Heart Disease UCI

Dominik Wolff

3 6 2019

Detecting heart desease risks

Abstract

This work analyzes risk factors for heart desease based on the "Heart Disease UCI dataset." The dataset is derived from Kaggle (https://www.kaggle.com/ronitf/heart-disease-uci/downloads/heart-disease-uci.zip/1). The dataset contains 13 features and the target variable. The "target" refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 1 (present). The 13 features are: 1. age 2. sex 3. chest pain type (4 values) 4. resting blood pressure 5. serum cholestoral in mg/dl 6. fasting blood sugar > 120 mg/dl 7. resting electrocardiographic results (values 0,1,2) 8. maximum heart rate achieved 9. exercise induced angina 10. oldpeak = ST depression induced by exercise relative to rest 11. the slope of the peak exercise ST segment 12. number of major vessels (0-3) colored by flourosopy 13. thal: 3 = normal; 6 = fixed defect; 7 = reversable defect

Load required packages

```r
knitr::opts_chunk$set(echo = FALSE)
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: tidyverse

## Warning: package 'tidyverse' was built under R version 3.5.2

## -- Attaching packages --------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.3.0

## Warning: package 'readr' was built under R version 3.5.2

## Warning: package 'forcats' was built under R version 3.5.2

## -- Conflicts ------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: caret

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```r
if(!require(lubridate)) install.packages("lubridate", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: lubridate

##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```r
if(!require(e1071)) install.packages("e1071", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: e1071
```

```r
if(!require(naivebayes)) install.packages("naivebayes", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: naivebayes

## Warning: package 'naivebayes' was built under R version 3.5.2
```

```r
if(!require(rmarkdown)) install.packages("rmarkdown", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: rmarkdown

## Warning: package 'rmarkdown' was built under R version 3.5.2
```

```r
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(ranger)) install.packages("ranger", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: ranger
```

```r
if(!require(rpart)) install.packages("rpart", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: rpart
```

Download heart desease dataset and prepare data

Source of the Data

https://www.kaggle.com/ronitf/heart-disease-uci/downloads/heart-disease-uci.zip/1. The data is downloaded from the web and unzipped. The csv file is saved to the working directory.

```r
tmpdir <-"C:/Users/Dominik/Downloads/"
setwd(tmpdir)

url <- "https://www.kaggle.com/ronitf/heart-disease-uci/downloads/heart-disease-uci.zip/1"
temp <- tempfile(tmpdir=tmpdir, fileext=".zip")
download.file(url, temp)
unzip(temp)
```

```
## Warning in unzip(temp): Fehler 1 während des Extrahierens aus Zipfile
```

```r
unlink(temp)
```

Prepare Data:

Load data from csv. Rename column names.

```
HeartDeaseaseData <- read.csv("C:/Users/Dominik/Downloads/heart.csv",header = TRUE, sep = ",",dec = "."
class(HeartDeaseaseData)
```

```
## [1] "data.frame"
```

```
dim(HeartDeaseaseData)
```

```
## [1] 303  14
```

```
names <- c("age",
            "sex",
            "ChestPainType",
            "RestingBloodPressure",
            "SerumCholestoral",
            "FastingBloodSugar",
            "RestingElectrocardiographic",
            "MaximumHeartRate",
            "ExerciseAngina",
            "STDepressionExercise",
            "SlopePeakExercise",
            "NumberMajorVessels",
            "thal",
            "heartdesease")

names(HeartDeaseaseData) <- names
head(HeartDeaseaseData)
```

```
##   age sex ChestPainType RestingBloodPressure SerumCholestoral
## 1  63   1             3                  145              233
## 2  37   1             2                  130              250
## 3  41   0             1                  130              204
## 4  56   1             1                  120              236
## 5  57   0             0                  120              354
## 6  57   1             0                  140              192
##   FastingBloodSugar RestingElectrocardiographic MaximumHeartRate
## 1                 1                           0              150
## 2                 0                           1              187
## 3                 0                           0              172
## 4                 0                           1              178
## 5                 0                           1              163
## 6                 0                           1              148
##   ExerciseAngina STDepressionExercise SlopePeakExercise NumberMajorVessels
## 1              0                  2.3                 0                  0
## 2              0                  3.5                 0                  0
## 3              0                  1.4                 2                  0
## 4              0                  0.8                 2                  0
## 5              1                  0.6                 2                  0
## 6              0                  0.4                 1                  0
##   thal heartdesease
## 1    1            1
## 2    2            1
## 3    2            1
## 4    2            1
## 5    2            1
## 6    1            1
```

```
#Check for missing values
sum(is.na(HeartDeaseaseData))
```

