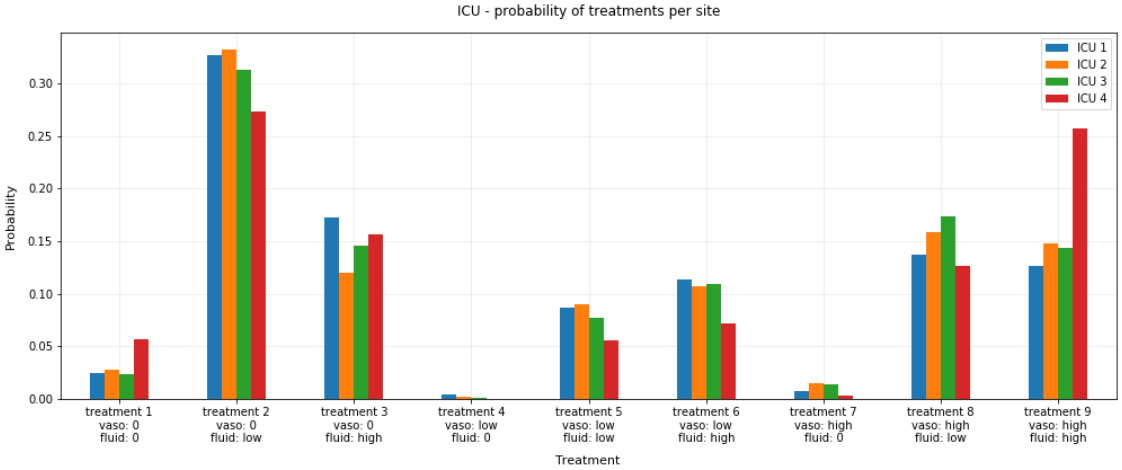


Figure 4.13: ICU - marginal probability of treatments during ICU stay per site.

4.3.4 Practice variation

Among sites. Figure 4.14 illustrates practice variation among ICUs as the expected difference in propensity for treatment t for two sites s_1, s_2 (see Equation 3.7). For example, the probability of receiving treatment 3 in ICU 1 was about 5% higher than in ICU 2 while the probability of receiving treatment 1 in ICU 1 was about 3% lower than in the ICU 4.

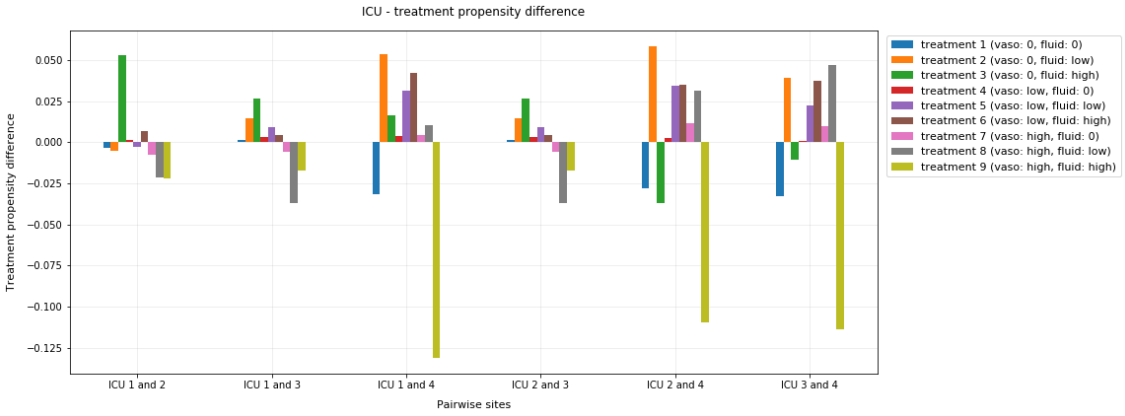


Figure 4.14: ICU - treatment propensity difference.

Among patients. Figure 4.15 illustrates the global practice variation distribution at a patient level (see Equations 3.8-3.11) and Table 4.10 displays statistics of this distribution.

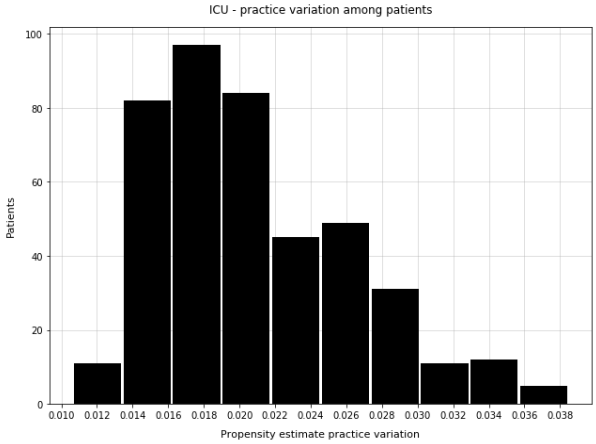


Figure 4.15: ICU - global practice variation distribution at a patient level.

Patients	Mean	Std	Min	Max
427	0.02084	0.00553	0.01064	0.03847

Table 4.10: ICU - statistics for the global practice variation distribution at a patient level.

Figure 4.16 illustrates the practice variation distribution at a patient level for each treatment and Table 4.11 displays statistics of these distributions.

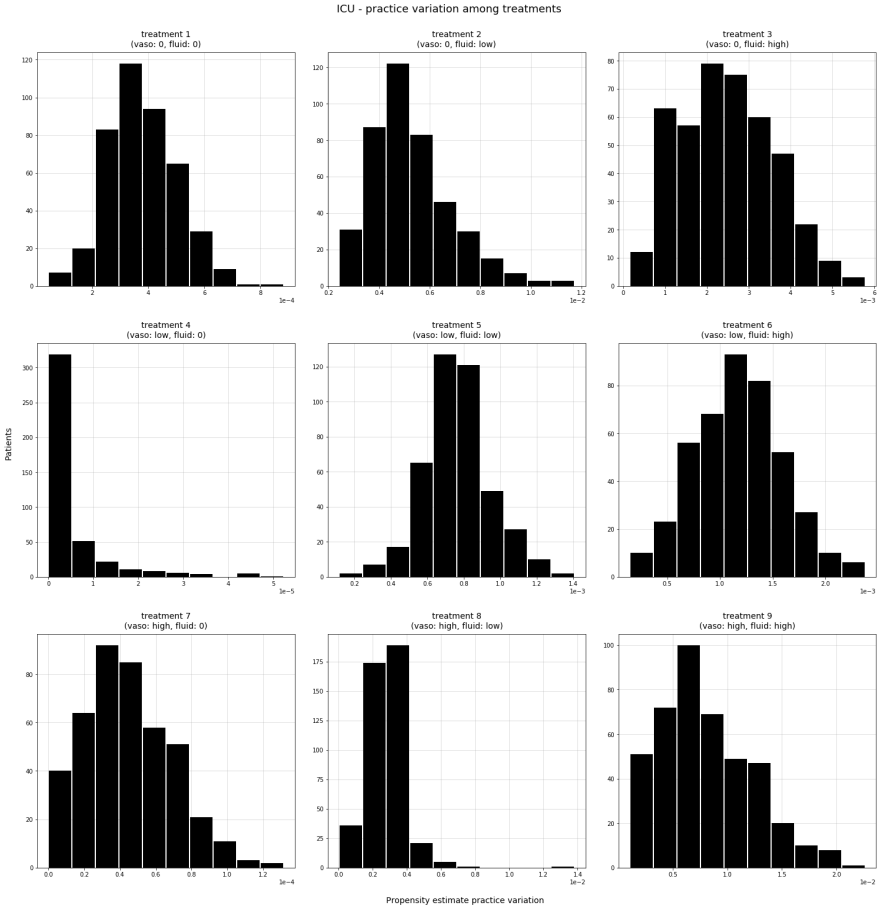


Figure 4.16: ICU - practice variation among patients for each treatment.

Treatment	Mean	Std	Min	Max
1 (vaso: 0, fluid: 0)	0.00038	0.00012	0.00004	0.00088
2 (vaso: 0, fluid: low)	0.00529	0.00161	0.00241	0.01172
3 (vaso: 0, fluid: high)	0.00247	0.00110	0.00016	0.00578
4 (vaso: low, fluid: 0)	0.00000	0.00001	0.00000	0.00005
5 (vaso: low, fluid: low)	0.00076	0.00018	0.00012	0.00141
6 (vaso: low, fluid: high)	0.00117	0.00041	0.00014	0.00237
7 (vaso: high, fluid: 0)	0.00004	0.00002	0.00000	0.00013
8 (vaso: high, fluid: low)	0.00273	0.00114	0.00002	0.01381
9 (vaso: high, fluid: high)	0.00798	0.00417	0.00106	0.02260

Table 4.11: ICU - practice variation among patients for each treatment.

Situate patients with high practice variation among others. Figure 4.17 represents a PCA with two components. The patients with 1% highest practice variation roughly lie on the periphery.

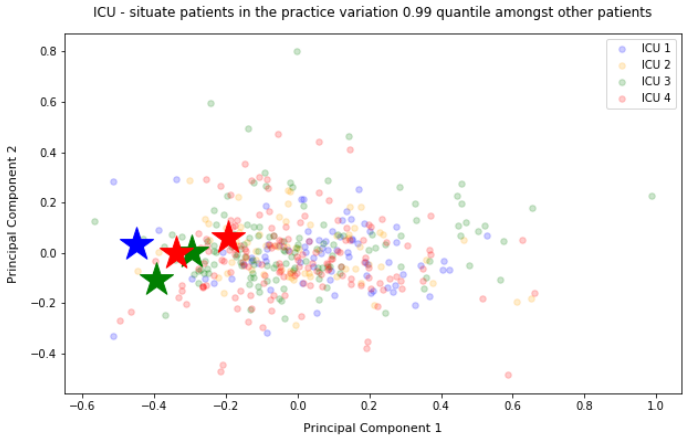


Figure 4.17: ICU - situate patients in the practice variation 0.99 quantile amongst other patients.

Figure 4.18 individualizes these patients from the distribution of every covariate in the dataset.

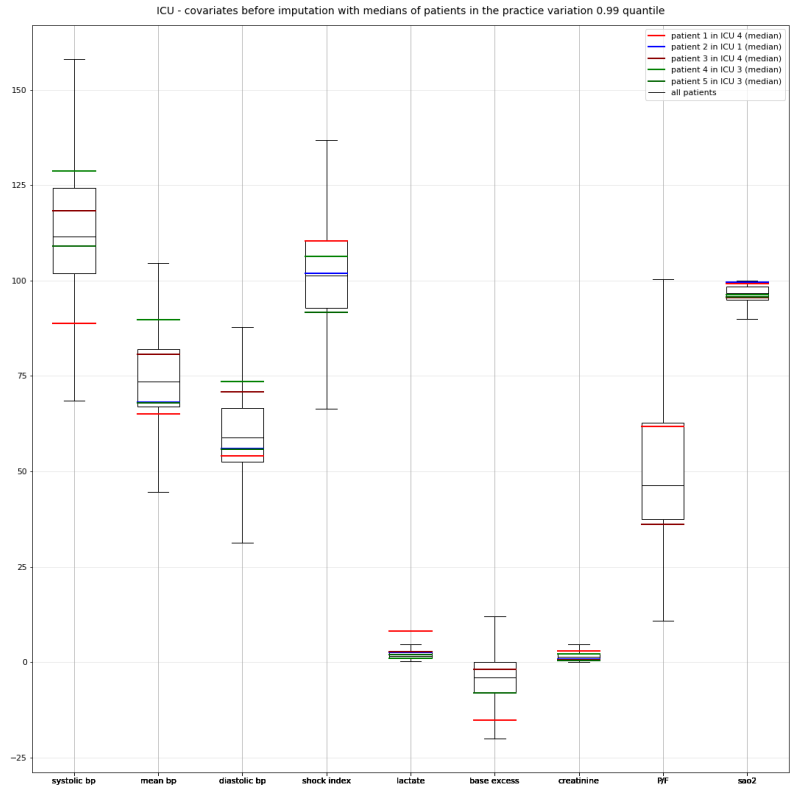


Figure 4.18: ICU - covariates before imputation with medians of the top five patients with highest practice variation.

Correlation with covariates. Figure 4.19 shows a correlation between practice variation and the first component of a PCA (Pearson's correlation = -0.599).

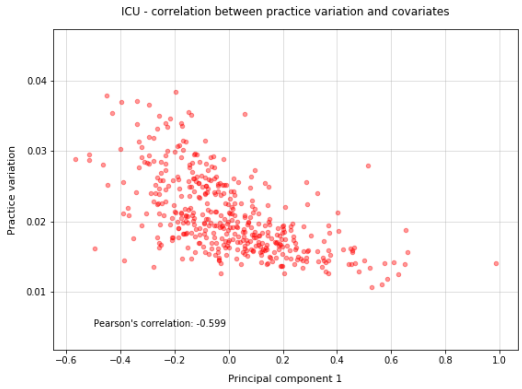


Figure 4.19: ICU - correlation between practice variation and principal component.

This correlation was not evident when we considered each covariate individually as shown in Figure 4.20.

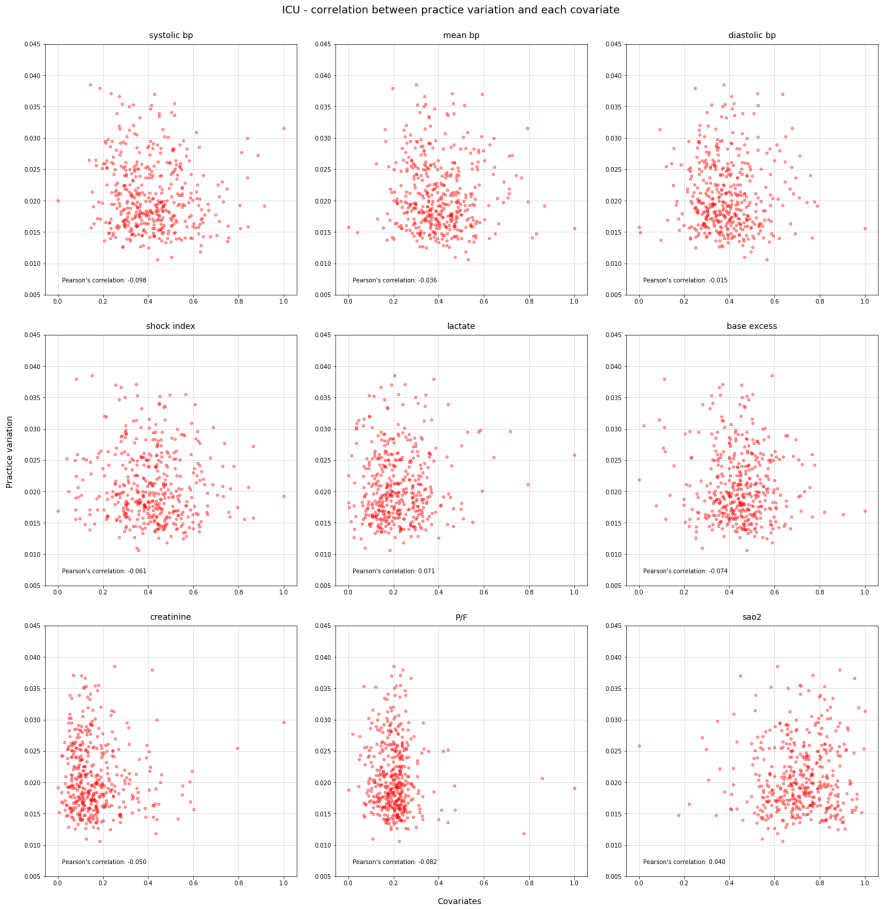


Figure 4.20: ICU - correlation between practice variation and each covariate.

Importance sampling. Figure 4.21 shows the distributions of the weights (see equation 3.10) for each learned policy and each treatment. For a given treatment and a given policy, a ratio greater than one (green) for most of the patients could indicate that this policy recommended this treatment and vice versa (red). For example, treatment 1 was recommended by the policy in ICU 4 but not by the policy in ICU 1.

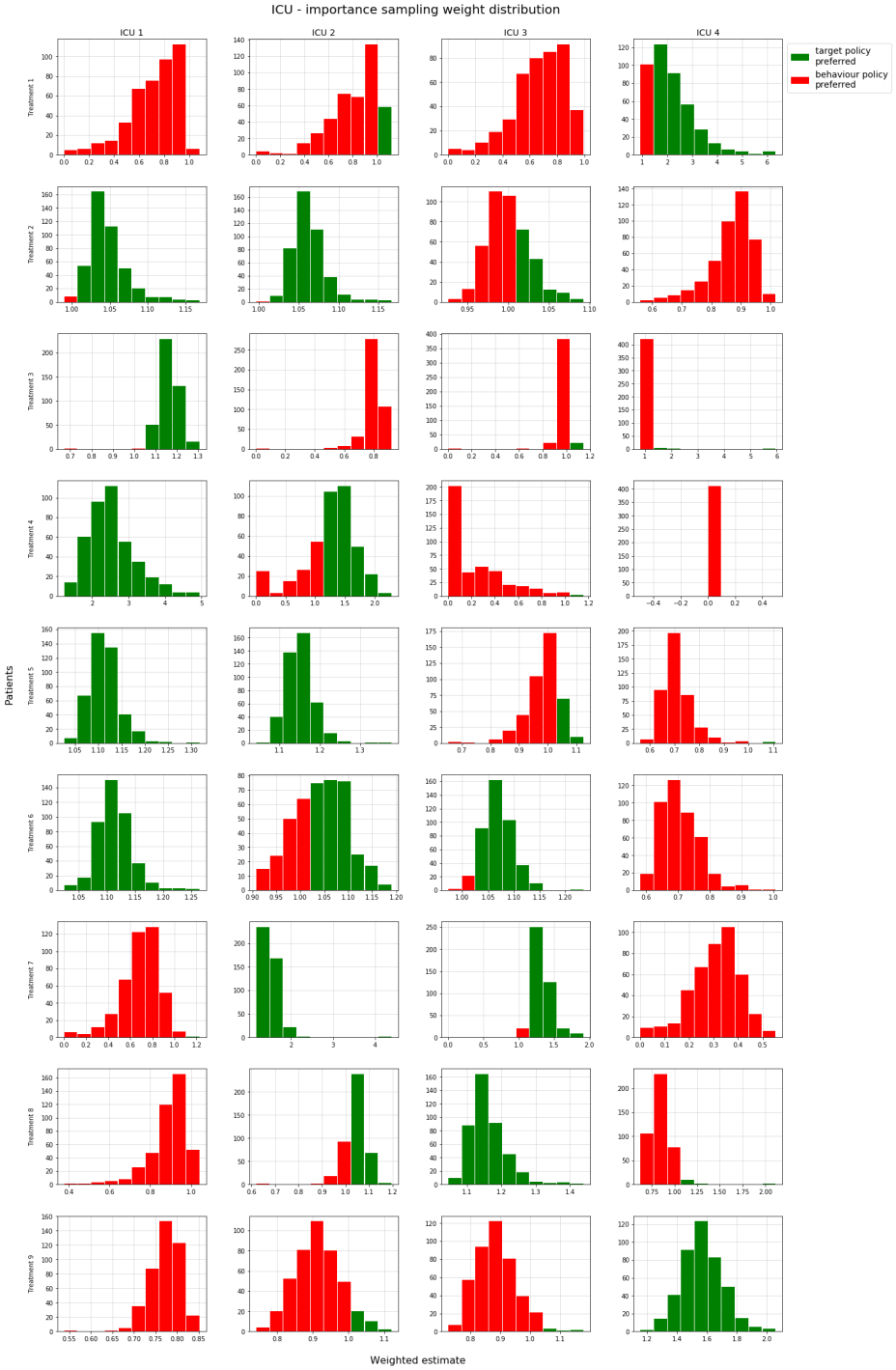


Figure 4.21: ICU - importance sampling weight distribution.

4.4 Hospital results

4.4.1 Estimator of site propensity

Training and calibration. The extracted and imputed datasets was split into training and testing sets with a split ratio of 80/20 without overlapping stays. The train set contained 669 patient stays with 9716 periods and the test set contained 168 stays with 2475 periods. These figures are identical for each imputed dataset and the complete figures with an overview of the train and test split on an hospital level can be found in Appendix B.2.

48 logistic regression estimators with distinct parameters were trained to predict ICUs given covariates for each imputed dataset. 16 models were not calibrated, 16 models were calibrating using Platt calibration and 16 models were calibrated using isotonic calibration. Among all these models, three dummy classifiers with stratified strategy were trained (uncalibrated, Platt and isotonic calibrated respectively). For each estimator, the performance prediction and the calibration quality were evaluated with BAC, ECE and MCE respectively after pooling the predictions from each imputed dataset as described above. The log loss was also retrieved. The dummy classifiers yielded a BAC of 0.25 (four classes) and served as baseline performance.

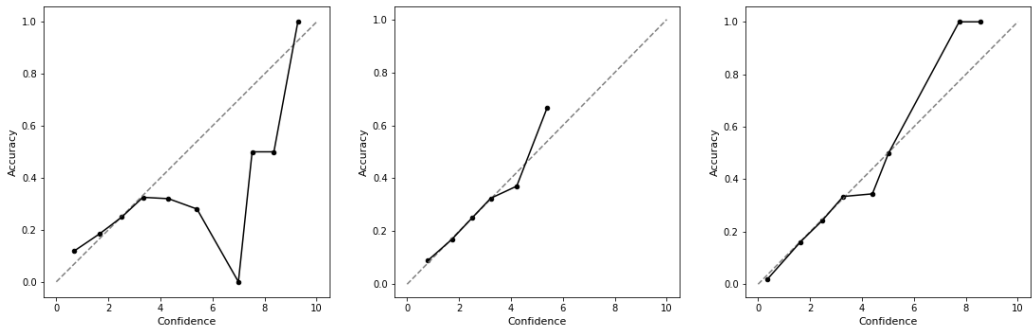
Table 4.12 displays the results of the best calibrated estimators regarding BAC (0.95 quantile) with information of their respective parameter settings and scores. Once again, the results showed that estimators calibrated with isotonic regression outperformed calibration with Platt scaling across all metrics on almost all parameter settings.

No.	Estimator class	Solver	Penalty	C	Calibration	BAC	ECE	MCE	Log loss
7	logistic regression	lbfgs	l2	10	isotonic	0.316	0.0209	0.225829	1.326
6	logistic regression	lbfgs	l2	7	isotonic	0.317	0.0258	0.497848	1.326
5	logistic regression	lbfgs	l2	3	isotonic	0.318	0.0314	0.489147	1.327
4	logistic regression	lbfgs	l2	6	isotonic	0.318	0.0315	0.477560	1.326
3	logistic regression	lbfgs	l2	8	isotonic	0.319	0.0213	0.501692	1.326
2	logistic regression	lbfgs	l2	2	isotonic	0.319	0.0323	0.247794	1.328
1	logistic regression	lbfgs	l2	4	isotonic	0.321	0.0300	0.480372	1.327

Table 4.12: Hospital - best candidates for the estimator of site propensity regarding BAC (0.95 quantile).

Model number 7 with parameters [solver: lbgfs, penalty: l2, c: 10] and isotonic calibration had the lowest MCE (0.225829) in the subgroup of models having the 5% highest BAC and was therefore chosen as the estimator of site propensity. It had also a BAC of 0.316.

Figure 4.22 represents the reliability diagram of model number 7 before calibration and after Platt and isotonic calibration.



(a) No calibration **(b)** Platt calibration **(c)** Isotonic calibration

Figure 4.22: Hospital - reliability diagrams for the logistic regression model with parameters [solver: lbgfs, penalty: l2, c: 10] before and after calibration. The model with isotonic calibration is selected as the estimator of site propensity.

Extracting comparable patients using the estimator of site propensity. Figure 4.23 represents the number of retained patients as a function of epsilon and delta. We visually estimated that an epsilon of 0.13 and a delta of 0.9 were sensible trade-offs for retaining a larger subset of stays (retention of around 93% of the patient ICU stays). Table 4.13 shows the number of patient ICU stays and periods retained per site for the next step.

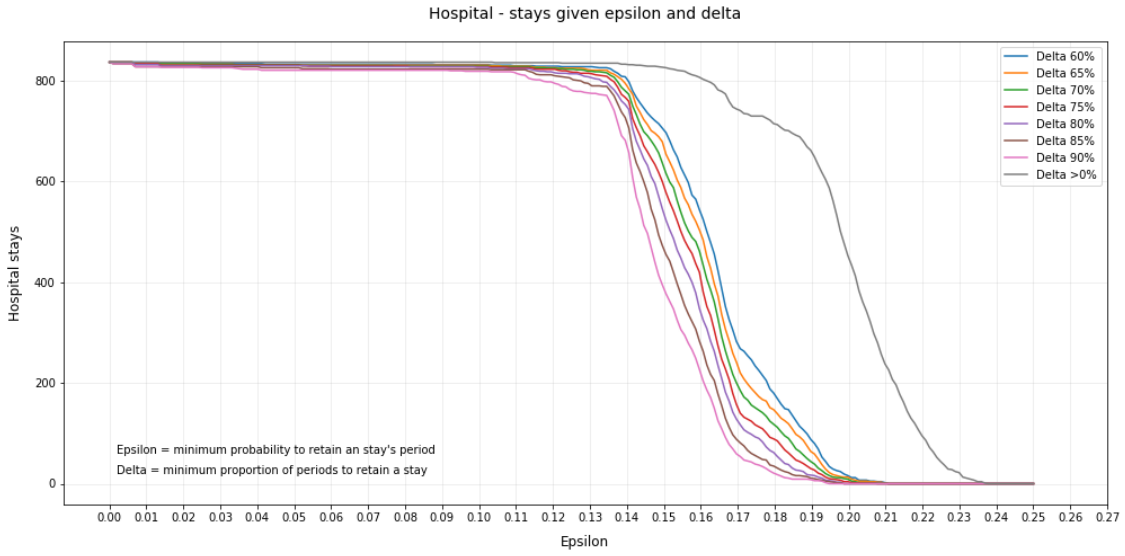


Figure 4.23: Hospital - stays given epsilon and delta.

