heart_failure_clinical_analysis

2022-09-07

Introduction

Cardiac diseases are the most common cause of death in the United States and across most of the world. Cardiovascular disorders include problems of both the heart muscle and blood vessels, which are further classified into coronary heart disease, cerebrovascular disease, and heart failure, among others. This research paper is going to focus on heart failure, lifestyle choices, and associated biomarkers.

There are two main types of heart failure based on the ejection fraction value. The ejection fraction is calculated by $EF = \frac{SV}{EDV} * 100$. Where (SV) stands for the amount of blood pumped out of the ventricle with each contraction (the stroke volume) divided by the end-diastolic volume (EDV), which is the total amount of blood in the ventricle. Normal ejection fraction levels are between 50% and 75%.

An ejection fraction smaller than 40% is known as heart failure due to reduced ejection fraction. When heart failure is present, but the ejection fraction is normal, it is known as heart failure with preserved ejection fraction. In our study, ejection fraction and serum creatine became the most important variables to predict death events. Lastly, medical data has a big problem of individual variability along unknown aspects of measuring data in heterogeneous documentation of information. Thus we had to dismiss some data instances.

Dataset Description

The data set contains 299 individual profiles along with 13 clinical features from the Faisalabad Institute of Cardiology in Punjab, Pakistan. All 299 patients had left ventricular systolic dysfunction. Some of the clinical features are binary. Some of them are not properly defined such as the high blood pressure binary feature. Kidney problems may mask or be associated with heart dysfunction, but the data set doesn’t provide any further information on it. CPK is relevant since, when a muscle tissue gets damaged, CPK gets into the blood which may indicate damage to the heart muscle. High levels of serum creatine may indicate kidney problems, which tend to be associated with heart problems. Low levels of sodium in the body may be an indication of heart failure. Some data points were curated away, such as the 45 year old person with heart problems and an ejection fraction of 80%. Which is an indication of a possible hypertrophic cardiomyopathy problem, a disease that we do not seek to investigate in this study.

Research Questions

Given the 13 clinical features that we contain in the data set, we sought to model the best predictors of heart failure. In turn, we also sought the best indicators that could ultimately lead to early death. Ultimately, what we want to know is whether we can use some biomarkers associated with low ejection fraction. Any strong association of ejection fraction with death is something this paper intends to discover. It would be convenient, besides ejection fraction, to seek any other biomarkers that have a strong prediction for death. Thus, the following questions we want to ask are:

(1) How does each variable correlate with Heart failure, or what is the relationship between each of these variables and a death event, which of these variables are associated with heart failure?

(2) If there are strong enough relationships between these variables, low ejection fractions, and death events, can we build a model to describe the relationships at play?

(3) Additionally, which variables are the most relevant in association with death events?

Feature definition

(The numeric variables are colored in blue and categorical variables are colored in orange) Age: In years, the age of the patient. Integer

Anemia: The decrease of red blood cells or hemoglobin. A factor with levels No (0) and Yes (1)

High blood pressure (HBP): If a patient has hypertension. A factor with levels No (0) and Yes (1)

Creatine phosphokinase (CPK): The level of the CPK enzyme in the blood (mcg/L). Integer

Diabetes: Presence of diabetes in the patient; no distinction between Type I and Type II. A factor with two levels No (0) and Yes (1)

Ejection fraction (EF):Percentage of blood leaving the heart at each contraction. Numeric

Platelets: Measurement of the number of platelets in the blood (kiloplatelets/mL). Numeric

Sex: Woman or man. A factor with two levels Woman (0) Man (1)

Serum creatinine: The level of serum creatinine in the blood (mg/dL). Numeric

Serum sodium: the level of serum sodium in the blood (mEq/L). Integer

Smoking: If a patient smokes. A factor with two levels NoSmoke (0) YesSmoke (1)

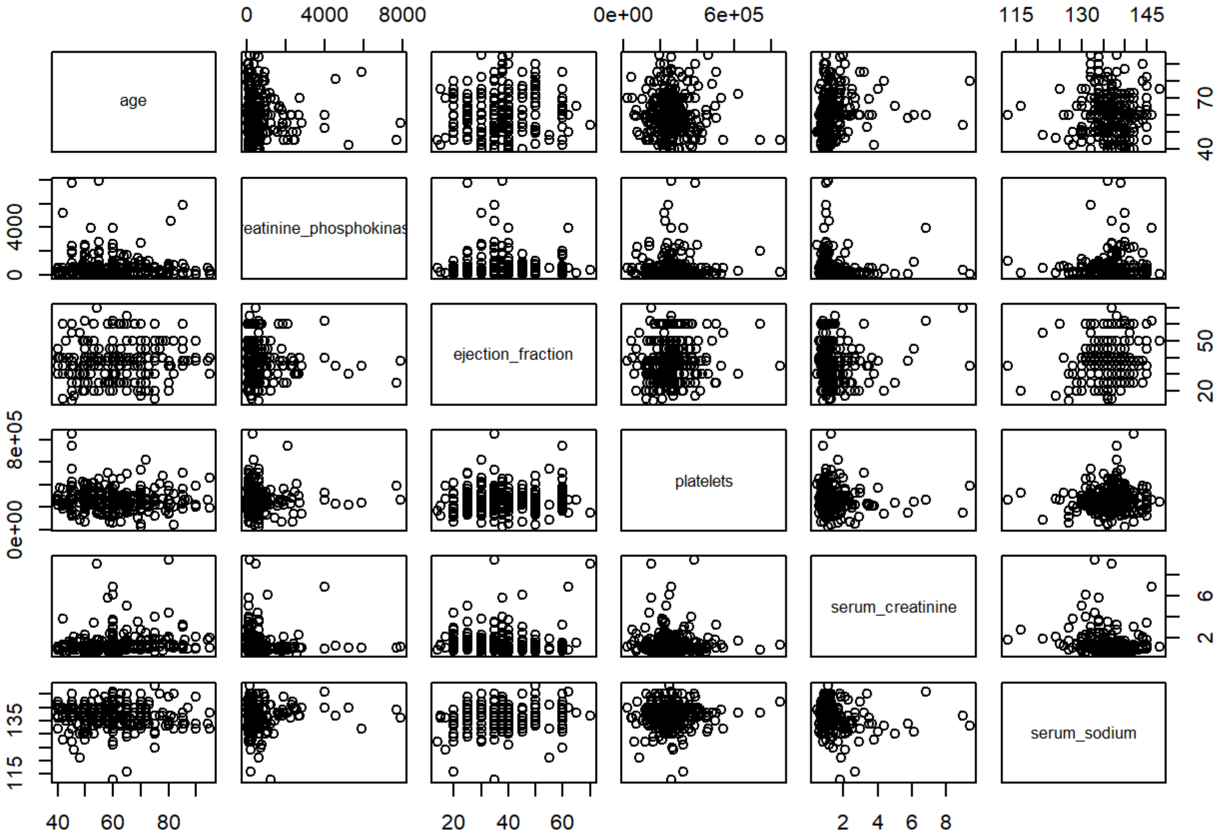
Time: In days, a patient's follow-up period length. Integer

Death event: If the patient deceased during the follow-up period. A factor with two levels Not-Deceased (0) Deceased (1)

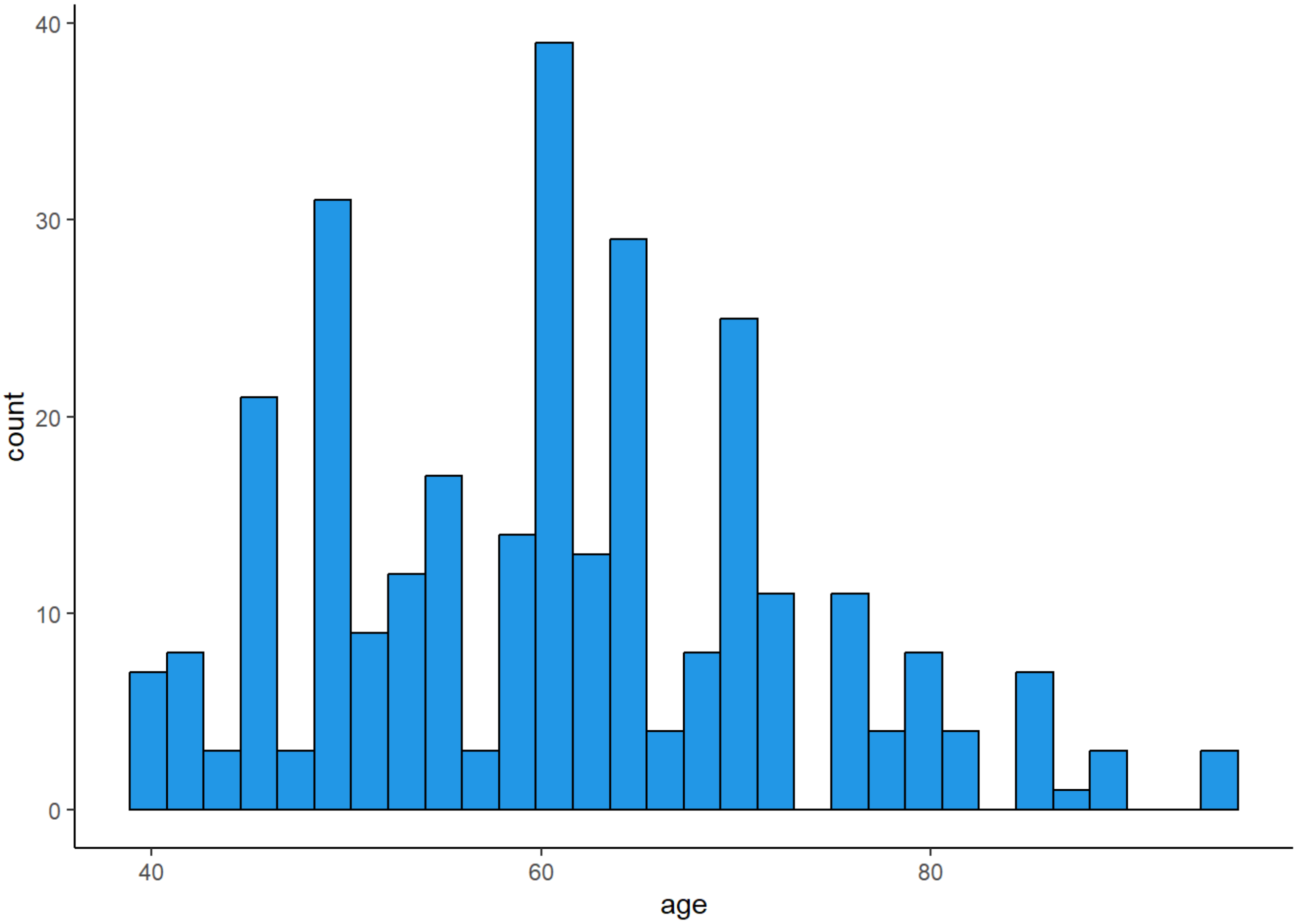
Importing data

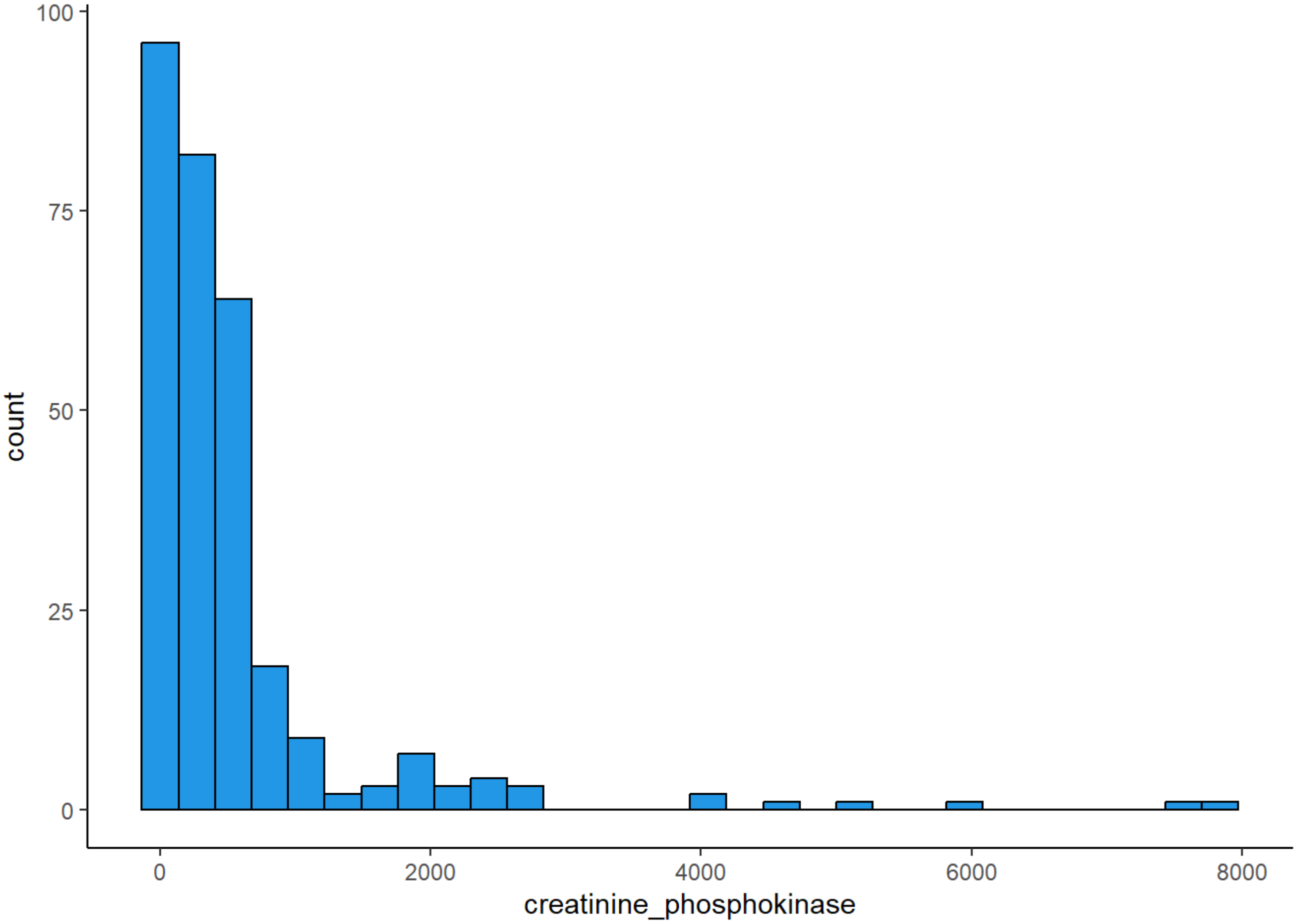
Data exploration - Summary of numerical data

