ADS1001 | Data Challenges 1

ICU Mortality Rates

For patients aged 70-90, what are the most common variables that contribute to their mortality?

ICU image via www.shutterstock.com.

Report completed by:

Fiona George [33154430]

Pooja Kampli [33154759]

Hajeong Lee [33154341]

Brandon Dinh [31451160]

TABLE OF **CONTENTS**

INTRODUCTION

The objective of this assignment is to develop a model prototype that can predict the mortality of ICU patients using the data from the PhysioNet Computing in Cardiology Challenge 2012.

This report investigates trends and patterns in the data, identifies correlations between variables and draws valid conclusions based on thoroughly reviewed evidence. Our main approach to the problem is logistic regression, a simple yet effective Machine Learning algorithm.

Pooja Kampli conducted half of the background research associated with the variables involved in our analysis. Apart from her individual section in the presentation, she was also involved in the coding process and data analysis. Fiona George assisted Pooja with the other half of the background research. In addition to organising the presentation slides and overall layout of the report, she coded the majority of the logistic regression model and contributed to the data analysis. Hajeong Lee carried out the preprocessing of the dataset and assisted with correcting coding errors. Furthermore, she covered the in-depth analysis of the results of our findings. Brandon Dinh covered the script for his individual section of the presentation. Additionally, there were many overlaps within our roles, with team members assisting and collaborating with each other in areas that required greater attention and scrutiny.

BACKGROUND

Supplied Data

We were supplied with the recorded data of a thousand ICU patients for their first 2 days of their stay in the ICU. All relevant variables were recorded according to the patient's needs, meaning that every single variable was not recorded for every single patient, instead a duplicate number was placed in all empty spaces where there was no variable recorded. There are a total of 42 variables recorded, 6 of these are collected in admission to the ICU, while the others are observed throughout the duration of the patient's stay.

Research on Variables

We conducted background research on each of the variables we chose to analyse for this project and how they can contribute to ICU mortality rates.

Heart rate

The heart rate of a patient is dependent on various factors including physical activities, ventilation type, body temperature, and so on. Very high heart rate levels have been linked to severity of diseases and mortality in critical patients in the ICU. Many studies and trials have shown that drugs and medication which include beta blockers and reduce heart rate can help reduce mortality in some patients. The normal heart rate is 60 - 100 bpm for healthy individuals above the age of 10.

Respiratory Rate

Respiratory rate is a vital indicator for a serious condition that could be potentially happening in the body. A study has shown that in a group of patients that had been admitted on the general wards in the hospital, half of them that were suffering from a serious issue had a respiratory rate higher than 24 breaths per minute and in another study, 21% of a group of patients with a respiratory rate between 25-29 breaths per minute assessed by a critical care outreach service died in hospital. Because breathing requires multiple systems of the body to execute properly, problems regarding the body can easily be found through a patient's breathing rate.

Temperature

The most important thing that temperature tests is whether or not a patient has a fever. Human temperature is typically around 36.5 and 37.2 degrees Celsius, and can vary depending on gender, food, fluid consumption, time of day, ect. However, having a fever which is one degree over the normal range is considered important as it is a sign of your body getting rid of bad bacteria in the body. Having a fever for an extended period of time or having more serious symptoms as well as a fever can signal more dire problems happening with the body.

Glucose

Glucose is the primary type of sugar and major source of energy for cells. Various different hormones, including insulin control levels of glucose in the blood, the normal levels ranging between 70 to 130 mg/dL. Glucose levels typically increase with age and extremely high levels are usually connected to cardiovascular disease-caused deaths.

Cholesterol

Cholesterol levels of a patient are usually dependent on the food they eat. Three quarters of cholesterol usually comes from the body producing it in the liver so the reason why someone has too much or not enough cholesterol is because of their diet. The normal range for cholesterol for a healthy person is recommended to be below 200 mg/dL (5.17mmol/L). The higher the levels of cholesterol the higher the probability of a risk in a heart attack or stroke. A third of ischaemic heart disease points to high cholesterol levels as a major contributing factor.