Site	Stays	Stays retained	Periods	Periods retained
Hospital 1	173	163	2280	2157
Hospital 2	207	186	2979	2660
Hospital 3	198	184	2704	2501
Hospital 4	259	242	4228	3955
	837	775	12191	11273

Table 4.13: Hospital - stays after subset.

4.4.2 The treatment variable

A treatment variable (1-9) was then computed for each period in the subset of patients retained. Figure 4.11 displays the marginal distribution of all nine treatment combinations obtained. The figure shows that treatments were clearly imbalanced as treatment 2 and 3 had more than two thousand occurrences, while treatment 4 and 7 only around 50. This indicates that fluid 0 combined with doses of vasopressor was rarely observed. These figures are identical for each imputed dataset and the complete figures with an overview of the marginal distribution of treatments on a hospital level can be found in Appendix B.4.

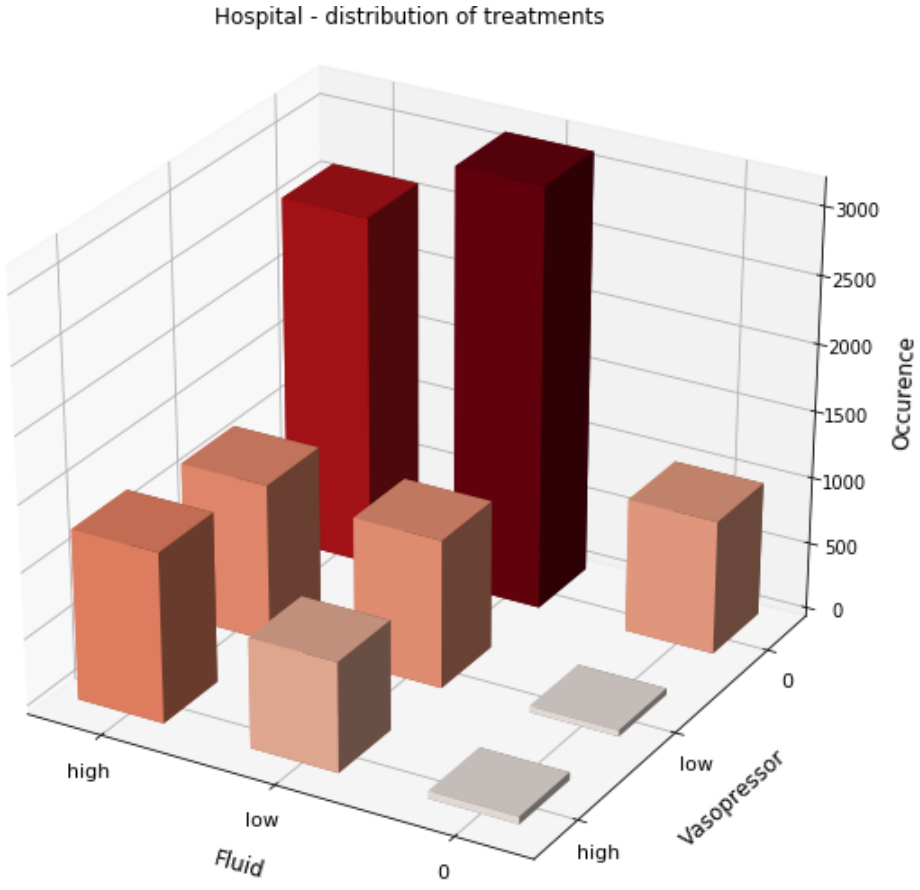


Figure 4.24: Hospital - marginal distribution of treatments.

4.4.3 Estimator of treatment propensity

Training and calibration. The estimator of treatment propensity was trained to predict treatments given covariates and sites. The sites were one-hot encoded and each imputed dataset therefore contained 13 features (9 covariates + 4 sites).

As in the first step, the imputed datasets were split into 80% train and 20% test data, without overlapping stays. The train set contained 620 patient stays with 9060 periods and the test set contained 155 stays with 2213 periods. These figures are identical for each imputed dataset and the complete figures with an overview of the train and test split on a hospital level can be found in Appendix B.6.

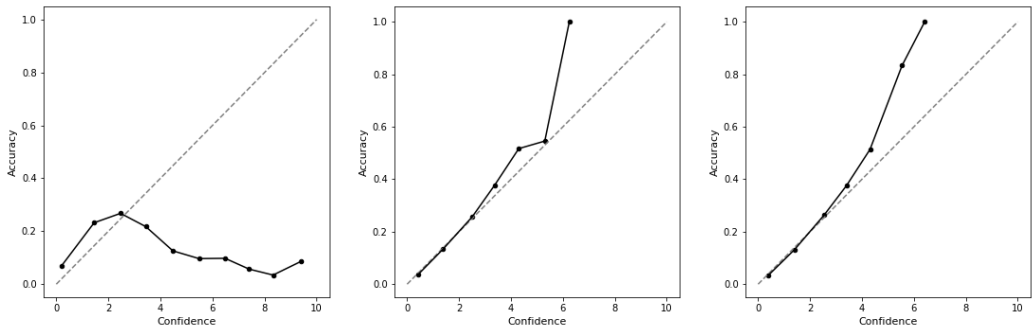
In the same way as for the first step, we trained 48 logistic regression estimators with distinct parameters for each imputed dataset. 16 models were not calibrated, 16 models were calibrating using Platt calibration and 16 models were calibrated using isotonic calibration. Among all these models, three dummy classifiers with stratified strategy were trained (uncalibrated, Platt and isotonic calibrated respectively). For each estimator, the performance prediction and the calibration quality were evaluated with BAC, ECE and MCE respectively after pooling the predictions from each imputed dataset as described above. The log loss was also retrieved. The dummy classifiers yielded a BAC of 0.11 (nine classes) and served as baseline performance.

No.	Estimator class	Solver	Penalty	C	Calibration	BAC	ECE	MCE	Log loss
8	logistic regression	lbfgs	l2	300	isotonic	0.202	0.0899	0.368877	1.645
7	logistic regression	lbfgs	l2	260	isotonic	0.202	0.0913	0.369331	1.645
6	logistic regression	lbfgs	l2	400	isotonic	0.202	0.0902	0.368812	1.645
5	logistic regression	lbfgs	l2	220	isotonic	0.203	0.0913	0.359259	1.645
4	logistic regression	lbfgs	l2	290	isotonic	0.203	0.0937	0.369316	1.645
3	logistic regression	lbfgs	l2	500	isotonic	0.203	0.0908	0.368743	1.645
2	logistic regression	lbfgs	l2	270	isotonic	0.203	0.0916	0.358627	1.645
1	logistic regression	lbfgs	none	none	isotonic	0.204	0.0931	0.368917	1.645

Table 4.14: Hospital - best candidates for the estimator of treatment propensity regarding BAC (0.95 quantile).

Model number 2 with parameters [solver: lbgfs, penalty: l2, c: 270] and isotonic calibration had the lowest MCE (0.358627) in the subgroup of models having the 5% highest BAC and was therefore chosen as the estimator of treatment propensity. It had also a BAC of 0.203.

Figure 4.25 represents the reliability diagram of model number 2 before calibration and after Platt and isotonic calibration.



(a) Logistic regression (solver: lbgfs, penalty: l2, c: 270)

(b) (g) after Platt calibration

(c) (g) after Isotonic calibration

Figure 4.25: Hospital - reliability diagrams for the logistic regression model with parameters [solver: lbgfs, penalty: l2, c: 10] before and after calibration. The model with isotonic calibration is selected as the estimator of treatment propensity.

Treatment policies. By sequentially modifying the one-hot encoding for sites in the imputed datasets, the estimator of treatment propensity could yield the propensity for treatments at each time steps for each site.

In Figure 4.26 the propensities for each treatment and each site were averaged over time and these aggregated propensities roughly matched the treatment distribution in the datasets.

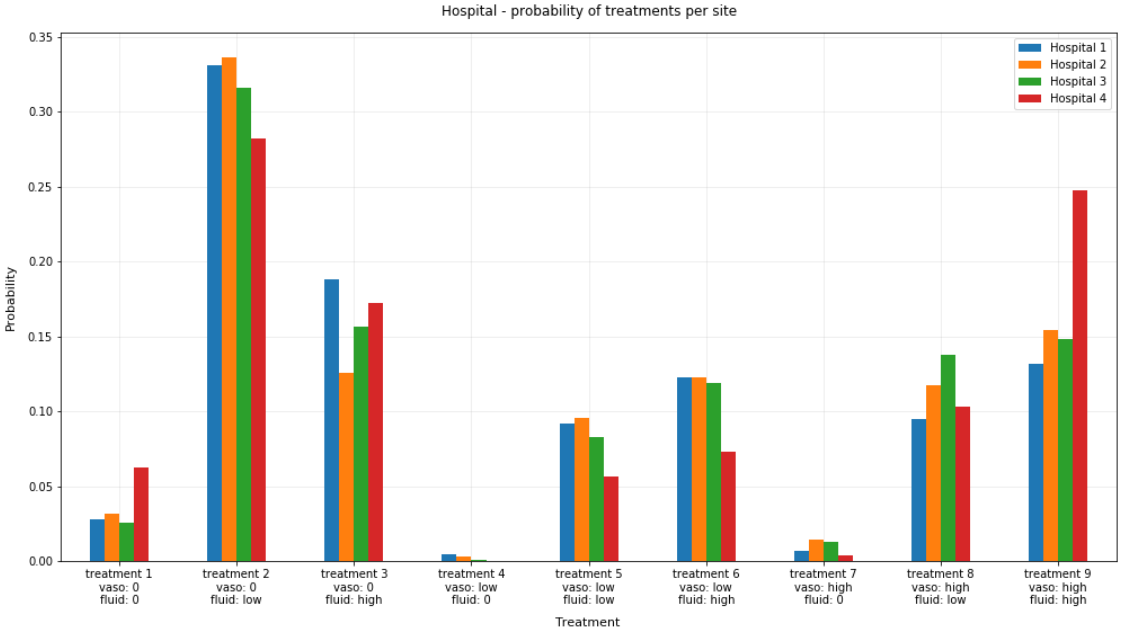


Figure 4.26: Hospital - marginal probability of treatments during ICU stay per site.

4.4.4 Practice variation

Among sites. Figure 4.27 illustrates practice variation among hospitals as the expected difference in propensity for treatment t for two sites s_1, s_2 (see Equation 3.7). For example, the probability of receiving treatment 3 in hospital 1 was about 6% higher than in ICU 2 while the probability of receiving treatment 1 in ICU 1 was about 3% lower than in the ICU 4.

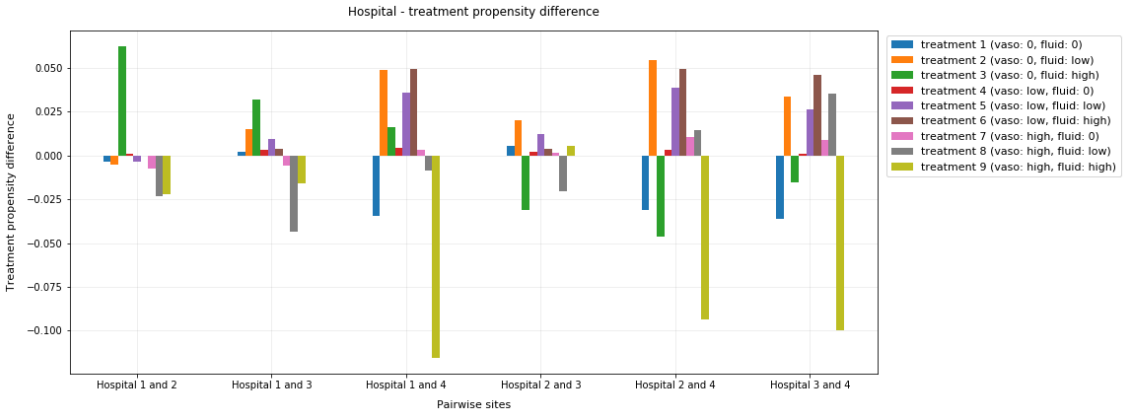


Figure 4.27: Hospital - treatment propensity difference.

Among patients. Figure 4.28 illustrates the global practice variation distribution at a patient level (see Equations 3.8-3.11) and the table 4.15 displays statistics of this distribution.

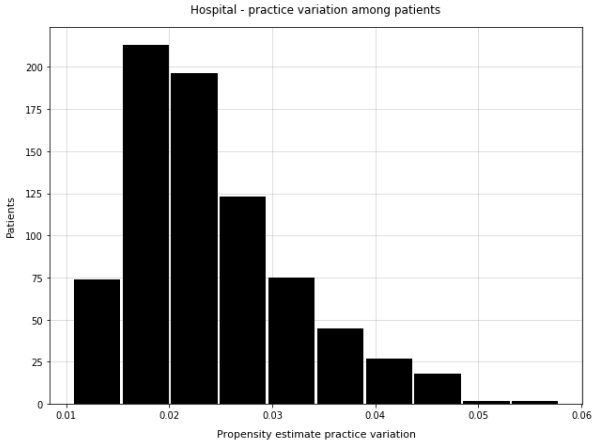


Figure 4.28: Hospital - global practice variation distribution at a patient level.

Patients	Mean	Std	Min	Max
775	0.02417	0.00806	0.01063	0.05786

Table 4.15: Hospital - statistics for the global practice variation distribution at a patient level.

Figure 4.29 illustrates the practice variation distribution at a patient level for each treatment and the table Table 4.16 displays statistics of these distributions.

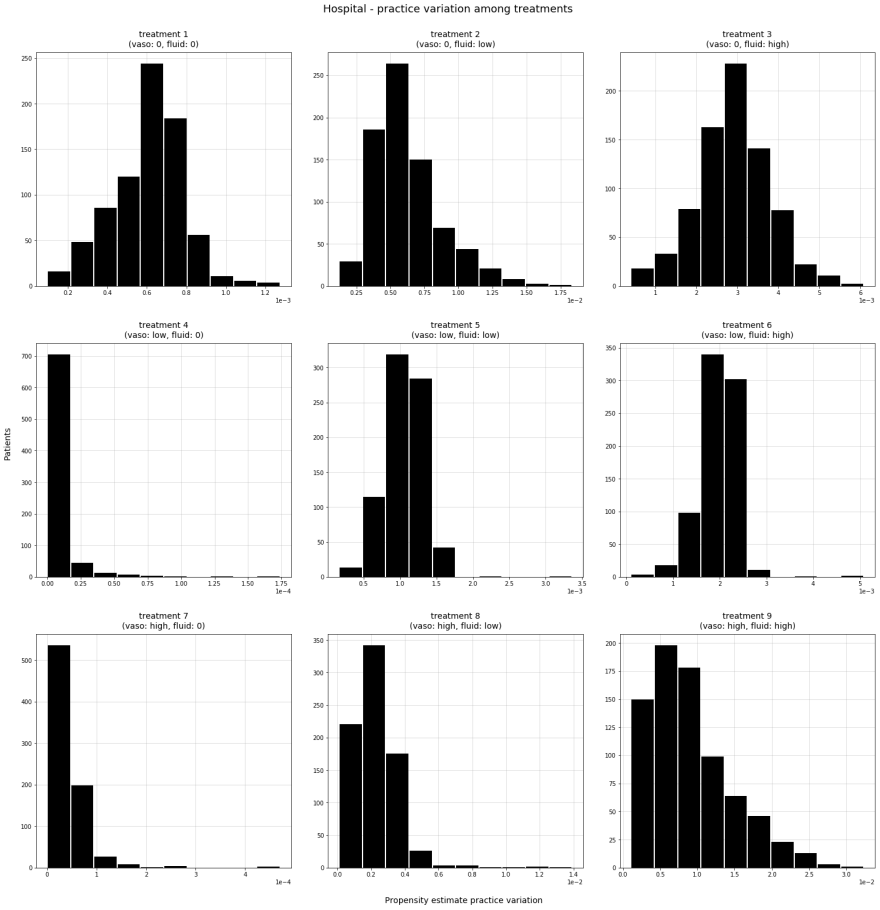


Figure 4.29: Hospital - practice variation among treatments.

Treatment	Mean	Std	Min	Max
1 (vaso: 0, fluid: 0)	0.00061	0.00018	0.00010	0.00127
2 (vaso: 0, fluid: low)	0.00622	0.00247	0.00123	0.01835
3 (vaso: 0, fluid: high)	0.00291	0.00088	0.00041	0.00609
4 (vaso: low, fluid: 0)	0.00001	0.00001	0.00000	0.00017
5 (vaso: low, fluid: low)	0.00106	0.00027	0.00016	0.00336
6 (vaso: low, fluid: high)	0.00198	0.00041	0.00010	0.00509
7 (vaso: high, fluid: 0)	0.00004	0.00004	0.00000	0.00047
8 (vaso: high, fluid: low)	0.00229	0.00130	0.00009	0.01392
9 (vaso: high, fluid: high)	0.00905	0.00547	0.00108	0.03239

Table 4.16: Hospital - practice variation among treatments.

Situate patients with high practice variation among others. Figure 4.30 represents a PCA with two components. The patients with 1% highest practice variation roughly lie on the periphery.

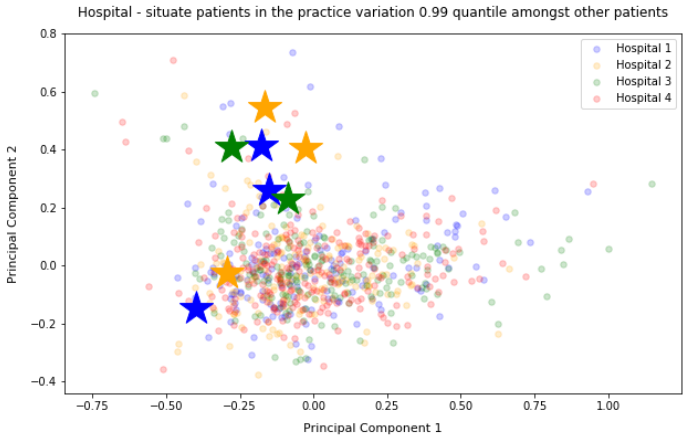


Figure 4.30: Hospital - situate patients in the practice variation 0.99 quantile amongst other patients.

Figure 4.31 individualizes five of these patients from the distribution of every covariate in the dataset.

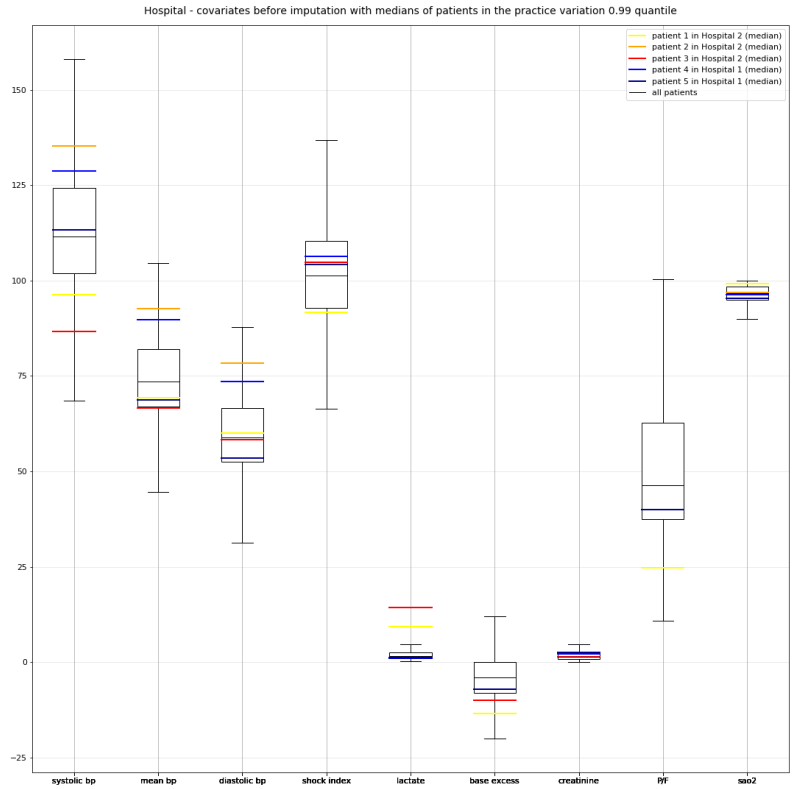


Figure 4.31: Hospital - covariates before imputation with medians of the top five patients with highest practice variation.

Correlation with covariates. Figure 4.32 shows a weak correlation between practice variation and the first component of a PCA (Pearson's correlation = -0.511).

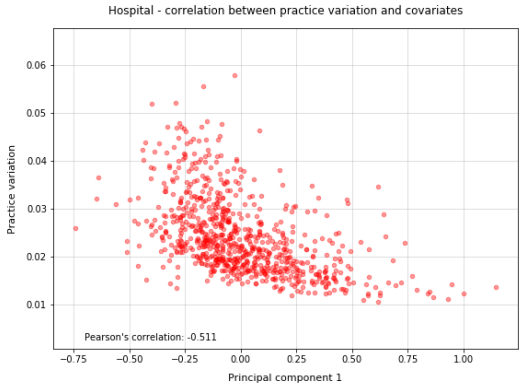


Figure 4.32: Hospital - correlation between practice variation and principal component.

This correlation was not evident when we considered each covariate individually as shown in Figure 4.33 .

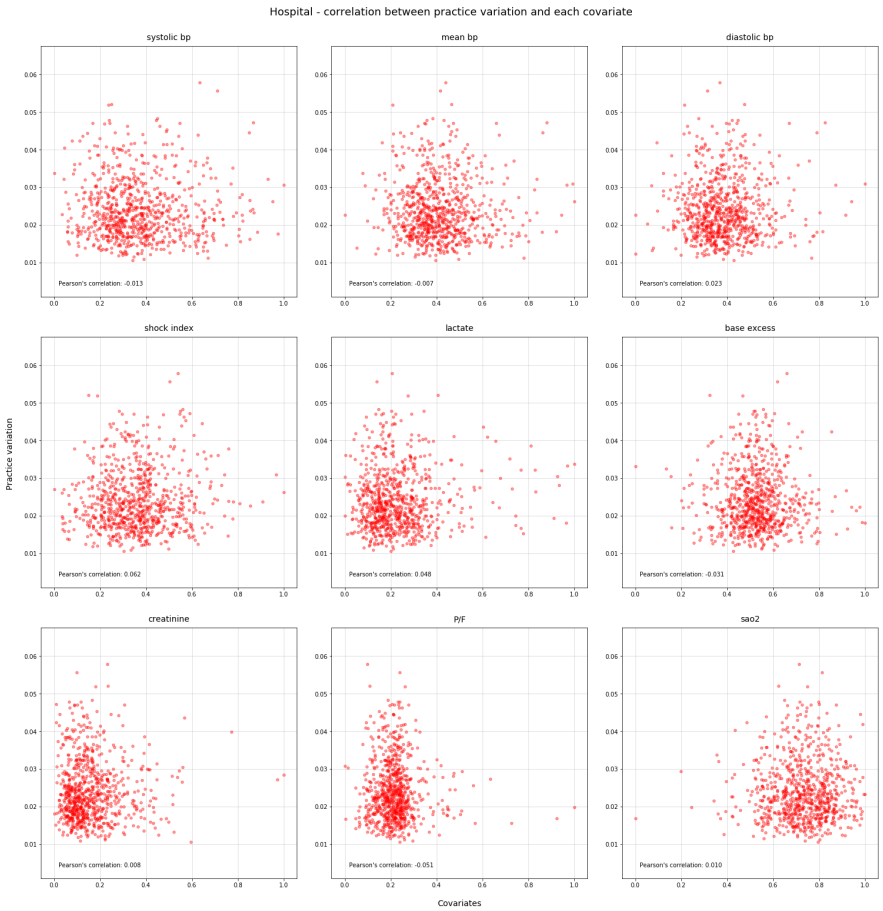


Figure 4.33: Hospital - correlation between practice variation and each covariate.

Importance sampling. Figure 4.34 shows the distributions of the weights (see equation 3.10) for each learned policy and each treatment. For a given treatment and a given policy, a ratio greater than one (green) for most of the patients could indicate that this policy recommended this treatment and vice versa (red). For example, treatment 1 was recommended by the policy in hospital 4 but not by the policy in hospital 1.

