Q1

The equation associated with ridge regression is: $y = X\beta + \epsilon$. In this equation, y represents the N*1 observation vectors of the dependent variable. X is the N*k matrix with K regressors. β is the K*1 vector of coefficients of regression and ϵ is the matrix of vector of errors.

The ordinary least square estimator $\hat{\beta}$ is a solution of the minimization problem which is:

$$\hat{\beta} = \arg min_b \sum_{i=1}^{N} (y_i - x_i b)^2$$

$$where\ x_i is\ the\ ith\ row\ of\ x\ and\ b\ and\ \hat{\beta}\ are\ K * 1\ column\ vector$$

When X has full rank the ordinary least square solution to the problem

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

The ridge estimator $\hat{\beta}_\lambda$ aims to solve an altered minimization problem which is expressed as:

$$\hat{\beta}_\lambda = \arg\min_b \sum_{i=1}^{N} (y_i - x_i b)^2 + \lambda \sum_{k=1}^{k} (b_k)^2$$

In this expression, lambda is a constant which is positive

In ridge estimation, a penalty is added to the criterion of least squares, the sum of squared residuals is being minimized as well as the squared norm of the coefficients of the vectors.

$$SSR = \sum_{i=1}^{N} (y_i - x_i b)^2$$

$$\|b\|^2 = \sum_{k=1}^{k} (b_k)^2$$

Basically, ridge regression penalizes coefficients that are large and the bigger the lambda, the bigger the penalty.

The solution to this minimization problem is expressed as

$$\hat{\beta}_\lambda = (X^T X + \lambda I)^{-1} X^T y$$

Where I is a K*K identity matrix. Compared to ordinary least squares estimation there is no need to assume that the design matrix X is of the full rank. The ridge estimator exists when X doesn't have a full rank as well.

References:

Taboga, Marco (2017). "Ridge regression", Lectures on probability theory and mathematical statistics, Third edition. Kindle Direct Publishing. Online appendix. https://www.statlect.com/fundamentals-of-statistics/ridge-regression.

Q2

```
# Hw 3 SMLE Q2 part 1


library(tidyverse)
library(caret)
library(glmnet)
#call data and renaming columns
hw3_question2 <- read.table("C:/Users/ppill/Desktop/R files/grocery.txt",header= FALSE)
names(hw3_question2) <- c('Y','X1','X2','X3')

#multiple linear regression
lmfit2 <- lm(Y~ X1+X2+X3, data = hw3_question2)
summary(lmfit2)
```

```
> #call data and renaming columns
> hw3_question2 <- read.table("C:/Users/ppill/Desktop/R files/grocery.txt",header= FALSE)
> names(hw3_question2) <- c('Y','X1','X2','X3')
>
> #multiple linear regression
> lmfit2 <- lm(Y~ X1+X2+X3, data = hw3_question2)
> summary(lmfit2)

Call:
lm(formula = Y ~ X1 + X2 + X3, data = hw3_question2)

Residuals:
    Min      1Q  Median      3Q     Max
-264.05 -110.73  -22.52   79.29  295.75

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.150e+03  1.956e+02  21.220  < 2e-16 ***
X1           7.871e-04  3.646e-04   2.159   0.0359 *
X2          -1.317e+01  2.309e+01  -0.570   0.5712
X3           6.236e+02  6.264e+01   9.954 2.94e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 143.3 on 48 degrees of freedom
Multiple R-squared:  0.6883,    Adjusted R-squared:  0.6689
F-statistic: 35.34 on 3 and 48 DF,  p-value: 3.316e-12
```

$$The\ equation\ for\ mutiple\ linear\ regression\ can\ be\ expressed\ as: Y = 4150 + 0.0007871X1 - 1.317\ X2 + 623.6\ X3$$

Residual standard error (sigma hat squared) = 143.3 on 48 degrees of freedom.