Invasive systolic arterial blood pressure (SysABP)

Invasive systolic arterial blood pressure is the pressure in the arteries as the heart beats and blood is pumped into the arteries. A healthy range in which it is recommended is between 90 to 120 mm Hg for adults. There are many contributing factors that can cause this to rise mainly by diet or lack of exercise. High systolic blood pressure can lead to an increased likelihood of strikes, kidney or heart disease. It is vital medications provided to control systolic hypertension do not cause the levels to drop too low as it can cause dizziness, syncope or organ failure.

Invasive diastolic arterial blood pressure (DiasABP)

Invasive diastolic arterial blood pressure is the pressure building on the arterial wall when the heart muscle relaxes, allowing the chambers to fill with blood. It occurs in the ventricular relaxation period when preparing for the next heart muscle contraction. The levels are known to decrease with age and the normal range is from 60-80 mm Hg for adults. When levels are too high it causes systolic hypertension which involves sleep apnea, endocrine and renovascular disorders. Whereas when

levels are too low, it can cause low levels of blood and oxygen in the heart or ischemia caused by medications causing the levels to lower.

Weight

Weight is both a general descriptor, recorded on admission, and a time series variable, often measured hourly, for estimating fluid balance. Obesity is associated with low-grade chronic inflammation and cardiometabolic diseases, also increasing the risk of infections and sepsis. It can also lead to diabetes, hypertension, coronary vascular disease and overall significantly increases risk of mortality. Additionally, very low weight is commonly associated with mortality for infants and children and a BMI of less than 18.5kg/㎡ has caused an increase in mortality rates.

Urine

Urine output is measured hourly and per kg of body weight. The normal human urine output ranges from 0.5mL to 1.5mL per hour. In-hospital mortality in intensive care patients is mostly associated with septic shock which is a life-threatening condition that happens when your blood pressure drops to a dangerously low level after an infection.

Correlation between variables

- Heart rate and respiratory rates are correlated as when the heart beat increases, breathing increases as well, sending more oxygen to the body as the heart uses more energy.
- Blood pressure readings are given in two numbers where the top number is the systolic pressure - the maximum pressure the heart exerts while beating. The bottom number is the diastolic pressure - the amount of pressure in the arteries between beats.
- The higher the weight of the individual, typically the higher amount of urine output.

DATA PREPARATION

Preprocessing

Initially, we used Google Colaboratory (Colab) for the coding aspects as it was much easier for team members to edit simultaneously as well as view progress without having to electronically send multiple versions of the Jupyter notebook.

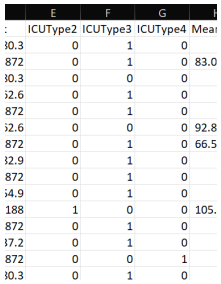
```
from google.colab import auth
auth.authenticate_user()

import gspread
from google.auth import default
creds, _ = default()

gc = gspread.authorize(creds)
```

```
worksheet = gc.open('Preprocessed ICU data').sheet1

# get_all_values gives a list of rows.
rows = worksheet.get_all_values()
print(rows)

import pandas as pd
df = pd.DataFrame.from_records(rows)
```

As we started to clean up the data, however, we realised that the program was not automatically detecting the column titles. Instead, they were recognised as raw values as part of the first row of the dataset. Upon consultation with Zach, we were able to figure out a solution:

```
#change the column titles to the ones in the first row
df.columns = df.iloc[0]
#remove the first row as they have already been converted to the titles
df = df.drop(index = 0, axis = 0)
```

The version of the dataset we received had already been partly preprocessed. Even so, it had separate columns for ICU Types 2, 3 and 4, but not for ICU Type 1. To overcome this, we prepared the following code:



	E	F	G	
:	ICUType2	ICUType3	ICUType4	Mea
80.3	0	1	0	
872	0	1	0	83.0
80.3	0	0	0	
52.6	0	1	0	
872	0	1	0	
52.6	0	0	0	92.8
872	0	1	0	66.5
52.9	0	1	0	
872	0	1	0	
54.9	0	1	0	
188	1	0	0	105.
872	0	1	0	
87.2	0	1	0	
872	0	0	1	
80.3	0	1	0	

```
icutype2 = df[df['ICUType2']==0]
icutype3 = icutype2[icutype2['ICUType3']==0]
icutype1 = icutype3[icutype3['ICUType4']==0]
icutype1.describe()
```

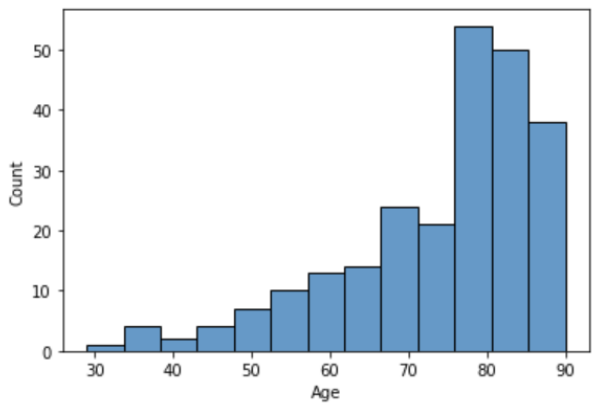
At this point, we moved our coding to Jupyter Notebook as Google Colab did not successfully run some of our codes, and the additional work to overcome those hiccups was an unnecessary inconvenience.

Essentially, what the code above did was create another dataframe ‘icytype1’, where the entries of patients (whole rows) from ICU Types 2, 3 and 4 were filtered out, leaving only the entries of patients who were admitted into ICU Type 1.

Secondly, we had to reduce the number of patients to those within a specific age range. The reason for this being our background research revealing that age can be another factor that could determine the health conditions of those being admitted into ICU.

```
ages = icutype1['Age'].unique()
sns.histplot(data=icutype1,x='Age')
ages
np.sort(ages)[::1]

icutype1.pivot_table(index='Age',values='Gender',aggfunc='count')
```

Age	Gender
29	1
34	1
36	2
37	1
42	1
43	1
44	1
46	2
47	1
49	2
50	2
51	1
52	2
53	1
54	2
56	4
57	3
58	3
59	1
60	2
61	7
62	2
63	1
64	6
65	2
66	3
67	3
68	5
69	4
70	8
71	4
72	5
72	5
73	5
74	4
75	7
76	6
77	17
78	9
79	11
80	11
81	8
82	7
83	12
84	11
85	12
86	6
87	3
88	7
89	5
90	17



From the table and histogram, we were able to gauge the range of the ages of the patients. The minimum was 29 and the maximum was 90. To prevent age from being an external influence, we decided to narrow our research question to those aged between 70 to 90, and consequently, create another dataframe 'agerange’ to work with.

```
agerange = icutype1[icutype1['Age']>69]
```

Lastly, we made an adjustment to the column title ‘In.hospital_death’ as it was causing problems with the logistic regression coding. This will be explained later in the report.

```
agerange.rename(columns = {"In.hospital_death":"In_hospital_death"},inplace = True)
```

Data Manipulation

After filtering our dataframe to better suit our research question, we went into more detail when assessing how we could extract information from the values provided for each variable.

Using urine output as a simple measurement to determine its effect on the mortality of patients did not seem to be a valid comparison. After carrying out further background research, we realised that we could use the correlation between weight and urine output to better assess the mortality rates.