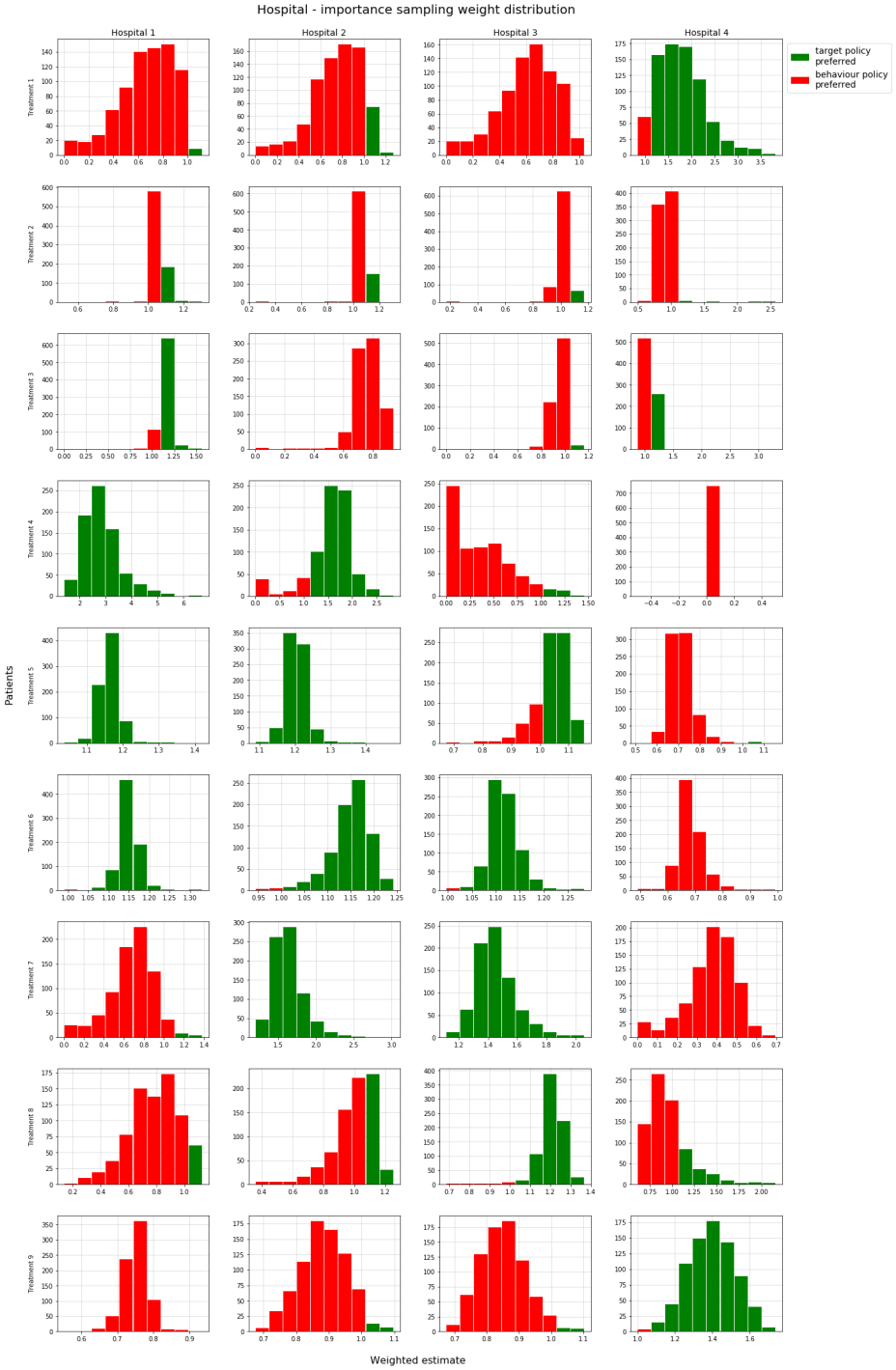


Figure 4.34: Hospital - importance sampling weight distribution.

5

Discussion

In this work, we showed the feasibility of learning site-specific treatment policies for the management of sepsis-induced circulatory failure by modeling the distribution of sites and treatments with logistic regression models.

6313 sepsis patients were retrieved from the eICU database. This number is consistent with previous studies and may vary from about 5000 to 11000 according to the stringency of sepsis definition [49, 50, 51]. However, we excluded half of the patients because the reliability and data completion could significantly vary across hospitals and/or ICUs [43]. We used a previously described method to estimate the quality of data in every site [26]. The final cohort contained 3174 sepsis patients but after selecting the four most represented hospitals and ICUs, this figure dropped to 837 and 573 respectively. To not further reduce the available data for analysis, we decided not to exclude readmitted patients (77 in total). A larger number of patients might have helped the second classifier to discriminate, especially for uncommon treatments such as high doses of vasopressor without fluid infusion.

We modelled practice variation among four ICUs and four hospitals. The hospitals contained a varying number of ICUs, between one and six. From a clinical perspective, the hospital analysis may include more heterogeneous patients and it may be more difficult to find similar patient to compare. We didn't face this issue in our work. Considering practice variation between hospitals was a way to generalize the concept and one could imagine investigating larger geographical zones like cities or regions.

The amount of fluid (ml) and the dose of vasopressor (mcg/kg/min) given to the patients are theoretically considered as continuous variables. We defined a discrete variable Treatment which could take nine values corresponding to nine combinations of fluid doses (0, low and high) and vasopressor doses (0, low and high). It would have been clinically more sensible to increase the number of combinations but it would have drastically reduced the number of patients per treatment class and the classifier would not have learned efficiently.

Merging together the nine covariates created many missing values. Indeed, vital signs such as blood pressure, SaO2 and heart rate were automatically recorded every 5 minutes while the other biological parameters such as PaO2, lactate, base excess and creatinine were only ordered when medically needed. This pattern of

missingness could correspond to missing at random (MAR) values [33]) and we decided therefore to impute missing values using linear interpolation, LOCF and MICE methods.

Since our data consisted in multidimensional discrete time series, we made the assumption that adjacent observations were similar to one another. Thus, we performed first linear imputation and forward filling. Linear interpolation was preferred for vital signs whereas forward filling was used for biological parameters. This distinction might intuitively match the cognitive process of clinicians when it comes to take medical decisions. Indeed, vital signs trends are considered more labile while biological parameters are thought more stable over time. Before imputation, vital signs and biological parameters had a missingness rate ranging from 25 to 65% and over 99% respectively. Imputation was performed with a maximum time range of 48 hours. This time limit should be clinically relevant and adapted to each feature. We chose first a unique long time interval for all the covariates for practical reasons but we didn't get the time to investigate the consequences of such a uniform choice. After this first imputation step, the missingness rate for vital signs and biological parameters ranged from 0.3 to 14% and from 13 to 54% respectively. We finally performed a multiple imputation step using the Scikit-Learn implementation of the original MICE algorithm written in R. Multiple imputation has become an increasingly popular imputation method mainly because it accounts for uncertainty due to missing data but also because it is easy to use [52]. We set the number of imputation to five as it is usually recommended when the primary interest is on the point estimates [35].

In observational data, sites and treatments are not randomly assigned to patients. Thus, differences in treatment outcomes between sites may be caused by factors or confounders that predict sites or treatments rather than sites or treatments themselves. In practice, the clinical profile of admitted patients may substantially vary between hospitals because of socio-economic inequalities and those with sicker patients may exhibit worse outcomes. To minimize the bias due to confounders, site-specific treatment policies for the management of sepsis-induced circulatory failure were learned by modeling the distribution of sites and treatments using logistic regression models. We defined a cohort of comparable patients across sites who had approximately the same probability of being hospitalized in every site given a set of clinical features and we then estimated the probability of receiving the treatments given each site. P. Rosenbaum and D. implemented the propensity score matching procedure in 1983 for reducing the risk of bias due to confounders when comparing the outcome between treated and non-treated groups [53]. In brief, this multi-step procedure estimates the propensity for treatment given clinical covariates for every patient and then matches each treated patient to one or more non-treated patients on propensity score using different possible methods. We used a similar approach to account for confounders. However, we performed a two-step regression because we wanted first to match comparable patients across sites and then to quantify the differences in received treatments in these patients. The advantage of this two-step procedure compared to a one-step approach which would have predicted both sites and treatments was the possibility of selecting the cohort of comparable patients

using a probability threshold irrespective of the received treatments.

The first logistic regression model learned to predict sites given covariates coded as discrete time series with 4-hours time steps and the covariates that were likely to be found in every site where used to select patients. As described in the method, the threshold α had to be sufficiently high to be confident about patient similarities but also low to retain enough patients for the next step. Moreover, a patient was represented by several covariates, depending on the length of stay. Selecting only the patients that had all their covariates fulfilling the α criteria would have drastically reduced the cohorts' size. With a threshold α between 10% (ICU) and 15% (hospitals) and a threshold δ of 90% we were able to create a sufficiently large cohort of "similar" patients across different sites in the database. Thus, about 75% (ICU) and 92% (hospitals) of the septic patients could be considered as comparable, i.e. having a probability of being hospitalized in every site greater than 10% and 15% respectively. (Answer to question I)

Another approach for retrieving comparable patients in observational data consists in computing pairwise similarity between patients using a specified distance metric such as Euclidean distance, cosine distance or Mahalanobis distance [54]. A short distance between two feature vectors implies high similarity between patients. However, this method could not be used to estimate site-specific treatment policies therefore we considered that it was more convenient and straightforward to train the same type of model throughout the project.

Although both classifiers were trained with time series data, the temporal aspect was not taking into account and covariates were assumed to be correlated with sites and treatment irrespective of their timing.

Due to time constraint, a total of 48 distinct parameter settings were tested and the performance och each logistic regression model $\mathrm{m_p}$ $p \in \{1, ..., 48\}$ was evaluated with BAC. These logistic regression models were parameterized with the multinomial option and a lbfgs solver. 48 parameter combinations were obtained from different penalties ['none', 'l2'] and regularization strength [0.00001, ... , 500].

The datasets used for predicting hospitals and ICUs were imbalanced [0.2, 0.25, 0.25, 0.3] and [0.2, 0.2, 0.3, 0.3] respectively. However, we assumed that imbalance was not significant and that resampling or generating synthetic samples like Synthetic Minority Over-sampling Technique (SMOTE) were not indicated. We opted instead for a performance metric that could deal with class imbalance such as BAC.

The datasets used for predicting treatments were clearly imbalanced. Two classes had more than one thousand observations, five classes had a few hundred ones and two classes had only a few dozen ones. We deemed that oversampling or performing SMOTE would change drastically the original distributions and giving too much weight to uncommon observations seemed inappropriate. Once again we preferred to use the metric BAC which accounts for the imbalance in classes by taking the average of recall obtained on each class.

We wanted our estimators to be reliable not only for class prediction but also for estimating propensity for each class. The best models had therefore to exhibit both high BAC and low MCE. We preferred MCE to ECE since we wanted models that minimize the maximum discrepancy between accuracy and confidence as recommended for safety critical applications.

In the ICU analysis, the best 5% estimators for predicting sites had a BAC around 0.34 (dummy classifier 0.25) and a MCE between 0.12 and 0.13 while the best 5% estimators for predicting treatments had an a BAC around 0.14 (dummy classifier 0.11) and a MCE between 0.45 and 0.50. The quite low performance for predicting site was anticipated given the rather similar covariates distribution across ICUs as shown in Figure 4.8. The even lower performance for predicting treatments was also foreseeable given that similar covariates had been selected by the estimator of site propensity. A similar pattern was found in the hospital analysis where the best 5% estimators for predicting sites had an accuracy around 0.32 (dummy classifier 0.25) and a MCE between 0.22 and 0.5 while the best 5% estimators for predicting treatments had a BAC around 0.2 (dummy classifier 0.11) and a MCE between 0.36. The slightly higher performance for predicting treatment in the hospital analysis compared to the ICU analysis (0.2 vs 0.14, respectively) might be explained by a larger dataset (11273 vs 5784 instances respectively).

For both analysis, i.e. for ICUs and for hospitals, we selected the final estimators f and g that had the lowest MCE among the models that exhibited the 5% best BAC.

We used our trained estimator of treatment propensity g to investigate the treatment policy for each site and we could reveal varying propensities for treatments between sites, i.e. we could identify distinct treatment practices between these sites. (Answer to question II). However, most of the differences in propensities for treatments between sites were less than 5%, for both the hospital and ICU analysis, and the clinical significance of such magnitude is unclear.

To analyse practice variation between patients, we made the weak assumption that a patient ICU stay could be summarized by the average value of all the time steps of the covariates. This unrealistic assumption introduced bias but considerably simplified the modeling of practice variation PV.

At a patient level, we defined PV for a given patient with given covariates as the distance between the likelihood to get a certain treatment t in a certain site s and the expectation of the likelihood to get the same treatment t over all the sites with the assumption that this patient had the same likelihood of being treated in s as he had in the actual dataset.

We could plot the distributions of PV over patients (Figures 4.15, 4.16, 4.28 and 4.29) and could individualize therefore the patients who had the 1% most unusual treatments, both for the hospital and ICU analysis (4 and 5 patients respectively).

Then we compared the covariates distributions of these particular patients with the whole cohort (Figure 4.18 and Figure 4.31) and could visually notice, at least for the

hospital sub-analysis, a certain degree of divergence. This pattern was less obvious for the ICU sub-analysis but the small number of patients limited the conclusions that could be drawn. It would be of interest to investigate if some additional features such as age and comorbidities could explain the different treatment regimen received by these particular patients.

We tried to visualise how PV among patients could be correlated to their covariates. We performed a principal component analysis (PCA) to convert the nine covariates into one component and we plotted patient's practice variation against the unique principal component.

The correlation was weak and almost equivalent between ICUs (0.59) and hospitals analysis (0.51). Then we separately plotted patient's practice variation against each covariate but no significant correlation could be revealed. We could imagine that for a particular covariate, it would exist a range of values (extreme values for examples) where practice variation would become more marked and revealing therefore more uncertainty in the clinical management.

Our work consisted in modeling practice variation in the management of circulatory failure during sepsis across sites. For each site the estimator g learned from the data a specific treatment policy. The next step should investigate which policy gives the best outcome such as mortality or ICU length of stay. Since deploying a bad policy would be costly and dangerous for patients, it is of upmost importance to correctly asses the performance of a learned policy before execution in the real world. Evaluating the outcomes obtained by a learned policy from data generated by another policy is called off-policy evaluation. Importance sampling is a popular method for off-policy evaluation in reinforcement learning setting that re-weights off-policy outcomes to account for differences in the likelihood of the outcomes between the learned policy and the behaviour policy that generates the data. This method gives unbiased estimates of the performance of a policy provided a sufficient number of samples [29]. However if the policy to evaluate and the behaviour policy differ significantly because clinicians made choices that the algorithm didn't recommend, the number of informative samples may be very small and the variance of the importance sampling estimator will increase. A reduced number of informative samples may therefore limit the reliability of policy evaluation [55]. Even if we didn't investigate any outcome, we plotted the distribution of the importance sampling weights over patients for the different treatment policies (Figure 4.21 and Figure 4.34) and could notice that these weight mostly lay around one, except for the more uncommon treatments. In reinforcement learning, the weights are calculated for every time step until the outcome and then multiplied all together. In sepsis studies, the mortality outcome may occur several days after admission resulting in long trajectories and very small final weights. Since we condensed an entire ICU stay which could contain up to 18 time steps into one instance, we could not observe the issue of reduced informative samples. Our work could be continued by computing the importance sampling weights for each time step and analysing precisely which profile of a patient's history are excluded by the re-weighted method.

6

Conclusion

Our work demonstrated the practicability of modeling practice variation among distinct sites in the management of sepsis-induced circulatory failure using retrospective data. We were able to identify comparable patients across different sites in the database and to observe distinct treatment practices between these sites. The next step would be to investigate this practice variation in terms of outcome such as mortality, that is to find the treatment policies which are associated with better survival. However, stronger assumptions should be made and the time aspect of the data should be taken into account. Finally, the quality of policy evaluation should be carefully assessed in particular the size of informative samples should be clearly documented.

Bibliography

[1] Mervyn Singer, Clifford S Deutschman, Christopher Warren Seymour, Manu Shankar-Hari, Djillali Annane, Michael Bauer, Rinaldo Bellomo, Gordon R Bernard, Jean-Daniel Chiche, Craig M Coopersmith, et al. The third international consensus definitions for sepsis and septic shock (sepsis-3). *Jama*, 315(8):801–810, 2016.

[2] Fabio Guarracino, Pietro Bertini, and Michael R Pinsky. Cardiovascular determinants of resuscitation from sepsis and septic shock. *Critical Care*, 23(1):1–13, 2019.

[3] Konrad Reinhart, Ron Daniels, Niranjan Kissoon, Flavia R Machado, Raymond D Schachter, and Simon Finfer. Recognizing sepsis as a global health priority—a who resolution. *New England Journal of Medicine*, 377(5):414–417, 2017.

[4] Kristina E Rudd, Sarah Charlotte Johnson, Kareha M Agesa, Katya Anne Shackelford, Derrick Tsoi, Daniel Rhodes Kievlan, Danny V Colombara, Kevin S Ikuta, Niranjan Kissoon, Simon Finfer, et al. Global, regional, and national sepsis incidence and mortality, 1990–2017: analysis for the global burden of disease study. *The Lancet*, 395(10219):200–211, 2020.

[5] Jean-Louis Vincent, John C Marshall, Silvio A Ñamendys-Silva, Bruno François, Ignacio Martin-Loeches, Jeffrey Lipman, Konrad Reinhart, Massimo Antonelli, Peter Pickkers, Hassane Njimi, et al. Assessment of the worldwide burden of critical illness: the intensive care over nations (icon) audit. *The lancet Respiratory medicine*, 2(5):380–386, 2014.

[6] Ezekiel J Emanuel, Govind Persad, Ross Upshur, Beatriz Thome, Michael Parker, Aaron Glickman, Cathy Zhang, Connor Boyle, Maxwell Smith, and James P Phillips. Fair allocation of scarce medical resources in the time of covid-19, 2020.

[7] Andrew Rhodes, Laura E Evans, Waleed Alhazzani, Mitchell M Levy, Massimo Antonelli, Ricard Ferrer, Anand Kumar, Jonathan E Sevransky, Charles L Sprung, Mark E Nunnally, et al. Surviving sepsis campaign: international guidelines for management of sepsis and septic shock: 2016. *Intensive care medicine*, 43(3):304–377, 2017.

[8] Michael D Cabana, Cynthia S Rand, Neil R Powe, Albert W Wu, Modena H

Wilson, Paul-Andre C Abboud, and Haya R Rubin. Why don't physicians follow clinical practice guidelines?: A framework for improvement. *Jama*, 282(15):1458–1465, 1999.

[9] Karl W Thomas. Adoption of sepsis bundles in the emergency room and intensive care unit: A model for quality improvement. *Critical care medicine*, 35(4):1210–1212, 2007.

[10] Frank M Brunkhorst, Christoph Engel, Max Ragaller, Tobias Welte, Rolf Rossaint, Herwig Gerlach, Konstantin Mayer, Stefan John, Frank Stuber, Norbert Weiler, et al. Practice and perception—a nationwide survey of therapy habits in sepsis. *Critical care medicine*, 36(10):2719–2725, 2008.

[11] Mitchell M Levy, R Phillip Dellinger, Sean R Townsend, Walter T Linde-Zwirble, John C Marshall, Julian Bion, Christa Schorr, Antonio Artigas, Graham Ramsay, Richard Beale, et al. The surviving sepsis campaign: results of an international guideline-based performance improvement program targeting severe sepsis. *Intensive care medicine*, 36(2):222–231, 2010.

[12] Tom J Pollard, Alistair EW Johnson, Jesse D Raffa, Leo A Celi, Roger G Mark, and Omar Badawi. The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific data*, 5:180178, 2018.

[13] David P Byar, Richard M Simon, William T Friedewald, James J Schlesselman, David L DeMets, Jonas H Ellenberg, Mitchell H Gail, and James H Ware. Randomized clinical trials: perspectives on some recent ideas. *New England Journal of Medicine*, 295(2):74–80, 1976.

[14] Nick Black. Why we need observational studies to evaluate the effectiveness of health care. *Bmj*, 312(7040):1215–1218, 1996.

[15] David S Jones and Scott H Podolsky. The history and fate of the gold standard. *The Lancet*, 385(9977):1502–1503, 2015.

[16] L Nelson Sanchez-Pinto, Yuan Luo, and Matthew M Churpek. Big data and data science in critical care. *Chest*, 154(5):1239–1248, 2018.

[17] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[18] Eric J Topol. High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine*, 25(1):44–56, 2019.

[19] Md Mohaimenul Islam, Tahmina Nasrin, Bruno Andreas Walther, Chieh-Chen Wu, Hsuan-Chia Yang, and Yu-Chuan Li. Prediction of sepsis patients using machine learning approach: a meta-analysis. *Computer methods and programs in biomedicine*, 170:1–9, 2019.

[20] Qingqing Mao, Melissa Jay, Jana L Hoffman, Jacob Calvert, Christopher Barton, David Shimabukuro, Lisa Shieh, Uli Chettipally, Grant Fletcher, Yaniv

Kerem, et al. Multicentre validation of a sepsis prediction algorithm using only vital sign data in the emergency department, general ward and icu. *BMJ open*, 8(1), 2018.

[21] David W Shimabukuro, Christopher W Barton, Mitchell D Feldman, Samson J Mataraso, and Ritankar Das. Effect of a machine learning-based severe sepsis prediction algorithm on patient survival and hospital length of stay: a randomised clinical trial. *BMJ open respiratory research*, 4(1), 2017.

[22] John P Donnelly, Monika M Safford, Nathan I Shapiro, John W Baddley, and Henry E Wang. Application of the third international consensus definitions for sepsis (sepsis-3) classification: a retrospective population-based cohort study. *The Lancet infectious diseases*, 17(6):661–670, 2017.

[23] Christopher W Seymour, Jason N Kennedy, Shu Wang, Chung-Chou H Chang, Corrine F Elliott, Zhongying Xu, Scott Berry, Gilles Clermont, Gregory Cooper, Hernando Gomez, et al. Derivation, validation, and potential treatment implications of novel clinical phenotypes for sepsis. *Jama*, 321(20):2003–2017, 2019.

[24] David B Antcliffe, Katie L Burnham, Farah Al-Beidh, Shalini Santhakumaran, Stephen J Brett, Charles J Hinds, Deborah Ashby, Julian C Knight, and Anthony C Gordon. Transcriptomic signatures in sepsis and a differential response to steroids. from the vanish randomized trial. *American journal of respiratory and critical care medicine*, 199(8):980–986, 2019.

[25] Matthieu Komorowski. Artificial intelligence in intensive care: are we there yet? *Intensive care medicine*, 45(9):1298–1300, 2019.

[26] Matthieu Komorowski, Leo A Celi, Omar Badawi, Anthony C Gordon, and A Aldo Faisal. The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nature medicine*, 24(11):1716–1720, 2018.

[27] Omer Gottesman, Fredrik Johansson, Joshua Meier, Jack Dent, Donghun Lee, Srivatsan Srinivasan, Linying Zhang, Yi Ding, David Wihl, Xuefeng Peng, et al. Evaluating reinforcement learning algorithms in observational health settings. *arXiv preprint arXiv:1805.12298*, 2018.

[28] Russell Jeter, Christopher Josef, Supreeth Shashikumar, and Shamim Nemati. Does the" artificial intelligence clinician" learn optimal treatment strategies for sepsis in intensive care? *arXiv preprint arXiv:1902.03271*, 2019.

[29] Doina Precup. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, page 80, 2000.

[30] Reuven Y Rubinstein and Dirk P Kroese. *Simulation and the Monte Carlo method*, volume 10. John Wiley & Sons, 2016.

[31] Katherine J Lee, Kate M Tilling, Rosie P Cornish, Roderick JA Little, Melanie L Bell, Els Goetghebeur, Joseph W Hogan, James R Carpenter, et al. Framework for the treatment and reporting of missing data in observational studies: The treatment and reporting of missing data in observational studies framework. *Journal of Clinical Epidemiology*, 134:79–88, 2021.

[32] Iris Eekhout, R Michiel de Boer, Jos WR Twisk, Henrica CW de Vet, and Martijn W Heymans. Missing data: a systematic review of how they are reported and handled. *Epidemiology*, 23(5):729–732, 2012.

[33] Donald B Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

[34] Mattias Albinsson and Erik Gillsbro. Imputation methods in dialysis data. *Master's Theses in Mathematical Sciences*, 2017.

[35] Stef Van Buuren. *Flexible imputation of missing data*. CRC press, 2018.

[36] Donald B Rubin. Multiple imputation for survey nonresponse, 1987.

[37] Angela M Wood, Patrick Royston, and Ian R White. The estimation and use of predictions for the assessment of model performance using large samples with multiply imputed data. *Biometrical Journal*, 57(4):614–632, 2015.

[38] Aniruddh Raghu, Omer Gottesman, Yao Liu, Matthieu Komorowski, Aldo Faisal, Finale Doshi-Velez, and Emma Brunskill. Behaviour policy estimation in off-policy policy evaluation: Calibration matters. *arXiv preprint arXiv:1807.01066*, 2018.

[39] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

[40] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[41] World Health Organization. International classification of diseases , 2021. [Online; accessed 26-May-2021].

[42] Jean-Louis Vincent, Arnaldo De Mendonça, Francis Cantraine, Rui Moreno, Jukka Takala, Peter M Suter, Charles L Sprung, Francis Colardyn, and Serge Blecher. Use of the sofa score to assess the incidence of organ dysfunction/failure in intensive care units: results of a multicenter, prospective study. *Critical care medicine*, 26(11):1793–1800, 1998.

[43] Heather M O'Halloran, Kenneth Kwong, Richard A Veldhoen, and David M Maslove. Characterizing the patients, hospitals, and data quality of the eicu collaborative research database. *Critical Care Medicine*, 48(12):1737–1743, 2020.