The boxed shows the coefficient estimation, standard error, t score and p values. It also shows the significance levels, residual standard error, f statistic with respect to degrees of freedom, multiple and adjusted r squared.

```r
#Q2 part 2
#predict data
z1 <- data.frame(hw3_question2$X1*hw3_question2$X2)
z2 <- data.frame(hw3_question2$X1*hw3_question2$X3)
z3 <- data.frame(hw3_question2$X2*hw3_question2$X3)
z4 <- data.frame(rnorm(52,30,30))
z5 <- data.frame(rnorm(52,7,1))

hw3_question2$z1 <- z1
hw3_question2$z2 <- z2
hw3_question2$z3 <- z3
hw3_question2$z4 <- z4
hw3_question2$z5 <- z5

#lasso regression

a <- hw3_question2$Y
b <- data.frame(hw3_question2$X1,hw3_question2$X2,hw3_question2$X3,hw3_question2$z1,
                hw3_question2$z2,hw3_question2$z3,hw3_question2$z4,hw3_question2$z5)

b_new <- as.matrix(b)

set.seed(123)
lambda <- 10^seq(-3,3, length(100))
cv_model <- cv.glmnet(b_new,a,alpha = 1,lambda = lambda ,nfolds = 5)
optimal_lambda <-cv_model$lambda.min
optimal_lambda

plot(cv_model)

#determine best coefficients of best model

optimal_model <- glmnet(b_new,a, alpha =1, lambda = optimal_lambda)
coef(optimal_model)

#no coefficient is shown for the predictors z1,z2 and z4 as the lasso regression....
#....has shrunk the coefficient all the way to zero. As the result these predictor
#variables were left from the model as it does not have much influence on it.
#source : https://www.statology.org/lasso-regression-in-r/

#determine sst and sse
a_predicted <- predict(optimal_model, s = optimal_lambda, newx = b_new)
sst <- sum((a - mean(a))^2)
sse <- sum((a_predicted-a)^2)
#find r squared

rsq <- 1 - sse/sst
rsq
```

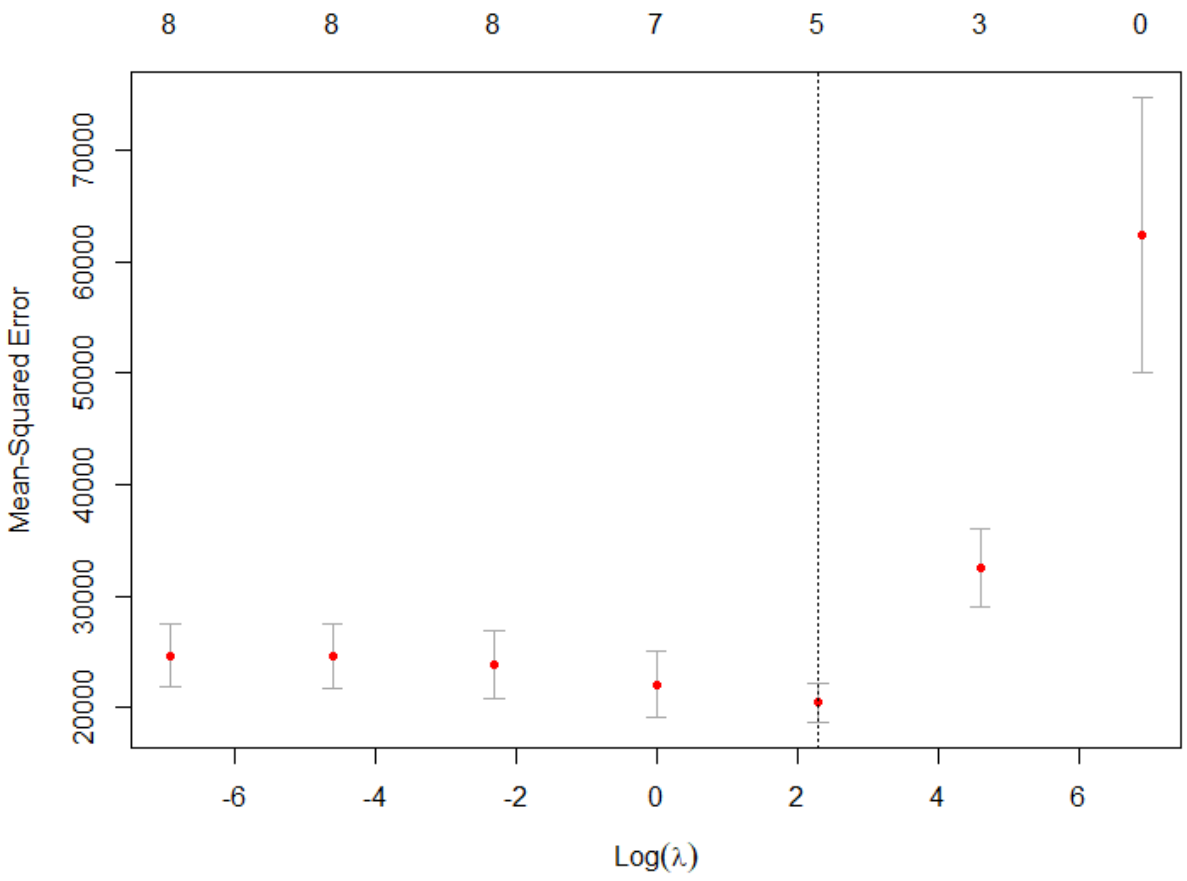
```
> #Q2 part 2
> #predict data
> z1 <- data.frame(hw3_question2$X1*hw3_question2$X2)
> z2 <- data.frame(hw3_question2$X1*hw3_question2$X3)
> z3 <- data.frame(hw3_question2$X2*hw3_question2$X3)
> z4 <- data.frame(rnorm(52,30,30))
> z5 <- data.frame(rnorm(52,7,1))
>
> hw3_question2$z1 <- z1
> hw3_question2$z2 <- z2
> hw3_question2$z3 <- z3
> hw3_question2$z4 <- z4
> hw3_question2$z5 <- z5
>
> #lasso regression
>
> a <- hw3_question2$Y
> b <- data.frame(hw3_question2$X1,hw3_question2$X2,hw3_question2$X3,hw3_question2$z1,
+                 hw3_question2$z2,hw3_question2$z3,hw3_question2$z4,hw3_question2$z5)
>
> b_new <- as.matrix(b)
>
> set.seed(123)
> lambda <- 10^seq( 3,-3, length(100))
> cv_model <- cv.glmnet(b_new,a,alpha = 1,lambda = lambda ,nfolds = 5)
> optimal_lambda <-cv_model$lambda.min
> optimal_lambda
[1] 10
```