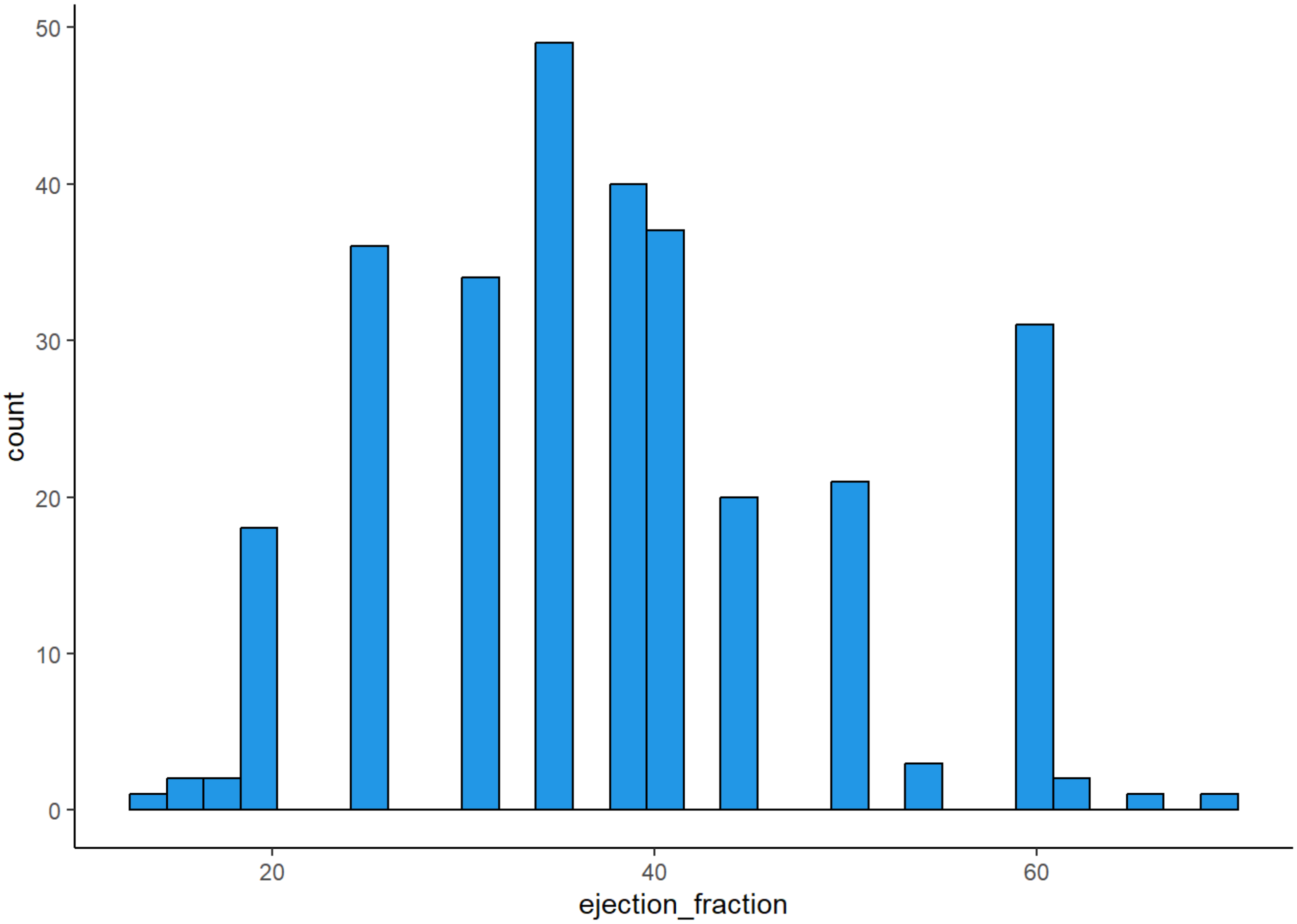
Correlation between numerical variables

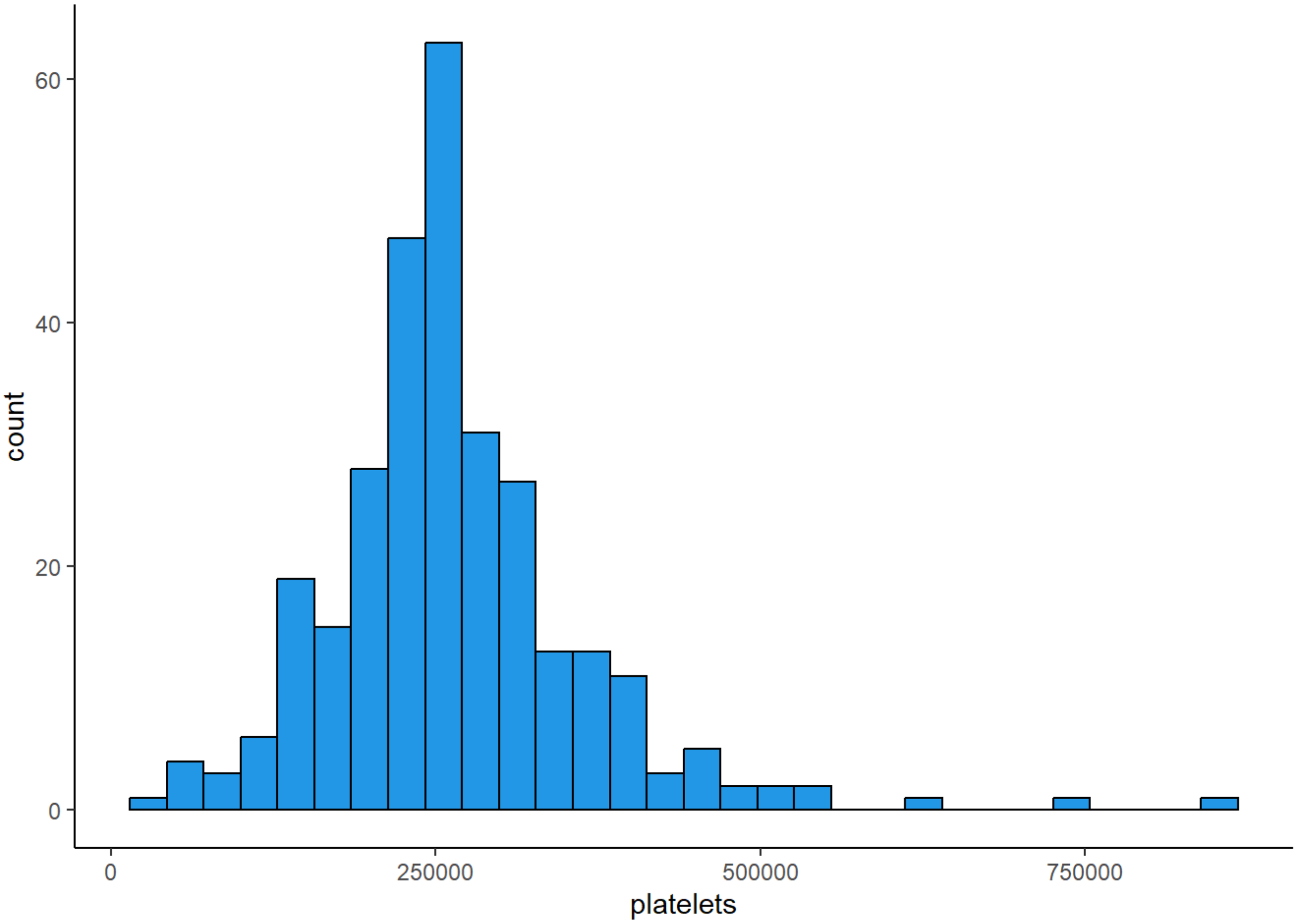


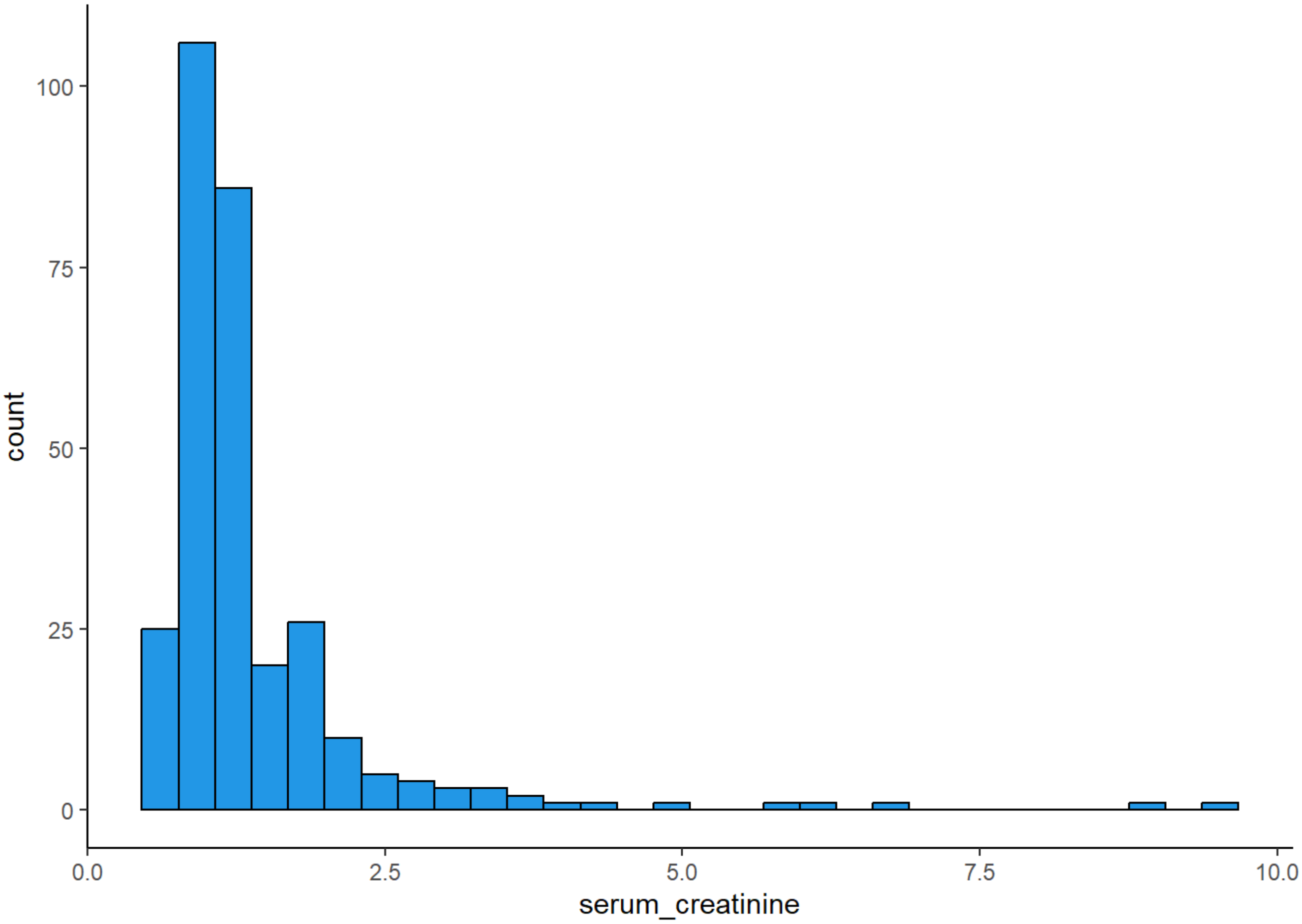
checking distribution of variables

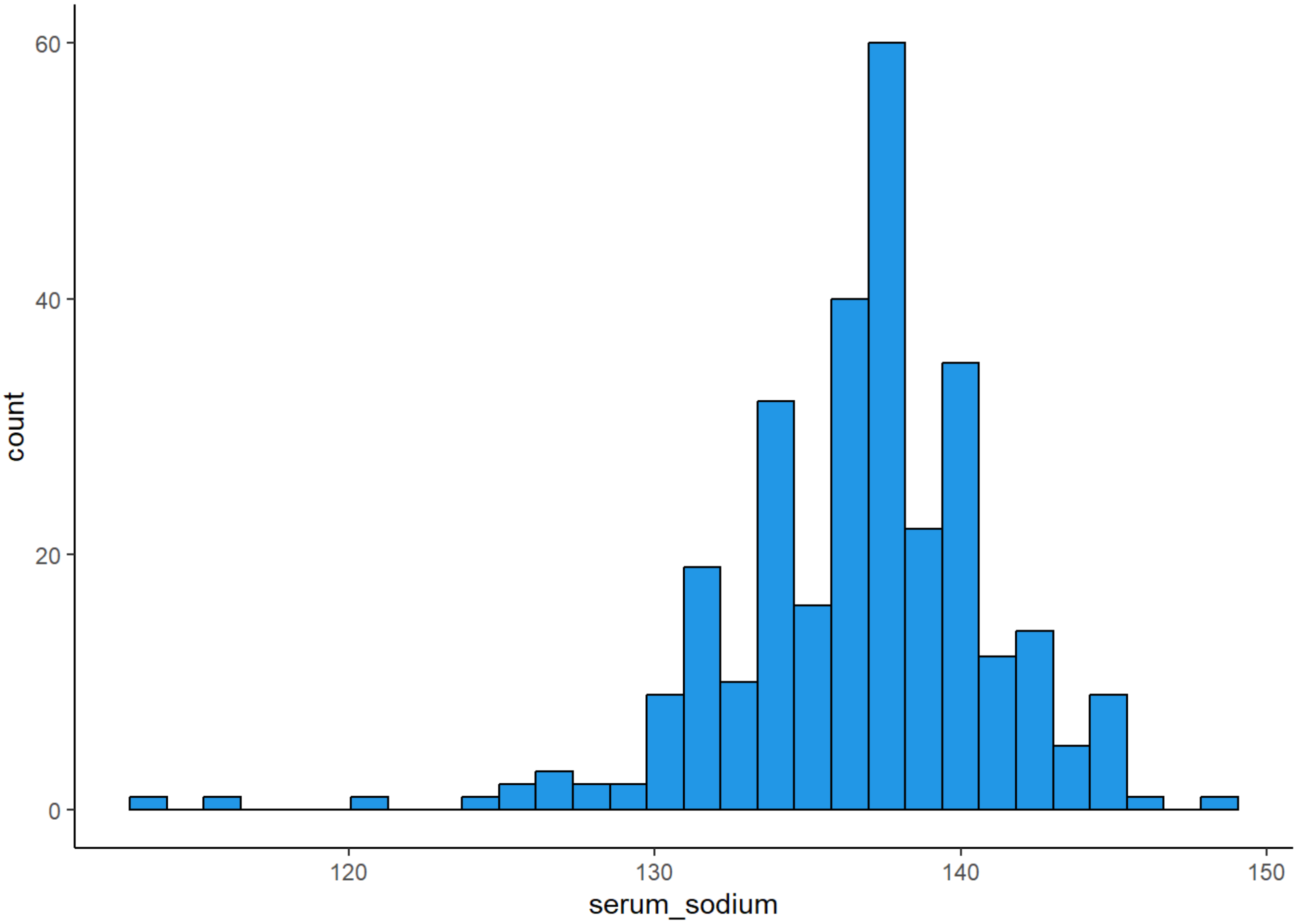

40
30
20
10
0
count
40
60
80
age


100
75
50
25
0
count
0
2000
4000
6000
8000
creatinine_phosphokinase


count
50
40
30
20
10
0
20
40
60
ejection_fraction

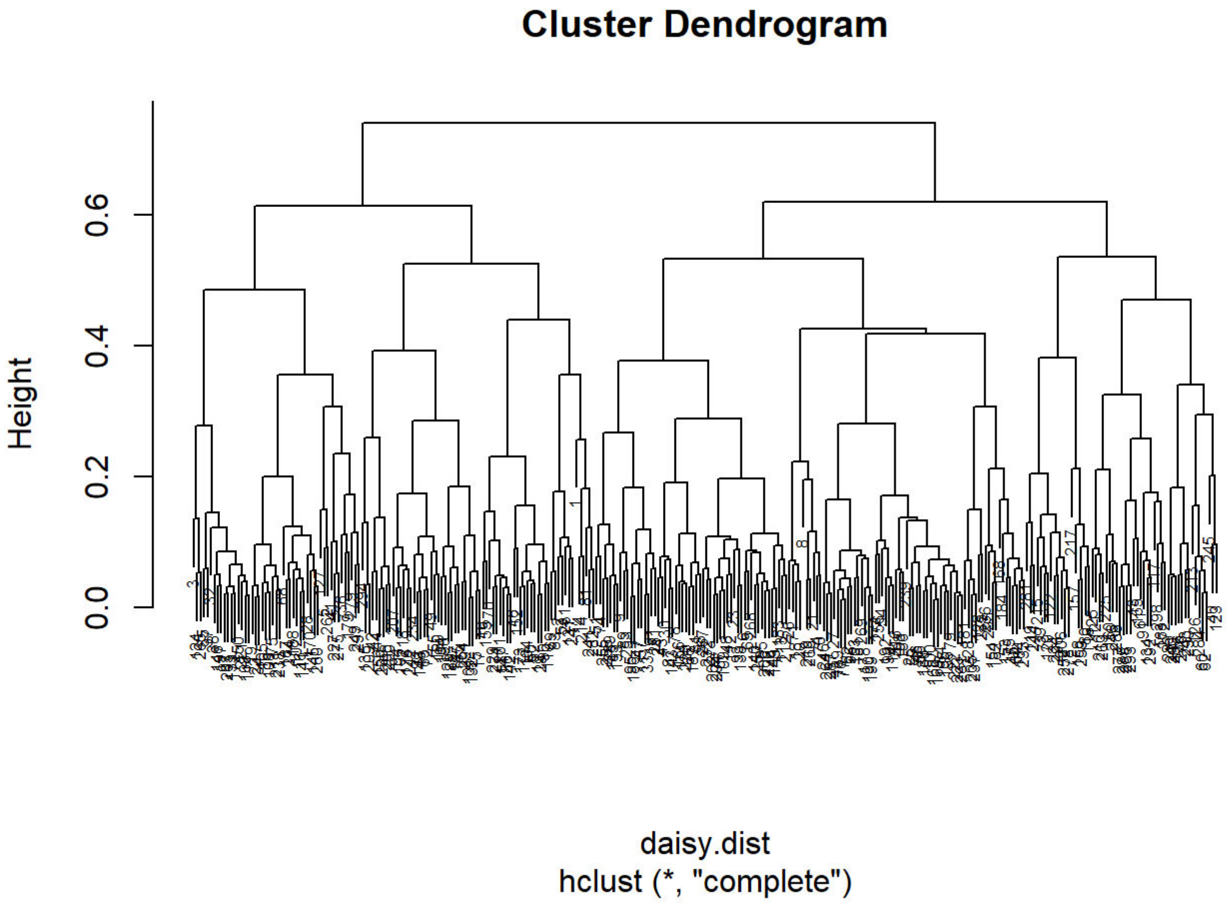

count
60
40
20
0
0
250000
500000
750000
platelets


count
100
75
50
25
0
0.0
2.5
5.0
7.5
10.0
serum_creatinine

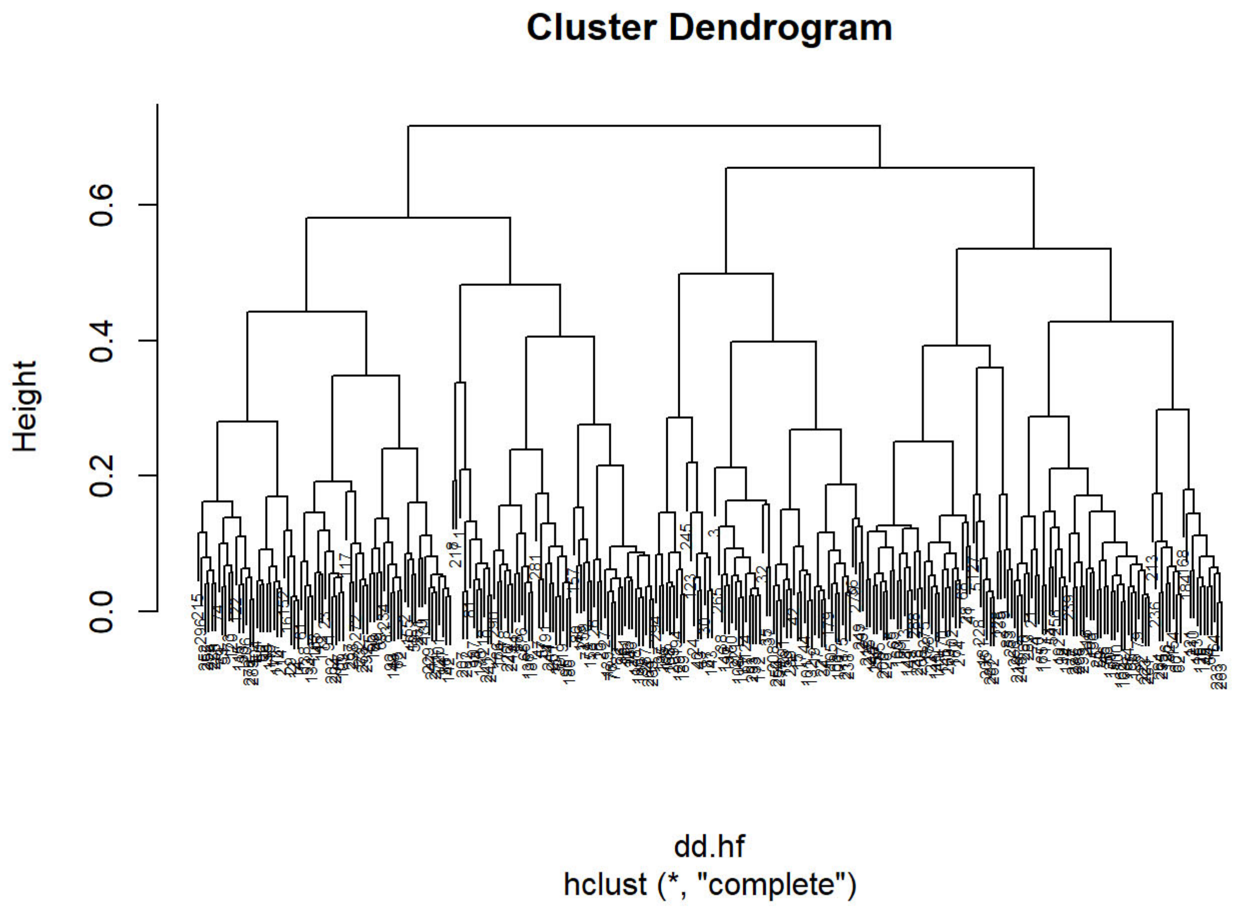

count
60
40
20
0
120
130
140
150
serum_sodium

Hierarchical Clustering

```
## Warning in daisy(Heart.Failure, metric = "gower"): binary variable(s) 2, 4, 6,
## 10, 11, 12 treated as interval scaled
```



```
## Warning in daisy(hf, metric = "gower"): binary variable(s) 2, 4, 6, 10, 11
## treated as interval scaled
```



Cutting tree into two main clusters and add column of prediction to dataframe

##explore two main clusters (alive and death)

Actual data

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   17.00   35.00   38.00   40.07   45.00   62.00
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   14.00   25.00   30.00   33.47   38.00   70.00
```

```
## [1] 73.01
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.500   0.900   1.000   1.185   1.200   6.100
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.600   1.075   1.300   1.836   1.900   9.400
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   25100  219250  262500  266674  302000  850000
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   47000  197500  258500  256381  311000  621000
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    30.0   109.0   244.5   539.8   582.0  5209.0
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    23.0   128.8   259.0   670.2   582.0  7861.0
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   113.0   135.2   137.0   137.2   140.0   148.0
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   116.0   133.0   135.5   135.4   138.2   146.0
```

Using hierarchical clustering data

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   14.00   30.00   35.00   36.75   40.00   62.00