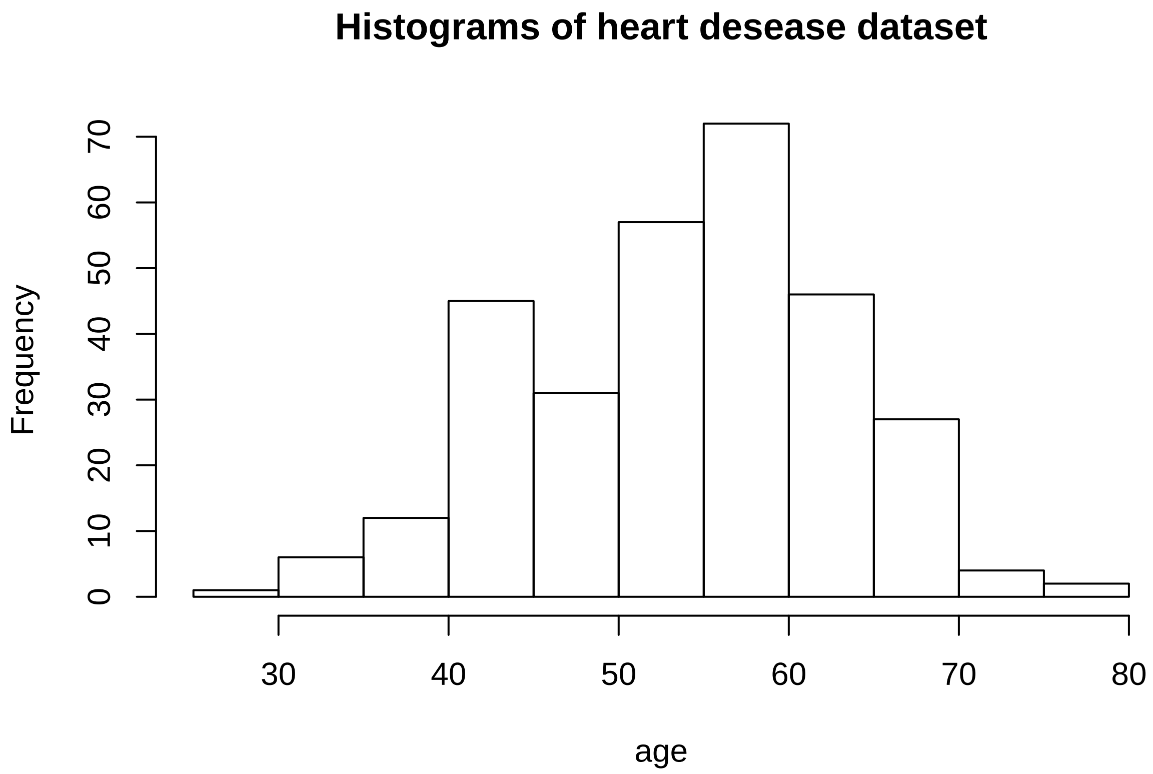
```
## [1] 0
```

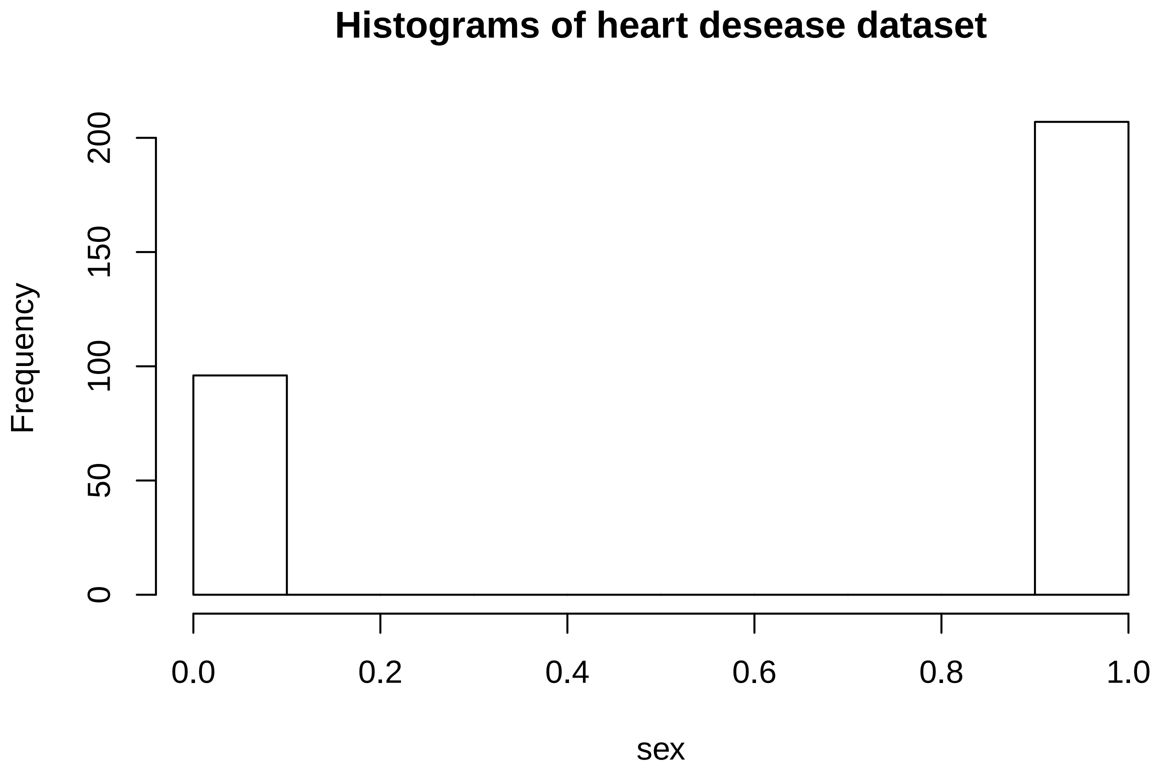
There are no missing values in the dataset.