Performance of 5-fold cross validation to assess prediction error of estimated model.

```
> plot(cv_model)
>
> #determine best coefficients of best model
>
> optimal_model <- glmnet(b_new,a, alpha =1, lambda = optimal_lambda)
> coef(optimal_model)
9 x 1 sparse Matrix of class "dgCMatrix"
                                              s0
(Intercept)                          4.076916e+03
hw3_question2.X1                     5.887994e-04
hw3_question2.X2                    -1.321987e+01
hw3_question2.X3                     1.952608e+02
hw3_question2.X1...hw3_question2.X2  .
hw3_question2.X1...hw3_question2.X3  .
hw3_question2.X2...hw3_question2.X3  5.195644e+01
rnorm.52..30..30.                    .
rnorm.52..7..1.                      1.971218e+01
>
> #no coefficient is shown for the predictors z1,z2 and z4 as the lasso regression....
> #....has shrunk the coefficient all the way to zero. As the result these predictor
> #variables were left from the model as it does not have much influence on it.
> #source : https://www.statology.org/lasso-regression-in-r/
>
> #determine sst and sse
> a_predicted <- predict(optimal_model, s = optimal_lambda, newx = b_new)
> sst <- sum((a - mean(a))^2)
> sse <- sum((a_predicted-a)^2)
> #find r squared
>
> rsq <- 1 - sse/sst
> rsq
[1] 0.7030669
```

R squared was found to be 0.7030669. Hence, the best model could explain around 70.31% of variation in the response values of training data.

Plot(cv_model):