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   15.00   30.00   38.00   39.79   45.00   70.00
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.600   0.900   1.100   1.375   1.400   9.400
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.500   0.900   1.100   1.424   1.400   9.000
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   25100  200000  254000  255646  298000  850000
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   62000  223000  263358  275289  318000  742000
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##      23     115     253     654     582    7861
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    52.0   121.0   244.0   470.3   582.0  3964.0
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   113.0   134.0   137.0   136.6   139.0   148.0
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   116.0   134.0   137.0   136.6   140.0   146.0
```

EF: both actual and clustered data showed that patients who did not die had a larger minimun EF than the patients that died

serum creatinine: both actual and clustered data showed that patients who did not die averaged a larger number of platelets than the patients that died

platelets: both actual and clustered data showed that patients who did not die averaged a larger number of platelets than the patients that died

creatinine phosphokinase: both actual and clustered data showed that patients who did not die averaged a smaller number of creatinine phosphokinase than the patients that died

serum sodium: neither data showed a that patients who did not die averaged a smaller number of creatinine phosphokinase than the patients that died

Confusion Matrix: Compare classification and actual data

```
## [1] 202
```

```
## [1] 96
```

```
## [1] 181
```

```
## [1] 117
```

```
## function (x)  .Primitive("names")
```

```
## [1] 6.853583
```

```
## [1] 93.14642
```

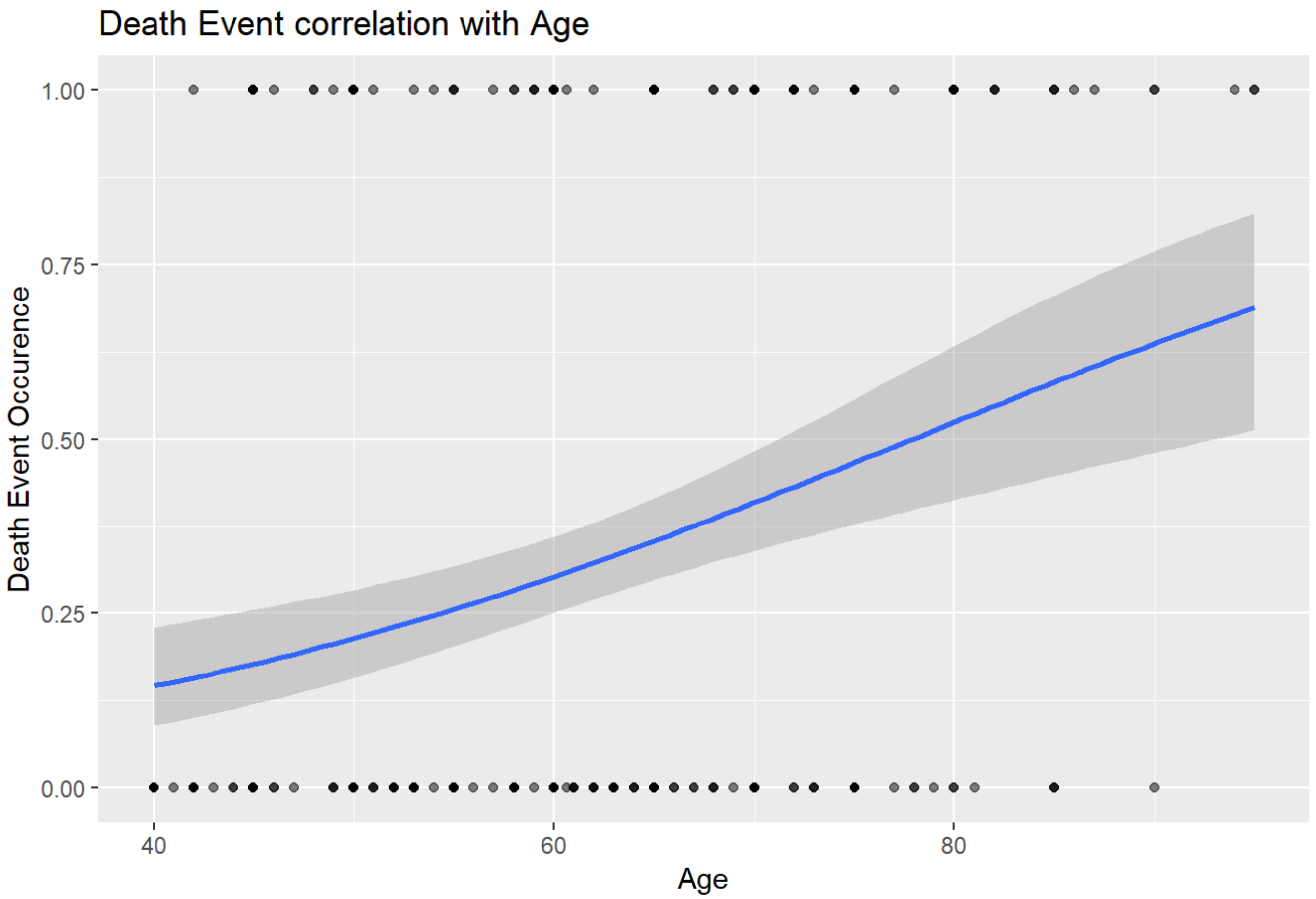
```
## confusion.matrx
##   0  22  96 203 
##   1   1   1   1
```

Data Preview Visualisation

The main objective of this analysis is to find predictors of death occurrences when dealing with heart failure. A logistical regression developed a model with coefficients. For this reason it was valuable to plot individual variables against death occurrences to see the general trends between them.