[44] William A Knaus, Elizabeth A Draper, Douglas P Wagner, and Jack E Zimmerman. Apache ii: a severity of disease classification system. *Critical care medicine*, 13(10):818–829, 1985.

[45] TJ Pollard et al. Mit-lcp/eicu-code: eicu-crd code repository. *Zenodo*, 2018.

[46] Derek C. Angus and Tom van der Poll. Severe sepsis and septic shock. *New England Journal of Medicine*, 369(9):840–851, 2013. PMID: 23984731.

[47] Shruti Goradia, Arwa Abu Sardaneh, Sujita W Narayan, Jonathan Penm, and

Asad E Patanwala. Vasopressor dose equivalence: A scoping review and suggested formula. *Journal of Critical Care*, 2020.

[48] Willem van den Boom, Michael Hoy, Jagadish Sankaran, Mengru Liu, Haroun Chahed, Mengling Feng, and Kay Choong See. The search for optimal oxygen saturation targets in critically ill patients: observational data from large icu databases. *Chest*, 157(3):566–573, 2020.

[49] Yu-Pu Wu and Julie C Lauffenburger. Effectiveness of corticosteroids in patients with sepsis or septic shock using the new third international consensus definitions (sepsis-3): A retrospective observational study. *Plos one*, 15(12):e0243149, 2020.

[50] Saskya Byerly, Joshua P Parreco, Hahn Soe-Lin, Jonathan J Parks, Eugenia E Lee, Ilya Shnaydman, Alejandro Mantero, D Dante Yeh, Nicholas Namias, and Rishi Rattan. Vitamin c and thiamine are associated with lower mortality in sepsis. *Journal of Trauma and Acute Care Surgery*, 89(1):111–117, 2020.

[51] Yide Li, Yingfang She, Le Fu, Ruitong Zhou, Wendi Xiang, and Liang Luo. Association between red cell distribution width and hospital mortality in patients with sepsis. *Journal of International Medical Research*, 49(4):03000605211004221, 2021.

[52] Panteha Hayati Rezvan, Katherine J Lee, and Julie A Simpson. The rise of multiple imputation: a review of the reporting and implementation of the method in medical research. *BMC medical research methodology*, 15(1):1–14, 2015.

[53] PAUL R. ROSENBAUM and DONALD B. RUBIN. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 04 1983.

[54] Leyu Dai, He Zhu, and Dianbo Liu. Patient similarity: methods and applications, 2020.

[55] Omer Gottesman, Fredrik Johansson, Matthieu Komorowski, Aldo Faisal, David Sontag, Finale Doshi-Velez, and Leo Anthony Celi. Guidelines for reinforcement learning in healthcare. *Nat Med*, 25(1):16–18, 2019.

A

Appendix 1

In this appendix we append the SQL codes used for extraction of data from the eICU database.

A.1 admission_diagnosis.sql

```
DROP TABLE IF EXISTS admission_diagnosis CASCADE;
CREATE TABLE admission_diagnosis AS

WITH
t1_adm AS -- admission diagnosis AS SEPSIS from admissiondx table (23 136)
(
    SELECT
        patientUnitStayID,
        admitdxname
    FROM
        admissiondx
    WHERE
        LOWER(admitdxname) LIKE '%sepsis%'
),

t2_adm AS
(
    SELECT
        t1_adm.patientUnitStayID,
        CASE WHEN t1_adm.admitdxname LIKE '%pulmonary%' THEN 'pulmonary'
         WHEN t1_adm.admitdxname LIKE '%renal%' THEN 'UTI_renal'
         WHEN t1_adm.admitdxname LIKE '%GI%' THEN 'GI'
         WHEN t1_adm.admitdxname LIKE '%cutaneous%' THEN 'soft_tissue'
         WHEN t1_adm.admitdxname LIKE '%gyneco%' THEN 'gyneco'
         WHEN t1_adm.admitdxname LIKE '%other%'
            OR t1_adm.admitdxname LIKE '%unknown%'
            OR t1_adm.admitdxname IS NULL THEN 'other_unknown' ELSE '0' END AS sepsis_type
    FROM t1_adm
)

SELECT * FROM t2_adm;
```

A.2 antibio_1.sql

```
DROP TABLE IF EXISTS antibio_1 CASCADE;
CREATE TABLE antibio_1 AS

WITH
t_medication as
(
    SELECT
        distinct patientUnitStayID,
        drugStartOffset AS antibio_time
    FROM
        medication
    WHERE
        lower(drugname) LIKE '%adoxa%' OR lower(drugname) LIKE '%ala-tet%'
        OR lower(drugname) LIKE '%alodox%' OR lower(drugname) LIKE '%amikacin%'
        OR lower(drugname) LIKE '%amikin%' OR lower(drugname) LIKE '%amoxicillin%'
        OR lower(drugname) LIKE '%clavulanate%' OR lower(drugname) LIKE '%ampicillin%'
        OR lower(drugname) LIKE '%augmentin%' OR lower(drugname) LIKE '%avelox%'
        OR lower(drugname) LIKE '%avidoxy%' OR lower(drugname) LIKE '%azactam%'
        OR lower(drugname) LIKE '%azithromycin%' OR lower(drugname) LIKE '%aztreonam%'
```

```
        OR lower(drugname) LIKE '%axetil%' OR lower(drugname) LIKE '%bactocill%'
        OR lower(drugname) LIKE '%bactrim%' OR lower(drugname) LIKE '%bethkis%'
        OR lower(drugname) LIKE '%biaxin%' OR lower(drugname) LIKE '%bicillin l-a%'
        OR lower(drugname) LIKE '%cayston%' OR lower(drugname) LIKE '%cefazolin%'
        OR lower(drugname) LIKE '%cedax%' OR lower(drugname) LIKE '%cefoxitin%'
        OR lower(drugname) LIKE '%ceftazidime%' OR lower(drugname) LIKE '%cefaclor%'
        OR lower(drugname) LIKE '%cefadroxil%' OR lower(drugname) LIKE '%cefdinir%'
        OR lower(drugname) LIKE '%cefditoren%' OR lower(drugname) LIKE '%cefepime%'
        OR lower(drugname) LIKE '%cefotetan%' OR lower(drugname) LIKE '%cefotaxime%'
        OR lower(drugname) LIKE '%cefpodoxime%' OR lower(drugname) LIKE '%cefprozil%'
        OR lower(drugname) LIKE '%ceftibuten%' OR lower(drugname) LIKE '%ceftin%'
        OR lower(drugname) LIKE '%cefuroxime%' OR lower(drugname) LIKE '%cefuroxime%'
        OR lower(drugname) LIKE '%cephalexin%' OR lower(drugname) LIKE '%chloramphenicol%'
        OR lower(drugname) LIKE '%cipro%' OR lower(drugname) LIKE '%ciprofloxacin%'
        OR lower(drugname) LIKE '%claforan%' OR lower(drugname) LIKE '%clarithromycin%'
        OR lower(drugname) LIKE '%cleocin%' OR lower(drugname) LIKE '%clindamycin%'
        OR lower(drugname) LIKE '%cubicin%' OR lower(drugname) LIKE '%dicloxacillin%'
        OR lower(drugname) LIKE '%doryx%' OR lower(drugname) LIKE '%doxycycline%'
        OR lower(drugname) LIKE '%duricef%' OR lower(drugname) LIKE '%dynacin%'
        OR lower(drugname) LIKE '%ery-tab%' OR lower(drugname) LIKE '%eryped%'
        OR lower(drugname) LIKE '%eryc%' OR lower(drugname) LIKE '%erythrocin%'
        OR lower(drugname) LIKE '%erythromycin%' OR lower(drugname) LIKE '%factive%'
        OR lower(drugname) LIKE '%flagyl%' OR lower(drugname) LIKE '%fortaz%'
        OR lower(drugname) LIKE '%furadantin%' OR lower(drugname) LIKE '%garamycin%'
        OR lower(drugname) LIKE '%gentamicin%' OR lower(drugname) LIKE '%kanamycin%'
        OR lower(drugname) LIKE '%keflex%' OR lower(drugname) LIKE '%ketek%'
        OR lower(drugname) LIKE '%levaquin%' OR lower(drugname) LIKE '%levofloxacin%'
        OR lower(drugname) LIKE '%lincocin%' OR lower(drugname) LIKE '%macrobid%'
        OR lower(drugname) LIKE '%macrodantin%' OR lower(drugname) LIKE '%maxipime%'
        OR lower(drugname) LIKE '%mefoxin%' OR lower(drugname) LIKE '%metronidazole%'
        OR lower(drugname) LIKE '%minocin%' OR lower(drugname) LIKE '%minocycline%'
        OR lower(drugname) LIKE '%monodox%' OR lower(drugname) LIKE '%monurol%'
        OR lower(drugname) LIKE '%morgidox%' OR lower(drugname) LIKE '%moxatag%'
        OR lower(drugname) LIKE '%moxifloxacin%' OR lower(drugname) LIKE '%myrac%'
        OR lower(drugname) LIKE '%nafcillin sodium%' OR lower(drugname) LIKE '%nicazel doxy 30%'
        OR lower(drugname) LIKE '%nitrofurantoin%' OR lower(drugname) LIKE '%noroxin%'
        OR lower(drugname) LIKE '%ocudox%' OR lower(drugname) LIKE '%ofloxacin%'
        OR lower(drugname) LIKE '%omnicef%' OR lower(drugname) LIKE '%oracea%'
        OR lower(drugname) LIKE '%oraxyl%' OR lower(drugname) LIKE '%oxacillin%'
        OR lower(drugname) LIKE '%pc pen vk%' OR lower(drugname) LIKE '%pce dispertab%'
        OR lower(drugname) LIKE '%panixine%' OR lower(drugname) LIKE '%pediazole%'
        OR lower(drugname) LIKE '%penicillin%' OR lower(drugname) LIKE '%periostat%'
        OR lower(drugname) LIKE '%pfizerpen%' OR lower(drugname) LIKE '%piperacillin%'
        OR lower(drugname) LIKE '%tazobactam%' OR lower(drugname) LIKE '%primsol%'
        OR lower(drugname) LIKE '%proquin%' OR lower(drugname) LIKE '%raniclor%'
        OR lower(drugname) LIKE '%rifadin%' OR lower(drugname) LIKE '%rifampin%'
        OR lower(drugname) LIKE '%rocephin%' OR lower(drugname) LIKE '%smz-tmp%'
        OR lower(drugname) LIKE '%septra%' OR lower(drugname) LIKE '%septra ds%'
        OR lower(drugname) LIKE '%septra%' OR lower(drugname) LIKE '%solodyn%'
        OR lower(drugname) LIKE '%spectracef%' OR lower(drugname) LIKE '%streptomycin sulfate%'
        OR lower(drugname) LIKE '%sulfadiazine%' OR lower(drugname) LIKE '%sulfamethoxazole%'
        OR lower(drugname) LIKE '%trimethoprim%' OR lower(drugname) LIKE '%sulfatrim%'
        OR lower(drugname) LIKE '%sulfisoxazole%' OR lower(drugname) LIKE '%suprax%'
        OR lower(drugname) LIKE '%synercid%' OR lower(drugname) LIKE '%tazicef%'
        OR lower(drugname) LIKE '%tetracycline%' OR lower(drugname) LIKE '%timentin%'
        OR lower(drugname) LIKE '%tobi%'  OR lower(drugname) LIKE '%tobramycin%'
        OR lower(drugname) LIKE '%trimethoprim%' OR lower(drugname) LIKE '%unasyn%'
        OR lower(drugname) LIKE '%vancocin%' OR lower(drugname) LIKE '%vancomycin%'
        OR lower(drugname) LIKE '%vantin%' OR lower(drugname) LIKE '%vibativ%'
        OR lower(drugname) LIKE '%vibra-tabs%' OR lower(drugname) LIKE '%vibramycin%'
        OR lower(drugname) LIKE '%zinacef%' OR lower(drugname) LIKE '%zithromax%'
        OR lower(drugname) LIKE '%zmax%' OR lower(drugname) LIKE '%zosyn%'
        OR lower(drugname) LIKE '%zyvox%'
)

SELECT * FROM t_medication
```

A.3 antibio_2.sql

```
DROP TABLE IF EXISTS antibio_2 CASCADE;
CREATE TABLE antibio_2 AS

WITH
t_infusion as
(
    SELECT
        distinct patientUnitStayID,
        infusionOffset as antibio_time
    FROM
        infusionDrug
    WHERE
        lower(drugname) LIKE '%adoxa%' OR lower(drugname) LIKE '%ala-tet%'
        OR lower(drugname) LIKE '%alodox%' OR lower(drugname) LIKE '%amikacin%'
        OR lower(drugname) LIKE '%amikin%' OR lower(drugname) LIKE '%amoxicillin%'
        OR lower(drugname) LIKE '%clavulanate%' OR lower(drugname) LIKE '%ampicillin%'
        OR lower(drugname) LIKE '%augmentin%' OR lower(drugname) LIKE '%avelox%'
        OR lower(drugname) LIKE '%avidoxy%' OR lower(drugname) LIKE '%azactam%'
        OR lower(drugname) LIKE '%azithromycin%' OR lower(drugname) LIKE '%aztreonam%'
        OR lower(drugname) LIKE '%axetil%' OR lower(drugname) LIKE '%bactocill%'
```

```
        OR lower(drugname) LIKE '%bactrim%' OR lower(drugname) LIKE '%bethkis%'
        OR lower(drugname) LIKE '%biaxin%' OR lower(drugname) LIKE '%bicillin l-a%'
        OR lower(drugname) LIKE '%cayston%' OR lower(drugname) LIKE '%cefazolin%'
        OR lower(drugname) LIKE '%cedax%' OR lower(drugname) LIKE '%cefoxitin%'
        OR lower(drugname) LIKE '%ceftazidime%' OR lower(drugname) LIKE '%cefaclor%'
        OR lower(drugname) LIKE '%cefadroxil%' OR lower(drugname) LIKE '%cefdinir%'
        OR lower(drugname) LIKE '%cefditoren%' OR lower(drugname) LIKE '%cefepime%'
        OR lower(drugname) LIKE '%cefotetan%' OR lower(drugname) LIKE '%cefotaxime%'
        OR lower(drugname) LIKE '%cefpodoxime%' OR lower(drugname) LIKE '%cefprozil%'
        OR lower(drugname) LIKE '%ceftibuten%' OR lower(drugname) LIKE '%ceftin%'
        OR lower(drugname) LIKE '%cefuroxime%' OR lower(drugname) LIKE '%cefuroxime%'
        OR lower(drugname) LIKE '%cephalexin%' OR lower(drugname) LIKE '%chloramphenicol%'
        OR lower(drugname) LIKE '%cipro%' OR lower(drugname) LIKE '%ciprofloxacin%'
        OR lower(drugname) LIKE '%claforan%' OR lower(drugname) LIKE '%clarithromycin%'
        OR lower(drugname) LIKE '%cleocin%' OR lower(drugname) LIKE '%clindamycin%'
        OR lower(drugname) LIKE '%cubicin%' OR lower(drugname) LIKE '%dicloxacillin%'
        OR lower(drugname) LIKE '%doryx%' OR lower(drugname) LIKE '%doxycycline%'
        OR lower(drugname) LIKE '%duricef%' OR lower(drugname) LIKE '%dynacin%'
        OR lower(drugname) LIKE '%ery-tab%' OR lower(drugname) LIKE '%eryped%'
        OR lower(drugname) LIKE '%eryc%' OR lower(drugname) LIKE '%erythrocin%'
        OR lower(drugname) LIKE '%erythromycin%' OR lower(drugname) LIKE '%factive%'
        OR lower(drugname) LIKE '%flagyl%' OR lower(drugname) LIKE '%fortaz%'
        OR lower(drugname) LIKE '%furadantin%' OR lower(drugname) LIKE '%garamycin%'
        OR lower(drugname) LIKE '%gentamicin%' OR lower(drugname) LIKE '%kanamycin%'
        OR lower(drugname) LIKE '%keflex%' OR lower(drugname) LIKE '%ketek%'
        OR lower(drugname) LIKE '%levaquin%' OR lower(drugname) LIKE '%levofloxacin%'
        OR lower(drugname) LIKE '%lincocin%' OR lower(drugname) LIKE '%macrobid%'
        OR lower(drugname) LIKE '%macrodantin%' OR lower(drugname) LIKE '%maxipime%'
        OR lower(drugname) LIKE '%mefoxin%' OR lower(drugname) LIKE '%metronidazole%'
        OR lower(drugname) LIKE '%minocin%' OR lower(drugname) LIKE '%minocycline%'
        OR lower(drugname) LIKE '%monodox%' OR lower(drugname) LIKE '%monurol%'
        OR lower(drugname) LIKE '%morgidox%' OR lower(drugname) LIKE '%moxatag%'
        OR lower(drugname) LIKE '%moxifloxacin%' OR lower(drugname) LIKE '%myrac%'
        OR lower(drugname) LIKE '%nafcillin sodium%' OR lower(drugname) LIKE '%nicazel doxy 30%'
        OR lower(drugname) LIKE '%nitrofurantoin%' OR lower(drugname) LIKE '%noroxin%'
        OR lower(drugname) LIKE '%ocudox%' OR lower(drugname) LIKE '%ofloxacin%'
        OR lower(drugname) LIKE '%omnicef%' OR lower(drugname) LIKE '%oracea%'
        OR lower(drugname) LIKE '%oraxyl%' OR lower(drugname) LIKE '%oxacillin%'
        OR lower(drugname) LIKE '%pc pen vk%' OR lower(drugname) LIKE '%pce dispertab%'
        OR lower(drugname) LIKE '%panixine%' OR lower(drugname) LIKE '%pediazole%'
        OR lower(drugname) LIKE '%penicillin%' OR lower(drugname) LIKE '%periostat%'
        OR lower(drugname) LIKE '%pfizerpen%' OR lower(drugname) LIKE '%piperacillin%'
        OR lower(drugname) LIKE '%tazobactam%' OR lower(drugname) LIKE '%primsol%'
        OR lower(drugname) LIKE '%proquin%' OR lower(drugname) LIKE '%raniclor%'
        OR lower(drugname) LIKE '%rifadin%' OR lower(drugname) LIKE '%rifampin%'
        OR lower(drugname) LIKE '%rocephin%' OR lower(drugname) LIKE '%smz-tmp%'
        OR lower(drugname) LIKE '%septra%' OR lower(drugname) LIKE '%septra ds%'
        OR lower(drugname) LIKE '%septra%' OR lower(drugname) LIKE '%solodyn%'
        OR lower(drugname) LIKE '%spectracef%' OR lower(drugname) LIKE '%streptomycin sulfate%'
        OR lower(drugname) LIKE '%sulfadiazine%' OR lower(drugname) LIKE '%sulfamethoxazole%'
        OR lower(drugname) LIKE '%trimethoprim%' OR lower(drugname) LIKE '%sulfatrim%'
        OR lower(drugname) LIKE '%sulfisoxazole%' OR lower(drugname) LIKE '%suprax%'
        OR lower(drugname) LIKE '%synercid%' OR lower(drugname) LIKE '%tazicef%'
        OR lower(drugname) LIKE '%tetracycline%' OR lower(drugname) LIKE '%timentin%'
        OR lower(drugname) LIKE '%tobi%'  OR lower(drugname) LIKE '%tobramycin%'
        OR lower(drugname) LIKE '%trimethoprim%' OR lower(drugname) LIKE '%unasyn%'
        OR lower(drugname) LIKE '%vancocin%' OR lower(drugname) LIKE '%vancomycin%'
        OR lower(drugname) LIKE '%vantin%' OR lower(drugname) LIKE '%vibativ%'
        OR lower(drugname) LIKE '%vibra-tabs%' OR lower(drugname) LIKE '%vibramycin%'
        OR lower(drugname) LIKE '%zinacef%' OR lower(drugname) LIKE '%zithromax%'
        OR lower(drugname) LIKE '%zmax%' OR lower(drugname) LIKE '%zosyn%'
        OR lower(drugname) LIKE '%zyvox%'
)

SELECT * FROM t_infusion;
```

A.4 antibio_3.sql

```
DROP TABLE IF EXISTS antibio_3 CASCADE;
CREATE TABLE antibio_3 AS

WITH
t_treatment AS
(
    SELECT
        distinct patientUnitStayID,
        treatmentoffset AS antibio_time
    FROM
        treatment
    WHERE
        lower(drugname) LIKE '%adoxa%' OR lower(drugname) LIKE '%ala-tet%'
        OR lower(drugname) LIKE '%alodox%' OR lower(drugname) LIKE '%amikacin%'
        OR lower(drugname) LIKE '%amikin%' OR lower(drugname) LIKE '%amoxicillin%'
        OR lower(drugname) LIKE '%clavulanate%' OR lower(drugname) LIKE '%ampicillin%'
        OR lower(drugname) LIKE '%augmentin%' OR lower(drugname) LIKE '%avelox%'
        OR lower(drugname) LIKE '%avidoxy%' OR lower(drugname) LIKE '%azactam%'
        OR lower(drugname) LIKE '%azithromycin%' OR lower(drugname) LIKE '%aztreonam%'
        OR lower(drugname) LIKE '%axetil%' OR lower(drugname) LIKE '%bactocill%'
        OR lower(drugname) LIKE '%bactrim%' OR lower(drugname) LIKE '%bethkis%'
```

```
        OR lower(drugname) LIKE '%biaxin%' OR lower(drugname) LIKE '%bicillin l-a%'
        OR lower(drugname) LIKE '%cayston%' OR lower(drugname) LIKE '%cefazolin%'
        OR lower(drugname) LIKE '%cedax%' OR lower(drugname) LIKE '%cefoxitin%'
        OR lower(drugname) LIKE '%ceftazidime%' OR lower(drugname) LIKE '%cefaclor%'
        OR lower(drugname) LIKE '%cefadroxil%' OR lower(drugname) LIKE '%cefdinir%'
        OR lower(drugname) LIKE '%cefditoren%' OR lower(drugname) LIKE '%cefepime%'
        OR lower(drugname) LIKE '%cefotetan%' OR lower(drugname) LIKE '%cefotaxime%'
        OR lower(drugname) LIKE '%cefpodoxime%' OR lower(drugname) LIKE '%cefprozil%'
        OR lower(drugname) LIKE '%ceftibuten%' OR lower(drugname) LIKE '%ceftin%'
        OR lower(drugname) LIKE '%cefuroxime%' OR lower(drugname) LIKE '%cefuroxime%'
        OR lower(drugname) LIKE '%cephalexin%' OR lower(drugname) LIKE '%chloramphenicol%'
        OR lower(drugname) LIKE '%cipro%' OR lower(drugname) LIKE '%ciprofloxacin%'
        OR lower(drugname) LIKE '%claforan%' OR lower(drugname) LIKE '%clarithromycin%'
        OR lower(drugname) LIKE '%cleocin%' OR lower(drugname) LIKE '%clindamycin%'
        OR lower(drugname) LIKE '%cubicin%' OR lower(drugname) LIKE '%dicloxacillin%'
        OR lower(drugname) LIKE '%doryx%' OR lower(drugname) LIKE '%doxycycline%'
        OR lower(drugname) LIKE '%duricef%' OR lower(drugname) LIKE '%dynacin%'
        OR lower(drugname) LIKE '%ery-tab%' OR lower(drugname) LIKE '%eryped%'
        OR lower(drugname) LIKE '%eryc%' OR lower(drugname) LIKE '%erythrocin%'
        OR lower(drugname) LIKE '%erythromycin%' OR lower(drugname) LIKE '%factive%'
        OR lower(drugname) LIKE '%flagyl%' OR lower(drugname) LIKE '%fortaz%'
        OR lower(drugname) LIKE '%furadantin%' OR lower(drugname) LIKE '%garamycin%'
        OR lower(drugname) LIKE '%gentamicin%' OR lower(drugname) LIKE '%kanamycin%'
        OR lower(drugname) LIKE '%keflex%' OR lower(drugname) LIKE '%ketek%'
        OR lower(drugname) LIKE '%levaquin%' OR lower(drugname) LIKE '%levofloxacin%'
        OR lower(drugname) LIKE '%lincocin%' OR lower(drugname) LIKE '%macrobid%'
        OR lower(drugname) LIKE '%macrodantin%' OR lower(drugname) LIKE '%maxipime%'
        OR lower(drugname) LIKE '%mefoxin%' OR lower(drugname) LIKE '%metronidazole%'
        OR lower(drugname) LIKE '%minocin%' OR lower(drugname) LIKE '%minocycline%'
        OR lower(drugname) LIKE '%monodox%' OR lower(drugname) LIKE '%monurol%'
        OR lower(drugname) LIKE '%morgidox%' OR lower(drugname) LIKE '%moxatag%'
        OR lower(drugname) LIKE '%moxifloxacin%' OR lower(drugname) LIKE '%myrac%'
        OR lower(drugname) LIKE '%nafcillin sodium%' OR lower(drugname) LIKE '%nicazel doxy 30%'
        OR lower(drugname) LIKE '%nitrofurantoin%' OR lower(drugname) LIKE '%noroxin%'
        OR lower(drugname) LIKE '%ocudox%' OR lower(drugname) LIKE '%ofloxacin%'
        OR lower(drugname) LIKE '%omnicef%' OR lower(drugname) LIKE '%oracea%'
        OR lower(drugname) LIKE '%oraxyl%' OR lower(drugname) LIKE '%oxacillin%'
        OR lower(drugname) LIKE '%pc pen vk%' OR lower(drugname) LIKE '%pce dispertab%'
        OR lower(drugname) LIKE '%panixine%' OR lower(drugname) LIKE '%pediazole%'
        OR lower(drugname) LIKE '%penicillin%' OR lower(drugname) LIKE '%periostat%'
        OR lower(drugname) LIKE '%pfizerpen%' OR lower(drugname) LIKE '%piperacillin%'
        OR lower(drugname) LIKE '%tazobactam%' OR lower(drugname) LIKE '%primsol%'
        OR lower(drugname) LIKE '%proquin%' OR lower(drugname) LIKE '%raniclor%'
        OR lower(drugname) LIKE '%rifadin%' OR lower(drugname) LIKE '%rifampin%'
        OR lower(drugname) LIKE '%rocephin%' OR lower(drugname) LIKE '%smz-tmp%'
        OR lower(drugname) LIKE '%septra%' OR lower(drugname) LIKE '%septra ds%'
        OR lower(drugname) LIKE '%septra%' OR lower(drugname) LIKE '%solodyn%'
        OR lower(drugname) LIKE '%spectracef%' OR lower(drugname) LIKE '%streptomycin sulfate%'
        OR lower(drugname) LIKE '%sulfadiazine%' OR lower(drugname) LIKE '%sulfamethoxazole%'
        OR lower(drugname) LIKE '%trimethoprim%' OR lower(drugname) LIKE '%sulfatrim%'
        OR lower(drugname) LIKE '%sulfisoxazole%' OR lower(drugname) LIKE '%suprax%'
        OR lower(drugname) LIKE '%synercid%' OR lower(drugname) LIKE '%tazicef%'
        OR lower(drugname) LIKE '%tetracycline%' OR lower(drugname) LIKE '%timentin%'
        OR lower(drugname) LIKE '%tobi%'  OR lower(drugname) LIKE '%tobramycin%'
        OR lower(drugname) LIKE '%trimethoprim%' OR lower(drugname) LIKE '%unasyn%'
        OR lower(drugname) LIKE '%vancocin%' OR lower(drugname) LIKE '%vancomycin%'
        OR lower(drugname) LIKE '%vantin%' OR lower(drugname) LIKE '%vibativ%'
        OR lower(drugname) LIKE '%vibra-tabs%' OR lower(drugname) LIKE '%vibramycin%'
        OR lower(drugname) LIKE '%zinacef%' OR lower(drugname) LIKE '%zithromax%'
        OR lower(drugname) LIKE '%zmax%' OR lower(drugname) LIKE '%zosyn%'
        OR lower(drugname) LIKE '%zyvox%'
)

SELECT * FROM t_treatment;
```