Q3

```
#IE 500 SMLE HW 3 Question 3

library(tidyverse)
library(caret)
library(glmnet)

#load data
#loading train data
train_data <- read.csv(file = 'train.air.csv', header = TRUE)
#test data
test_data <- read.csv(file = 'test.air.csv', header = TRUE)
#rescaling data to 0 and 1
library(scales)
#rescaling test data
new_testCO <- rescale(test_data$CO)
new_testC6H6 <- rescale(test_data$C6H6)
new_testNMHC <- rescale(test_data$NMHC)
new_testNox <- rescale(test_data$Nox)
new_testNO2 <- rescale(test_data$NO2)
new_testO3 <- rescale(test_data$O3)
new_testT <- rescale(test_data$T)
new_testRH <- rescale(test_data$RH)
new_testAH <- rescale(test_data$AH)

new_test <- data.frame(new_testCO,new_testC6H6,new_testNMHC,new_testNox,new_testNO2
                       ,new_testO3,new_testT,new_testRH,new_testAH)
#rescaling train data
new_trainCO <- rescale(train_data$CO)
new_trainC6H6 <- rescale(train_data$C6H6)
new_trainNMHC <- rescale(train_data$NMHC)
new_trainNox <- rescale(train_data$Nox)
new_trainNO2 <- rescale(train_data$NO2)
new_trainO3 <- rescale(train_data$O3)
new_trainT <- rescale(train_data$T)
new_trainRH <- rescale(train_data$RH)
new_trainAH <- rescale(train_data$AH)

new_train <- data.frame(new_trainCO,new_trainC6H6,new_trainNMHC,new_trainNox,new_trainNO2
                       ,new_trainO3,new_trainT,new_trainRH,new_trainAH)
```

```
#predictor variable
x<- model.matrix(new_trainCO~., new_train)[,-1]
#outcome variable
y <- new_train$new_trainCO
#computing ridge regression
set.seed(123)
cv <- cv.glmnet(x,y, alpha = 0)
#show best lambda value
cv$lambda.min
#fit final model on training data
ridge_model <- glmnet(x,y, alpha = 0, lambda = cv$lambda.min)
coef(ridge_model)
#predict test data
x_test_ridge <- model.matrix(new_testCO~., new_test)[,-1]
prediction_ridge <- ridge_model %>% predict(x_test_ridge) %>% as.vector()
# model performance metrics
library(Metrics)
data.frame(RMSE_ridge = rmse(prediction_ridge, new_test$new_testCO),
           Rsquared_ridge = R2(prediction_ridge,new_test$new_testCO)
           )
#computing lasso regression
set.seed(123)
cv <- cv.glmnet(x,y, alpha = 1)
#show best lambda value
cv$lambda.min
#fit final model on training data
lasso_model <- glmnet(x,y, alpha = 1, lambda = cv$lambda.min)
coef(lasso_model)
#predict test data
x_test_lasso <- model.matrix(new_testCO~., new_test)[,-1]
prediction_lasso <- lasso_model %>% predict(x_test_lasso) %>% as.vector()
# model performance metrics
data.frame(
  RMSE_lasso = rmse(prediction_lasso, new_test$new_testCO),
  Rsquare_lasso = R2(prediction_lasso,new_test$new_testCO)
)
#computing elastic net regression
set.seed(123)
cv <- cv.glmnet(x,y, alpha = 0.3)
#show best lambda value
cv$lambda.min
#fit final model on training data
elasticnet_model <- glmnet(x,y, alpha = 0.3, lambda = cv$lambda.min)
coef(elasticnet_model)
#predict test data
x_test_elasticnet <- model.matrix(new_testCO~., new_test)[,-1]
prediction_elasticnet <- elasticnet_model %>% predict(x_test_elasticnet) %>% as.vector()
# model performance metrics
data.frame(
  RMSE_elasticnet = rmse(prediction_elasticnet, new_test$new_testCO),
  Rsquare_elasticnet = R2(prediction_elasticnet,new_test$new_testCO)
)

#The lasso model will be selected to present the prediction for Carbon Monoxide
#as it has a lower RMSE compared to ridge and elastic net regression
```