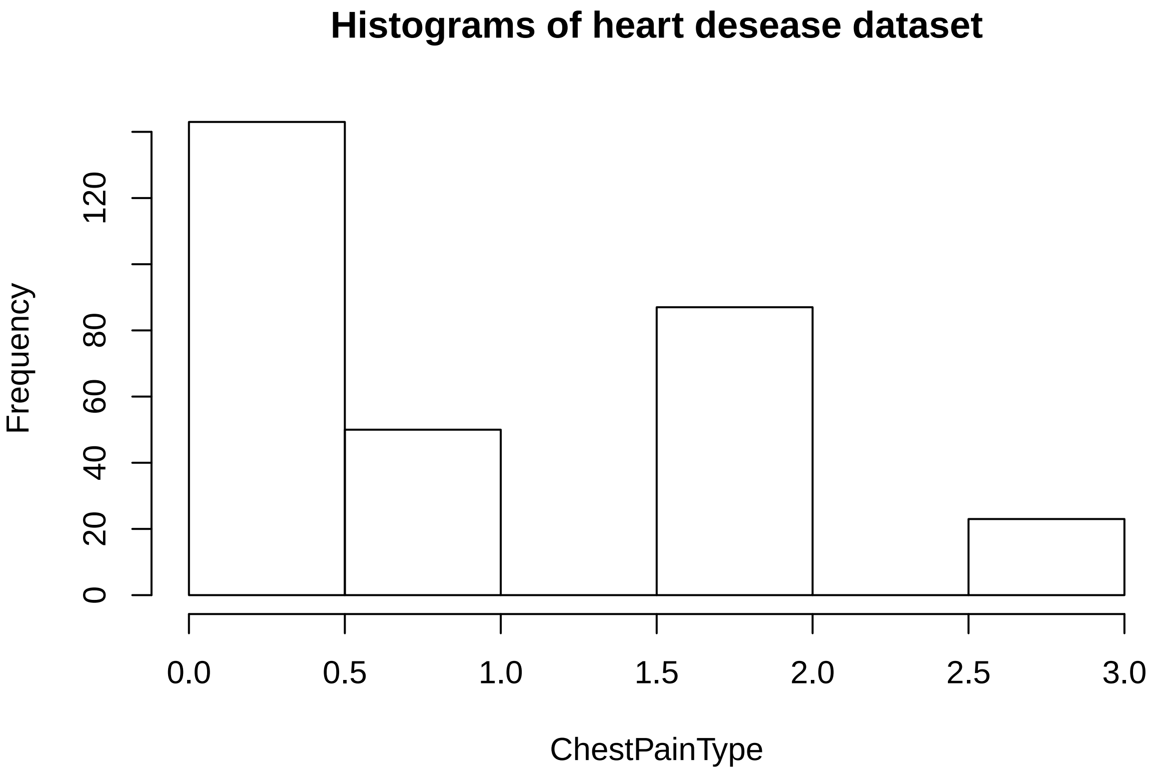
Explanatory Data Analysis

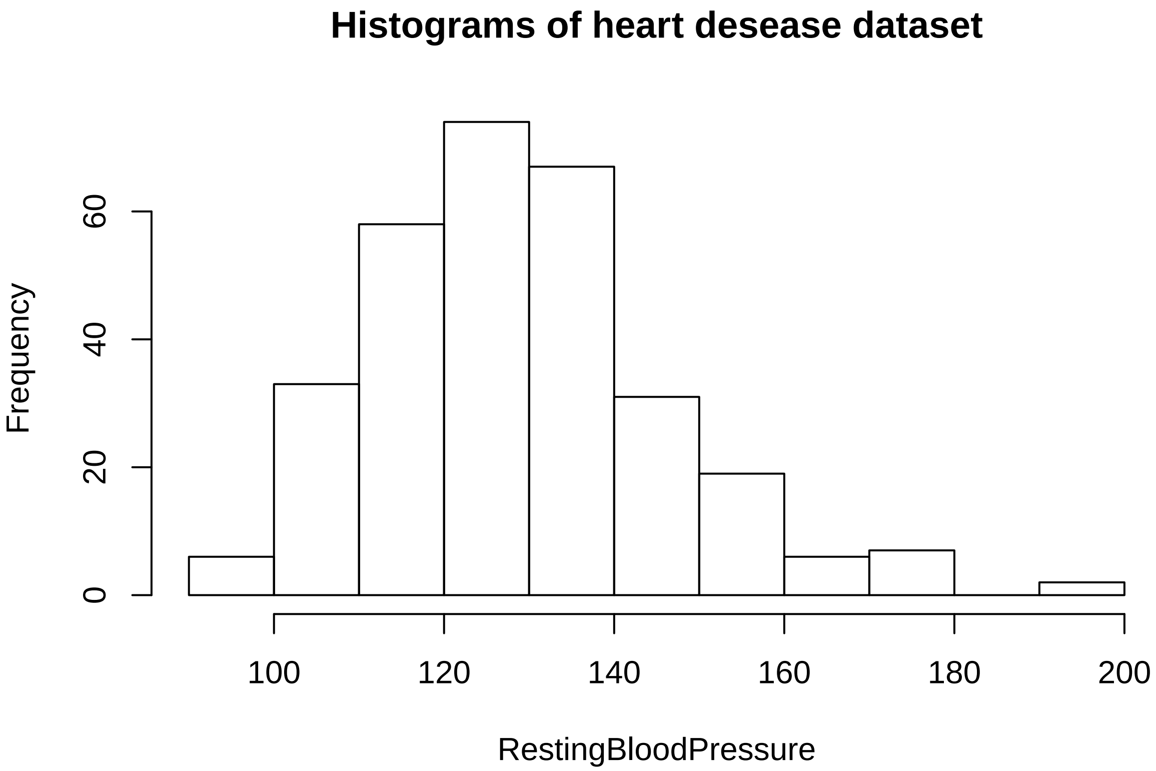
Plot histograms for all variables to analyze the distribution

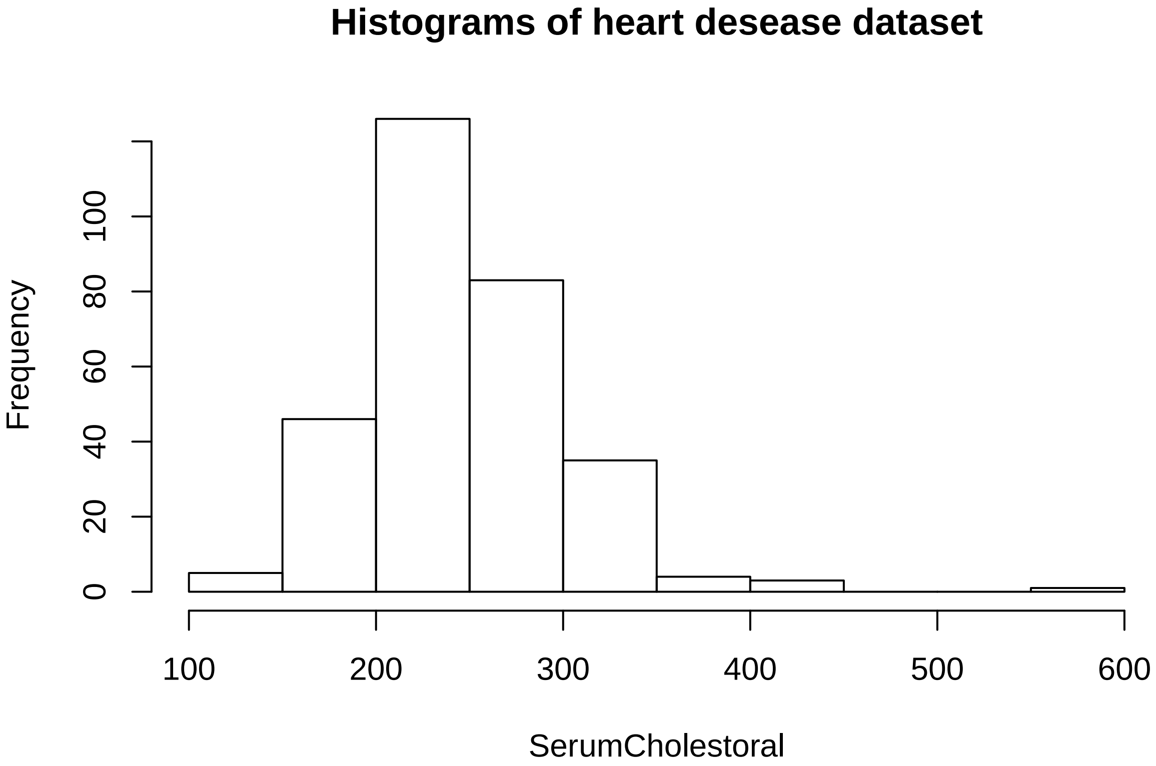
```
for (i in (1:ncol(HeartDeaseaseData))){
hist(HeartDeaseaseData[,i], xlab=names(HeartDeaseaseData[i]), main="Histograms of heart desease dataset
}
```