Firstly, an obvious prediction is that as a patient’s age increases, so do their chances of experiencing a death event. This is confirmed by the general linear model graph of death event correlation with age.

```
## `geom_smooth()` using formula = 'y ~ x'
```

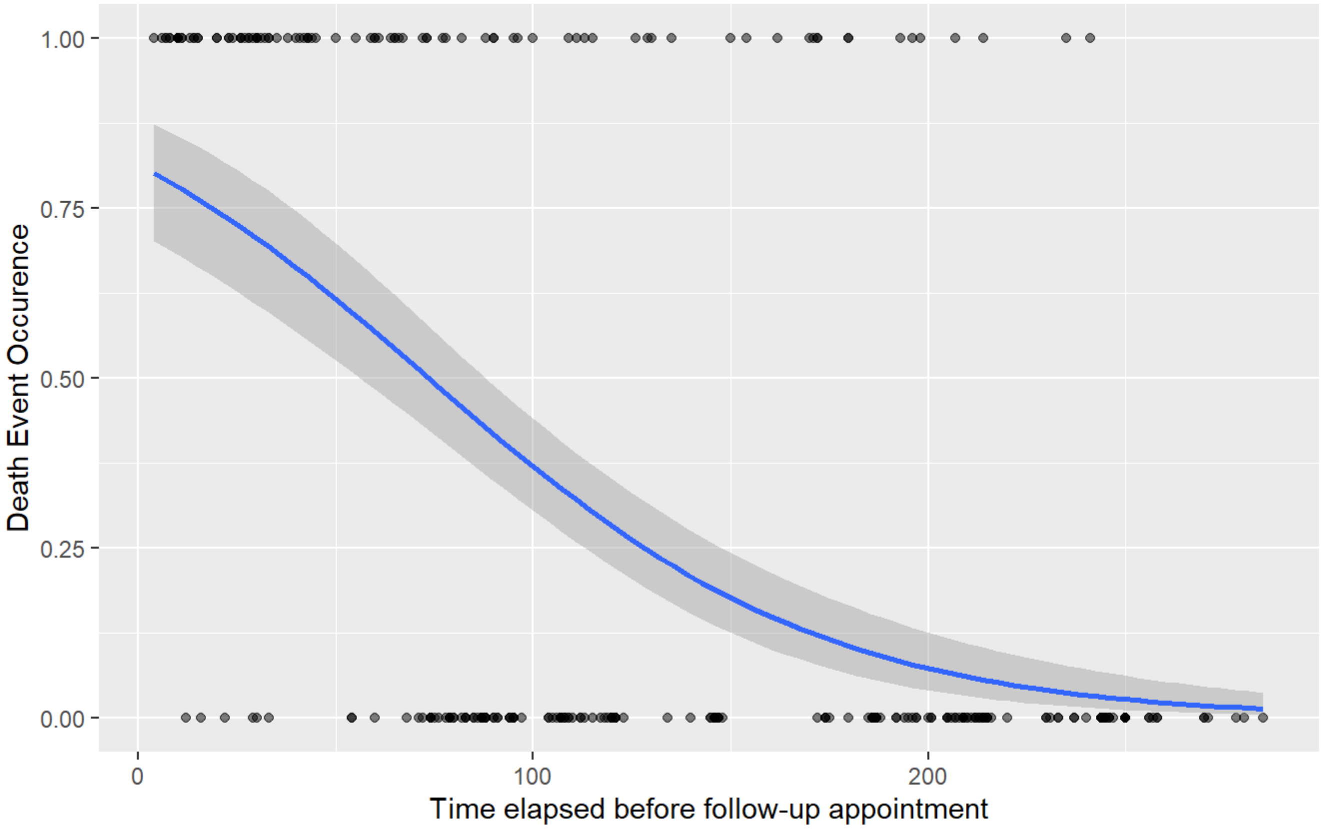


As expected, there is strong correlation between age and death events.

A second obvious correlation is time elapsed before a follow up appointment. As expected, as a patient’s condition is less severe, their follow up appointment can be postponed to a later date as seen in the graph below.

```
## `geom_smooth()` using formula = 'y ~ x'
```

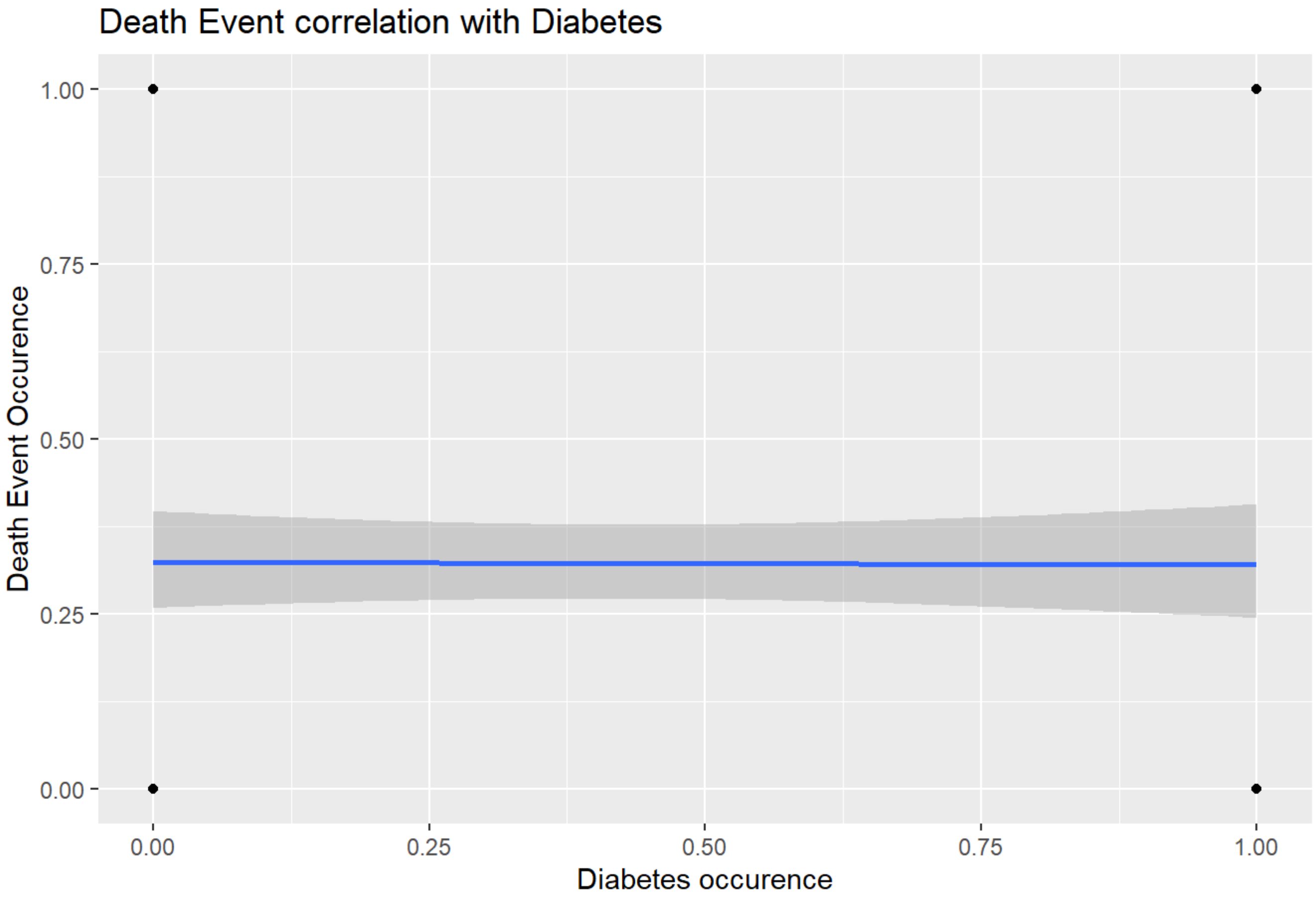
Death Event correlation with Time Elapsed Before Follow-up Appointment



As a patient lives longer, the time for a follow-up can increase. For this reason this is likely causation rather than correlation. Furthermore, a case's severity would warrant a sooner follow-up.

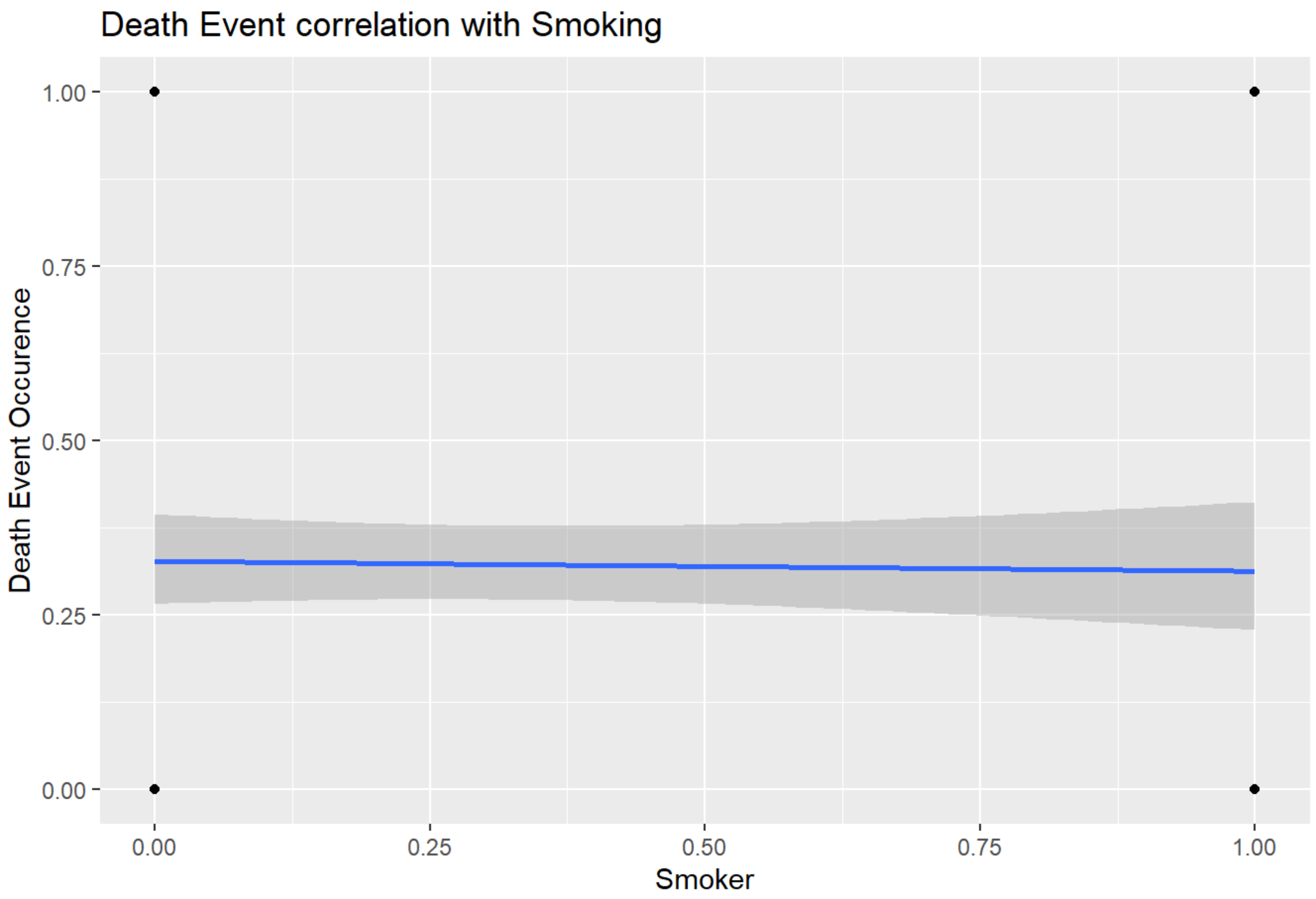
Two very interesting observations of coefficients and correlations were during the comparison of diabetes with fatal heart failure, as well as smoking. Both seemed to have little to no effect on fatality of heart failure. In fact, patients who smoked had an insignificant decrease in fatality of heart failure.

```
## `geom_smooth()` using formula = 'y ~ x'
```



Diabetes interestingly is not a strong predictor of death events due to heart failure