Ridge, lasso and elastic ridge regression in violet, grey and yellow boxes respectively. It is assumed that alpha value for lasso is 0.3 (between 0 and 1).

```
> #computing ridge regression
> set.seed(123)
> cv <- cv.glmnet(x,y, alpha = 0)
> #show best lambda value
> cv$lambda.min
[1] 0.01334379
> #fit final model on training data
> ridge_model <- glmnet(x,y, alpha = 0, lambda = cv$lambda.min)
> coef(ridge_model)
9 x 1 sparse Matrix of class "dgCMatrix"
                      s0
(Intercept)   -0.02146104
new_trainC6H6  0.41546898
new_trainNMHC  0.24646576
new_trainNox   0.15264101
new_trainNO2   0.05690027
new_trainO3    0.06151206
new_trainT    -0.03153857
new_trainRH    0.01419318
new_trainAH   -0.01004718
> #predict test data
> x_test_ridge <- model.matrix(new_testCO~., new_test)[,-1]
> prediction_ridge <- ridge_model %>% predict(x_test_ridge) %>% as.vector()
> # model performance metrics

> data.frame(RMSE_ridge = rmse(prediction_ridge, new_test$new_testCO),
+            Rsquared_ridge = R2(prediction_ridge,new_test$new_testCO)
+            )
  RMSE_ridge Rsquared_ridge
1 0.04370738      0.8717504
> #computing lasso regression
```

```
> #computing lasso regression
> set.seed(123)
> cv <- cv.glmnet(x,y, alpha = 1)
> #show best lambda value
> cv$lambda.min
[1] 0.0005023851
> #fit final model on training data
> lasso_model <- glmnet(x,y, alpha = 1, lambda = cv$lambda.min)
> coef(lasso_model)
9 x 1 sparse Matrix of class "dgCMatrix"
                      s0
(Intercept)    0.01632347
new_trainC6H6  0.81022074
new_trainNMHC  .
new_trainNox   0.11709114
new_trainNO2   0.07398231
new_trainO3    .
new_trainT    -0.05607151
new_trainRH    0.01152857
new_trainAH    .
> #predict test data
> x_test_lasso <- model.matrix(new_testCO~., new_test)[,-1]
> prediction_lasso <- lasso_model %>% predict(x_test_lasso) %>% as.vector()
> # model performance metrics
> data.frame(
+   RMSE_lasso = rmse(prediction_lasso, new_test$new_testCO),
+   Rsquare_lasso = R2(prediction_lasso,new_test$new_testCO)
+ )
  RMSE_lasso Rsquare_lasso
1 0.04270161     0.8762554
```

```
> #computing elastic net regression
> set.seed(123)
> cv <- cv.glmnet(x,y, alpha = 0.3)
> #show best lambda value
> cv$lambda.min
[1] 0.0002605162
> #fit final model on training data
> elasticnet_model <- glmnet(x,y, alpha = 0.3, lambda = cv$lambda.min)
> coef(elasticnet_model)
9 x 1 sparse Matrix of class "dgCMatrix"
                      s0
(Intercept)    0.011780217
new_trainC6H6  0.789581812
new_trainNMHC  0.027551951
new_trainNox   0.115599085
new_trainNO2   0.074210801
new_trainO3   -0.005639725
new_trainT    -0.054615296
new_trainRH    0.016582815
new_trainAH   -0.004837267
> #predict test data
> x_test_elasticnet <- model.matrix(new_testCO~., new_test)[,-1]
> prediction_elasticnet <- elasticnet_model %>% predict(x_test_elasticnet) %>% as.vector()
> # model performance metrics
> data.frame(
+   RMSE_elasticnet = rmse(prediction_elasticnet, new_test$new_testCO),
+   Rsquare_elasticnet = R2(prediction_elasticnet,new_test$new_testCO)
+ )
  RMSE_elasticnet Rsquare_elasticnet
1      0.04276571          0.8761902
>
> #The lasso model will be selected to present the prediction for Carbon Monoxide
> #as it has a lower RMSE compared to ridge and elastic net regression
```

Selection of Model according to RMSE. Lasso model has least deviation between values of predictor and values observed hence it is suitable for predicting the carbon monoxide concentration.

```
  RMSE_ridge Rsquared_ridge
1 0.04370738      0.8717504
```

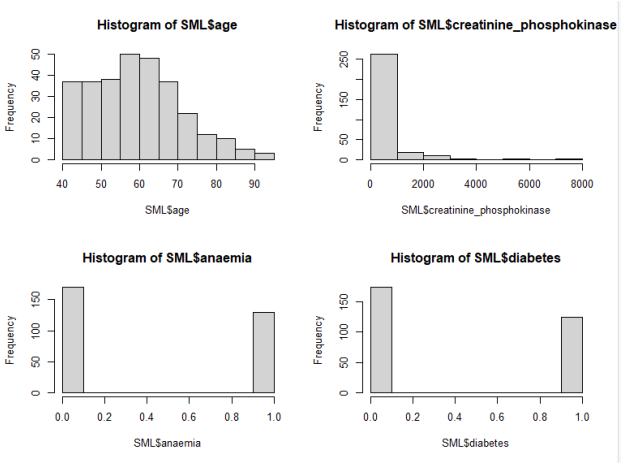
```
  RMSE_lasso Rsquare_lasso
1 0.04270161     0.8762554
```