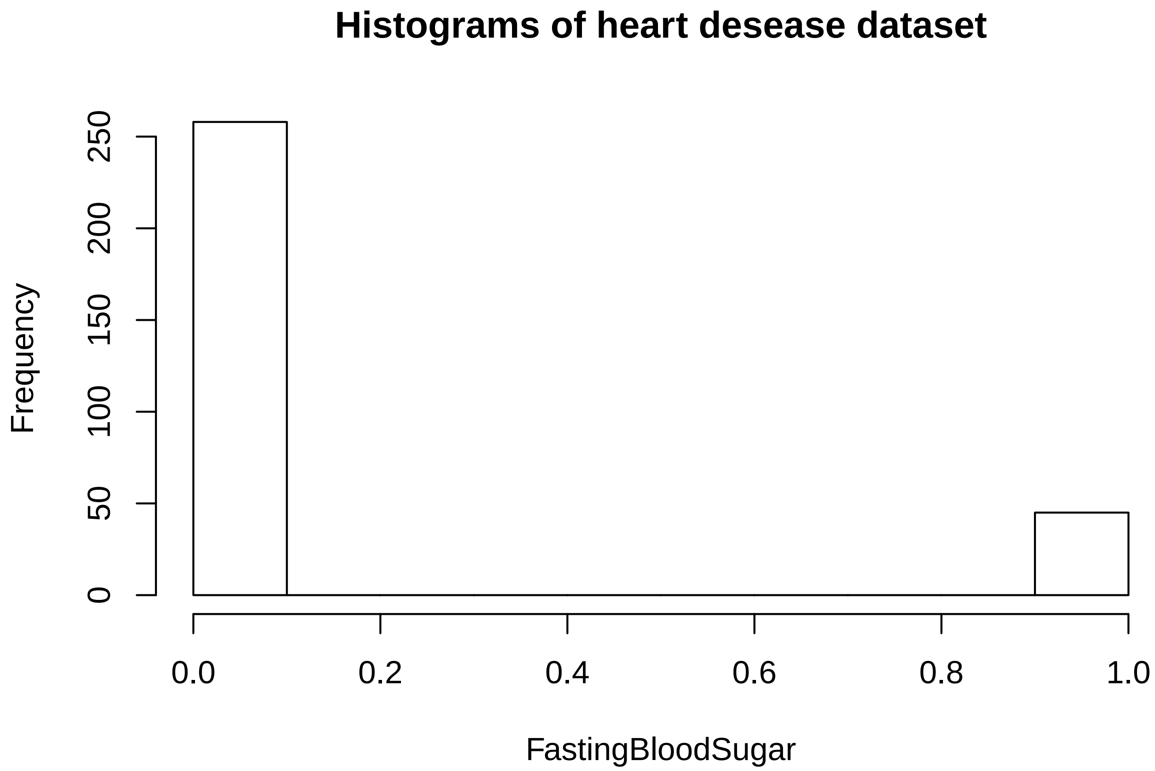


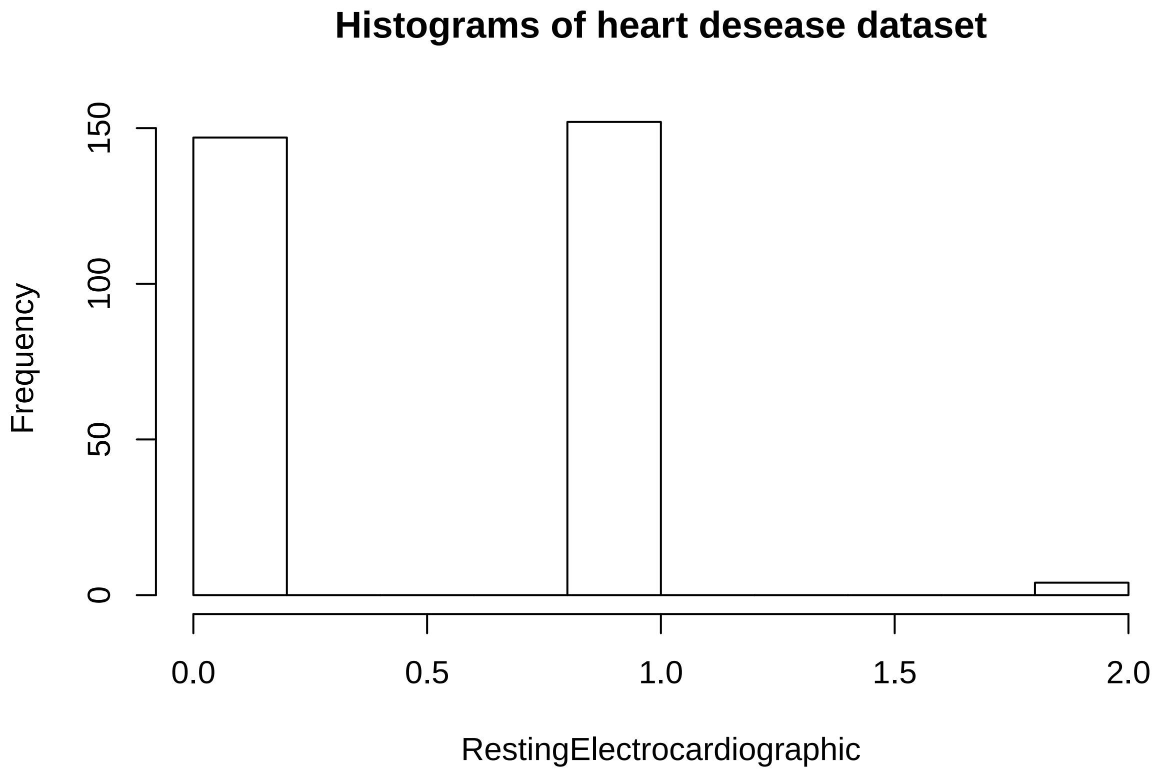

Histograms of heart desease dataset
Frequency
0
50
100
150
200
0.0
0.2
0.4
0.6
0.8
1.0
sex

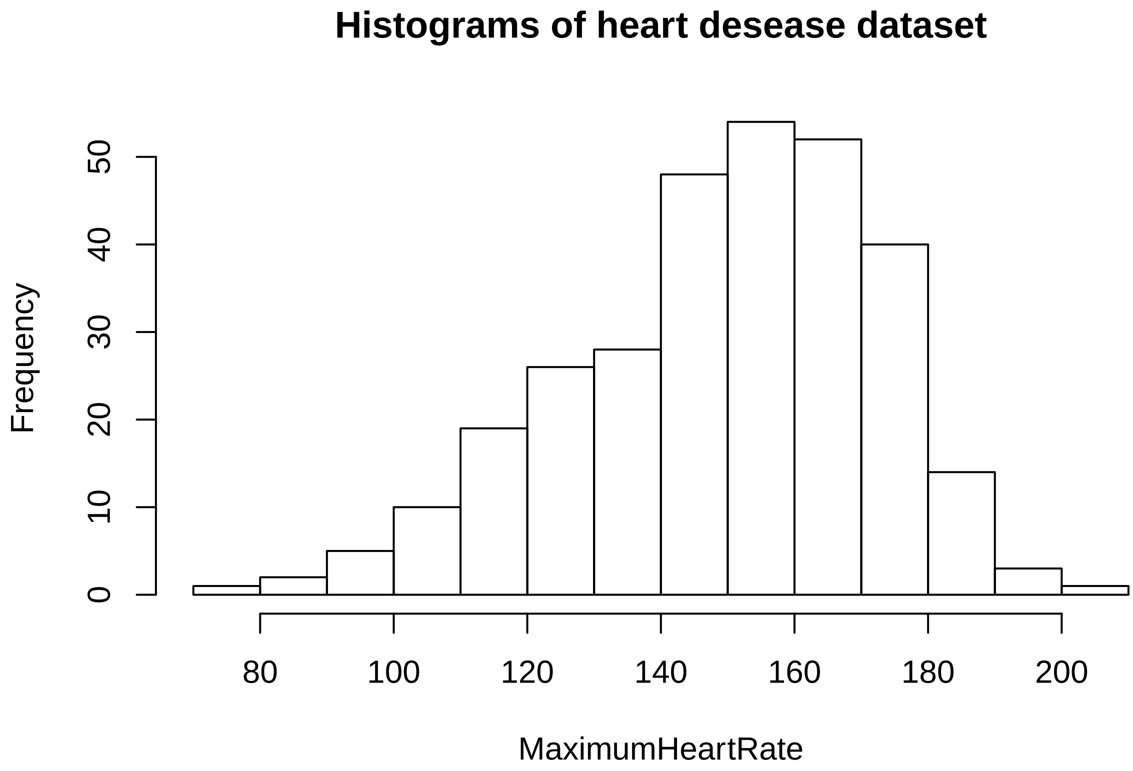

Histograms of heart desease dataset
Frequency
0
20
40
60
80
120
0.0
0.5
1.0
1.5
2.0
2.5
3.0
ChestPainType