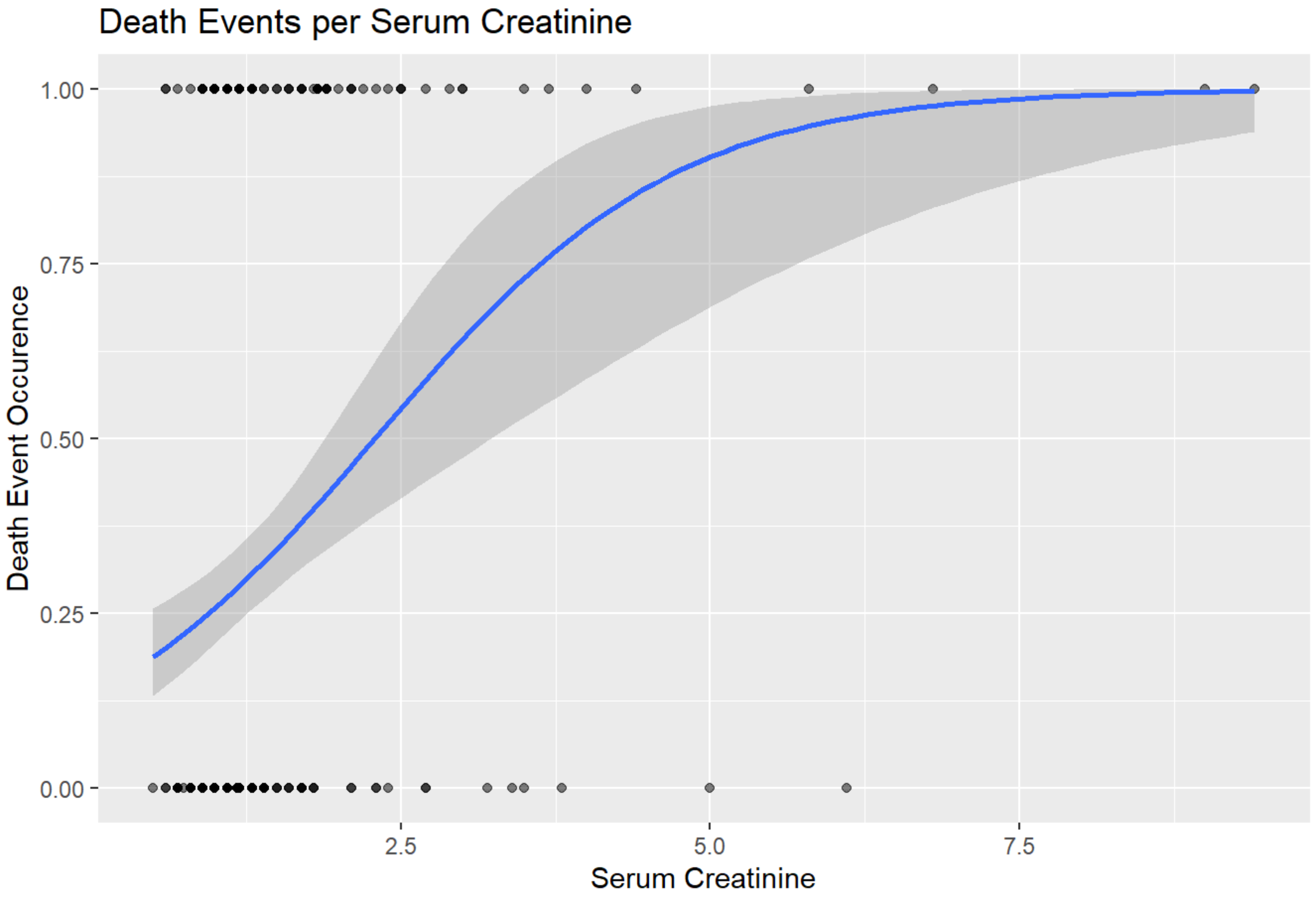
```
## `geom_smooth()` using formula = 'y ~ x'
```



Smoking apparently has a positive impact in preventing death occurences, although minimal.

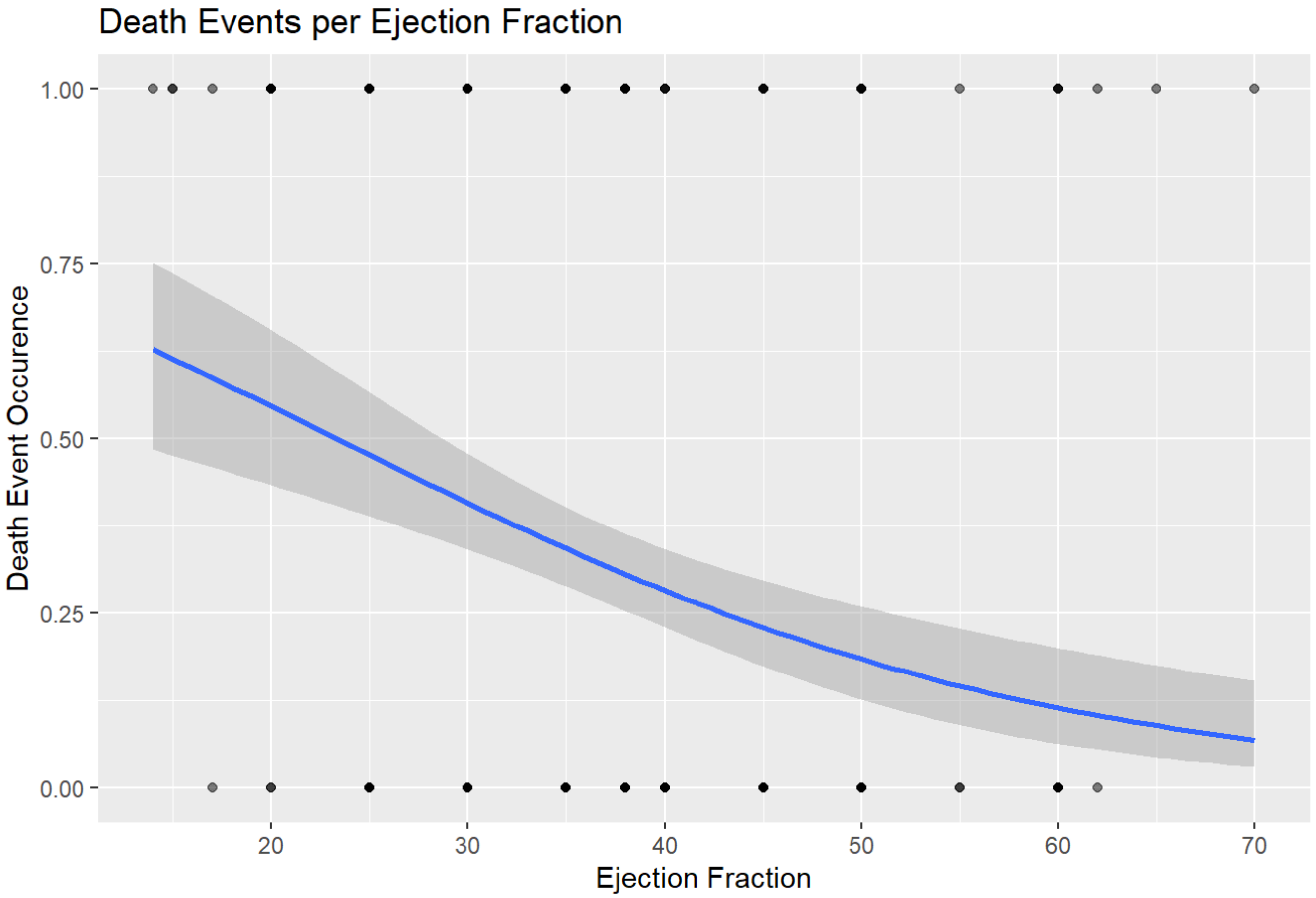
Finally, two most significant variables, serum creatinine and ejection fraction, both provided a reliable basis for predicting fatality of heart failure.

```
## `geom_smooth()` using formula = 'y ~ x'
```



As blood serum creatinine increases, so does a patient's likelihood of a death event.

```
## `geom_smooth()` using formula = 'y ~ x'
```



Ejection Fraction has an inverse relation with the chances of a death event

Logistic Regression

After previewing the data in the plots above, a logistic regression was ran in order to create a logistical model. Logistic regressions have the form of $e^{b_0+b_1X_1+...+b_nX_n}$ whose probability is between 0 and 1. The data was subset in an 80:20 training:testing sampling proportionally to the population in terms of dead to living. This left 19 death events and 40 survivals. A logistical regression was then carried out and a logistical model was produced.

```
##
## Call:
## glm(formula = DEATH_EVENT ~ ., family = binomial, data = data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2835  -0.4991  -0.1750   0.4099   2.0698
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               1.572e+01  7.193e+00   2.186  0.02885 *
## age                       4.825e-02  1.846e-02   2.613  0.00897 **
## anaemia                  -1.974e-01  4.393e-01  -0.449  0.65308
## creatinine_phosphokinase  2.486e-04  1.970e-04   1.262  0.20693
## diabetes                  1.506e-01  4.231e-01   0.356  0.72179
## ejection_fraction        -1.711e-01  3.226e-02  -5.304 1.13e-07 ***
## high_blood_pressure      -4.005e-02  4.459e-01  -0.090  0.92843
## platelets                -4.087e-07  2.288e-06  -0.179  0.85823
## serum_creatinine          4.935e-01  2.364e-01   2.087  0.03686 *
## serum_sodium             -8.771e-02  4.888e-02  -1.794  0.07276 .
## sex                      -2.915e-01  4.890e-01  -0.596  0.55104
## smoking                  -1.608e-01  4.688e-01  -0.343  0.73153
## time                     -2.106e-02  3.664e-03  -5.749 8.99e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 300.42  on 238  degrees of freedom
## Residual deviance: 158.77  on 226  degrees of freedom
## AIC: 184.77
##
## Number of Fisher Scoring iterations: 6
```

The logistic model is as follows:

A confusion matrix and misclassification rate were then calculated for the test data.

```
## [1] "Confusion Matrix"
```

```
##                       True Death Events
## Predicted Death Events  0  1
##                  ALIVE 40 18
##                  DEAD   0  1
```

```
## [1] "Misclassification Error Rate"
```

```
## [1] 0.3050847
```

Another logistic regression using the two most significant variables, serum creatinine and ejection fraction, was run. This produced an even more accurate model. An anova was then run on both models, indicating that using just serum creatinine and ejection fraction as predictors provides a much more robust and precise model. The model is as follows:

Possible BEST MODEL EVAH

```
## 
## Call:
## glm(formula = DEATH_EVENT ~ ejection_fraction + serum_creatinine, 
##     family = binomial, data = data_train)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.4282  -0.6801  -0.4641   0.7693   2.1384  
## 
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)    
## (Intercept)        2.95787    0.79697   3.711 0.000206 ***
## ejection_fraction -0.14641    0.02389  -6.129 8.87e-10 ***
## serum_creatinine   0.71931    0.20240   3.554 0.000379 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 300.42  on 238  degrees of freedom
## Residual deviance: 232.82  on 236  degrees of freedom
## AIC: 238.82
## 
## Number of Fisher Scoring iterations: 4
```

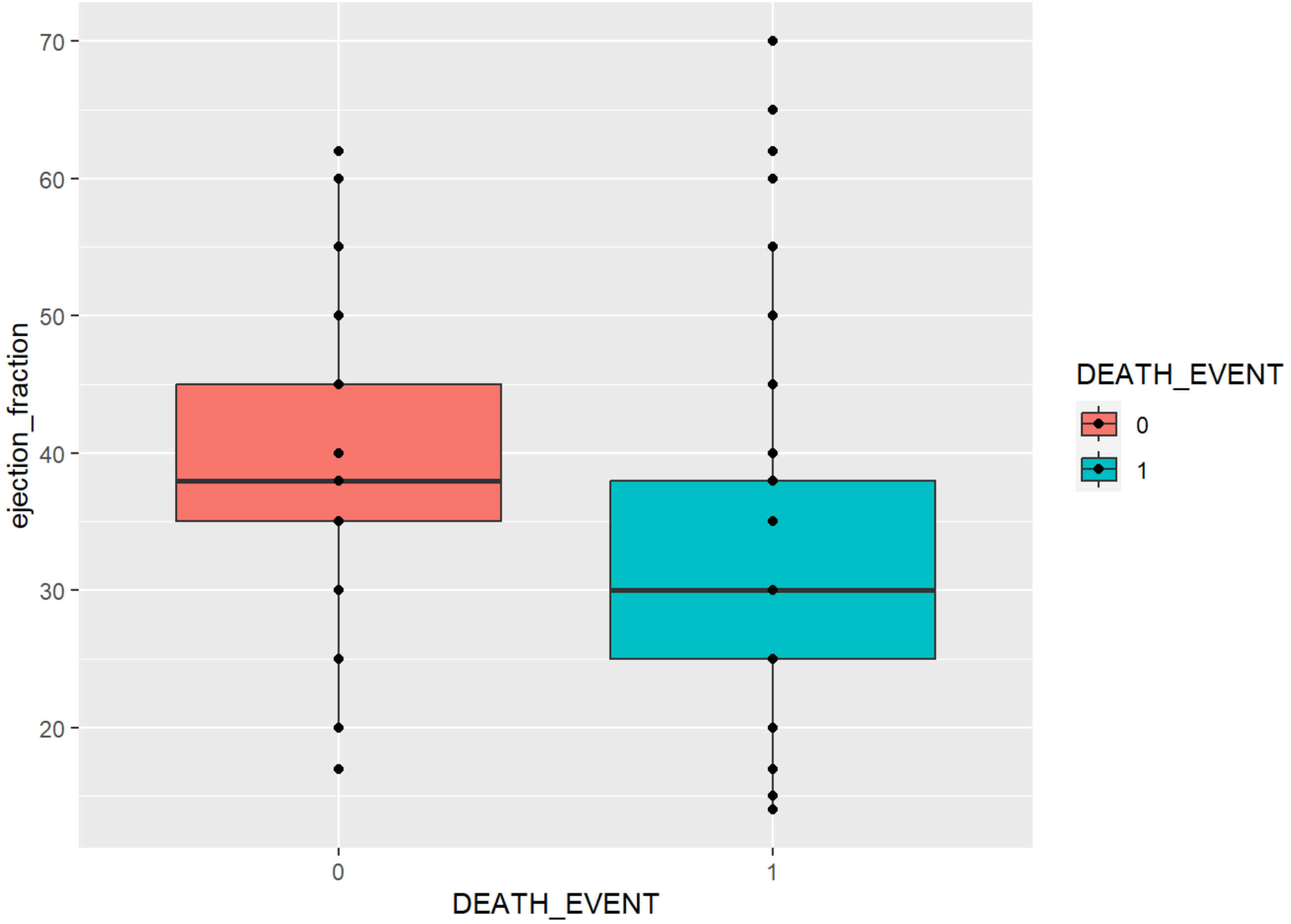
```
##                       True Death Events
## Predicted Death Events  0  1
##                  ALIVE 40 19
```