A.5 antibio_4.sql

```
DROP TABLE IF EXISTS antibio_4 CASCADE;
CREATE TABLE antibio_4 AS

WITH
t_intakeoutput AS
(
    SELECT
        distinct patientUnitStayID,
        intakeOutputOffset AS antibio_time
    FROM
        intakeoutput
    WHERE
        lower(drugname) LIKE '%adoxa%' OR lower(drugname) LIKE '%ala-tet%'
        OR lower(drugname) LIKE '%alodox%' OR lower(drugname) LIKE '%amikacin%'
        OR lower(drugname) LIKE '%amikin%' OR lower(drugname) LIKE '%amoxicillin%'
        OR lower(drugname) LIKE '%clavulanate%' OR lower(drugname) LIKE '%ampicillin%'
        OR lower(drugname) LIKE '%augmentin%' OR lower(drugname) LIKE '%avelox%'
        OR lower(drugname) LIKE '%avidoxy%' OR lower(drugname) LIKE '%azactam%'
        OR lower(drugname) LIKE '%azithromycin%' OR lower(drugname) LIKE '%aztreonam%'
        OR lower(drugname) LIKE '%axetil%' OR lower(drugname) LIKE '%bactocill%'
        OR lower(drugname) LIKE '%bactrim%' OR lower(drugname) LIKE '%bethkis%'
        OR lower(drugname) LIKE '%biaxin%' OR lower(drugname) LIKE '%bicillin l-a%'
```

```
        OR lower(drugname) LIKE '%cayston%' OR lower(drugname) LIKE '%cefazolin%'
        OR lower(drugname) LIKE '%cedax%' OR lower(drugname) LIKE '%cefoxitin%'
        OR lower(drugname) LIKE '%ceftazidime%' OR lower(drugname) LIKE '%cefaclor%'
        OR lower(drugname) LIKE '%cefadroxil%' OR lower(drugname) LIKE '%cefdinir%'
        OR lower(drugname) LIKE '%cefditoren%' OR lower(drugname) LIKE '%cefepime%'
        OR lower(drugname) LIKE '%cefotetan%' OR lower(drugname) LIKE '%cefotaxime%'
        OR lower(drugname) LIKE '%cefpodoxime%' OR lower(drugname) LIKE '%cefprozil%'
        OR lower(drugname) LIKE '%ceftibuten%' OR lower(drugname) LIKE '%ceftin%'
        OR lower(drugname) LIKE '%cefuroxime%' OR lower(drugname) LIKE '%cefuroxime%'
        OR lower(drugname) LIKE '%cephalexin%' OR lower(drugname) LIKE '%chloramphenicol%'
        OR lower(drugname) LIKE '%cipro%' OR lower(drugname) LIKE '%ciprofloxacin%'
        OR lower(drugname) LIKE '%claforan%' OR lower(drugname) LIKE '%clarithromycin%'
        OR lower(drugname) LIKE '%cleocin%' OR lower(drugname) LIKE '%clindamycin%'
        OR lower(drugname) LIKE '%cubicin%' OR lower(drugname) LIKE '%dicloxacillin%'
        OR lower(drugname) LIKE '%doryx%' OR lower(drugname) LIKE '%doxycycline%'
        OR lower(drugname) LIKE '%duricef%' OR lower(drugname) LIKE '%dynacin%'
        OR lower(drugname) LIKE '%ery-tab%' OR lower(drugname) LIKE '%eryped%'
        OR lower(drugname) LIKE '%eryc%' OR lower(drugname) LIKE '%erythrocin%'
        OR lower(drugname) LIKE '%erythromycin%' OR lower(drugname) LIKE '%factive%'
        OR lower(drugname) LIKE '%flagyl%' OR lower(drugname) LIKE '%fortaz%'
        OR lower(drugname) LIKE '%furadantin%' OR lower(drugname) LIKE '%garamycin%'
        OR lower(drugname) LIKE '%gentamicin%' OR lower(drugname) LIKE '%kanamycin%'
        OR lower(drugname) LIKE '%keflex%' OR lower(drugname) LIKE '%ketek%'
        OR lower(drugname) LIKE '%levaquin%' OR lower(drugname) LIKE '%levofloxacin%'
        OR lower(drugname) LIKE '%lincocin%' OR lower(drugname) LIKE '%macrobid%'
        OR lower(drugname) LIKE '%macrodantin%' OR lower(drugname) LIKE '%maxipime%'
        OR lower(drugname) LIKE '%mefoxin%' OR lower(drugname) LIKE '%metronidazole%'
        OR lower(drugname) LIKE '%minocin%' OR lower(drugname) LIKE '%minocycline%'
        OR lower(drugname) LIKE '%monodox%' OR lower(drugname) LIKE '%monurol%'
        OR lower(drugname) LIKE '%morgidox%' OR lower(drugname) LIKE '%moxatag%'
        OR lower(drugname) LIKE '%moxifloxacin%' OR lower(drugname) LIKE '%myrac%'
        OR lower(drugname) LIKE '%nafcillin sodium%' OR lower(drugname) LIKE '%nicazel doxy 30%'
        OR lower(drugname) LIKE '%nitrofurantoin%' OR lower(drugname) LIKE '%noroxin%'
        OR lower(drugname) LIKE '%ocudox%' OR lower(drugname) LIKE '%ofloxacin%'
        OR lower(drugname) LIKE '%omnicef%' OR lower(drugname) LIKE '%oracea%'
        OR lower(drugname) LIKE '%oraxyl%' OR lower(drugname) LIKE '%oxacillin%'
        OR lower(drugname) LIKE '%pc pen vk%' OR lower(drugname) LIKE '%pce dispertab%'
        OR lower(drugname) LIKE '%panixine%' OR lower(drugname) LIKE '%pediazole%'
        OR lower(drugname) LIKE '%penicillin%' OR lower(drugname) LIKE '%periostat%'
        OR lower(drugname) LIKE '%pfizerpen%' OR lower(drugname) LIKE '%piperacillin%'
        OR lower(drugname) LIKE '%tazobactam%' OR lower(drugname) LIKE '%primsol%'
        OR lower(drugname) LIKE '%proquin%' OR lower(drugname) LIKE '%raniclor%'
        OR lower(drugname) LIKE '%rifadin%' OR lower(drugname) LIKE '%rifampin%'
        OR lower(drugname) LIKE '%rocephin%' OR lower(drugname) LIKE '%smz-tmp%'
        OR lower(drugname) LIKE '%septra%' OR lower(drugname) LIKE '%septra ds%'
        OR lower(drugname) LIKE '%septra%' OR lower(drugname) LIKE '%solodyn%'
        OR lower(drugname) LIKE '%spectracef%' OR lower(drugname) LIKE '%streptomycin sulfate%'
        OR lower(drugname) LIKE '%sulfadiazine%' OR lower(drugname) LIKE '%sulfamethoxazole%'
        OR lower(drugname) LIKE '%trimethoprim%' OR lower(drugname) LIKE '%sulfatrim%'
        OR lower(drugname) LIKE '%sulfisoxazole%' OR lower(drugname) LIKE '%suprax%'
        OR lower(drugname) LIKE '%synercid%' OR lower(drugname) LIKE '%tazicef%'
        OR lower(drugname) LIKE '%tetracycline%' OR lower(drugname) LIKE '%timentin%'
        OR lower(drugname) LIKE '%tobi%'  OR lower(drugname) LIKE '%tobramycin%'
        OR lower(drugname) LIKE '%trimethoprim%' OR lower(drugname) LIKE '%unasyn%'
        OR lower(drugname) LIKE '%vancocin%' OR lower(drugname) LIKE '%vancomycin%'
        OR lower(drugname) LIKE '%vantin%' OR lower(drugname) LIKE '%vibativ%'
        OR lower(drugname) LIKE '%vibra-tabs%' OR lower(drugname) LIKE '%vibramycin%'
        OR lower(drugname) LIKE '%zinacef%' OR lower(drugname) LIKE '%zithromax%'
        OR lower(drugname) LIKE '%zmax%' OR lower(drugname) LIKE '%zosyn%'
        OR lower(drugname) LIKE '%zyvox%'
)

SELECT * FROM t_intakeoutput;
```

A.6 antibio_1234.sql

```
DROP TABLE IF EXISTS antibio_1234 CASCADE;
CREATE TABLE antibio_1234 AS

WITH

t1 AS
(
    SELECT
        patientunitstayid,
        antibio_time
    FROM
        antibio_1

    UNION

    SELECT
        patientunitstayid,
        antibio_time
    FROM
        antibio_2

    UNION
```

```
    SELECT
        patientunitstayid,
        antibio_time
    FROM
        antibio_3

    UNION

    SELECT
        patientunitstayid,
        antibio_time
    FROM
        antibio_4
)

SELECT * FROM t1;
```

A.7 bg_sepsis_subset.sql

```
DROP TABLE IF EXISTS bg_sepsis_subset CASCADE;
CREATE TABLE bg_sepsis_subset as -- get blood gas measures

WITH
t1 AS
(
    SELECT
        pivoted_bg.patientunitstayid,
        chartoffset AS observationoffset,
        fio2,
        pao2,
        baseexcess
    FROM
        pivoted_bg, sepsis_subset

    WHERE
         pivoted_bg.patientunitstayid = sepsis_subset.patientunitstayid
         AND chartoffset < 4321
    ORDER BY
        sepsis_subset.patientunitstayid, pivoted_bg.chartoffset
)

SELECT * FROM t1;
```

A.8 hospi_unique.sql

```
DROP TABLE IF EXISTS hospi_unique CASCADE;
CREATE TABLE hospi_unique AS

WITH
t1 AS
(
    SELECT
        patient.uniquepid
    FROM
        sepsis_subset,
        patient
    WHERE
        sepsis_subset.patientunitstayid = patient.patientunitstayid
        AND (
        sepsis_subset.wardid = 369
        OR sepsis_subset.wardid = 413
        OR sepsis_subset.wardid = 376
        OR sepsis_subset.wardid = 391
        OR sepsis_subset.wardid = 312
        OR sepsis_subset.wardid = 324
        OR sepsis_subset.wardid = 408
        OR sepsis_subset.wardid = 1029
        OR sepsis_subset.wardid = 1026
        OR sepsis_subset.wardid = 1032
        OR sepsis_subset.wardid = 1039
        OR sepsis_subset.wardid = 1027
        OR sepsis_subset.wardid = 1035)
    ORDER BY
        patient.uniquepid
),

t2 AS
(
    SELECT

        count(t1.uniquepid) AS n
    FROM
```

```
        t1
    GROUP BY
        t1.uniquepid
    ORDER BY
        n DESC

),

t3 AS
(
    SELECT
        t2.n,
        count(t2.n) AS nb
    FROM
        t2
    GROUP BY
        t2.n
    ORDER BY
        nb DESC

)

SELECT * FROM t1;
```

A.9 ICD_Codes_9_10.sql

```
DROP TABLE IF EXISTS ICD_Codes_9_10 CASCADE;
CREATE TABLE ICD_Codes_9_10 AS

WITH
t1 AS
(
    SELECT
        patientUnitStayID,
        ICD9Code,
        diagnosisOffset,
        diagnosisPriority
    FROM
        diagnosis
    WHERE ICD9Code LIKE '%995.92%' OR ICD9Code LIKE '%785.52%'
    OR ICD9Code LIKE '%R65.20%' OR ICD9Code LIKE '%R65.21%'

)

SELECT * FROM t1
```

A.10 ICU_unique.sql

```
DROP TABLE IF EXISTS ICU_unique CASCADE;
CREATE TABLE ICU_unique AS

WITH
t1 AS
(
    SELECT
        patient.uniquepid
    FROM
        sepsis_subset,
        patient
    WHERE
        sepsis_subset.patientunitstayid = patient.patientunitstayid
        AND (sepsis_subset.wardid = 369
        OR sepsis_subset.wardid = 413
        OR sepsis_subset.wardid = 347
        OR sepsis_subset.wardid = 337)
    ORDER BY
        patient.uniquepid
),

t2 AS
(
    SELECT
        count(t1.uniquepid) AS n
    FROM
        t1
    GROUP BY
        t1.uniquepid
    ORDER BY
        n DESC

),
```

```
t3 AS
(
    SELECT
        t2.n,
        count(t2.n) AS nb
    FROM
        t2
    GROUP BY
        t2.n
    ORDER BY
        nb DESC

)

SELECT * FROM t3;
```

A.11 input_fluid.sql

```
DROP TABLE IF EXISTS input_fluid CASCADE;
CREATE TABLE input_fluid AS

WITH
t1 AS
(
    SELECT distinct
        sepsis_subset.patientunitstayid,
        intakeoutputoffset,
        intaketotal,
        dialysistotal
    FROM
        intakeoutput,
        sepsis_subset
    WHERE
         intakeoutput.patientunitstayid  = sepsis_subset.patientunitstayid
    ORDER BY
        sepsis_subset.patientunitstayid, intakeoutputoffset
),

t2 AS
(
    SELECT
        t1.patientunitstayid,
        t1.intakeoutputoffset,
        CASE
            WHEN t1.dialysistotal > 0 THEN t1.intaketotal + t1.dialysistotal
            ELSE t1.intaketotal
        END AS intaketotal
    FROM
        t1
    ORDER BY
        t1.patientunitstayid, t1.intakeoutputoffset
),

t3 AS
(
    SELECT
        t2.patientunitstayid,
        t2.intakeoutputoffset,
        t2.intaketotal
    FROM
        t2
    WHERE
        t2.intakeoutputoffset < 4321
    ORDER BY
        t2.patientunitstayid, t2.intakeoutputoffset
)

SELECT * from t3;
```

A.12 input_output_fluid.sql

```
DROP TABLE IF EXISTS input_output_fluid CASCADE;
CREATE TABLE input_output_fluid AS

WITH
t1 AS
(
    SELECT distinct
        sepsis_subset.patientunitstayid,
        intakeoutputoffset,
        intaketotal,
        outputtotal,
```

```
        dialysistotal
    FROM
        intakeoutput,
        sepsis_subset
    WHERE
         intakeoutput.patientunitstayid  = sepsis_subset.patientunitstayid
    ORDER BY
        sepsis_subset.patientunitstayid, intakeoutputoffset
),

t2 AS
(
    SELECT
        t1.patientunitstayid,
        t1.intakeoutputoffset,
        CASE
            WHEN dialysistotal < 0 THEN outputtotal + dialysistotal
            ELSE outputtotal
        END AS outputtotal,
        CASE
            WHEN t1.dialysistotal > 0 THEN t1.intaketotal + t1.dialysistotal
            ELSE t1.intaketotal
        END AS intaketotal
    FROM
        t1

    ORDER BY
        t1.patientunitstayid, t1.intakeoutputoffset
),

t3 AS
(
    SELECT
        t2.patientunitstayid,
        t2.intakeoutputoffset AS observationoffset,
        t2.intaketotal,
        t2.outputtotal
    FROM
        t2
    WHERE
        t2.intakeoutputoffset < 4321
    ORDER BY
        t2.patientunitstayid, t2.intakeoutputoffset
)

SELECT * from t3;
```

A.13 lab_sepsis_subset.sql

```
DROP TABLE IF EXISTS lab_sepsis_subset CASCADE;
CREATE TABLE lab_sepsis_subset as

WITH
t1 AS
(
    SELECT
        pivoted_lab.patientunitstayid,
        chartoffset AS observationoffset,
        creatinine,
        lactate
    FROM
        pivoted_lab,
        sepsis_subset
    WHERE
        sepsis_subset.patientunitstayid = pivoted_lab.patientunitstayid
        AND chartoffset < 4321
    ORDER BY
        sepsis_subset.patientunitstayid, pivoted_lab.chartoffset
)

SELECT * FROM t1;
```

A.14 output_fluid.sql

```
DROP TABLE IF EXISTS output_fluid CASCADE;
CREATE TABLE output_fluid AS

WITH
t1 AS
(
    SELECT distinct
        sepsis_subset.patientunitstayid,
```

```
        intakeoutputoffset,
        outputtotal,
        dialysistotal

    FROM
        intakeoutput,
        sepsis_subset
    WHERE
         intakeoutput.patientunitstayid  = sepsis_subset.patientunitstayid
    ORDER BY
        sepsis_subset.patientunitstayid, intakeoutputoffset
),

t2 AS
(
    SELECT
        t1.patientunitstayid,
        t1.intakeoutputoffset,
        CASE
            WHEN dialysistotal < 0 THEN outputtotal + dialysistotal
            ELSE outputtotal
        END AS outputtotal
    FROM
        t1
    ORDER BY
        t1.patientunitstayid, t1.intakeoutputoffset
),

t3 AS
(
    SELECT
        t2.patientunitstayid,
        t2.intakeoutputoffset,
        t2.outputtotal
    FROM
        t2
    WHERE
        t2.intakeoutputoffset < 4321
    ORDER BY
        t2.patientunitstayid, t2.intakeoutputoffset
)

SELECT * from t3;
```

A.15 pivoted_bg.sql

```
DROP TABLE IF EXISTS pivoted_bg CASCADE;
CREATE TABLE pivoted_bg AS -- get blood gas measures
WITH vw0 AS
(
  SELECT
      patientunitstayid
    , labname
    , labresultoffset
    , labresultrevisedoffset
  FROM lab
  WHERE labname in
  (
        'paO2'
      , 'paCO2'
      , 'pH'
      , 'FiO2'
      , 'anion gap'
      , 'Base Deficit'
      , 'Base Excess'
      , 'PEEP'
  )
  GROUP BY patientunitstayid, labname, labresultoffset, labresultrevisedoffset
  HAVING count(distinct labresult)<=1
)
-- get the last lab to be revised
, vw1 AS
(
  SELECT
      lab.patientunitstayid
    , lab.labname
    , lab.labresultoffset
    , lab.labresultrevisedoffset
    , lab.labresult
    , ROW_NUMBER() OVER
        (
          PARTITION BY lab.patientunitstayid, lab.labname, lab.labresultoffset
          ORDER BY lab.labresultrevisedoffset DESC
        ) AS rn
  FROM lab
  INNER JOIN vw0
    ON  lab.patientunitstayid = vw0.patientunitstayid
    AND lab.labname = vw0.labname
    AND lab.labresultoffset = vw0.labresultoffset
```

```
    AND lab.labresultrevisedoffset = vw0.labresultrevisedoffset
  WHERE
      (lab.labname = 'paO2' and lab.labresult >= 15 and lab.labresult <= 720)
  OR (lab.labname = 'paCO2' and lab.labresult >= 5 and lab.labresult <= 250)
  OR (lab.labname = 'pH' and lab.labresult >= 6.5 and lab.labresult <= 8.5)
  OR (lab.labname = 'FiO2' and lab.labresult >= 0.2 and lab.labresult <= 1.0)
  OR (lab.labname = 'FiO2' and lab.labresult >= 20 and lab.labresult <= 100)
  OR (lab.labname = 'anion gap' and lab.labresult >= 0 and lab.labresult <= 300)
  OR (lab.labname = 'Base Deficit' and lab.labresult >= -100 and lab.labresult <= 100)
  OR (lab.labname = 'Base Excess' and lab.labresult >= -100 and lab.labresult <= 100)
  OR (lab.labname = 'PEEP' and lab.labresult >= 0 and lab.labresult <= 60)
)
SELECT
    patientunitstayid
  , labresultoffset AS chartoffset
  -- the aggregate (max()) only ever applies to 1 value due to the WHERE clause
  , MAX(case
        when labname != 'FiO2' then null
        when labresult >= 20 then labresult/100.0
      else labresult end) AS fio2
  , MAX(case when labname = 'paO2' then labresult else null end) AS pao2
  , MAX(case when labname = 'paCO2' then labresult else null end) AS paco2
  , MAX(case when labname = 'pH' then labresult else null end) AS pH
  , MAX(case when labname = 'anion gap' then labresult else null end) AS aniongap
  , MAX(case when labname = 'Base Deficit' then labresult else null end) AS basedeficit
  , MAX(case when labname = 'Base Excess' then labresult else null end) AS baseexcess
  , MAX(case when labname = 'PEEP' then labresult else null end) AS peep
FROM vw1
WHERE rn = 1
GROUP BY patientunitstayid, labresultoffset
ORDER BY patientunitstayid, labresultoffset;
```