So, we created another column 'Mean_UrinePKG.x' which contained the mean urine output (mL) per kg of weight of each patient.

```
agerange['Mean_UrinePKG.x'] = agerange['Mean_Urine.x'].div(agerange['Mean_Weight.x'])
agerange['Mean_UrinePKG.y'] = agerange['Mean_Urine.y'].div(agerange['Mean_Weight.y'])
agerange.sort_values(by=['Mean_UrinePKG.x'],ascending=False)
agerange.index[42]
agerange = agerange.drop(agerange.index[42])

agerange.sort_values(by=['Mean_UrinePKG.x'],ascending=False)
```

By using the new values to create the logistic regression model for the effect of urine on the mortality of patients, we not only satisfied the medical accuracy of our approach, but also improved the validity of our analysis.

Next, we subsetted the DataFrame further, taking out only the variables we would be using from the set of Day 1 entries (first day of patient admission into ICU Type 1).

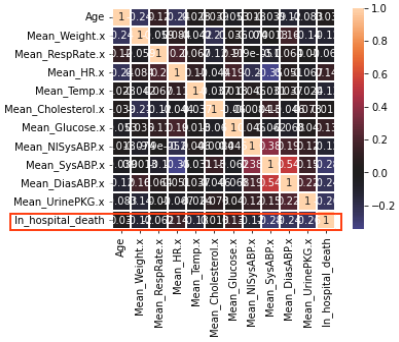
```
variables = agerange[['Age','In_hospital_death','Mean_Weight.x','Mean_Weight.y','Mean_RespRate.x','Mean_RespRate.y','Mean_HR
variables1 = variables[['Age','Mean_Weight.x','Mean_RespRate.x','Mean_HR.x','Mean_Temp.x','Mean_Cholesterol.x','Mean_Glucose

.y','Mean_HR.x','Mean_HR.y','Mean_Temp.x','Mean_Temp.y','Mean_Cholesterol.x','Mean_Cholesterol.y','Mean_Glucose.x','Mean_Gluc
Mean_Glucose.x','Mean_NISysABP.x','Mean_SysABP.x', 'Mean_DiasABP.x','Mean_UrinePKG.x', 'In_hospital_death']]

se.y','Mean_NISysABP.x','Mean_NISysABP.y','Mean_SysABP.x','Mean_SysABP.y','Mean_DiasABP.x','Mean_DiasABP.y','Mean_UrinePKG.x'

','Mean_DiasABP.y','Mean_UrinePKG.x','Mean_UrinePKG.y']]
```

As a result, 'variables1' became the final, processed DataFrame we decided to work with.

DATA PROCESSING

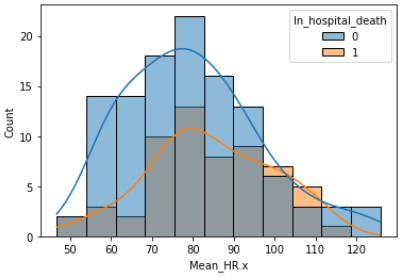
Exploratory Data Analysis

After filtering the data we needed and labelling it ‘variables1’, we assessed how we can use the filtered data to answer our research question.



We used a heatmap to visually represent which variables had more impact on the mortality of the patients in ICU type 1, within the age range of 70-90. Having a visual representation helped us to see with one image the more prominent variables that we could choose to focus on.

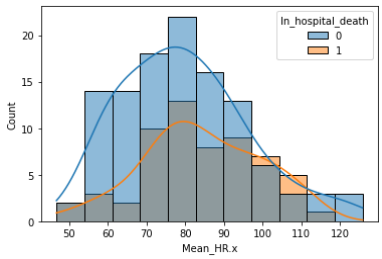
However, the differences were very subtle in the ‘In_hospital_deaths’ row, as the colours were very close to each other, barely noticeable by eyes. Hence, we decided to perform logistic regression on each variable, rather than to pick out variables from the heatmap alone.