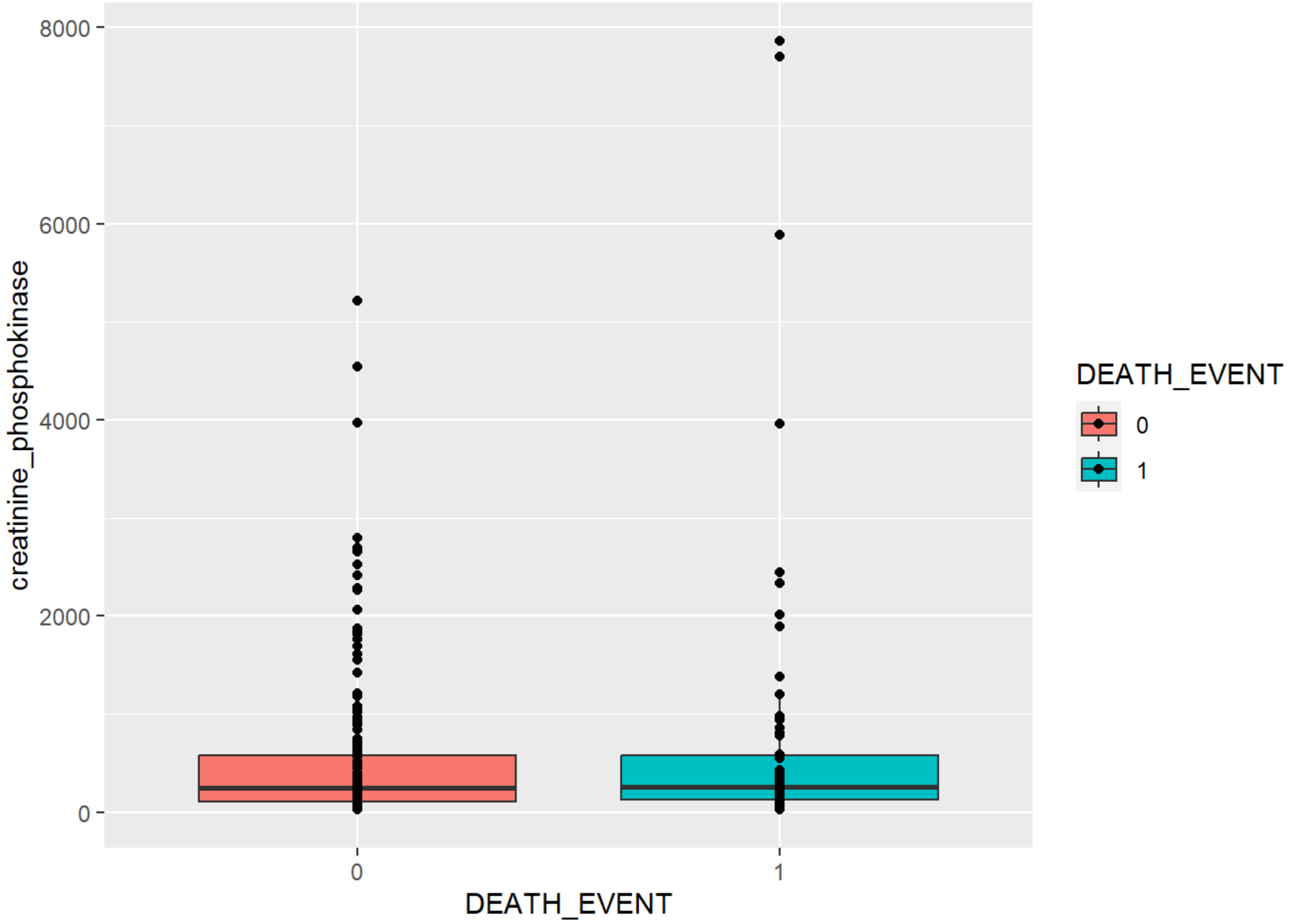
```
## [1] "Misclassification Error Rate"
```

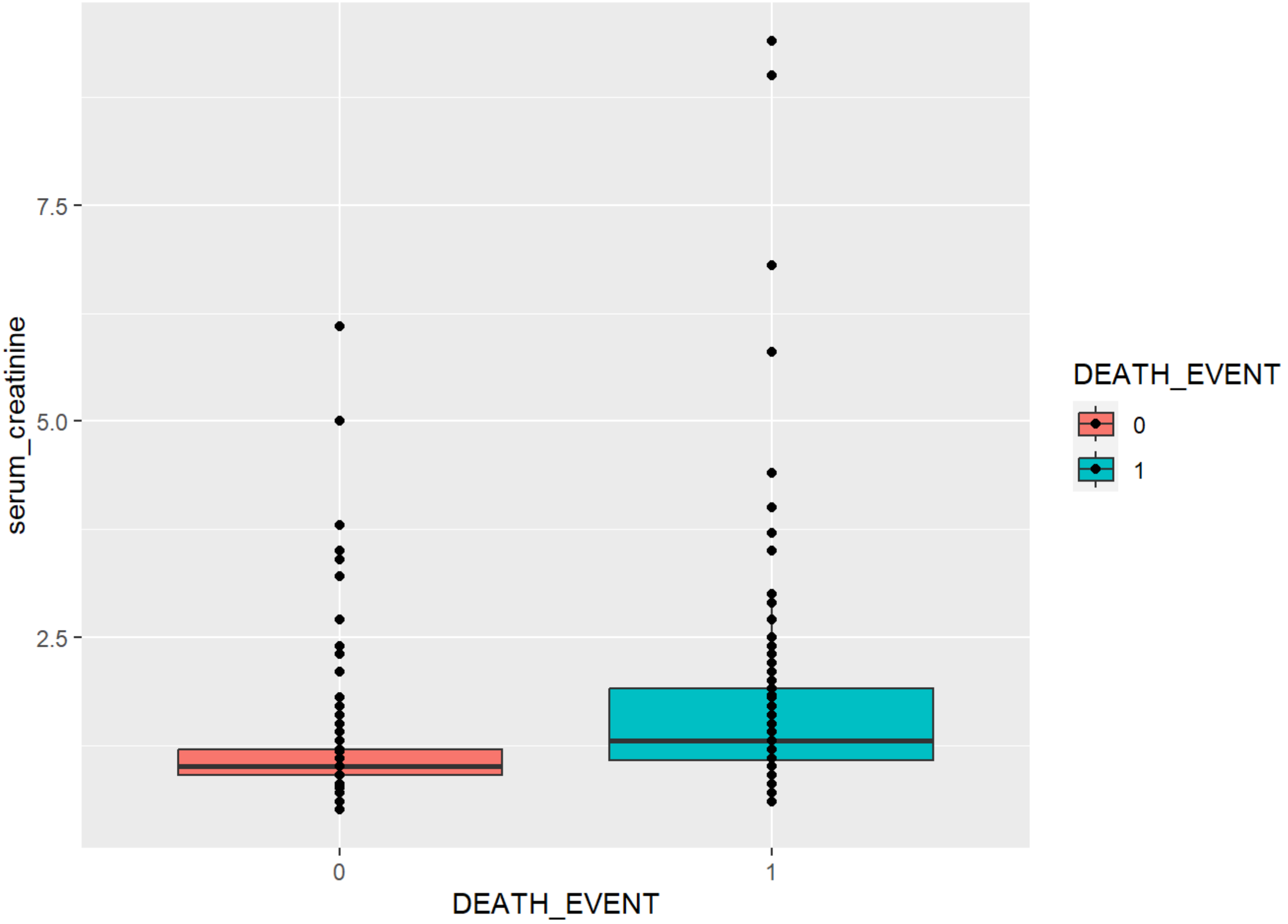
```
## [1] 0.3220339
## Analysis of Deviance Table
##
## Model 1: DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + diabetes +
##     ejection_fraction + high_blood_pressure + platelets + serum_creatinine +
##     serum_sodium + sex + smoking + time
## Model 2: DEATH_EVENT ~ ejection_fraction + serum_creatinine
##   Resid. Df Resid. Dev  Df Deviance Pr(>Chi)
## 1       226     158.77
## 2       236     232.82 -10   -74.05 7.28e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

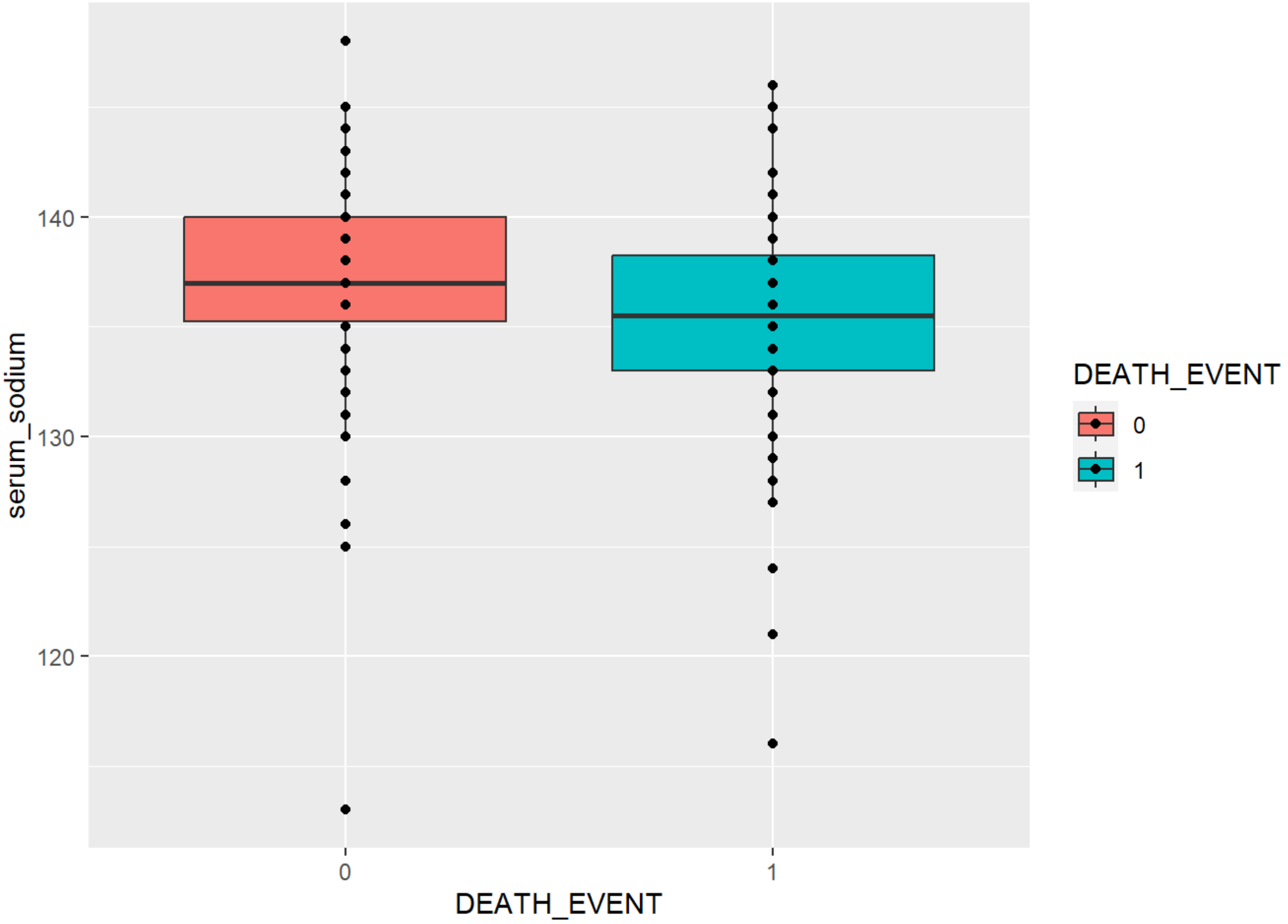
```
## Analysis of Deviance Table
##
## Model 1: DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + diabetes +
##     ejection_fraction + high_blood_pressure + platelets + serum_creatinine +
##     serum_sodium + sex + smoking + time
## Model 2: DEATH_EVENT ~ ejection_fraction + serum_creatinine
##   Resid. Df Resid. Dev  Df Deviance
## 1       226     158.77
## 2       236     232.82 -10   -74.05
```

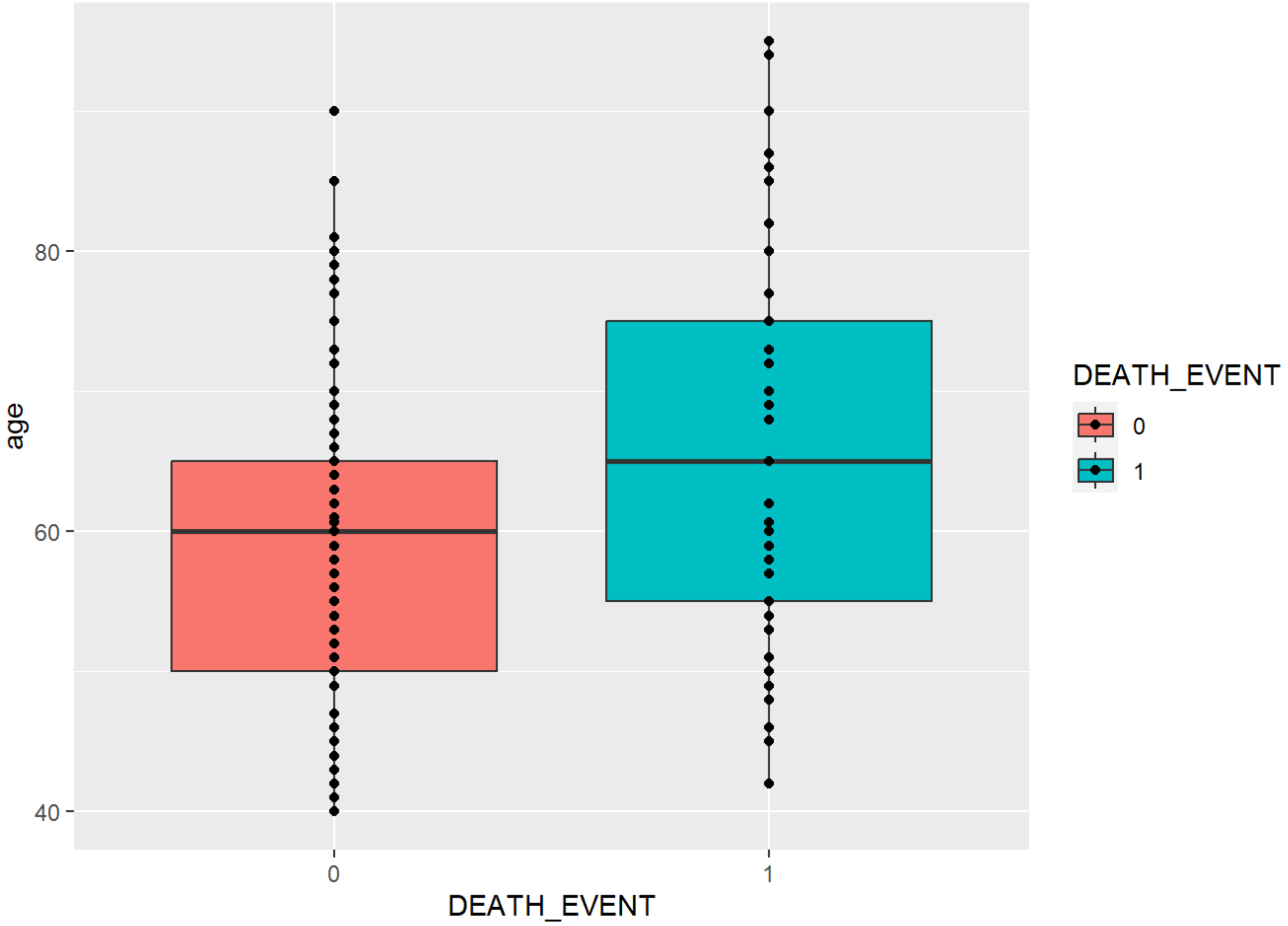
Data Preview Boxplot


70
60
50
40
30
20
ejection_fraction
0
1
DEATH_EVENT
DEATH_EVENT
0
1


8000
6000
4000
2000
0
creatinine_phosphokinase
0
1
DEATH_EVENT
DEATH_EVENT
0
1


serum_creatinine
7.5
5.0
2.5
0
1
DEATH_EVENT
DEATH_EVENT
0
1


serum_sodium
140
130
120
0
1
DEATH_EVENT
DEATH_EVENT
0
1



#LDA Part1