A.16 pivoted_lab.sql

```
DROP TABLE IF EXISTS pivoted_lab CASCADE;
CREATE TABLE pivoted_lab AS -- remove duplicate labs if they exist at the same time
with vw0 AS
(
  SELECT
      patientunitstayid
    , labname
    , labresultoffset
    , labresultrevisedoffset
  FROM lab
  WHERE labname in
  (
      'albumin'
    , 'total bilirubin'
    , 'BUN'
    , 'calcium'
    , 'chloride'
    , 'creatinine'
    , 'bedside glucose', 'glucose'
    , 'bicarbonate' -- HCO3
    , 'Total CO2'
    , 'Hct'
    , 'Hgb'
    , 'PT - INR'
    , 'PTT'
    , 'lactate'
    , 'platelets x 1000'
    , 'potassium'
    , 'sodium'
    , 'WBC x 1000'
    , '-bands'
    -- Liver enzymes
    , 'ALT (SGPT)'
    , 'AST (SGOT)'
    , 'alkaline phos.'
  )
  GROUP BY patientunitstayid, labname, labresultoffset, labresultrevisedoffset
  HAVING count(distinct labresult)<=1
)
-- get the last lab to be revised
, vw1 AS
(
  SELECT
      lab.patientunitstayid
    , lab.labname
    , lab.labresultoffset
    , lab.labresultrevisedoffset
    , lab.labresult
    , ROW_NUMBER() OVER
        (
          PARTITION BY lab.patientunitstayid, lab.labname, lab.labresultoffset
          ORDER BY lab.labresultrevisedoffset DESC
        ) AS rn
  FROM lab
  INNER JOIN vw0
```

```
    ON  lab.patientunitstayid = vw0.patientunitstayid
    AND lab.labname = vw0.labname
    AND lab.labresultoffset = vw0.labresultoffset
    AND lab.labresultrevisedoffset = vw0.labresultrevisedoffset
  -- only valid lab values
  WHERE
       (lab.labname = 'albumin' AND lab.labresult >= 0.5 AND lab.labresult <= 6.5)
    OR (lab.labname = 'total bilirubin' AND lab.labresult >= 0.2 AND lab.labresult <= 70.175)
    OR (lab.labname = 'BUN' AND lab.labresult >= 1 AND lab.labresult <= 280)
    OR (lab.labname = 'calcium' AND lab.labresult > 0 AND lab.labresult <= 9999)
    OR (lab.labname = 'chloride' AND lab.labresult > 0 AND lab.labresult <= 9999)
    OR (lab.labname = 'creatinine' AND lab.labresult >= 0.1 AND lab.labresult <= 28.28)
    OR (lab.labname in ('bedside glucose', 'glucose') AND lab.labresult >= 25 AND lab.labresult <= 1500)
    OR (lab.labname = 'bicarbonate' AND lab.labresult >= 0 AND lab.labresult <= 9999)
    OR (lab.labname = 'Total CO2' AND lab.labresult >= 0 AND lab.labresult <= 9999)
    -- will convert hct unit to fraction later
    OR (lab.labname = 'Hct' AND lab.labresult >= 5 AND lab.labresult <= 75)
    OR (lab.labname = 'Hgb' AND lab.labresult >  0 AND lab.labresult <= 9999)
    OR (lab.labname = 'PT - INR' AND lab.labresult >= 0.5 AND lab.labresult <= 15)
    OR (lab.labname = 'lactate' AND lab.labresult >= 0.1 AND lab.labresult <= 30)
    OR (lab.labname = 'platelets x 1000' AND lab.labresult >  0 AND lab.labresult <= 9999)
    OR (lab.labname = 'potassium' AND lab.labresult >= 0.05 AND lab.labresult <= 12)
    OR (lab.labname = 'PTT' AND lab.labresult >  0 AND lab.labresult <= 500)
    OR (lab.labname = 'sodium' AND lab.labresult >= 90 AND lab.labresult <= 215)
    OR (lab.labname = 'WBC x 1000' AND lab.labresult > 0 AND lab.labresult <= 100)
    OR (lab.labname = '-bands' AND lab.labresult >= 0 AND lab.labresult <= 100)
    OR (lab.labname = 'ALT (SGPT)' AND lab.labresult > 0)
    OR (lab.labname = 'AST (SGOT)' AND lab.labresult > 0)
    OR (lab.labname = 'alkaline phos.' AND lab.labresult > 0)
)
SELECT
     patientunitstayid
   , labresultoffset AS chartoffset
   , MAX(CASE WHEN labname = 'albumin' THEN labresult else NULL END) AS albumin
   , MAX(CASE WHEN labname = 'total bilirubin' THEN labresult else NULL END) AS bilirubin
   , MAX(CASE WHEN labname = 'BUN' THEN labresult else NULL END) AS BUN
   , MAX(CASE WHEN labname = 'calcium' THEN labresult else NULL END) AS calcium
   , MAX(CASE WHEN labname = 'chloride' THEN labresult else NULL END) AS chloride
   , MAX(CASE WHEN labname = 'creatinine' THEN labresult else NULL END) AS creatinine
   , MAX(CASE WHEN labname in ('bedside glucose', 'glucose') THEN labresult else NULL END) AS glucose
   , MAX(CASE WHEN labname = 'bicarbonate' THEN labresult else NULL END) AS bicarbonate
   , MAX(CASE WHEN labname = 'Total CO2' THEN labresult else NULL END) AS TotalCO2
   , MAX(CASE WHEN labname = 'Hct' THEN labresult else NULL END) AS hematocrit
   , MAX(CASE WHEN labname = 'Hgb' THEN labresult else NULL END) AS hemoglobin
   , MAX(CASE WHEN labname = 'PT - INR' THEN labresult else NULL END) AS INR
   , MAX(CASE WHEN labname = 'lactate' THEN labresult else NULL END) AS lactate
   , MAX(CASE WHEN labname = 'platelets x 1000' THEN labresult else NULL END) AS platelets
   , MAX(CASE WHEN labname = 'potassium' THEN labresult else NULL END) AS potassium
   , MAX(CASE WHEN labname = 'PTT' THEN labresult else NULL END) AS ptt
   , MAX(CASE WHEN labname = 'sodium' THEN labresult else NULL END) AS sodium
   , MAX(CASE WHEN labname = 'WBC x 1000' THEN labresult else NULL END) AS wbc
   , MAX(CASE WHEN labname = '-bands' THEN labresult else NULL END) AS bands
   , MAX(CASE WHEN labname = 'ALT (SGPT)' THEN labresult else NULL END) AS alt
   , MAX(CASE WHEN labname = 'AST (SGOT)' THEN labresult else NULL END) AS ast
   , MAX(CASE WHEN labname = 'alkaline phos.' THEN labresult else NULL END) AS alp
FROM vw1
WHERE rn = 1
GROUP BY patientunitstayid, labresultoffset
order by patientunitstayid, labresultoffset;
```

A.17 quality2014_2015_vf.sql

```
DROP TABLE IF EXISTS quality2014_2015_vf CASCADE;
CREATE TABLE quality2014_2015_vf AS

WITH
t_infusion AS
(
    SELECT
        patientUnitStayID,
        infusionoffset AS vaso_time
    FROM
        infusionDrug
    WHERE
        lower(drugname) like '%epinephrine%'
        or lower(drugname) like 'epi (mcg/min)'
        or lower(drugname) like '%norepinephrine%'
        or lower(drugname) like '%levoph%'
        or lower(drugname) like '%phenylephrine%'
        or lower(drugname) like '%neo-synephrine%'
        or lower(drugname) like '%neosynephrine%'
        or lower(drugname) like '%neosynsprine%'
        or lower(drugname) like '%synephrine%'
        or lower(drugname) like '%vasopressin%'
    ORDER BY patientUnitStayID
),

t2_2014v AS
(
```

```
    SELECT
        t_infusion.patientUnitStayID,
        hospitalDischargeYear,
        hospitalid,
        t_infusion.vaso_time,
        round(unitdischargeoffset,3) AS icu_los_min
    FROM
        t_infusion,
        patient
    WHERE
        t_infusion.patientUnitStayID = patient.patientUnitStayID
        AND hospitalDischargeYear = 2014
    ORDER BY t_infusion.patientUnitStayID, t_infusion.vaso_time
 ),

t3_2014v AS
 (
    SELECT
        t2_2014v.hospitalDischargeYear,
        t2_2014v.patientUnitStayID,
        t2_2014v.hospitalid,
        count(t2_2014v.vaso_time) AS nb,
        t2_2014v.icu_los_min
    FROM
        t2_2014v
    GROUP BY
        t2_2014v.hospitalDischargeYear, t2_2014v.patientUnitStayID, t2_2014v.hospitalid, t2_2014v.icu_los_min
    ORDER BY t2_2014v.patientUnitStayID
),

t4_2014v AS
 (
    SELECT
        t3_2014v.hospitalDischargeYear,
        t3_2014v.patientUnitStayID,
        t3_2014v.hospitalid,
        t3_2014v.nb,
        t3_2014v.icu_los_min,
        CASE
            WHEN t3_2014v.icu_los_min = 0 THEN 0
            ELSE (t3_2014v.nb/t3_2014v.icu_los_min)*1440
        END AS nb_day
    FROM
        t3_2014v
    GROUP BY
        t3_2014v.hospitalDischargeYear, t3_2014v.patientUnitStayID, t3_2014v.hospitalid, t3_2014v.nb, t3_2014v.icu_lo
    ORDER BY t3_2014v.patientUnitStayID
),

t5_2014v AS
 (
    SELECT
        t4_2014v.hospitalDischargeYear,
        t4_2014v.hospitalid,
        AVG(t4_2014v.nb_day) AS mean_dose_day
    FROM
        t4_2014v
    GROUP BY
        t4_2014v.hospitalDischargeYear, t4_2014v.hospitalid
    ORDER BY t4_2014v.hospitalid
),

t6_2014v AS
 (
    SELECT
        t5_2014v.hospitalDischargeYear,
        t5_2014v.hospitalid,
        t5_2014v.mean_dose_day,
        CASE WHEN t5_2014v.mean_dose_day >= 6 THEN 1 ELSE 0 END AS good_quality
    FROM
        t5_2014v
    ORDER BY
        t5_2014v.hospitalid
),

t7_2014v AS
 (
    SELECT
        t6_2014v.hospitalDischargeYear,
        t6_2014v.hospitalid
    FROM
        t6_2014v
    WHERE
        good_quality = 1
    ORDER BY
        t6_2014v.hospitalid
),

t1_2014f AS -- Every time patient has an intake-Output observation
(
    SELECT
        distinct patientUnitStayID,
        intakeOutputOffset
    FROM
        intakeoutput
```

```
    ORDER BY patientUnitStayID, intakeOutputOffset
),

t2_2014f AS
(
    SELECT
        t1_2014f.patientUnitStayID,
        hospitalDischargeYear,
        hospitalid,
        t1_2014f.intakeOutputOffset,
        round(unitdischargeoffset,3) AS icu_los_min
    FROM
        t1_2014f,
        patient
    WHERE
        t1_2014f.patientUnitStayID = patient.patientUnitStayID
        AND hospitalDischargeYear = 2014
    ORDER BY t1_2014f.patientUnitStayID, t1_2014f.intakeOutputOffset
 ),

t3_2014f AS
 (
    SELECT
        t2_2014f.hospitalDischargeYear,
        t2_2014f.patientUnitStayID,
        t2_2014f.hospitalid,
        count(t2_2014f.intakeOutputOffset) AS nb,
        t2_2014f.icu_los_min
    FROM
        t2_2014f
    GROUP BY
        t2_2014f.hospitalDischargeYear, t2_2014f.patientUnitStayID, t2_2014f.hospitalid, t2_2014f.icu_los_min
    ORDER BY t2_2014f.patientUnitStayID
),

t4_2014f AS
 (
    SELECT
        t3_2014f.hospitalDischargeYear,
        t3_2014f.patientUnitStayID,
        t3_2014f.hospitalid,
        t3_2014f.nb,
        t3_2014f.icu_los_min,
        CASE WHEN t3_2014f.icu_los_min = 0 THEN 0 ELSE (t3_2014f.nb/t3_2014f.icu_los_min)*1440 END AS nb_day
    FROM
        t3_2014f
    GROUP BY
        t3_2014f.hospitalDischargeYear, t3_2014f.patientUnitStayID, t3_2014f.hospitalid, t3_2014f.nb, t3_2014f.icu_lo
    ORDER BY t3_2014f.patientUnitStayID
),

t5_2014f AS
 (
    SELECT
        t4_2014f.hospitalDischargeYear,
        t4_2014f.hospitalid,
        AVG(t4_2014f.nb_day) AS mean_observ_day
    FROM
        t4_2014f
    GROUP BY
        t4_2014f.hospitalDischargeYear, t4_2014f.hospitalid
    ORDER BY t4_2014f.hospitalid
),

t6_2014f AS
 (
    SELECT
        t5_2014f.hospitalDischargeYear,
        t5_2014f.hospitalid,
        t5_2014f.mean_observ_day,
        CASE WHEN t5_2014f.mean_observ_day >= 6 THEN 1 ELSE 0 END AS good_quality
    FROM
        t5_2014f
    ORDER BY
        t5_2014f.hospitalid
),

t7_2014f AS
(
    SELECT
        t6_2014f.hospitalDischargeYear,
        t6_2014f.hospitalid
    FROM
        t6_2014f
    WHERE
        t6_2014f.good_quality = 1
    ORDER BY
        t6_2014f.hospitalid
),

t_2014_vf AS
(
    SELECT
        t7_2014v.hospitalDischargeYear,
        t7_2014v.hospitalid
```

```
    FROM
        t7_2014v
    INTERSECT
    SELECT
        t7_2014f.hospitalDischargeYear,
        t7_2014f.hospitalid
    FROM
        t7_2014f
),

t2_2015v AS
(
    SELECT
        t_infusion.patientUnitStayID,
        hospitalDischargeYear,
        hospitalid,
        t_infusion.vaso_time,
        round(unitdischargeoffset,3) AS icu_los_min
    FROM
        t_infusion,
        patient
    WHERE
        t_infusion.patientUnitStayID = patient.patientUnitStayID
        AND hospitalDischargeYear = 2015
    ORDER BY t_infusion.patientUnitStayID, t_infusion.vaso_time
 ),

t3_2015v AS
 (
    SELECT
        t2_2015v.hospitalDischargeYear,
        t2_2015v.patientUnitStayID,
        t2_2015v.hospitalid,
        count(t2_2015v.vaso_time) AS nb,
        t2_2015v.icu_los_min
    FROM
        t2_2015v
    GROUP BY
        t2_2015v.hospitalDischargeYear, t2_2015v.patientUnitStayID, t2_2015v.hospitalid, t2_2015v.icu_los_min
    ORDER BY t2_2015v.patientUnitStayID
),

t4_2015v AS
 (
    SELECT
        t3_2015v.hospitalDischargeYear,
        t3_2015v.patientUnitStayID,
        t3_2015v.hospitalid,
        t3_2015v.nb,
        t3_2015v.icu_los_min,
        CASE
            WHEN t3_2015v.icu_los_min = 0 THEN 0
            ELSE (t3_2015v.nb/t3_2015v.icu_los_min)*1440
        END AS nb_day
    FROM
        t3_2015v
    GROUP BY
        t3_2015v.hospitalDischargeYear, t3_2015v.patientUnitStayID, t3_2015v.hospitalid, t3_2015v.nb, t3_2015v.icu_lo
    ORDER BY t3_2015v.patientUnitStayID
),

t5_2015v AS
 (
    SELECT
        t4_2015v.hospitalDischargeYear,
        t4_2015v.hospitalid,
        AVG(t4_2015v.nb_day) AS mean_dose_day
    FROM
        t4_2015v
    GROUP BY
        t4_2015v.hospitalDischargeYear, t4_2015v.hospitalid
    ORDER BY t4_2015v.hospitalid
),

t6_2015v AS
 (
    SELECT
        t5_2015v.hospitalDischargeYear,
        t5_2015v.hospitalid,
        t5_2015v.mean_dose_day,
        CASE WHEN t5_2015v.mean_dose_day >= 6 THEN 1 ELSE 0 END AS good_quality
    FROM
        t5_2015v
    ORDER BY
        t5_2015v.hospitalid
),

t7_2015v AS
 (
    SELECT
        t6_2015v.hospitalDischargeYear,
        t6_2015v.hospitalid
    FROM
        t6_2015v
    WHERE
```

```
        good_quality = 1
    ORDER BY
        t6_2015v.hospitalid
),

t1_2015f AS -- Every time patient has an intake-Output observation
(
    SELECT
        distinct patientUnitStayID,
        intakeOutputOffset
    FROM
        intakeoutput
    ORDER BY patientUnitStayID, intakeOutputOffset
),

t2_2015f AS
(
    SELECT
        t1_2015f.patientUnitStayID,
        hospitalDischargeYear,
        hospitalid,
        t1_2015f.intakeOutputOffset,
        round(unitdischargeoffset,3) AS icu_los_min
    FROM
        t1_2015f,
        patient
    WHERE
        t1_2015f.patientUnitStayID = patient.patientUnitStayID
        AND hospitalDischargeYear = 2015
    ORDER BY t1_2015f.patientUnitStayID, t1_2015f.intakeOutputOffset
 ),

t3_2015f AS
 (
    SELECT
        t2_2015f.hospitalDischargeYear,
        t2_2015f.patientUnitStayID,
        t2_2015f.hospitalid,
        count(t2_2015f.intakeOutputOffset) AS nb,
        t2_2015f.icu_los_min
    FROM
        t2_2015f
    GROUP BY
        t2_2015f.hospitalDischargeYear, t2_2015f.patientUnitStayID, t2_2015f.hospitalid, t2_2015f.icu_los_min
    ORDER BY t2_2015f.patientUnitStayID
),

t4_2015f AS
 (
    SELECT
        t3_2015f.hospitalDischargeYear,
        t3_2015f.patientUnitStayID,
        t3_2015f.hospitalid,
        t3_2015f.nb,
        t3_2015f.icu_los_min,
        CASE
            WHEN t3_2015f.icu_los_min = 0 THEN 0
            ELSE (t3_2015f.nb/t3_2015f.icu_los_min)*1440
        END AS nb_day
    FROM
        t3_2015f
    GROUP BY
        t3_2015f.hospitalDischargeYear, t3_2015f.patientUnitStayID, t3_2015f.hospitalid, t3_2015f.nb, t3_2015f.icu_lo
    ORDER BY t3_2015f.patientUnitStayID
),

t5_2015f AS
 (
    SELECT
        t4_2015f.hospitalDischargeYear,
        t4_2015f.hospitalid,
        AVG(t4_2015f.nb_day) AS mean_observ_day
    FROM
        t4_2015f
    GROUP BY
        t4_2015f.hospitalDischargeYear, t4_2015f.hospitalid
    ORDER BY t4_2015f.hospitalid
),

t6_2015f AS
 (
    SELECT
        t5_2015f.hospitalDischargeYear,
        t5_2015f.hospitalid,
        t5_2015f.mean_observ_day,
        CASE WHEN t5_2015f.mean_observ_day >= 6 THEN 1 ELSE 0 END AS good_quality
    FROM
        t5_2015f
    ORDER BY
        t5_2015f.hospitalid
),

t7_2015f AS
(
    SELECT
```

```
        t6_2015f.hospitalDischargeYear,
        t6_2015f.hospitalid
    FROM
        t6_2015f
    WHERE
        t6_2015f.good_quality = 1
    ORDER BY
        t6_2015f.hospitalid
),

t_2015_vf AS
(
    SELECT
        t7_2015v.hospitalDischargeYear,
        t7_2015v.hospitalid
    FROM
        t7_2015v
    INTERSECT
    SELECT
        t7_2015f.hospitalDischargeYear,
        t7_2015f.hospitalid
    FROM
        t7_2015f
),

t_2014_2015_vf AS
(
    SELECT
        t_2014_vf.hospitalDischargeYear,
        t_2014_vf.hospitalid
    FROM
        t_2014_vf
    UNION
    SELECT
        t_2015_vf.hospitalDischargeYear,
        t_2015_vf.hospitalid
    FROM
        t_2015_vf
)

SELECT * FROM t_2014_2015_vf;
```

A.18 sepsis_distribution_hospital.sql

```
DROP TABLE IF EXISTS sepsis_distribution_hospital CASCADE;
CREATE TABLE sepsis_distribution_hospital AS

WITH
t1 AS
(
    SELECT
        hospitalid,
        count(hospitalid) AS nb_sepsis_patients_hospital
    FROM
        sepsis_total
    GROUP BY
        hospitalid
)

SELECT * FROM t1
```

A.19 sepsis_distribution_icu.sql

```
DROP TABLE IF EXISTS sepsis_distribution_icu CASCADE;
CREATE TABLE sepsis_distribution_icu AS

WITH
t1 AS
(
    SELECT
        wardid,
        count(wardid) AS nb_sepsis_patients_wardid
    FROM
        sepsis_total
    GROUP BY
        wardid

)

SELECT * FROM t1
```

A.20 sepsis_largest_hospital.sql

```
DROP TABLE IF EXISTS sepsis_largest_hospital CASCADE;
CREATE TABLE sepsis_largest_hospital AS

WITH
t0 AS -- select the 5 biggest hospitals
(
    SELECT
        hospitalid,
        nb_sepsis_patients_hospital
    FROM
        sepsis_total
    GROUP BY
        hospitalid,
        nb_sepsis_patients_hospital
    ORDER BY
        nb_sepsis_patients_hospital
    DESC
    LIMIT 5
),

t1 AS -- retrieve the patients for those 5 hospitals
(
    SELECT
        sepsis_total.patientUnitStayID,
        sepsis_total.sofa_score,
        sepsis_total.sepsis_type,
        sepsis_total.hospitalid,
        sepsis_total.hospitalDischargeYear,
        sepsis_total.wardid,
        sepsis_total.nb_sepsis_patients_hospital
    FROM
        sepsis_total,
        t0
    WHERE
        sepsis_total.hospitalid = t0.hospitalid
    ORDER BY
        sepsis_total.nb_sepsis_patients_hospital
    DESC

),

t2 AS
(
    SELECT
        t1.wardid,
        unitType,
        t1.nb_sepsis_patients_hospital,
        t1.patientUnitStayID,
        t1.sofa_score,
        t1.sepsis_type,
        t1.hospitalid,
        t1.hospitalDischargeYear,
        hospital.numbedscategory,
        hospital.teachingstatus,
        hospital.region,
        CASE
            WHEN age LIKE '%>%' THEN '90'
            ELSE age
        END AS age, -- age > 89 years are arbitrarily set to 90 years
        CASE
            WHEN gender = 'Female' THEN 'female'
            WHEN gender = 'Male' THEN 'male'
            ELSE 'unknown'
        END AS gender,
        CASE
            WHEN ethnicity = 'African American' THEN 'African_American'
            WHEN ethnicity = 'Asian' THEN 'Asian'
            WHEN ethnicity = 'Caucasian' THEN 'Caucasian'
            WHEN ethnicity = 'Hispanic' THEN 'Hispanic'
            WHEN ethnicity = 'Native American' THEN 'Native_American'
            ELSE 'Ethnicity_Other_Unknown'
        END AS ethnicity,
        CASE
            WHEN admissionWeight > 0 THEN admissionWeight
            ELSE NULL
        END AS admissionWeight, -- remove weight = 0
        CASE
            WHEN admissionHeight = 0 THEN NULL -- remove height = 0
            WHEN admissionHeight > 0 AND admissionHeight < 3 THEN admissionHeight * 100 -- convert (suspected) centim
            ELSE admissionHeight
        END AS admissionHeight,
        CASE
            WHEN unitAdmitSource = 'Emergency Department' OR unitAdmitSource = 'Chest Pain Center'
            OR unitAdmitSource = 'Observation' THEN 'adm_ED'
            WHEN unitAdmitSource = 'Direct Admit' THEN 'adm_direct'
            WHEN unitAdmitSource = 'Floor' OR unitAdmitSource = 'Acute Care/Floor' THEN 'adm_floor'
            WHEN unitAdmitSource = 'Operating Room' OR unitAdmitSource = 'Recovery Room'
            OR unitAdmitSource = 'PACU' THEN 'adm_OR_Recovery'
            WHEN unitAdmitSource = 'ICU to SDU' OR unitAdmitSource = 'ICU'
            OR unitAdmitSource = 'Other ICU' THEN 'adm_ICU'
            WHEN unitAdmitSource = 'Step-Down Unit (SDU)' THEN 'adm_SDU'
```

```
            WHEN unitAdmitSource = 'Other Hospital' THEN 'adm_other_hosp'
            WHEN unitAdmitSource = 'Other' OR unitAdmitSource IS NULL THEN 'adm_unknown'
            ELSE 'adm_unknown'
        END AS unitAdmitSource,
        ROUND(unitdischargeoffset/60) AS icu_los_hours,
        unitDischargeStatus,
        hospitaldischargestatus
    FROM
        t1,
        patient,
        Hospital
    WHERE
        t1.patientUnitStayID = patient.patientUnitStayID
        AND patient.hospitalID = hospital.hospitalID
),

t_infusion AS -- ICU stays with vasopressors
(
    SELECT
        patientUnitStayID
    FROM
        infusionDrug
    WHERE
        lower(drugname) LIKE '%epinephrine%'
        OR  lower(drugname) LIKE 'epi (mcg/min)'
        OR  lower(drugname) LIKE '%norepinephrine%'
        OR  lower(drugname) LIKE '%levoph%'
        OR  lower(drugname) LIKE '%phenylephrine%'
        OR  lower(drugname) LIKE '%synephrine%'
        OR  lower(drugname) LIKE '%dopamine%'
        OR  lower(drugname) LIKE '%vasopressin%'
    GROUP BY patientUnitStayID
),

t3 AS
(
    SELECT
        t2.patientUnitStayID,
        1 AS vaso
    FROM
        t2, t_infusion
    WHERE
        t2.patientUnitStayID = t_infusion.patientUnitStayID
),

t4 AS
(
    SELECT
        t2.wardid,
        t2.unitType,
        t2.nb_sepsis_patients_hospital,
        t2.patientUnitStayID,
        t2.sofa_score,
        t2.sepsis_type,
        t2.hospitalid,
        t2.hospitalDischargeYear,
        t2.numbedscategory,
        t2.teachingstatus,
        t2.region,
        t2.age, -- age > 89 years are arbitrarily set to 90 years
        t2.gender,
        t2.ethnicity,
        t2.admissionWeight, -- remove weight = 0
        t2.admissionHeight,
        t2.unitAdmitSource,
        t2.icu_los_hours,
        t2.unitDischargeStatus,
        t2.hospitaldischargestatus,
        t3.vaso
    FROM
        t2 LEFT JOIN t3
        ON t2.patientUnitStayID = t3.patientUnitStayID
),

t5 AS
(
    SELECT
        t4.wardid,
        t4.unitType,
        t4.nb_sepsis_patients_hospital,
        t4.patientUnitStayID,
        t4.sofa_score,
        t4.sepsis_type,
        t4.hospitalid,
        t4.hospitalDischargeYear,
        t4.numbedscategory,
        t4.teachingstatus,
        t4.region,
        t4.age, -- age > 89 years are arbitrarily set to 90 years
        t4.gender,
        t4.ethnicity,
        t4.admissionWeight, -- remove weight = 0
        t4.admissionHeight,
        t4.unitAdmitSource,
        t4.icu_los_hours,
```

```
        t4.unitDischargeStatus,
        t4.hospitaldischargestatus,
        CASE WHEN t4.vaso = 1 THEN 1 ELSE 0 END AS vasopressor
    FROM
        t4
)

SELECT * FROM t5;
```