Initially, our plan was to divide the values of the data into whether they belonged within the normal range, i.e. whether the values were that of a healthy human or not, but our first task was to create a plot for each of these data. We used the seaborn.histplot function to produce these plots, using the variables as the x-axes, and the count as the y-axis. We differentiated whether the patient died in the hospital or not by the colours. This is an example of the histograms that we produced from the data. The orange plot signifies the number of patients who died in the hospital, whereas the blue signifies those who did not. The colour that looks brown on the graph is a result of the two colours overlapping. We produced the graphs for each of the variables.

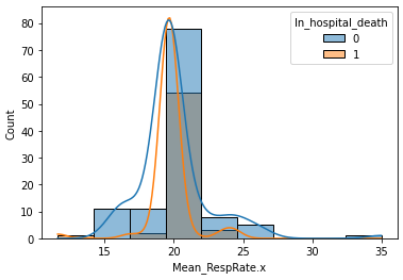
Interpreting Results

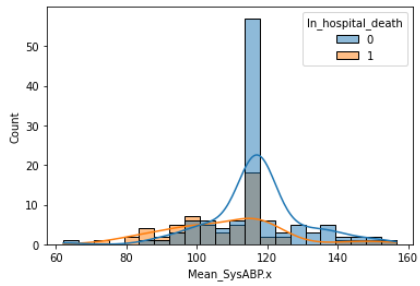
From these plots, we were given a general idea of what the trends were for each variable, and how many people survived in certain ranges of the variables. Some graphs were easily interpreted, due to the somewhat evenly distributed plots, but others were harder to interpret due to a significant number of unrecorded values that were replaced by the mean values of the variable.



The graph on the left is the histogram for Heart Rate (abbr. HR) with mortality as the dependent variable on the y-axis. It is a visible trend that there are the most number of total patients that had a HR within the range of 60-100 BPM. However, in the 97-111 BPM range, the clear orange colour above the brown signifies that there were more patients that died in that range than those who survived.

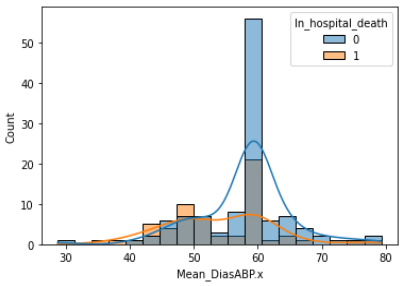
The graph on the right is the histogram for Respiratory Rate (abbr. RespRate) with the number of patients on the y-axis, blue signifying that of those who survived and orange signifying that of those who died in the hospital. Many patients had an unmeasured RespRate and the data was put down as the mean value. This explains the abnormal height of the peak at 19.67 on the x-axis.



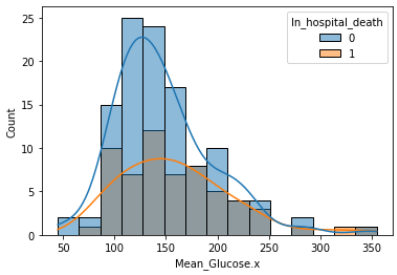


The graph on the left is the histogram for Invasive Systolic Arterial Blood Pressure (abbr. SysABP), with the number of patients on the y-axis. Similarly to the RespRate histogram, the high peak at 117 due to many patients' values not being measured and being replaced with the mean value instead. It is still clear that an extraordinarily high portion of those with low SysABP values died in the hospital, as signified by the clear orange bars below the 100 mark.