```
## Call:
## lda(DEATH_EVENT ~ ., data = heartdata_train)
##
## Prior probabilities of groups:
##         0         1
## 0.6778243 0.3221757
##
## Group means:
##        age  anaemia1 creatinine_phosphokinase diabetes1 ejection_fraction
## 0 58.24486 0.3888889                 581.8272 0.4259259          35.81481
## 1 64.22944 0.4675325                 709.4545 0.3896104          28.51948
##   high_blood_pressure1 platelets serum_creatinine serum_sodium      sex1
## 0            0.3024691  262796.5         1.199321     137.1049 0.6728395
## 1            0.4155844  259348.9         1.774545     135.0909 0.6883117
##    smoking1
## 0 0.3395062
## 1 0.3376623
##
## Coefficients of linear discriminants:
##                                    LD1
## age                       3.708173e-02
## anaemia1                  1.935956e-01
## creatinine_phosphokinase  2.367766e-04
## diabetes1                -9.221764e-02
## ejection_fraction        -1.186200e-01
## high_blood_pressure1      3.613816e-01
## platelets                 4.710628e-07
## serum_creatinine          4.356034e-01
## serum_sodium             -3.269267e-02
## sex1                     -1.562860e-01
## smoking1                  5.671355e-02
```

```
##
##      0  1
##   0 40  0
##   1 19  0
```

```
## [1] 0.6779661
## Call:
## lda(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = heartdata_train)
##
## Prior probabilities of groups:
##         0         1
## 0.6778243 0.3221757
##
## Group means:
##        age ejection_fraction serum_creatinine
## 0 58.24486          35.81481         1.199321
## 1 64.22944          28.51948         1.774545
##
## Coefficients of linear discriminants:
##                           LD1
## age                0.03953177
## ejection_fraction -0.12475696
## serum_creatinine   0.45240271
```

```
##
##      0  1
##   0 40  0
##   1 19  0
```

```
## [1] 0.6779661
```

```
## Call:
## lda(DEATH_EVENT ~ ejection_fraction + serum_creatinine, data = heartdata_train)
##
## Prior probabilities of groups:
##         0         1
## 0.6778243 0.3221757
##
## Group means:
##   ejection_fraction serum_creatinine
## 0          35.81481         1.199321
## 1          28.51948         1.774545
##
## Coefficients of linear discriminants:
##                          LD1
## ejection_fraction -0.1274683
## serum_creatinine   0.5771350
```

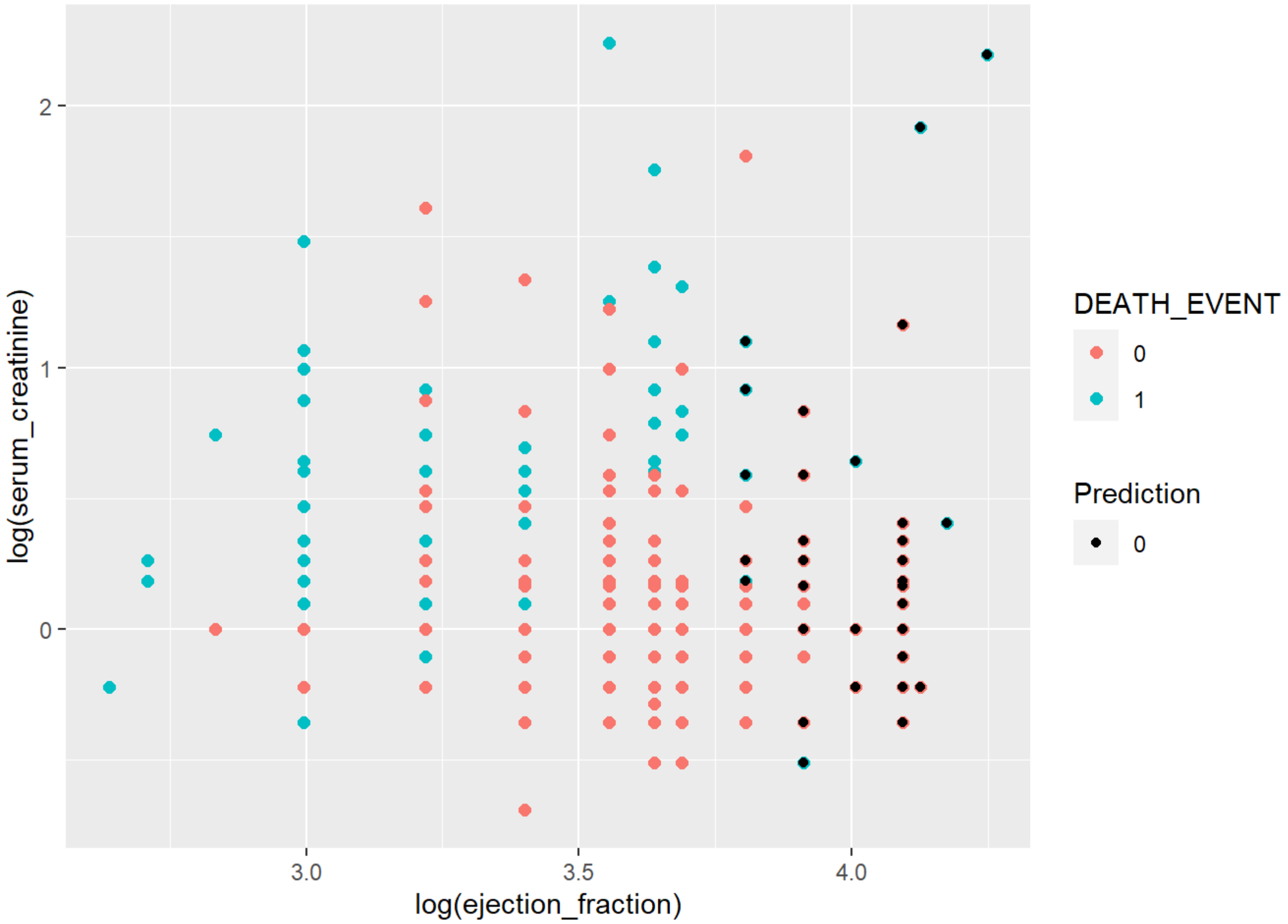
```
##
##      0  1
##   0 40  0
##   1 19  0
## [1] 0.6779661
```

#LDA Part2

```
## 
##      0  1
##   0 40 19
##   1  0  0
```

```
## [1] 0.6779661
```

```
## Call:
## lda(DEATH_EVENT ~ ejection_fraction + serum_creatinine, data = data_lda_visual, 
##     cv = TRUE)
## 
## Prior probabilities of groups:
##         0         1 
## 0.6778243 0.3221757 
## 
## Group means:
##   ejection_fraction serum_creatinine
## 0          35.81481         1.199321
## 1          28.51948         1.774545
## 
## Coefficients of linear discriminants:
##                          LD1
## ejection_fraction -0.1274683
## serum_creatinine   0.5771350
```



```
## Call:
## lda(DEATH_EVENT ~ ejection_fraction + serum_creatinine, data = data_lda_visual,
##     cv = TRUE)
##
## Prior probabilities of groups:
##         0         1
## 0.6778243 0.3221757
##
## Group means:
##   ejection_fraction serum_creatinine
## 0          35.81481         1.199321
## 1          28.51948         1.774545
##
## Coefficients of linear discriminants:
##                          LD1
## ejection_fraction -0.1274683
## serum_creatinine   0.5771350
```

K-fold Validation for LDA model:

```
## Linear Discriminant Analysis
##
## 298 samples
##   3 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 239, 238, 239, 238, 238
## Resampling results:
##
##   Accuracy   Kappa
##   0.7551412  0.3755989
```

```
## Linear Discriminant Analysis
##
## 298 samples
##   2 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 239, 238, 239, 239, 237
## Resampling results:
##
##   Accuracy   Kappa
##   0.7418311  0.3037359
```

```
## Linear Discriminant Analysis
##
## 298 samples
##  11 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 238, 238, 239, 239, 238
## Resampling results:
##
##   Accuracy   Kappa
##   0.7450282  0.359463
```

```
knitr::opts_chunk$set(echo = FALSE)
require(MASS)
library(readr)
heart_failure_clinical_records_dataset <- read.csv(file.choose(), header=T)
heartdata = heart_failure_clinical_records_dataset
require(ggplot2)
library(caret)
library(lattice)
library(sampling)
library(tidyverse)
library(GGally)
library(scatterplot3d)
library(ROCR)
library(cluster)

Heart.Failure <- read.csv(file.choose(), header=T)
Heart.Failure<- Heart.Failure[-12] # remove time

hf_numerical <- Heart.Failure[,-c(2,4,6,10,11,12)] # only numerical variables
sum_hf <- summary(hf_numerical)
# correlation
cor_matrix <- abs(cor(hf_numerical))


plot(hf_numerical)
# check distribution of data by plotting histograms (one variable)
require(ggplot2)
p<-ggplot(data = Heart.Failure) + theme_bw() +
  theme(panel.border = element_blank(), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"))

# histograms to check distribution
p+geom_histogram(mapping = aes(age), bins = 30, fill =  4, color = "black")
p+geom_histogram(mapping = aes(creatinine_phosphokinase), bins = 30, fill =  4, color = "black")
p+geom_histogram(mapping = aes(ejection_fraction), bins = 30, fill =  4, color = "black")
p+geom_histogram(mapping = aes(platelets), bins = 30, fill =  4, color = "black")
p+geom_histogram(mapping = aes(serum_creatinine), bins = 30, fill =  4, color = "black")
p+geom_histogram(mapping = aes(serum_sodium), bins = 30, fill =  4, color = "black")

daisy.dist <- daisy(Heart.Failure, metric = "gower")
hc.daisy.gower <- hclust(daisy.dist, method = "complete")
plot(hc.daisy.gower, cex = 0.5)

hf <- Heart.Failure[,-12]
dd.hf <- daisy(hf, metric = "gower")
hc.hf <- hclust(dd.hf, method = "complete")
plot(hc.hf, cex = 0.5)

# cut three into two main groups
hc.daisy.gower.cut <- cutree(hc.daisy.gower, k=2)

# change 2 to 0 in hc.daisy.gower.cut such that:
```

```
## 0: not in dying group
## 1: in dying group
hc.daisy.gower.cut[which(hc.daisy.gower.cut == 2)] <- 0

# add column of classified patients to data
Heart.Failure <- cbind(Heart.Failure, hc.daisy.gower.cut)



# alive
hc.EF.0 <- Heart.Failure$ejection_fraction[which(Heart.Failure$DEATH_EVENT == 0)]
# death
hc.EF.1 <- Heart.Failure$ejection_fraction[which(Heart.Failure$DEATH_EVENT == 1)]
summary(hc.EF.0)
summary(hc.EF.1)
73.01
# serum creatinine
# alive
hc.se.cre.0 <- Heart.Failure$serum_creatinine[which(Heart.Failure$DEATH_EVENT == 0)]
# death
hc.se.cre.1 <- Heart.Failure$serum_creatinine[which(Heart.Failure$DEATH_EVENT == 1)]
summary(hc.se.cre.0)
summary(hc.se.cre.1)

# platelets
# alive
hc.platelet.0 <- Heart.Failure$platelets[which(Heart.Failure$DEATH_EVENT == 0)]
# death
hc.platelet.1 <- Heart.Failure$platelets[which(Heart.Failure$DEATH_EVENT == 1)]
summary(hc.platelet.0)
summary(hc.platelet.1)

# creatinine phosphokinase
# alive
hc.phospho.0 <- Heart.Failure$creatinine_phosphokinase[which(Heart.Failure$DEATH_EVENT == 0)]
# death
hc.phospho.1 <- Heart.Failure$creatinine_phosphokinase[which(Heart.Failure$DEATH_EVENT == 1)]
summary(hc.phospho.0)
summary(hc.phospho.1)

# serum sodium : not differences
# alive
hc.sodium.0 <- Heart.Failure$serum_sodium[which(Heart.Failure$DEATH_EVENT == 0)]
# death
hc.sodium.1 <- Heart.Failure$serum_sodium[which(Heart.Failure$DEATH_EVENT == 1)]
summary(hc.sodium.0)
summary(hc.sodium.1)
# alive
hc.dg.EF.0 <- Heart.Failure$ejection_fraction[which(hc.daisy.gower.cut == 0)]
# death
hc.dg.EF.1 <- Heart.Failure$ejection_fraction[which(hc.daisy.gower.cut == 1)]
summary(hc.dg.EF.0)
summary(hc.dg.EF.1)
```

```
# serum creatinine
# alive
hc.dg.se.cre.0 <- Heart.Failure$serum_creatinine[which(hc.daisy.gower.cut == 0)]
# death
hc.dg.se.cre.1 <- Heart.Failure$serum_creatinine[which(hc.daisy.gower.cut == 1)]
summary(hc.dg.se.cre.0)
summary(hc.dg.se.cre.1)

# platelets
# alive
hc.dg.platelet.0 <- Heart.Failure$platelets[which(hc.daisy.gower.cut == 0)]
# death
hc.dg.platelet.1 <- Heart.Failure$platelets[which(hc.daisy.gower.cut == 1)]
summary(hc.dg.platelet.0)
summary(hc.dg.platelet.1)