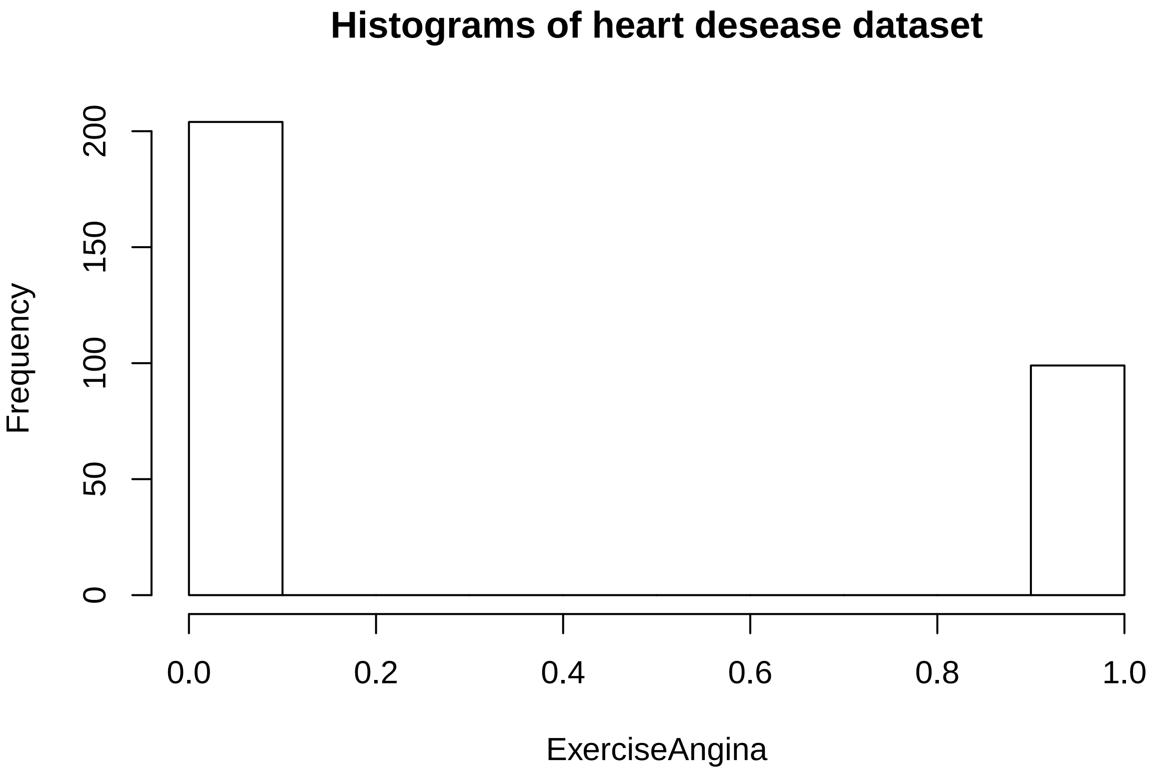

Histograms of heart desease dataset
Frequency
0
20
40
60
100
120
140
160
180
200
RestingBloodPressure

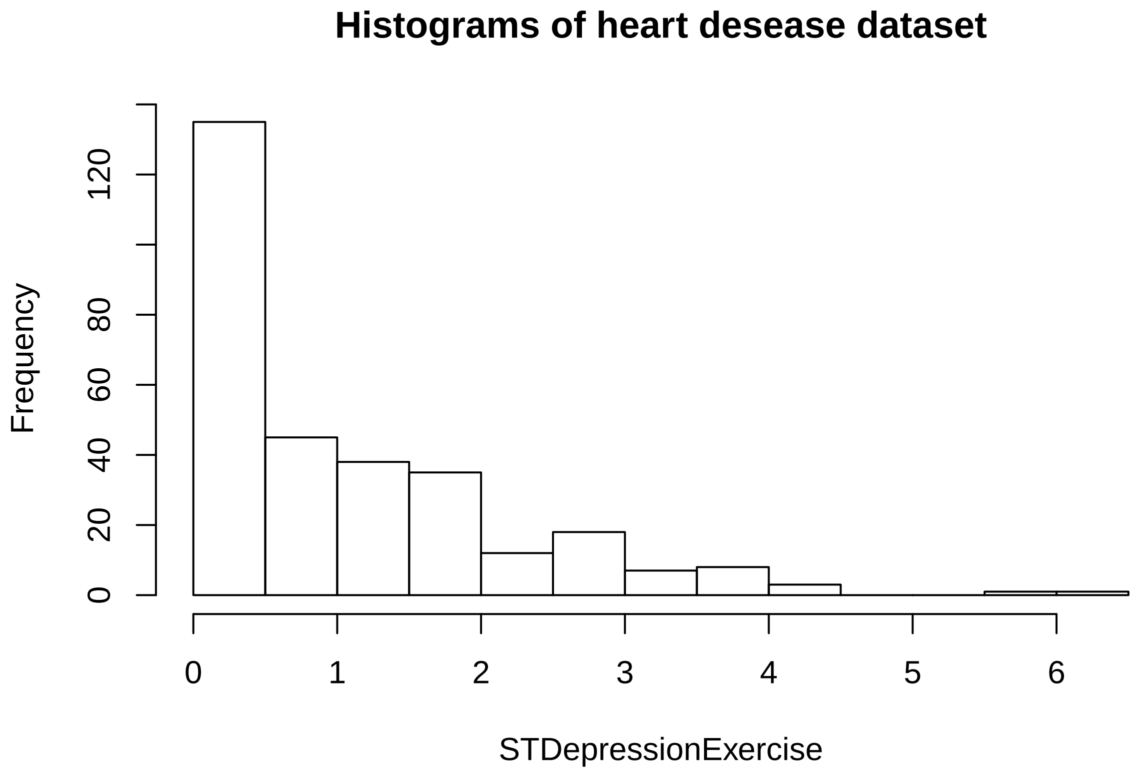

Histograms of heart desease dataset
Frequency
0
20
40
60
80
100
100
200
300
400
500
600
SerumCholestoral