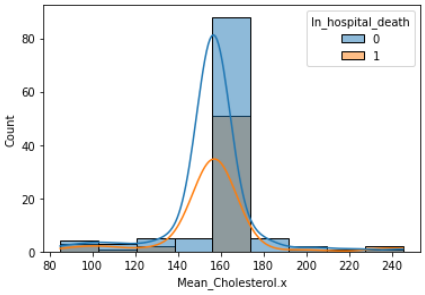
The graph on the right is the histogram for Invasive Diastolic Arterial Blood Pressure (abbr. DiasABP), with the number of patients on the y-axis. As DiasABP is highly correlated with SysABP, it is no surprise that many patients had no measuring of DiasABP either, explaining the high peak at 59 on the x-axis. Similarly to SysABP, those with low DiasABP values are more likely



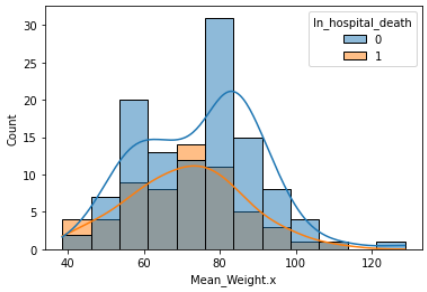
to die. The blood pressure value (e.g. 120/90, 90/60) is a value that consists of SysABP as the numerator and the DiasABP as the denominator.



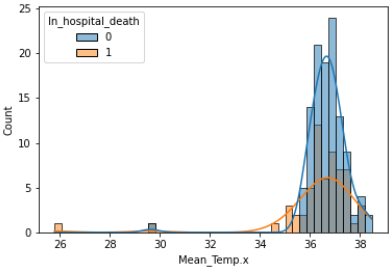
The graph on the left is the histogram for Glucose, with the number of patients on the y-axis. Although glucose levels were measured for most of the patients, as shown by the lack of any tall peaks, near to no range had more patients who died than those who survived. Most ranges that had a significant number of deaths also had a significant number more who survived in that range, sometimes even adding up to double the number of deaths. From the graph alone, we can hypothesise that the regression correlation will be relatively low compared to the other variables.



The graph on the right is the histogram for Cholesterol levels, with the number of patients on the y-axis. Similarly to RespRate, a significant number of patients had unrecorded values for this variable, resulting in the insanely tall peak at 156 on the x-axis. There are no values or ranges that show a significant increase in the number of deaths when judging from the graph, especially when compared to the number of surviving patients in the same range. There is no clear orange to be seen on the graph, hence we can guess that cholesterol levels will not affect the mortality rate as much as some other factors.

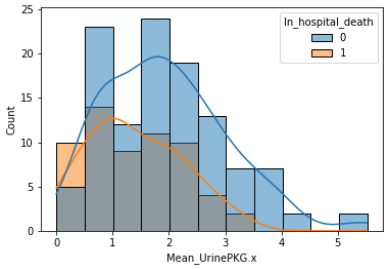


The graph on the left is the histogram for weights of patients, with the number of patients on the y-axis. While most patients' weights were recorded, resulting in a relatively even spread of values, weights do not hold meaning without any supplementary data, such as height. As a result, weights do not show much correlation with mortality rates in ICU. This is mostly due to the effects of other factors, such as height and muscle mass. While height was given but we decided against using it to find further information, muscle mass of the patients were not given in the csv file that we received, so further analysis was not possible.



The graph on the right is the histogram for body temperature (abbr. Temp), with the number of patients on the y-axis. While the majority of the patients lie within the standard range of 36-37 degrees celsius, there are some with extremely low or high

temperatures. For example, below the 36 degree mark, there are a significant number of patients who died. This is mostly due to any temperature below 35 degree is considered hypothermia, causing deaths.



The graph on the left is the histogram for urine per kilogram of body weight (abbr. UrinePKG), with the number of patients on the y-axis. While big amounts of UrinePKG is not necessarily a problem as the graph shows that more patients with high UrinePKG values survive than die. However, low urine levels are concerning as there are a significantly larger number, almost double the number, of patients who die with values lower than the standard range of 0.5-1.5mL/kg when compared with those who survived within that range.