# creatinine phosphokinase
# alive
hc.dg.phospho.0 <- Heart.Failure$creatinine_phosphokinase[which(hc.daisy.gower.cut == 0)]
# death
hc.dg.phospho.1 <- Heart.Failure$creatinine_phosphokinase[which(hc.daisy.gower.cut == 1)]
summary(hc.dg.phospho.0)
summary(hc.dg.phospho.1)

# serum sodium : not differences
# alive
hc.dg.sodium.0 <- Heart.Failure$serum_sodium[which(hc.daisy.gower.cut == 0)]
# death
hc.dg.sodium.1 <- Heart.Failure$serum_sodium[which(hc.daisy.gower.cut == 1)]
summary(hc.dg.sodium.0)
summary(hc.dg.sodium.1)


## TN
sum(Heart.Failure$DEATH_EVENT == 0) # 203 patients don't die
## TP
sum(Heart.Failure$DEATH_EVENT == 1) # 96 patients don't die

# hierarchical clustering:
sum(Heart.Failure$hc.daisy.gower.cut == 0) # 225 patients don't die
sum(Heart.Failure$hc.daisy.gower.cut == 1) # 74 patients don't die

# FN: how many TP were classified as Negative
## how many 1 in DEATHEVENTS were classified as 0 in Daisy.Gower

false_negatives <- 0

# FP: How many TN were classified as Positive
## how many 0 in DEATH EVENTS were classifed as 1 in Daisy.Gower
false_positives <- 22
```

```
tn_fp <- c(203, 22)
fn_tp <- c(0, 96)
confusion.matrx <- (rbind(tn_fp,fn_tp))
names
accuracy <- (96 + 203)/(203+22+96)
Err_rate <- 1-accuracy
Err_rate*100
accuracy*100
table(confusion.matrx)
ggplot(heart_failure_clinical_records_dataset, aes(y = DEATH_EVENT, x = age))+
  geom_point(alpha = .5)+
  stat_smooth(method = "glm", se = TRUE, method.args = list(family=binomial))+
  labs(title = "Death Event correlation with Age", caption = "As expected, there is strong corre
lation between age and death events.") + xlab("Age") +
  ylab("Death Event Occurence") + theme(plot.margin = unit(c(0.2,0.2,0.2,0.2),"cm"))
ggplot(heart_failure_clinical_records_dataset, aes(y = DEATH_EVENT, x = time))+
  geom_point(alpha = .5)+
  stat_smooth(method = "glm", se = TRUE, method.args = list(family=binomial))+
  labs(title = "Death Event correlation with Time Elapsed Before Follow-up Appointment", caption
= "As a patient lives longer, the time for a follow-up can increase. For this reason this is lik
ely
       causation rather than correlation. Furthermore, a case's severity would warrant a sooner
follow-up.") + xlab("Time elapsed before follow-up appointment") +
  ylab("Death Event Occurence")

ggplot(heart_failure_clinical_records_dataset, aes(y = DEATH_EVENT, x = diabetes))+
  geom_point(alpha = .5)+
  stat_smooth(method = "glm", se = TRUE, method.args = list(family=binomial))+
  labs(title = "Death Event correlation with Diabetes", caption = "Diabetes interestingly is not
a strong predictor of death events due to heart failure") + xlab("Diabetes occurence") +
  ylab("Death Event Occurence")

ggplot(heart_failure_clinical_records_dataset, aes(y = DEATH_EVENT, x = smoking))+
  geom_point(alpha = .5)+
  stat_smooth(method = "glm", se = TRUE, method.args = list(family=binomial))+
  labs(title = "Death Event correlation with Smoking", caption = "Smoking apparently has a posit
ive impact in preventing death occurences, although minimal.") + xlab("Smoker") +
  ylab("Death Event Occurence")



ggplot(heart_failure_clinical_records_dataset, aes(y = DEATH_EVENT, x = serum_creatinine))+
  geom_point(alpha = .5)+
  stat_smooth(method = "glm", se = TRUE, method.args = list(family=binomial))+
  labs(title = "Death Events per Serum Creatinine", caption = "As blood serum creatinine increas
es, so does a patient's likelihood of a death event.") + xlab("Serum Creatinine") +
  ylab("Death Event Occurence")
ggplot(heart_failure_clinical_records_dataset, aes(y = DEATH_EVENT, x = ejection_fraction))+
  geom_point(alpha = .5)+
  stat_smooth(method = "glm", se = TRUE, method.args = list(family=binomial))+
  labs(title = "Death Events per Ejection Fraction", caption = "Ejection Fraction has an inverse
relation with the chances of a death event") + xlab("Ejection Fraction") +
  ylab("Death Event Occurence")
```

```
miscaccuracy <- function(model, train, test) {
  smry <- summary(model)
  predicted <- ifelse(predict(model, test, type= "response") > .5, "DEAD", "ALIVE")
  cnfsnmtrx <- table(
    predicted,
    test$DEATH_EVENT,
    dnn = c("Predicted Death Events","True Death Events"))
  print("Summary, Misclassification Error Rate and Accuracy Rate, and Confusion Matrix")
  msclass <- 1-sum(diag(cnfsnmtrx))/sum(cnfsnmtrx)
  acrcy <- sum(diag(cnfsnmtrx))/sum(cnfsnmtrx)
  outp <- list(smry,msclass, acrcy, cnfsnmtrx)
  return(outp)

}

index_dead = which(heartdata$DEATH_EVENT == 1)[1:19]
index_alive = which(heartdata$DEATH_EVENT == 0)[1:40]

data_test = as.data.frame(heartdata[c(index_dead, index_alive),])
data_train = as.data.frame(heartdata[-c(index_dead, index_alive),])


logreg1 <- glm(DEATH_EVENT ~. , data = data_train, family = binomial)
summary(logreg1)
prob <- predict(logreg1, type = "response")
predicted <- ifelse(predict(logreg1, data_test, type = "response")>.5,
                    "DEAD","ALIVE")

confusion_matrix <- table(
  predicted,
  data_test$DEATH_EVENT,
  dnn = c("Predicted Death Events","True Death Events"))
print("Confusion Matrix")
confusion_matrix
print("Misclassification Error Rate")
1-sum(diag(confusion_matrix))/sum(confusion_matrix)


logreg <- glm(DEATH_EVENT ~ ejection_fraction + serum_creatinine, data = data_train, family = bi
nomial)
summary(logreg)
prob <- predict(logreg, type = "response")
predicted <- ifelse(predict(logreg, data_test, type = "response")>.5,
                    "DEAD","ALIVE")


confusion_matrix <- table(
  predicted,
  data_test$DEATH_EVENT,
  dnn = c("Predicted Death Events","True Death Events"))
confusion_matrix
print("Misclassification Error Rate")
1-sum(diag(confusion_matrix))/sum(confusion_matrix)
```

```
anova(logreg1, logreg, test = "Chisq")
anova(logreg1, logreg)
```

```
print_est = function(m, caption) {
  t_coef = coefficients(summary(m))[,1, drop=F]
  rownames(t_coef) = sapply(rownames(t_coef), convert_beta_string)
  colnames(t_coef)[1] <- "Value"
  t_coef = rbind(t_coef , "$s^2$" = summary(m)$sigma^2)
  knitr::kable(t_coef, align = "l", caption = caption)
}
```

```
#!!! the time variable is removed here!!!
heartdata <- heartdata[,-12]
#mutate 'DEATH_EVENT' to factor instead of numeric!
heartdata <- heartdata %>%
  mutate_at('DEATH_EVENT',as.factor)
heartdata <- heartdata %>%
  mutate_at('diabetes',as.factor)
heartdata <- heartdata %>%
  mutate_at('high_blood_pressure',as.factor)
heartdata <- heartdata %>%
  mutate_at('anaemia',as.factor)
heartdata <- heartdata %>%
  mutate_at('sex',as.factor)
heartdata <- heartdata %>%
  mutate_at('smoking',as.factor)
ggplot(data= heartdata, aes(x = DEATH_EVENT, y=ejection_fraction, fill=DEATH_EVENT))+
  geom_boxplot()+
  geom_point()
ggplot(data= heartdata, aes(x = DEATH_EVENT, y=creatinine_phosphokinase,fill=DEATH_EVENT))+
  geom_boxplot()+
  geom_point()
ggplot(data= heartdata, aes(x = DEATH_EVENT, y=serum_creatinine,fill=DEATH_EVENT))+
  geom_boxplot()+
  geom_point()
ggplot(data= heartdata, aes(x = DEATH_EVENT, y=serum_sodium,fill=DEATH_EVENT))+
  geom_boxplot()+
  geom_point()
ggplot(data= heartdata, aes(x = DEATH_EVENT, y=age, fill=DEATH_EVENT))+
  geom_boxplot()+
  geom_point()
#Set up training and testing dataset
```

```
index_dead = which(heartdata$DEATH_EVENT == 1)[1:19]
index_alive = which(heartdata$DEATH_EVENT == 0)[1:40]

heartdata_test = as.data.frame(heartdata[c(index_dead, index_alive),])
heartdata_train = as.data.frame(heartdata[-c(index_dead, index_alive),])
# Full model
lda_full <- lda(DEATH_EVENT~., data=heartdata_train)
lda_full
lda_full_pred <- predict(lda_full,heartdata_test)
table(heartdata_test$DEATH_EVENT, lda_full_pred$class)
sum(diag(table(heartdata_test$DEATH_EVENT, lda_full_pred$class)))/sum(table(heartdata_test$DEATH_EVENT, lda_full_pred$class))

# 3s model (DEATH_EVENT~age+ejection_fraction+serum_cristinine)
lda_3 <- lda(DEATH_EVENT~age+ejection_fraction+serum_creatinine, data=heartdata_train)
lda_3
lda_3_pred <- predict(lda_3,heartdata_test)
table(heartdata_test$DEATH_EVENT, lda_3_pred$class)
sum(diag(table(heartdata_test$DEATH_EVENT, lda_3_pred$class)))/sum(table(heartdata_test$DEATH_EVENT, lda_3_pred$class))

#2s model (DEATH_EVENT~ejection_fraction+serum_cristinine)
lda_2 <- lda(DEATH_EVENT~ejection_fraction+serum_creatinine, data=heartdata_train)
lda_2
lda_2_pred <- predict(lda_2,heartdata_test)
table(heartdata_test$DEATH_EVENT, lda_2_pred$class)
sum(diag(table(heartdata_test$DEATH_EVENT, lda_2_pred$class)))/sum(table(heartdata_test$DEATH_EVENT, lda_2_pred$class))
#set up for dataframe which only contain 'ejection_fraction', 'serum_cristinine' and 'DEATH_EVENT'.
data_lda_visual <- heartdata_train[, c(5,8,12)]
data_lda_visual_test <- heartdata_test[,c(5,8,12)]

#Build up lda model and do prediction
data_lda_visual_fit <- lda(DEATH_EVENT~ejection_fraction+serum_creatinine, data = data_lda_visual,cv=TRUE)
data_lda_visual_pred <- predict(data_lda_visual_fit,data_lda_visual_test)

#Consufion matrix construction and calculate accuracy for the model
table(data_lda_visual_pred$class,heartdata_test$DEATH_EVENT)
sum(diag(table(data_lda_visual_pred$class,heartdata_test$DEATH_EVENT)))/sum(table(data_lda_visual_pred$class,heartdata_test$DEATH_EVENT))


# Calculate the slope and intercept for decision boundary
pi1_h <- data_lda_visual_fit$prior[1]
pi2_h <- data_lda_visual_fit$prior[2]
m1_h <- data_lda_visual_fit$means[1,]
m2_h <- data_lda_visual_fit$means[2,]

n1 <- length(which(heartdata_train$DEATH_EVENT == 0))
n2 <- length(which(heartdata_train$DEATH_EVENT == 1))
```

```
S_hat <- ((n1-1)*cov(subset(heartdata_train,DEATH_EVENT==0)[,c(5,8)])+(n2-1)*cov(subset(heartdata_train,DEATH_EVENT==1)[,c(5,8)]))/(n1+n2-2)

S_h_inv <- solve(S_hat)
b1 <- S_h_inv %*% (m2_h-m1_h)
b0 <- log(pi1_h/pi2_h)-.5*t(m1_h)%*%S_h_inv%*%m1_h+.5*t(m2_h)%*%S_h_inv%*%m2_h

data_lda_visual_fit


# Plot the prediction using lda model which is trained by heartdata_train
ggplot(data= heartdata[,c(5,8,12)], aes(x = log(ejection_fraction), y= log(serum_creatinine))) +
  geom_point(aes(color = DEATH_EVENT),size=2)+
  geom_point(data = data_lda_visual_test[,-3],aes( x = log(ejection_fraction), y =log(serum_creatinine),shape = data_lda_visual_pred$class))+
  labs(shape = 'Prediction')+
   geom_abline(slope=-b1[1]/b1[2],intercept=-b0/b1[2], color = 'blue')

# Calculate the slope and intercept for decision boundary
pi1_h <- data_lda_visual_fit$prior[1]
pi2_h <- data_lda_visual_fit$prior[2]
m1_h <- data_lda_visual_fit$means[1,]
m2_h <- data_lda_visual_fit$means[2,]

n1 <- length(which(heartdata_train$DEATH_EVENT == 0))
n2 <- length(which(heartdata_train$DEATH_EVENT == 1))

S_hat <- ((n1-1)*cov(subset(heartdata_train,DEATH_EVENT==0)[,c(5,8)])+(n2-1)*cov(subset(heartdata_train,DEATH_EVENT==1)[,c(5,8)]))/(n1+n2-2)

S_h_inv <- solve(S_hat)
b1 <- S_h_inv %*% (m2_h-m1_h)
b0 <- log(pi1_h/pi2_h)-.5*t(m1_h)%*%S_h_inv%*%m1_h+.5*t(m2_h)%*%S_h_inv%*%m2_h

data_lda_visual_fit
#caret package is used here!
ctrl <- trainControl(method = "cv", number = 5)

#cross validation for 3s model (5-fold)
lda_cv_3 <- train(DEATH_EVENT ~ ejection_fraction + serum_creatinine+age, data = heartdata, method = "lda", trControl = ctrl)
print(lda_cv_3)


#lda_full
#cross validation for 2s model (5-fold)
lda_cv_2 <-train(DEATH_EVENT ~ ejection_fraction + serum_creatinine, data = heartdata, method = "lda", trControl = ctrl)
print(lda_cv_2)

#cross validation for full model (5-fold)
```

```
lda_cv_F <- train(DEATH_EVENT ~ ., data = heartdata, method = "lda", trControl = ctrl)
print(lda_cv_F)
```