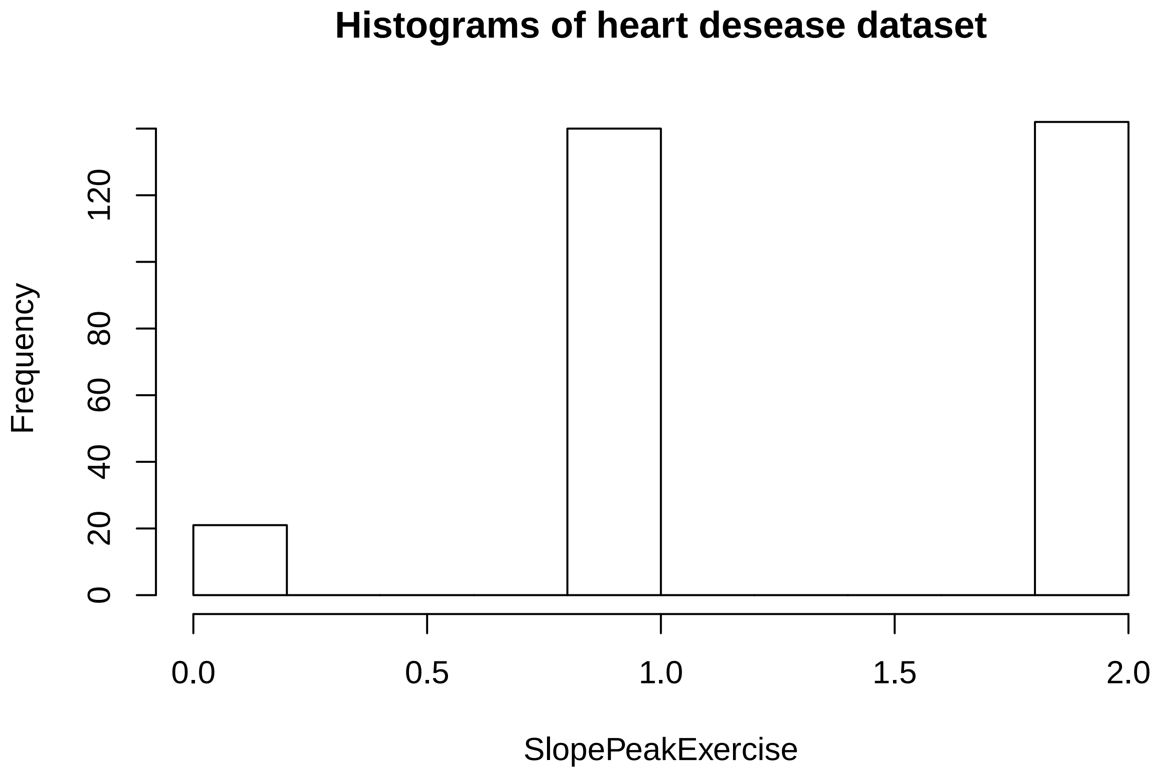

Histograms of heart desease dataset
Frequency
0
50
100
150
200
250
0.0
0.2
0.4
0.6
0.8
1.0
FastingBloodSugar

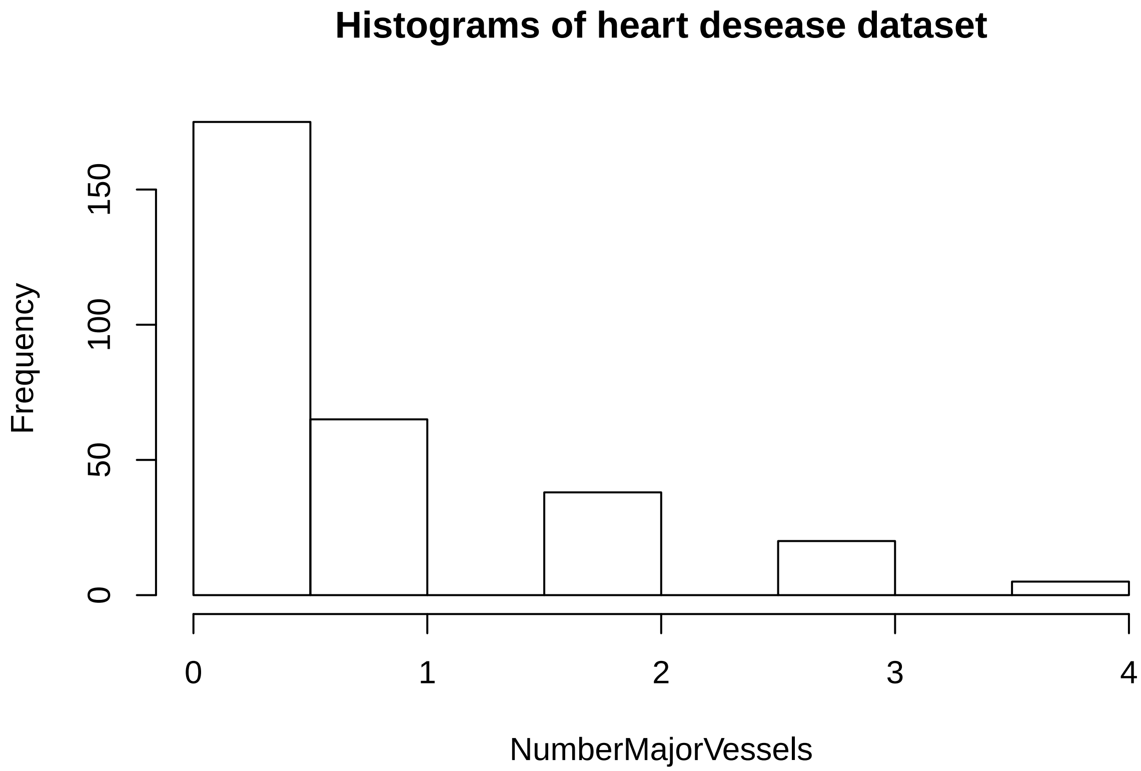

Histograms of heart desease dataset
Frequency
0
50
100
150
0.0
0.5
1.0
1.5
2.0
RestingElectrocardiographic