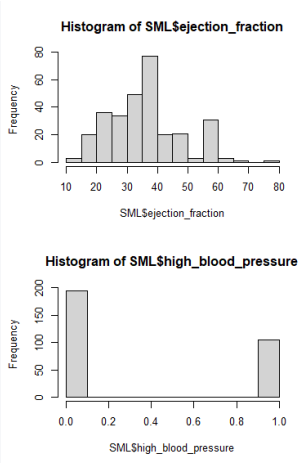
Q4

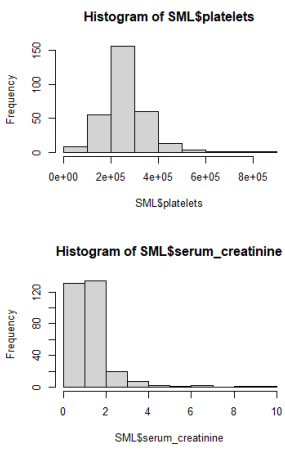
Part 1

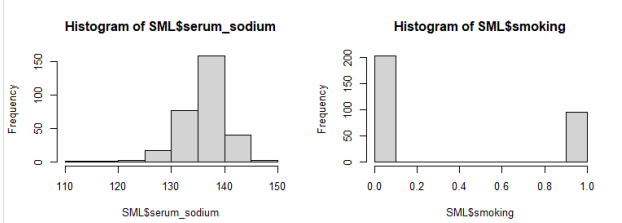
The dataset from Kaggle.com is a model of predicting mortality from heart failure of 300 patients based in hospitals in Pakistan. Cardiovascular diseases (CVDs) the primary cause of death, killing around 18 million people a year accounting for 31% of deaths globally. Heart failure is a consequence of CVDs and has no cure but if alterations are made to lifestyle, the risk of heart failure can be reduced drastically. Examples include cutting smoking and alcohol use, having an active lifestyle and managing a balanced life. People who suffer from CVDs or are at risk of CVDs need immediate diagnosis and treatment. The dataset contains factors such as age, anaemia, creatinine phosphokinase, diabetes, ejection fraction, high blood pressure, platelets, serum creatinine, serum sodium, sex, smoking, time (of follow up with

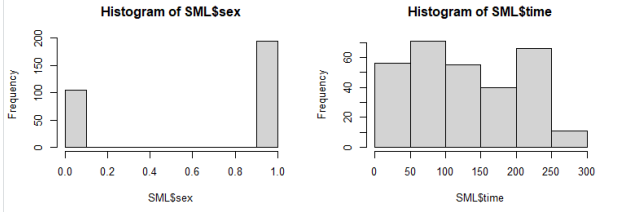
patient) and death event. The death event is the independent variable and the rest of the factors are dependent variables. The data from the set was visualized using histograms, bar plots and pie charts.

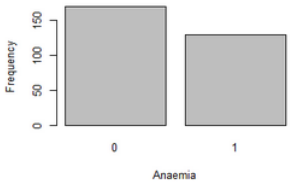


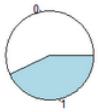


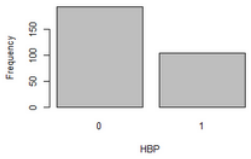


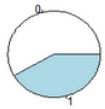


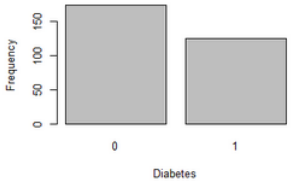


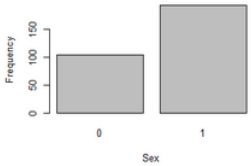












Part 2

The objective of this study is to determine the key factors that affect heart failure so that it becomes preventable and effectively save lives. This will be done using logistical regression, t-test and ANOVA. The correlation and covariance of the factors of dependent and independent variables will be analyzed by statistical means. Logistic regression will be used to model the prediction of mortality after heart failure versus the factors that impact heart failure in patients. Logistic regression will be used as we are trying to predict if the patient will be dead or alive (binary factor/ dependent variable).

The equation (equation for logistical regression) that will be appropriate for this model will be expressed as:

$$\ell = \log_b \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

Other calculations will also be conducted such as root mean square error, mean square error, ANOVA, mean absolute percentage error, r-squared, adjust r-squared and heat maps. These will be in the form:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

$$M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

$$R^2 = 1 - \frac{RSS}{TSS}$$

$$\text{Adjusted R Squared Formula} = 1 - \left[\frac{(1 - R^2) \times (n - 1)}{(n - k - 1)}\right]$$

$$F = \frac{\Sigma n_j(\bar{X}_j - \bar{X})^2/(k-1)}{\Sigma\Sigma(X - \bar{X}_j)^2/(N-k)}$$

df_1 = k-1 and df_2=N-k,

After splitting data into testing and training model, the death event shall be predicted relative to the factors, and we will use this data to counter mortality from occurring.