Modelling

For the modelling of our data, we decided to use logistic regression rather than linear regression, as our y variable, "In_hospital_death", was categorical. The two categories were whether they died in hospital (1) or they did not (0). We set each of our factors as the x variables, then conducted the logistic regression models.

```
X = agerange[["Mean_HR.x"]]
Y = agerange['In_hospital_death']

# split into a training set with 80% of the data, and a testing set as the remainder
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)

logistic = LogisticRegression(fit_intercept=True) # instantatiate the linear regression model
logistic.fit(X_train,Y_train) # fit the data to the model

print("Intercept is",np.round(logistic.intercept_,2))
print("Coefficients are",np.round(logistic.coef_, 2))
```

This is the code we used to perform the logistic regression on Heart Rate. We used the logistic regression function from the 'sklearn' library, as provided in the week 10 notebook. We then split 80% of our x and y variables into a training set and the remaining into our testing set and named the variables 'X_train', 'Y_train', 'X_test', 'Y_test' respectively. Following the split, we initiated the logistic regression model and named the model 'logistic'. We then fit the training set into the model, which was used to give the intercept and the coefficient of our model.

```
#above 1: decreasing the odds | below 1: increasing the odds
print("Intercept is",np.round(np.exp(logistic.intercept_),5))
print("Coefficients are",np.round(np.exp(logistic.coef_), 2))
```

Then to be more concise with the mortality rates with the odds of the patient dying, we exponentiated the logistic intercept and the coefficient with the 'exp' function from the 'numpy' library. The exponentiated coefficients signify the change in odds for each unit change of each variable. We then interpreted these values in context. For example, the exponentiated coefficient for Heart Rate is 1.02, so the odds of a patient dying increases by 0.02 for every one BPM change in Heart Rate. Then we repeated the same process for all the remaining variables to perform the modelling on them as well as obtain the odds from the exponentiated coefficients.

The values that we obtained for each variable were as follows:

Variable	Exponentiated intercept	Exponentiated coefficient
Heart Rate	0.14292	1.02
Respiratory Rate	0.5165	1.0
SysABP	37.04566	0.96
DiasABP	8.67267	0.95
Weight	1.56939	0.99
Cholesterol	0.99997	1.0
Glucose	0.16869	1.01
Temperature	618344.92715	0.68
Urine	1.37541	0.56

When the results above were obtained, some values made us question the code. For example, the exponentiated intercept for temperature is over 6 digits, which is quite odd, considering that most other numbers lie between 0 and 2. However, upon searching for errors, none were detected, so we had to keep the number. Despite acknowledging its problem, we decided to keep the value as we were more interested in the coefficient to answer our research question - which variables affected the mortality the most. Upon assessing the coefficients, we reached the conclusion which stated that the variables with the highest correlation were urine output and temperature - decreasing by 0.44 and 0.32 respectively for every change in their respective units, degree celsius and mL.

Testing and Deployment

```
pred_probs = logistic.predict_proba(X_train) #get predicted probabilities
pred_probs = pd.DataFrame(pred_probs)
pred_probs["death"] = np.where(pred_probs[1] > 0.5, 1, 0)
pred_probs["true"] = Y_train.reset_index().In_hospital_death

pred_probs
```

After we completed the modelling, we used the 'predict_proba' function on the model to obtain predicted probabilities. These probabilities are determined by the model and the probabilities are what the model predicts the probability of the death of each patient. After we used the 'predict_proba' function, we made a dataframe from the probabilities, labelled 'pred_probs'. To the dataframe, we added a column titled 'death' and appended 1 to the column if the probability of their death is more than 0.5, and added 0 if it was less than 0.5. Another column titled 'true' was added and this column was the 'In_hospital_death' column from 'Y_train'.

Originally, the 'In_hospital_death' column was named 'In.hospital_death'. However, while we were attempting this particular step, our code refused to work with the full stop in its name, as it read the second part as a separate function, not a part of the name. Therefore, we had to use the df.rename() function to rename the 'In.hospital_death' column to 'In_hospital_death' to prevent unnecessary confusion. This code compared the prediction to the actual survival of the patient, giving us the dataframe on the right for the model we used for Heart Rate. Our next task was to use the predictions and measure its accuracy.