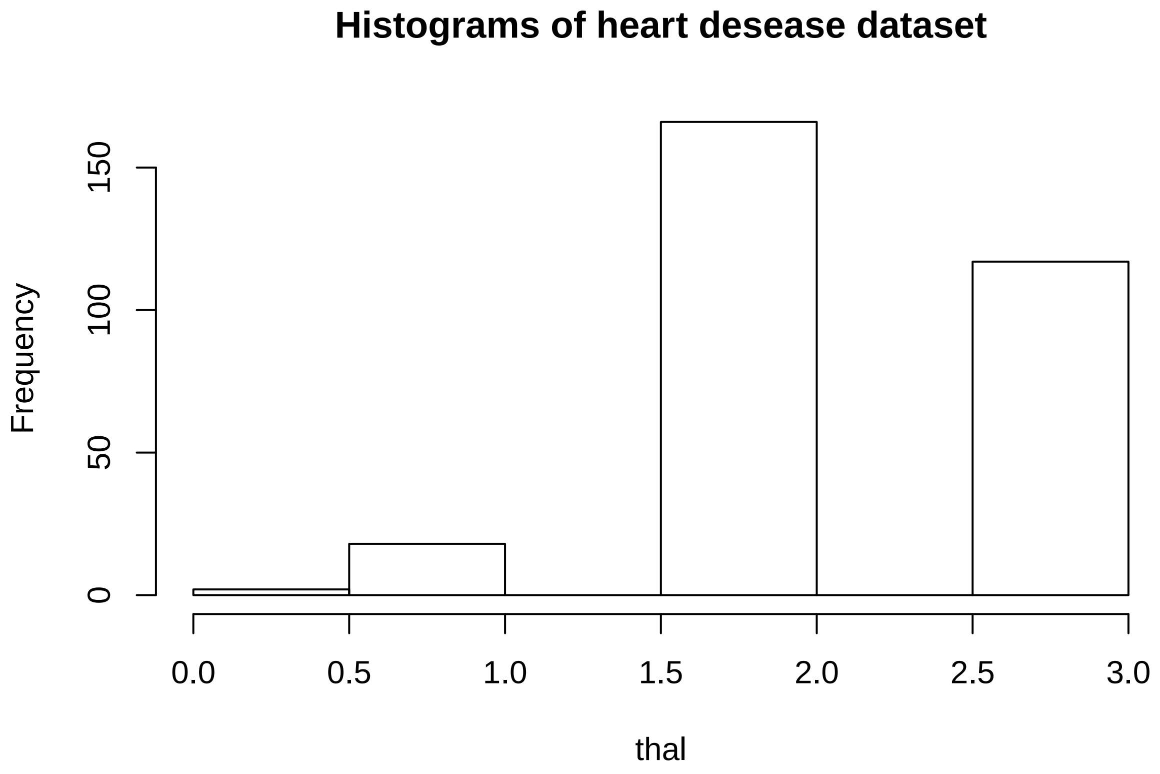

Histograms of heart desease dataset
Frequency
0
10
20
30
40
50
80
100
120
140
160
180
200
MaximumHeartRate

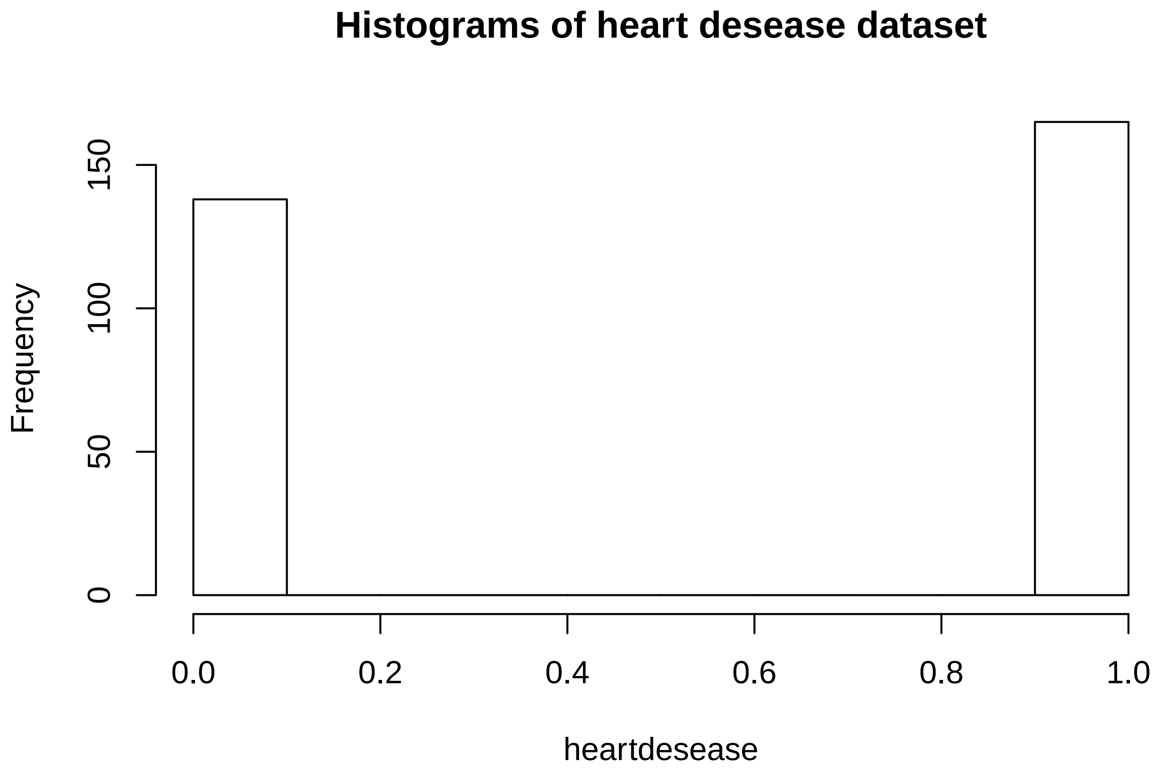

Histograms of heart desease dataset
Frequency
0
50
100
150
200
0.0
0.2
0.4
0.6
0.8
1.0
ExerciseAngina


Histograms of heart desease dataset
Frequency
0
20
40
60
80
120
0
1
2
3
4
5
6
STDepressionExercise


Histograms of heart desease dataset
Frequency
0
20
40
60
80
120
0.0
0.5
1.0
1.5
2.0
SlopePeakExercise


Histograms of heart desease dataset
Frequency
0
50
100
150
0
1
2
3
4
NumberMajorVessels


Histograms of heart desease dataset
Frequency
0
50
100
150
0.0
0.5
1.0
1.5
2.0
2.5
3.0
thal



# Show a data summary

```
summary(HeartDeaseaseData)
```

```
##       age             sex          ChestPainType   RestingBloodPressure
##  Min.   :29.00   Min.   :0.0000   Min.   :0.000   Min.   : 94.0
##  1st Qu.:47.50   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:120.0
##  Median :55.00   Median :1.0000   Median :1.000   Median :130.0
##  Mean   :54.37   Mean   :0.6832   Mean   :0.967   Mean   :131.6
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000   Max.   :3.000   Max.   :200.0
##  SerumCholestoral FastingBloodSugar RestingElectrocardiographic
##  Min.   :126.0    Min.   :0.0000    Min.   :0.0000
##  1st Qu.:211.0    1st Qu.:0.0000    1st Qu.:0.0000
##  Median :240.0    Median :0.0000    Median :1.0000
##  Mean   :246.3    Mean   :0.1485    Mean   :0.5281
##  3rd Qu.:274.5    3rd Qu.:0.0000    3rd Qu.:1.0000
##  Max.   :564.0    Max.   :1.0000    Max.   :2.0000
##  MaximumHeartRate ExerciseAngina   STDepressionExercise SlopePeakExercise
##  Min.   : 71.0    Min.   :0.0000   Min.   :0.00         Min.   :0.000
##  1st Qu.:133.5    1st Qu.:0.0000   1st Qu.:0.00         1st Qu.:1.000
##  Median :153.0    Median :0.0000   Median :0.80         Median :1.000
##  Mean   :149.6    Mean   :0.3267   Mean   :1.04         Mean   :1.399
##  3rd Qu.:166.0    3rd Qu.:1.0000   3rd Qu.:1.60         3rd Qu.:2.000
##  Max.   :202.0    Max.   :1.0000   Max.   :6.20         Max.   :2.000
##  NumberMajorVessels      thal        heartdesease
##  Min.   :0.0000     Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.0000     1st Qu.:2.000   1st Qu.:0.0000
##  Median :0.0000     Median :2.000   Median :1.0000
##  Mean   :0.7294     Mean   :2.314   Mean   :0.5446
##  3rd Qu.:1.0000     3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :4.0000     Max.   :3.000   Max.   :1.0000
```