A.21 sepsis_largest_icu.sql

```
DROP TABLE IF EXISTS sepsis_largest_icu CASCADE;
CREATE TABLE sepsis_largest_icu AS

WITH
t0 AS -- select the 5 biggest ICU
(
    SELECT
        wardid,
        nb_sepsis_patients_wardid
    FROM
        sepsis_total
    GROUP BY
        wardid,
        nb_sepsis_patients_wardid
    ORDER BY
        nb_sepsis_patients_wardid
    DESC
    LIMIT 5
),

t1 AS -- retrieve the patients for those 5 ICUs
(
    SELECT
        sepsis_total.patientUnitStayID,
        sepsis_total.sofa_score,
        sepsis_total.sepsis_type,
        sepsis_total.hospitalid,
        sepsis_total.hospitalDischargeYear,
        sepsis_total.wardid,
        sepsis_total.nb_sepsis_patients_wardid
    FROM
        sepsis_total,
        t0
    WHERE
        sepsis_total.wardid = t0.wardid
    ORDER BY
        sepsis_total.nb_sepsis_patients_wardid
    DESC

),

t2 AS
(
    SELECT
        t1.wardid,
        unitType,
        t1.nb_sepsis_patients_wardid,
        t1.patientUnitStayID,
        t1.sofa_score,
        t1.sepsis_type,
        t1.hospitalid,
        t1.hospitalDischargeYear,
        hospital.numbedscategory,
        hospital.teachingstatus,
        hospital.region,
        CASE
            WHEN age LIKE '%>%' THEN '90'
            ELSE age
        END AS age, -- age > 89 years are arbitrarily set to 90 years
        CASE
            WHEN gender = 'Female' THEN 'female'
            WHEN gender = 'Male' THEN 'male'
            ELSE 'unknown'
        END AS gender,
        CASE
            WHEN ethnicity = 'African American' THEN 'African_American'
            WHEN ethnicity = 'Asian' THEN 'Asian'
            WHEN ethnicity = 'Caucasian' THEN 'Caucasian'
            WHEN ethnicity = 'Hispanic' THEN 'Hispanic'
            WHEN ethnicity = 'Native American' THEN 'Native_American'
            ELSE 'Ethnicity_Other_Unknown'
        END AS ethnicity,
        CASE
            WHEN admissionWeight > 0 THEN admissionWeight
            ELSE NULL
        END AS admissionWeight, -- remove weight = 0
        CASE
            WHEN admissionHeight = 0 THEN NULL -- remove height = 0
```

```
            -- convert (suspected) centimeter to meter
            WHEN admissionHeight > 0 AND admissionHeight < 3 THEN admissionHeight * 100
            ELSE admissionHeight
        END AS admissionHeight,
        CASE
            WHEN unitAdmitSource = 'Emergency Department' OR unitAdmitSource = 'Chest Pain Center'
            OR unitAdmitSource = 'Observation' THEN 'adm_ED'
            WHEN unitAdmitSource = 'Direct Admit' THEN 'adm_direct'
            WHEN unitAdmitSource = 'Floor' OR unitAdmitSource = 'Acute Care/Floor' THEN 'adm_floor'
            WHEN unitAdmitSource = 'Operating Room' OR unitAdmitSource = 'Recovery Room'
            OR unitAdmitSource = 'PACU' THEN 'adm_OR_Recovery'
            WHEN unitAdmitSource = 'ICU to SDU' OR unitAdmitSource = 'ICU'
            OR unitAdmitSource = 'Other ICU' THEN 'adm_ICU'
            WHEN unitAdmitSource = 'Step-Down Unit (SDU)' THEN 'adm_SDU'
            WHEN unitAdmitSource = 'Other Hospital' THEN 'adm_other_hosp'
            WHEN unitAdmitSource = 'Other' OR unitAdmitSource IS NULL THEN 'adm_unknown'
            ELSE 'adm_unknown'
        END AS unitAdmitSource,
        ROUND(unitdischargeoffset/60) AS icu_los_hours,
        unitDischargeStatus,
        hospitaldischargestatus
    FROM
        t1,
        patient,
        Hospital
    WHERE
        t1.patientUnitStayID = patient.patientUnitStayID
        AND patient.hospitalID = hospital.hospitalID
),

t_infusion AS -- ICU stays with vasopressors
(
    SELECT
        patientUnitStayID
    FROM
        infusionDrug
    WHERE
        lower(drugname) LIKE '%epinephrine%'
        OR lower(drugname) LIKE 'epi (mcg/min)'
        OR lower(drugname) LIKE '%norepinephrine%'
        OR lower(drugname) LIKE '%levoph%'
        OR lower(drugname) LIKE '%phenylephrine%'
        OR lower(drugname) LIKE '%synephrine%'
        OR lower(drugname) LIKE '%dopamine%'
        OR lower(drugname) LIKE '%vasopressin%'
    GROUP BY patientUnitStayID
),

t3 AS
(
    SELECT
        t2.patientUnitStayID,
        1 AS vaso
    FROM
        t2, t_infusion
    WHERE
        t2.patientUnitStayID = t_infusion.patientUnitStayID
),

t4 AS
(
    SELECT
        t2.wardid,
        t2.unitType,
        t2.nb_sepsis_patients_wardid,
        t2.patientUnitStayID,
        t2.sofa_score,
        t2.sepsis_type,
        t2.hospitalid,
        t2.hospitalDischargeYear,
        t2.numbedscategory,
        t2.teachingstatus,
        t2.region,
        t2.age, -- age > 89 years are arbitrarily set to 90 years
        t2.gender,
        t2.ethnicity,
        t2.admissionWeight, -- remove weight = 0
        t2.admissionHeight,
        t2.unitAdmitSource,
        t2.icu_los_hours,
        t2.unitDischargeStatus,
        t2.hospitaldischargestatus,
        t3.vaso
    FROM
        t2 LEFT JOIN t3
        ON t2.patientUnitStayID = t3.patientUnitStayID
),

t5 AS
(
    SELECT
        t4.wardid,
        t4.unitType,
        t4.nb_sepsis_patients_wardid,
        t4.patientUnitStayID,
```

```
        t4.sofa_score,
        t4.sepsis_type,
        t4.hospitalid,
        t4.hospitalDischargeYear,
        t4.numbedscategory,
        t4.teachingstatus,
        t4.region,
        t4.age, -- age > 89 years are arbitrarily set to 90 years
        t4.gender,
        t4.ethnicity,
        t4.admissionWeight, -- remove weight = 0
        t4.admissionHeight,
        t4.unitAdmitSource,
        t4.icu_los_hours,
        t4.unitDischargeStatus,
        t4.hospitaldischargestatus,
        CASE
            WHEN t4.vaso = 1 THEN 1
            ELSE 0
        END AS vasopressor
    FROM
        t4
)

SELECT * FROM t5;
```

A.22 sepsis_subset.sql

```
DROP TABLE IF EXISTS sepsis_subset CASCADE;
CREATE TABLE sepsis_subset AS

-- 3174 ICU stays in sepsis_total
-- The five biggest ICUs: 369 413 347 601 337
-- The five biggest Hospitals: 420 167 176 157 243
    -- In hospital 420, ICUs: 1029 1026 1032 1039 1027 1035
    -- In hospital 167, ICUs: 413 324 408
    -- In hospital 176, ICUs: 376 391 312
    -- In hospital 157, ICU: 369
    -- In hospital 243, ICUs: 601 607 609 594
-- ICUs with bad vasopressor data completion (concentration...): 601 607 609 594
-- At the end we select a subset of sepsis_total with ICUs
-- where vasopressor drugrates are available:
-- The "four" biggest ICUs: 369 413 347 337
-- The "four" biggest Hospitals: 420 167 176 157
    -- In hospital 420, ICUs: 1029 1026 1032 1039 1027 1035
    -- In hospital 167, ICUs: 413 324 408
    -- In hospital 176, ICUs: 376 391 312
    -- In hospital 157, ICU: 369
------ ICUs retained for analysis:
-- 369 413 347 337 376 391 312 324 408 1029 1026 1032 1039 1027 1035
-- 1088 ICU stays in sepsis_subset

WITH
t1 AS
(
    SELECT
        patientUnitStayID,
        sofa_score,
        sepsis_type,
        hospitalid,
        hospitalDischargeYear,
        wardid,
        unitDischargeOffset
    FROM
        sepsis_total
    WHERE
        wardid = 369
        OR wardid = 413 OR wardid = 347 OR wardid = 337
        OR wardid = 376 OR wardid = 391 OR wardid = 312
        OR wardid = 324 OR wardid = 408 OR wardid = 1029
        OR wardid = 1026 OR wardid = 1032 OR wardid = 1039
        OR wardid = 1027 OR wardid = 1035
),

t2 AS
(
    SELECT
        t1.patientUnitStayID,
        t1.sofa_score,
        t1.sepsis_type,
        t1.hospitalid,
        t1.hospitalDischargeYear,
        t1.wardid,
        t1.unitDischargeOffset
    FROM
        t1
)

SELECT * FROM t2;
```

A.23 sepsis_subset_vital_aperiodic.sql

```
DROP TABLE IF EXISTS sepsis_subset_vital_aperiodic CASCADE;
CREATE TABLE sepsis_subset_vital_aperiodic AS

WITH
t1 AS
(
    SELECT
        vitalaperiodic.patientunitstayid,
        vitalaperiodic.observationoffset,
        vitalaperiodic.noninvasivesystolic,
        vitalaperiodic.noninvasivemean,
        vitalaperiodic.noninvasivediastolic
    FROM
        sepsis_subset, vitalaperiodic
    WHERE
        sepsis_subset.patientunitstayid = vitalaperiodic.patientunitstayid
        AND vitalaperiodic.observationoffset < 4321
    ORDER BY
        vitalaperiodic.patientunitstayid, vitalaperiodic.observationoffset
)

SELECT * FROM t1;
```

A.24 sepsis_subset_vital_periodic.sql

```
DROP TABLE IF EXISTS sepsis_subset_vital_periodic CASCADE;
CREATE TABLE sepsis_subset_vital_periodic AS

WITH
t1 AS
(
    SELECT
        vitalperiodic.patientunitstayid,
        vitalperiodic.observationoffset,
        vitalperiodic.systemicsystolic,
        vitalperiodic.systemicmean,
        vitalperiodic.systemicdiastolic,
        vitalperiodic.heartrate,
        vitalperiodic.sao2
    FROM
        sepsis_subset, vitalperiodic
    WHERE
        sepsis_subset.patientunitstayid = vitalperiodic.patientunitstayid
        AND vitalperiodic.observationoffset < 4321
    ORDER BY
        vitalperiodic.patientunitstayid, vitalperiodic.observationoffset
)

SELECT * FROM t1;
```

A.25 sepsis_total.sql

```
DROP TABLE IF EXISTS sepsis_total CASCADE;
CREATE TABLE sepsis_total AS

WITH
-------------------------------------------------------------------------------
-- At least one ICD Code 9 or 10 for sepsis or septic shock during ICU stay
-------------------------------------------------------------------------------
t1 AS -- 13936 ICU stays
(
    SELECT
        distinct patientUnitStayID

    FROM
        ICD_Codes_9_10
),
-------------------------------------------------------------------------------
-- SOFA score > 1 first day of ICU stay
-------------------------------------------------------------------------------
t2 AS -- 140 716 ICU stays
(
    SELECT
        patientUnitStayID,
        sofatotal AS sofa_score
    FROM
        sofa
    WHERE
```

```
        sofatotal > 1
),
-------------------------------------------------------------------------------
-- ICU stays with antibiotic in [day-1 ; day+1]
-------------------------------------------------------------------------------
t3 AS -- 63 862 ICU stays
(
    SELECT
        distinct patientUnitStayID
    FROM
        antibio_1234
    WHERE
        antibio_time between -1440 and 1440
),
-------------------------------------------------------------------------------
-- ICU stays with ICD Codes, SOFA, antibiotics and admission diagnosis = sepsis
-------------------------------------------------------------------------------
t4 AS -- 6313 ICU stays
(
    SELECT
        t1.patientUnitStayID,
        sofa_score,
        admission_diagnosis.sepsis_type
    FROM
        t1
        INNER JOIN t2
        ON t1.patientUnitStayID = t2.patientUnitStayID
        INNER JOIN t3
        ON t1.patientUnitStayID = t3.patientUnitStayID
        INNER JOIN admission_diagnosis
        ON t1.patientUnitStayID = admission_diagnosis.patientUnitStayID
),
-------------------------------------------------------------------------------
-- ICU stays with ICD Codes, SOFA, antibiotics and admission diagnosis = sepsis
-- and in hospitals with good data quality
-------------------------------------------------------------------------------
t5 AS
(
    SELECT
        t4.patientUnitStayID,
        t4.sofa_score,
        t4.sepsis_type,
        patient.hospitalid,
        patient.hospitalDischargeYear,
        patient.wardid,
        patient.unitDischargeOffset
    FROM
        t4,
        patient
    WHERE
        t4.patientUnitStayID = patient.patientUnitStayID
),
t6 AS -- 3174 ICU stays
(
    SELECT
        t5.patientUnitStayID,
        t5.sofa_score,
        t5.sepsis_type,
        t5.hospitalid,
        t5.hospitalDischargeYear,
        t5.wardid,
        t5.unitDischargeOffset
    FROM
        t5,
        quality2014_2015_vf
    WHERE
        t5.hospitalDischargeYear = quality2014_2015_vf.hospitalDischargeYear
        AND t5.hospitalid = quality2014_2015_vf.hospitalid
),
-------------------------------------------------------------------
---------------------- LIMITATION ---------------------------------
-------------------------------------------------------------------
t7 AS -- Flag ICUstays with care limitation (1) such as: ----291
            -- Comfort measures only
            -- No vasopressors/inotropes
            -- No augmentation of care
(
    SELECT
        t6.patientUnitStayID,
        t6.unitDischargeOffset,
        t6.sofa_score,
        t6.sepsis_type,
        t6.hospitalid,
        t6.hospitalDischargeYear,
        t6.wardid,
        cplgroup,
        cplitemvalue,
        CASE
            WHEN (cplgroup = 'Care Limitation' AND cplitemvalue = 'Comfort measures only') THEN 1
            WHEN (cplgroup = 'Care Limitation' AND cplitemvalue = 'No vasopressors/inotropes') THEN 1
            WHEN (cplgroup = 'Care Limitation' AND cplitemvalue = 'No augmentation of care') THEN 1
            ELSE 0
        END AS care_limitation
    FROM
        t6
```

```
        LEFT OUTER JOIN
        careplangeneral
        ON t6.patientUnitStayID = careplangeneral.patientUnitStayID
),
t8 AS
(
    SELECT
        t7.patientUnitStayID,
        t7.unitDischargeOffset,
        t7.sofa_score,
        t7.sepsis_type,
        t7.hospitalid,
        t7.hospitalDischargeYear,
        t7.wardid,
        SUM(t7.care_limitation) AS care_limitation
    FROM
        t7
    GROUP BY
        t7.patientUnitStayID,
        t7.unitDischargeOffset,
        t7.sofa_score,
        t7.sepsis_type,
        t7.hospitalid,
        t7.hospitalDischargeYear,
        t7.wardid
    ORDER BY
        t7.patientUnitStayID
),
t9 AS
(
    SELECT
        t8.patientUnitStayID,
        t8.unitDischargeOffset,
        t8.sofa_score,
        t8.sepsis_type,
        t8.hospitalid,
        t8.hospitalDischargeYear,
        t8.wardid,
        CASE WHEN care_limitation > 0 THEN 1 ELSE 0 END AS care_limitation
    FROM
        t8
    ORDER BY
        t8.patientUnitStayID
),
t10 AS -- Flag patients with age < 18 (1) ----1
(
    SELECT
        t9.patientUnitStayID,
        t9.unitDischargeOffset,
        t9.sofa_score,
        t9.sepsis_type,
        t9.hospitalid,
        t9.hospitalDischargeYear,
        t9.wardid,
        t9.care_limitation,
        age,
        CASE
            WHEN (age='0' OR age='1' OR age='2' OR age='3' OR age='4' OR age='5'
            OR age='6' OR age='7' OR age='8' OR age='9' OR age='10'
            OR age='11' OR age='12' OR age='13' OR age='14' OR age='15'
            OR age='16' OR age='17') THEN 1
            ELSE 0
        END AS age_limitation
    FROM
        t9,
        patient
    WHERE
        t9.patientUnitStayID = patient.patientUnitStayID

    ORDER BY
        t9.patientUnitStayID
),
t11 AS -- Flag patients with unitDischargeOffset < 360 min (1) ---- 34
(
    SELECT
        t10.patientUnitStayID,
        t10.unitDischargeOffset,
        CASE WHEN t10.unitDischargeOffset < 360 THEN 1 ELSE 0 END AS LOS_limitation,
        t10.sofa_score,
        t10.sepsis_type,
        t10.hospitalid,
        t10.hospitalDischargeYear,
        t10.wardid,
        t10.care_limitation,
        t10.age,
        t10.age_limitation
    FROM
        t10
    ORDER BY
        t10.patientUnitStayID
),
---------- EXCLUSION ----------------------------------------------------
t12 AS
(
    SELECT
```

```
        t11.patientUnitStayID,
        t11.unitDischargeOffset,
        t11.LOS_limitation,
        t11.sofa_score,
        t11.sepsis_type,
        t11.hospitalid,
        t11.hospitalDischargeYear,
        t11.wardid,
        t11.care_limitation,
        t11.age,
        t11.age_limitation
    FROM
        t11
    WHERE
        t11.care_limitation = 0
        AND t11.age_limitation = 0
        AND t11.los_limitation = 0
    ORDER BY
        t11.patientUnitStayID
),
-------------------------------------------------------------------------------
-- ICU stays with ICD Codes, SOFA, antibiotics and admission diagnosis = sepsis
-- and in hospitals with good data quality
-- and excluded patients
-------------------------------------------------------------------------------
-- Number of sepsis patients/ICU and sepsis patients/hospital
-------------------------------------------------------------------------------
t13 AS
(
    SELECT
        t12.wardid,
        COUNT(t12.wardid) AS nb_sepsis_patients_wardid

    FROM
        t12
    GROUP BY wardid
),
t14 AS
(
    SELECT
        t12.patientUnitStayID,
        t12.age,
        t12.sofa_score,
        t12.sepsis_type,
        t12.hospitalid,
        t12.hospitalDischargeYear,
        t12.wardid,
        t12.unitDischargeOffset,
        t13.nb_sepsis_patients_wardid
    FROM
        t13,
        t12
    WHERE
        t12.wardid = t13.wardid
),
t15 AS
(
    SELECT
        t12.hospitalid,
        COUNT(t12.hospitalid) AS nb_sepsis_patients_hospital
    FROM
        t12
    GROUP BY hospitalid
),

t16 AS
(
    SELECT
        t14.patientUnitStayID,
        t14.sofa_score,
        t14.sepsis_type,
        t14.hospitalid,
        t14.hospitalDischargeYear,
        t14.wardid,
        t14.nb_sepsis_patients_wardid,
        t15.nb_sepsis_patients_hospital,
        t14.unitDischargeOffset
    FROM
        t14,
        t15
    WHERE
        t14.hospitalid = t15.hospitalid

),
------------------------ JUST TO RETRIEVE THE NUMBER OF UNIQUE PATIENTS ------------------------
t17 AS
(
    SELECT
        patient.uniquepid
    FROM
        t16,
        patient
    WHERE
        t16.patientunitstayid = patient.patientunitstayid
    ORDER BY
```

```
        patient.uniquepid
),
t18 AS
(
    SELECT
        count(t17.uniquepid) AS n
    FROM
        t17
    GROUP BY
        t17.uniquepid
    ORDER BY
        n DESC

),
t19 AS
(
    SELECT
        t18.n,
        count(t18.n) AS nb
    FROM
        t18
    GROUP BY
        t18.n
    ORDER BY
        nb DESC

)

SELECT * FROM t16; -- 2820 ICU stays
```

A.26 sofa.sql

```
DROP TABLE IF EXISTS sofa CASCADE;
CREATE TABLE sofa AS -- SOFA score, first ICU day for every ICU stay.

WITH
cohort1 AS (
SELECT * FROM patient),
sofa AS
(
SELECT
    pt.patientunitstayid,
    sofacardiovasc.sofa_cv
    + sofarespi.sofa_respi
    + sofarenal.sofarenal
    + sofaGCSliverplatelets.sofacoag
    + sofaGCSliverplatelets.sofaliver
    + sofaGCSliverplatelets.sofacns
    AS sofatotal
FROM cohort1 pt
LEFT OUTER JOIN sofacardiovasc
ON pt.patientunitstayid=sofacardiovasc.Patientunitstayid
LEFT OUTER JOIN sofarespi
ON pt.patientunitstayid= sofarespi.Patientunitstayid
LEFT OUTER JOIN sofarenal
ON pt.patientunitstayid= sofarenal.Patientunitstayid
LEFT OUTER JOIN sofaGCSliverplatelets
ON pt.patientunitstayid= sofaGCSliverplatelets.Patientunitstayid
ORDER BY pt.patientunitstayid
)

SELECT * FROM sofa;
```

A.27 sofacardiovasc.sql

```
DROP TABLE IF EXISTS sofacardiovasc CASCADE;
CREATE TABLE sofacardiovasc AS -- SOFA cardiovasc score, first ICU day for every ICU stay.
-- Average admission weight = 83.93 kg

WITH
cohort1 AS (
SELECT * FROM patient),
t1 AS -- MAP
(
WITH tt1 AS
(
SELECT patientunitstayid,
min(CASE WHEN noninvasivemean IS NOT NULL THEN noninvasivemean ELSE NULL END) AS map
FROM vitalaperiodic
WHERE observationoffset between -1440 AND 1440
GROUP BY patientunitstayid
),
```

```
tt2 AS
(
SELECT patientunitstayid,
min(CASE WHEN systemicmean IS NOT NULL THEN systemicmean ELSE NULL END) AS map
FROM vitalperiodic
WHERE observationoffset between -1440 AND 1440
GROUP BY patientunitstayid
)
SELECT pt.patientunitstayid, CASE WHEN tt1.map IS NOT NULL THEN tt1.map
when tt2.map IS NOT NULL THEN tt2.map
ELSE NULL END AS map
FROM patient pt
LEFT OUTER JOIN tt1
ON tt1.patientunitstayid=pt.patientunitstayid
LEFT OUTER JOIN tt2
ON tt2.patientunitstayid=pt.patientunitstayid
ORDER BY pt.patientunitstayid
),
t2 AS --DOPAMINE
(
SELECT
    distinct patientunitstayid,
    max(
        CASE
            WHEN lower(drugname) LIKE '%(ml/hr)%' AND drugrate NOT LIKE '%UD%' AND drugrate NOT LIKE '%Date%'
            THEN round(cast(drugrate AS numeric)/3.14,3)
            -- rate in ml/h * 1600 mcg/ml / 83.93 kg / 60 min, to convert in mcg/kg/min
            WHEN lower(drugname) LIKE '%(mcg/kg/min)%' AND drugrate NOT LIKE '%OFF%'
            THEN cast(drugrate AS numeric)
            ELSE NULL
        END) AS dopa

FROM infusiondrug id
WHERE lower(drugname) LIKE '%dopamine%' AND infusionoffset between -120 AND 1440
AND drugrate<>'' AND drugrate<>'.'
GROUP BY patientunitstayid
ORDER BY patientunitstayid
),
t3 AS  --NOREPI
(
SELECT
    distinct patientunitstayid,
    max(CASE
        WHEN lower(drugname) LIKE '%(ml/hr)%'
        AND drugrate<>''
        AND drugrate<>'.'
        AND drugrate NOT LIKE '%UD%'
        THEN round(cast(drugrate AS numeric)/314.8,3)
        -- rate in ml/h * 16 mcg/ml / 83.93 kg / 60 min, to convert in mcg/kg/min
        WHEN lower(drugname) LIKE '%(mcg/min)%'
        AND drugrate<>''
        AND drugrate<>'.'
        AND drugrate NOT LIKE '%OFF%'
        AND drugrate NOT LIKE '%Documentation undone%'
        THEN round(cast(drugrate AS numeric)/83.93 ,3)-- divide by 83.93 kg
        WHEN lower(drugname) LIKE '%(mcg/kg/min)%'
        AND drugrate<>''
        AND drugrate<>'.'
        THEN cast(drugrate AS numeric)
        ELSE NULL END ) AS norepi
FROM infusiondrug id
WHERE lower(drugname) LIKE '%norepinephrine%'  AND infusionoffset between -120 AND 1440
AND drugrate<>'' AND drugrate<>'.'
GROUP BY patientunitstayid
ORDER BY patientunitstayid
),
t4 AS  --DOBUTAMINE
(
SELECT
    distinct patientunitstayid,
    1 AS dobu
FROM infusiondrug id
WHERE lower(drugname) LIKE '%dobutamin%'
    AND drugrate <>''
    AND drugrate<>'.'
    AND drugrate <>'0'
    AND infusionoffset between -120 AND 1440
ORDER BY patientunitstayid
),
sofacv AS
(
SELECT
    pt.patientunitstayid,
    t1.map,
    t2.dopa,
    t3.norepi,
    t4.dobu,
    (CASE WHEN dopa>=15 OR norepi>0.1 THEN 4
    WHEN dopa>5 OR (norepi>0 AND norepi <=0.1) THEN 3
    WHEN dopa<=5 OR dobu > 0 THEN 2
    WHEN map <70 THEN 1
    ELSE 0 END) AS SOFA_cv --COMPUTE SOFA CV
FROM
    cohort1 pt
    LEFT OUTER JOIN t1
```

```
    ON t1.patientunitstayid=pt.patientunitstayid
    LEFT OUTER JOIN t2
    ON t2.patientunitstayid=pt.patientunitstayid
    LEFT OUTER JOIN t3
    ON t3.patientunitstayid=pt.patientunitstayid
    LEFT OUTER JOIN t4
    ON t4.patientunitstayid=pt.patientunitstayid
    ORDER BY pt.patientunitstayid
)

SELECT * FROM sofacv;
```