	0	1	death	true
0	0.672771	0.327229	0	1
1	0.701395	0.298605	0	0
2	0.623648	0.376352	0	1
3	0.612900	0.387100	0	1
4	0.753373	0.246627	0	0
...	...	...	...	...
134	0.545107	0.454893	0	0
135	0.615660	0.384340	0	0
136	0.517382	0.482618	0	0
137	0.677912	0.322088	0	1
138	0.605012	0.394988	0	0

When we were measuring the accuracy of our model, we used the 'accuracy_score' function from the 'sklearn.metrics' library. First, we used the function on our data in 'pred_probs', the predicted probabilities for our training data. This was the code that we used to do this.

```
from sklearn.metrics import accuracy_score
accuracy_score(pred_probs['true'],pred_probs['death'])
```

Then, we conducted the same functions on our unseen data, i.e. the test data. We first repeated the production of a new dataframe with the predicted probability, predicted mortality and the actual mortality, and named it 'test_preds'. The column with the predicted death was titled 'test_death' and the actual mortality 'true'. Then repeated the 'accuracy_score' function on the test data as well.

```
test_preds = logistic.predict_proba(X_test)
test_preds = pd.DataFrame(test_preds)
test_preds['test_death'] = np.where(test_preds[1] > 0.5, 1, 0)
test_preds["true"] = Y_test.reset_index().In_hospital_death
accuracy_score(test_preds['true'],test_preds['test_death'])
```

We repeated these steps for each of our variables and the results were as follows:

Variable	Training data accuracy	Test data accuracy
Heart Rate	0.6258992805755396	0.6857142857142857
Respiratory Rate	0.6474820143884892	0.6857142857142857
SysABP	0.697841726618705	0.7142857142857143
DiasABP	0.6618705035971223	0.7142857142857143
Weight	0.6474820143884892	0.6857142857142857
Cholesterol	0.6474820143884892	0.6857142857142857
Glucose	0.6618705035971223	0.6571428571428571
Temperature	0.6690647482014388	0.6571428571428571
Urine	0.6762589928057554	0.7428571428571429

While the variables with the highest correlation were body temperature and urine output, the accuracy of the models were not necessarily reflective of the correlations. For example, the variable with the highest accuracy score was urine output, much like the order of the correlations, however, the accuracy scores that followed were Invasive Systolic and Diastolic Arterial Blood Pressures.

A multiple regression model was also attempted to observe the effect of these variables as a whole. However, due to multiple coding errors that we encountered, we decided to drop that venture.

CONCLUSION

The objective of this assignment was achieved with the sufficient external research and experimenting with the data from PhysioNet Computing in Cardiology Challenge 2012.

The research was conducted on patients in ICU Type 1, the coronary care unit, who are between 70 and 90 years of age. From the 42 variables that were recorded in the Preprocessed ICU data.csv file, we shortlisted 9 of the variables to focus on. These were chosen based on the effects each variable had on the patients' mortality rates as well as guidance from the mentor, Christoph Bergmeir. The shortlisted variables were as follows: Heart Rate, Respiratory Rate, Invasive Systolic and Diastolic Arterial Blood Pressures, Cholesterol, Glucose, Weight, Body Temperature and Urine Output Levels.

Upon conducting logistic regression on each of the chosen variables and analysing the results, we concluded that the variable that had the greatest correlation with mortality was urine output, with an exponentiated coefficient of 0.56. Following urine output was body temperature, with an exponentiated coefficient of 0.68.

However, upon assessing the accuracy of each model, we found that while urine output has a high accuracy score, body temperature does not. Rather, the models for the two blood pressures performed better in accuracy.

In conclusion, our research showed that the two most influential features on the mortality of a patient in ICU Type 1 between the ages 70 and 90 is urine output.