A.28 sofaGCSliverplatelets.sql

```
DROP TABLE IF EXISTS sofaGCSliverplatelets CASCADE;
CREATE TABLE sofaGCSliverplatelets AS
-- SOFA GCS, liver, platelets score, first ICU day for every ICU stay.

WITH
cohort1 AS
(
    SELECT * FROM patient
),
t1 AS --GCS
(
    SELECT
        patientunitstayid,
        sum(CAST(physicalexamvalue AS numeric)) AS gcs
    FROM physicalexam pe
    WHERE (lower(physicalexampath) LIKE '%gcs/eyes%'
        OR lower(physicalexampath) LIKE '%gcs/verbal%'
        OR lower(physicalexampath) LIKE '%gcs/motor%')
        AND physicalexamoffset between -1440 AND 1440
    GROUP BY patientunitstayid, physicalexamoffset
),
t1bis AS
(
SELECT t1.patientunitstayid, MIN(t1.gcs) AS gcs
FROM t1
GROUP BY patientunitstayid
),
t2 AS
(
    SELECT
        pt.patientunitstayid,
        max(
            CASE
                WHEN lower(labname) LIKE 'total bili%' THEN labresult
                ELSE null
            END) AS bili, --BILI
        min(
            CASE
                WHEN lower(labname) LIKE 'platelet%' THEN labresult
                ELSE null
            END) AS plt --PLATELETS
    FROM patient pt
    LEFT OUTER JOIN lab
    ON pt.patientunitstayid=lab.patientunitstayid
    WHERE labresultoffset between -1440 AND 1440
    GROUP BY pt.patientunitstayid
),
sofaGCSliverplatelets AS
(
    SELECT
        distinct pt.patientunitstayid,
        min(t1bis.gcs) AS gcs,
        max(t2.bili) AS bili,
        min(t2.plt) AS plt,
        max(
            CASE
                WHEN plt<20 THEN 4
                WHEN plt<50 THEN 3
                WHEN plt<100 THEN 2
                WHEN plt<150 THEN 1
                ELSE 0
            END) AS sofacoag,
        max(
            CASE
                WHEN bili>12 THEN 4
                WHEN bili>6 THEN 3
                WHEN bili>2 THEN 2
                WHEN bili>1.2 THEN 1
                ELSE 0
            END) AS sofaliver,
        max(
            CASE
                WHEN gcs=15 THEN 0
                WHEN gcs>=13 THEN 1
                WHEN gcs>=10 THEN 2
```

```
                WHEN  gcs>=6 THEN  3
                WHEN  gcs>=3 THEN  4
                ELSE 0
            END) AS sofacns
    FROM cohort1 pt
    LEFT OUTER JOIN t1bis
    ON t1bis.patientunitstayid=pt.patientunitstayid
    LEFT OUTER JOIN t2
    ON t2.patientunitstayid=pt.patientunitstayid
    GROUP BY pt.patientunitstayid, t1bis.gcs, t2.bili, t2.plt
    ORDER BY pt.patientunitstayid
)

SELECT * FROM sofaGCSliverplatelets
```

A.29 sofarenal.sql

```
DROP TABLE IF EXISTS sofarenal CASCADE;
CREATE TABLE sofarenal AS
-- SOFA renal score, first IUCU day for every ICU stay.

WITH
cohort1 AS (
SELECT * FROM patient),
t1 AS --CREATININE
(
SELECT
    pt.patientunitstayid,
    max(
        CASE
            WHEN lower(labname) LIKE 'creatin%'
                THEN labresult
                ELSE NULL
            END) AS creat
FROM patient pt
LEFT OUTER JOIN lab
ON pt.patientunitstayid=lab.patientunitstayid
WHERE labresultoffset between -1440 AND 1440
GROUP BY pt.patientunitstayid
),
t2 AS --UO
(
WITH uotemp AS
(
SELECT
    patientunitstayid,
    CASE WHEN dayz=1 THEN sum(outputtotal) ELSE NULL END AS uod1
FROM
    (
        SELECT distinct patientunitstayid,
        intakeoutputoffset,
        outputtotal,
        (CASE
            WHEN (intakeoutputoffset) between -120 AND 1440 THEN 1
            ELSE NULL
        END) AS dayz
        FROM intakeoutput
        WHERE intakeoutputoffset between 0 AND 5760
        ORDER BY patientunitstayid, intakeoutputoffset
    ) AS temp
GROUP BY patientunitstayid, temp.dayz
)
SELECT
    pt.patientunitstayid,
    max(
        CASE
            WHEN uod1 is not NULL THEN uod1
            ELSE NULL
        END) AS UO
FROM patient pt
LEFT OUTER JOIN uotemp
ON uotemp.patientunitstayid=pt.patientunitstayid
GROUP BY pt.patientunitstayid
),
sofarenal AS
(
SELECT
    pt.patientunitstayid, -- t1.creat, t2.uo,
    (CASE --COMPUTE SOFA RENAL
        WHEN uo <200 or creat>5 THEN 4
        WHEN uo <500 or creat >3.5 THEN 3
        WHEN creat between 2 AND 3.5 THEN 2
        WHEN creat between 1.2 AND 2 THEN 1
        ELSE 0
    END) AS sofarenal
FROM cohort1 pt
LEFT OUTER JOIN t1
ON t1.patientunitstayid=pt.patientunitstayid
LEFT OUTER JOIN t2
```

```
ON t2.patientunitstayid=pt.patientunitstayid
ORDER BY pt.patientunitstayid
)

SELECT * FROM sofarenal
```

A.30 sofarespi.sql

```
DROP TABLE IF EXISTS sofarespi CASCADE;
CREATE TABLE sofarespi AS
-- SOFA respi, first ICU day fOR every ICU stay.

WITH
cohort1 AS
(
    SELECT * FROM patient
),
tempo2 AS
(
WITH tempo1 as
(
WITH t1 AS --FIO2 FROM respchart
(
SELECT *
FROM
(
SELECT
    distinct patientunitstayid,
    max(cast(respchartvalue AS numeric)) AS rcfio2
FROM respiratorycharting
WHERE respchartoffset between -120 AND 1440 AND respchartvalue <> ''
    AND respchartvalue NOT LIKE '%C%'
    AND respchartvalue NOT LIKE '%O%'
    AND respchartvalue NOT LIKE '%S%'
    AND respchartvalue NOT LIKE '%o%'
    AND respchartvalue NOT LIKE '%T%'
    AND respchartvalue NOT LIKE '%H%'
    AND respchartvalue NOT LIKE '%Y%'
    AND respchartvalue NOT LIKE '%N%'
    AND respchartvalue NOT LIKE '%C%'
    AND respchartvalue NOT LIKE '%%%'
GROUP BY patientunitstayid
) AS tempo
WHERE rcfio2 >20 -- many values are liters per minute!
ORDER BY patientunitstayid
),
t2 AS --FIO2 FROM nursecharting
(
SELECT
    distinct patientunitstayid,
    max(cast(nursingchartvalue AS numeric)) AS ncfio2
FROM nursecharting nc
WHERE lower(nursingchartcelltypevallabel) LIKE '%fio2%' AND nursingchartentryoffset between -120 AND 1440
GROUP BY patientunitstayid
),
t3 AS --sao2 FROM vitalperiodic
(
SELECT
    patientunitstayid,
    min(CASE WHEN sao2 IS NOT NULL THEN sao2 ELSE null END) AS sao2
FROM vitalperiodic
WHERE observationoffset between -1440 AND 1440
GROUP BY patientunitstayid
),
t4 AS --pao2 FROM lab
(
SELECT
    patientunitstayid,
    min(CASE WHEN lower(labname) LIKE 'pao2%' THEN labresult ELSE null END) AS pao2
FROM lab
WHERE labresultoffset between -1440 AND 1440
GROUP BY patientunitstayid
),
t5 AS --airway type combining 3 sources (1=invasive)
(
WITH t1 AS --airway type FROM respcare (1=invasive) (by resp therapist!!)
(
SELECT
    distinct patientunitstayid,
    max(CASE WHEN airwaytype in ('Oral ETT','Nasal ETT','Tracheostomy') THEN 1 ELSE NULL END) AS airway
    -- either invasive airway OR NULL
FROM respiratorycare
WHERE respcarestatusoffset between -1440 AND 1440
GROUP BY patientunitstayid
),
t2 AS --airway type FROM respcharting (1=invasive)
(
SELECT
    distinct patientunitstayid,
```

```
    1 AS ventilator
FROM respiratorycharting rc
WHERE respchartvalue LIKE '%ventilator%'
OR respchartvalue LIKE '%vent%'
OR respchartvalue LIKE '%bipap%'
OR respchartvalue LIKE '%840%'
OR respchartvalue LIKE '%cpap%'
OR respchartvalue LIKE '%drager%'
OR respchartvalue LIKE 'mv%'
OR respchartvalue LIKE '%servo%'
OR respchartvalue LIKE '%peep%'
AND respchartoffset between -1440 AND 1440
GROUP BY patientunitstayid
),

t3 AS --airway type FROM treatment (1=invasive)
(
SELECT
    distinct patientunitstayid,
    max(CASE WHEN treatmentstring in
('pulmonary|ventilatiON AND oxygenation|mechanical ventilation',
'pulmonary|ventilatiON AND oxygenation|tracheal suctioning',
'pulmonary|ventilatiON AND oxygenation|ventilatOR weaning',
'pulmonary|ventilatiON AND oxygenation|mechanical ventilation|assist controlled',
'pulmonary|radiologic procedures / bronchoscopy|endotracheal tube',
'pulmonary|ventilatiON AND oxygenation|oxygen therapy (> 60%)',
'pulmonary|ventilatiON AND oxygenation|mechanical ventilation|tidal volume 6-10 ml/kg',
'pulmonary|ventilatiON AND oxygenation|mechanical ventilation|volume controlled',
'surgery|pulmonary therapies|mechanical ventilation',
'pulmonary|surgery / incisiON AND drainage of thorax|tracheostomy',
'pulmonary|ventilatiON AND oxygenation|mechanical ventilation|synchronized intermittent',
'pulmonary|surgery / incisiON AND drainage of thorax|tracheostomy|
 performed during current admissiON fOR ventilatory support',
'pulmonary|ventilatiON AND oxygenation|ventilatOR weaning|active',
'pulmonary|ventilatiON AND oxygenation|mechanical ventilation|pressure controlled',
'pulmonary|ventilatiON AND oxygenation|mechanical ventilation|pressure support',
'pulmonary|ventilatiON AND oxygenation|ventilatOR weaning|slow',
'surgery|pulmonary therapies|ventilatOR weaning',
'surgery|pulmonary therapies|tracheal suctioning',
'pulmonary|radiologic procedures / bronchoscopy|reintubation',
'pulmonary|ventilatiON AND oxygenation|lung recruitment maneuver',
'pulmonary|surgery / incisiON AND drainage of thorax|tracheostomy|planned',
'surgery|pulmonary therapies|ventilatOR weaning|rapid',
'pulmonary|ventilatiON AND oxygenation|prone position',
'pulmonary|surgery / incisiON AND drainage of thorax|tracheostomy|conventional',
'pulmonary|ventilatiON AND oxygenation|mechanical ventilation|permissive hypercapnea',
'surgery|pulmonary therapies|mechanical ventilation|synchronized intermittent',
'pulmonary|medications|neuromuscular blocking agent',
'surgery|pulmonary therapies|mechanical ventilation|assist controlled',
'pulmonary|ventilatiON AND oxygenation|mechanical ventilation|volume assured',
'surgery|pulmonary therapies|mechanical ventilation|tidal volume 6-10 ml/kg',
'surgery|pulmonary therapies|mechanical ventilation|pressure support',
'pulmonary|ventilatiON AND oxygenation|non-invasive ventilation',
'pulmonary|ventilatiON AND oxygenation|non-invasive ventilation|face mask',
'pulmonary|ventilatiON AND oxygenation|non-invasive ventilation|nasal mask',
'pulmonary|ventilatiON AND oxygenation|mechanical ventilation|non-invasive ventilation',
'pulmonary|ventilatiON AND oxygenation|mechanical ventilation|non-invasive ventilation|face mask',
'surgery|pulmonary therapies|non-invasive ventilation',
'surgery|pulmonary therapies|non-invasive ventilation|face mask',
'pulmonary|ventilatiON AND oxygenation|mechanical ventilation|non-invasive ventilation|nasal mask',
'surgery|pulmonary therapies|non-invasive ventilation|nasal mask',
'surgery|pulmonary therapies|mechanical ventilation|non-invasive ventilation',
'surgery|pulmonary therapies|mechanical ventilation|non-invasive ventilation|face mask'
) THEN 1  ELSE NULL END) AS interface
FROM treatment
WHERE treatmentoffset between -1440 AND 1440
GROUP BY patientunitstayid
ORDER BY patientunitstayid-- , treatmentoffset
),
t4 as
(
SELECT
    distinct patientunitstayid,
    max(CASE WHEN cplitemvalue LIKE '%Intubated%' THEN 1 ELSE NULL END) AS airway
    -- either invasive airway OR NULL
FROM careplangeneral
WHERE cplitemoffset between -1440 AND 1440
GROUP BY patientunitstayid
)

SELECT pt.patientunitstayid,
CASE
    WHEN t1.airway IS NOT NULL OR t2.ventilatOR IS NOT NULL OR t3.interface IS NOT NULL THEN 1
    ELSE null
END AS mechvent --summarize
FROM cohort1 pt
LEFT OUTER JOIN t1
ON t1.patientunitstayid=pt.patientunitstayid
LEFT OUTER JOIN t2
ON t2.patientunitstayid=pt.patientunitstayid
LEFT OUTER JOIN t3
ON t3.patientunitstayid=pt.patientunitstayid
LEFT OUTER JOIN t4
ON t4.patientunitstayid=pt.patientunitstayid
)
```

```
SELECT pt.patientunitstayid, t3.sao2, t4.pao2,
(CASE
    WHEN t1.rcfio2>20 THEN t1.rcfio2
    WHEN t2.ncfio2 >20 THEN t2.ncfio2
    WHEN t1.rcfio2=1 OR t2.ncfio2=1 THEN 100
    ELSE null END) AS fio2, t5.mechvent
FROM cohort1 pt
LEFT OUTER JOIN t1
ON t1.patientunitstayid=pt.patientunitstayid
LEFT OUTER JOIN t2
ON t2.patientunitstayid=pt.patientunitstayid
LEFT OUTER JOIN t3
ON t3.patientunitstayid=pt.patientunitstayid
LEFT OUTER JOIN t4
ON t4.patientunitstayid=pt.patientunitstayid
LEFT OUTER JOIN t5
ON t5.patientunitstayid=pt.patientunitstayid
)
SELECT *,
coalesce(pao2,100)/coalesce(coalesce(nullif(fio2,0),21),fio2,21) AS pf,
coalesce(sao2,100)/coalesce(coalesce(nullif(fio2,0),21),fio2,21) AS sf
FROM tempo1
),
sofarespi AS
(
SELECT patientunitstayid,
(CASE WHEN pf <1 OR sf <0.67 THEN 4  --COMPUTE SOFA RESPI
WHEN pf between 1 AND 2 OR sf between 0.67 AND 1.41 THEN 3
WHEN pf between 2 AND 3 OR sf between 1.42 AND 2.2 THEN 2
WHEN pf between 3 AND 4 OR sf between 2.21 AND 3.01 THEN 1
WHEN pf > 4 OR sf> 3.01 THEN 0 ELSE 0 END ) AS SOFA_respi
FROM tempo2
ORDER BY patientunitstayid
)

SELECT * FROM sofarespi
```

A.31 vaso.sql

```
DROP TABLE IF EXISTS vaso CASCADE;
CREATE TABLE vaso AS

WITH
t1 AS
(
    SELECT
        patientUnitStayID,
        drugname,
        infusionoffset,
        drugRate,
        infusionRate,
        drugAmount,
        volumeOfFluid
    FROM
        infusionDrug
    WHERE
        lower(drugname) like '%epinephrine%'
        or lower(drugname) like 'epi (mcg/min)'
        or lower(drugname) like '%norepinephrine%'
        or lower(drugname) like '%levoph%'
        or lower(drugname) like '%phenylephrine%'
        or lower(drugname) like '%neo-synephrine%'
        or lower(drugname) like '%neosynephrine%'
        or lower(drugname) like '%neosynsprine%'
        or lower(drugname) like '%synephrine%'
        or lower(drugname) like '%vasopressin%'
    ORDER BY patientUnitStayID
),
t2 AS (
    SELECT
        t1.patientUnitStayID,
        t1.drugname,
        t1.infusionoffset,
        t1.drugRate,
        t1.infusionRate,
        t1.drugAmount,
        t1.volumeOfFluid
    FROM
        t1,
        sepsis_total
    WHERE
        t1.patientUnitStayID = sepsis_total.patientUnitStayID
),
t3 AS
(    SELECT
        t2.patientUnitStayID,
        patient.wardid,
        t2.drugname,
        t2.infusionoffset,
```

```
        t2.drugRate,
        t2.infusionRate,
        t2.drugAmount,
        t2.volumeOfFluid
    FROM
        t2,
        patient
    WHERE
        t2.patientUnitStayID = patient.patientUnitStayID
)

SELECT * FROM t3;
```

A.32 vaso_subset_icu.sql

```
DROP TABLE IF EXISTS vaso_subset_icu CASCADE;
CREATE TABLE vaso_subset_icu AS

WITH
t1 AS
(
    SELECT
        patientUnitStayID,
        wardid,
        drugname,
        infusionoffset,
        drugRate,
        infusionRate,
        drugAmount,
        volumeOfFluid
    FROM
        vaso
    WHERE

        (wardid = 369 AND drugRate IS NOT NULL
        AND LOWER(drugRate) NOT LIKE ''
        AND LOWER(drugRate) NOT LIKE '0')
        OR
        (wardid = 413 AND drugRate IS NOT NULL
        AND LOWER(drugRate) NOT LIKE ''
        AND LOWER(drugRate) NOT LIKE '0')
        OR
        (wardid = 347 AND drugRate IS NOT NULL
        AND LOWER(drugRate) NOT LIKE ''
        AND LOWER(drugRate) NOT LIKE '0')
        OR
        (wardid = 337 AND drugRate IS NOT NULL
        AND LOWER(drugRate) NOT LIKE ''
        AND LOWER(drugRate) NOT LIKE '0')
        OR
        (wardid = 376 AND drugRate IS NOT NULL
        AND LOWER(drugRate) NOT LIKE ''
        AND LOWER(drugRate) NOT LIKE '0')
        OR
        (wardid = 391 AND drugRate IS NOT NULL
        AND LOWER(drugRate) NOT LIKE ''
        AND LOWER(drugRate) NOT LIKE '0')
        OR
        (wardid = 312 AND drugRate IS NOT NULL
        AND LOWER(drugRate) NOT LIKE ''
        AND LOWER(drugRate) NOT LIKE '0')
        OR
        (wardid = 324 AND drugRate IS NOT NULL
        AND LOWER(drugRate) NOT LIKE ''
        AND LOWER(drugRate) NOT LIKE '0')
        OR
        (wardid = 408 AND drugRate IS NOT NULL
        AND LOWER(drugRate) NOT LIKE ''
        AND LOWER(drugRate) NOT LIKE '0')
        OR
        (wardid = 1029 AND drugRate IS NOT NULL
        AND LOWER(drugRate) NOT LIKE ''
        AND LOWER(drugRate) NOT LIKE '0')
        OR
        (wardid = 1026 AND drugRate IS NOT NULL
        AND LOWER(drugRate) NOT LIKE ''
        AND LOWER(drugRate) NOT LIKE '0')
        OR
        (wardid = 1032 AND drugRate IS NOT NULL
        AND LOWER(drugRate) NOT LIKE ''
        AND LOWER(drugRate) NOT LIKE '0')
        OR
        (wardid = 1039 AND drugRate IS NOT NULL
        AND LOWER(drugRate) NOT LIKE ''
        AND LOWER(drugRate) NOT LIKE '0')
        OR
        (wardid = 1027 AND drugRate IS NOT NULL
        AND LOWER(drugRate) NOT LIKE ''
        AND LOWER(drugRate) NOT LIKE '0')
```

```
        OR
        (wardid = 1035 AND drugRate IS NOT NULL
        AND LOWER(drugRate) NOT LIKE ''
        AND LOWER(drugRate) NOT LIKE '0')
    ORDER BY
        patientUnitStayID, infusionoffset
),
t2 AS
(
    SELECT
        t1.patientUnitStayID,
        t1.wardid,
        t1.drugname,
        t1.infusionoffset,
        t1.drugRate,
        t1.infusionRate,
        t1.drugAmount,
        t1.volumeOfFluid
    FROM
        t1
    WHERE
        t1.infusionoffset < 4321
)

SELECT * FROM t2;
```

B

Appendix 2

In this appendix we present supplementary information for both the ICU and hospital analysis in each section.

B.1 Estimator of treatment propensity

Table B.1 and Table B.2 displays the number of patient stays and the total number of periods per site after splitting the extracted dataset into training and testing sets with a split ratio of 80/20 without overlapping stays. These figures are identical for each imputed dataset.

Dataset	Stays	Periods	Covariates
Train	458	6330	9
ICU 1	90	1251	9
ICU 2	95	1352	9
ICU 3	142	1857	9
ICU 4	131	1870	9
Test	115	1596	9
ICU 1	22	318	9
ICU 2	23	322	9
ICU 3	31	423	9
ICU 4	39	533	9
Total (train & test)	573	7926	9

Table B.1: ICU - train and test data for the estimator of site propensity

Dataset	Stays	Periods	Covariates
Train	669	9716	9
Hospital 1	139	1819	9
Hospital 2	171	2491	9
Hospital 3	148	1962	9
Hospital 4	211	3444	9
Test	168	2475	9
Hospital 1	34	461	9
Hospital 2	36	488	9
Hospital 3	50	742	9
Hospital 4	48	784	9
Total (train & test)	837	12191	9

Table B.2: Hospital - train and test data for the estimator of site propensity

B.2 The treatment variable

Table B.3 and Table B.4 shows the marginal distribution of treatments across sites in the subset of sepsis patients retained after the first step. The marginal distribution is similar for each imputed dataset since the treatment variable was not imputed with MCE.

Site	Treatment								
	1	2	3	4	5	6	7	8	9
ICU 1	100	292	307	8	100	118	6	70	52
ICU 2	103	248	185	6	131	97	4	81	89
ICU 3	157	438	399	5	144	210	7	138	154
ICU 4	255	649	505	0	173	157	1	95	300
Total	615	1627	1396	19	548	582	18	384	595

Table B.3: ICU - marginal distribution of treatments

Site	Treatment								
	1	2	3	4	5	6	7	8	9
Hospital 1	196	623	490	7	184	234	12	208	203
Hospital 2	274	789	605	3	241	215	5	144	384
Hospital 3	216	646	562	10	269	279	24	225	270
Hospital 4	317	1083	964	26	420	457	15	251	422
Total	1003	3141	2621	46	1114	1185	56	828	1279

Table B.4: Hospital - marginal distribution of treatments

B.3 Estimator of treatment propensity

Table B.5 and Table B.6 displays the distribution of patient stays, periods and treatments over the sites for the training and the testing sets.

Dataset	Stays	Periods	Treatment									Features
			1	2	3	4	5	6	7	8	9	
Train	341	4584	497	1229	1112	18	470	479	12	317	450	13
ICU 1	64	875	84	225	265	7	95	97	2	54	46	13
ICU 2	57	770	93	200	142	6	111	77	4	69	68	13
ICU 3	100	1268	130	301	325	5	113	163	5	117	109	13
ICU 4	120	1671	190	503	380	0	151	142	1	77	227	13
Test	86	1200	118	398	284	1	78	103	6	67	145	13
ICU 1	13	178	16	67	42	1	5	21	4	16	6	13
ICU 2	11	174	10	48	43	0	20	20	0	12	21	13
ICU 3	30	384	27	137	74	0	31	47	2	21	45	13
ICU 4	32	464	65	146	125	0	22	15	0	18	73	13
Total (train & test)	427	5784	615	1627	1396	19	548	582	18	384	595	13

Table B.5: ICU - train and test data for the estimator of treatment propensity (including treatment information)

Dataset	Stays	Periods	Treatment									Features
			1	2	3	4	5	6	7	8	9	
Train	620	9060	1	2	3	4	5	6	7	8	9	13
Hospital 1	127	1668	180	477	362	6	140	183	11	174	135	13
Hospital 2	158	2262	245	672	516	3	194	174	4	128	326	13
Hospital 3	142	1938	154	496	420	8	200	221	22	188	229	13
Hospital 4	193	3192	258	868	793	15	326	387	9	195	341	13
Test	155	2213	1	2	3	4	5	6	7	8	9	13
Hospital 1	36	489	16	146	128	1	44	51	1	34	68	13
Hospital 2	28	398	29	117	89	0	47	41	1	16	58	13
Hospital 3	42	563	62	150	142	2	69	58	2	37	41	13
Hospital 4	49	763	59	215	171	11	94	70	6	56	81	13
Total (train & test)	775	11273	1003	3141	2621	46	1114	1185	56	828	1279	13

Table B.6: Hospital - train and test data for the estimator of treatment propensity (including treatment information)