Find risk factors for heart desease

Compute correlation coefficients with target variable for all features

```
correlations <- NULL
for (i in (1:(ncol(HeartDeaseaseData)-1))){
correlations[i] <- cor(HeartDeaseaseData[,i],HeartDeaseaseData$heartdesease)
}

names(correlations) <-names(HeartDeaseaseData[-ncol(HeartDeaseaseData)])
as.data.frame(correlations)
```

```
##                              correlations
## age                           -0.22543872
## sex                           -0.28093658
## ChestPainType                  0.43379826
## RestingBloodPressure          -0.14493113
## SerumCholestoral              -0.08523911
## FastingBloodSugar             -0.02804576
## RestingElectrocardiographic    0.13722950
## MaximumHeartRate               0.42174093
## ExerciseAngina                -0.43675708
## STDepressionExercise          -0.43069600
## SlopePeakExercise              0.34587708
## NumberMajorVessels            -0.39172399
## thal                          -0.34402927
```

```
sort(correlations, decreasing=TRUE)
```

```
##               ChestPainType            MaximumHeartRate
##                  0.43379826                  0.42174093
##           SlopePeakExercise RestingElectrocardiographic
##                  0.34587708                  0.13722950
##           FastingBloodSugar            SerumCholestoral
##                 -0.02804576                 -0.08523911
##        RestingBloodPressure                         age
##                 -0.14493113                 -0.22543872
##                         sex                        thal
##                 -0.28093658                 -0.34402927
##          NumberMajorVessels        STDepressionExercise
##                 -0.39172399                 -0.43069600
##              ExerciseAngina
##                 -0.43675708
```

Findings from correlation analysis:

*The Chest Pain Type as well as the maximum heart rate are important risk factors for heart deseases

Setup Training and Testing Datasets

Covert target variable to factor Shuffle data *Split into training data (70%) and test data (30%)* check dimensions

```r
ColsToFactors <- c(
            #"sex",
            #"ChestPainType",
            #"FastingBloodSugar",
            #"RestingElectrocardiographic",
            #"ExerciseAngina",
            #"SlopePeakExercise",
            #"NumberMajorVessels",
            #"thal",
            "heartdesease")

HeartDeaseaseData[,ColsToFactors] <- as.factor(HeartDeaseaseData[,ColsToFactors])


RowIndices <- sample((1:nrow(HeartDeaseaseData)))
HeartDeaseaseData <- HeartDeaseaseData[RowIndices,]
split <- floor(0.7*nrow(HeartDeaseaseData))
training <- HeartDeaseaseData[(1:split),]
test <- HeartDeaseaseData[((split+1):nrow(HeartDeaseaseData)),]

dim(training)
```

```
## [1] 212  14
```

```r
dim(test)
```

```
## [1] 91 14
```

Apply Machine Learning Models on dataset

- Logistic Regression
- PCA Regression
- Ridge, LASSO and Elastic Net Regression
- Random Forest Model

Logistic regression

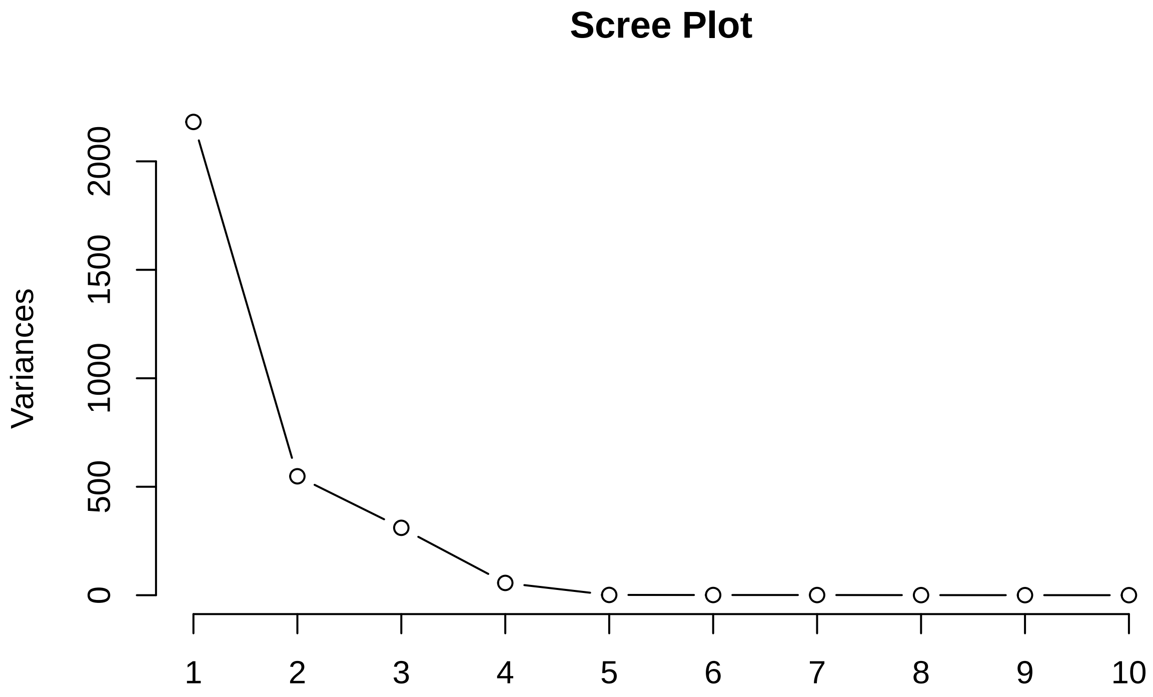
```r
LogReg_model <- train(as.factor(heartdesease)~., data=training,  method="glm")

  # Out-of-Sample predictions
  LogReg_predictions <-
    predict(
      LogReg_model,
      newdata = test,  type="prob")

  # Out-of-sample forecast error:
  confusionMatrix(
    as.factor(as.numeric(LogReg_predictions[,2]>0.5)),
    as.factor(test$heartdesease))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 27  6
##          1  9 49
##
##                Accuracy : 0.8352
##                  95% CI : (0.7427, 0.9047)
##     No Information Rate : 0.6044
##     P-Value [Acc > NIR] : 1.684e-06
##
##                   Kappa : 0.6503
##  Mcnemar's Test P-Value : 0.6056
##
##             Sensitivity : 0.7500
##             Specificity : 0.8909
##          Pos Pred Value : 0.8182
##          Neg Pred Value : 0.8448
##              Prevalence : 0.3956
##          Detection Rate : 0.2967
##    Detection Prevalence : 0.3626
##       Balanced Accuracy : 0.8205
##
##        'Positive' Class : 0
##
```

Logistic PCA regression

—-

```r
training.pca <- prcomp(training[,-ncol(training)])

# Scree-Plot to determine number of PCA factors
plot(training.pca , type="l", main="Scree Plot")
```



```
# steep decline of the slope until the 5th PCA facor -> Include 5 PCA factors in PCA regression model

# Set Training Parameter: Cross Validation & number of PCA Factors
  #                         for PCA Regression
  fitControlPCA <-
    trainControl(method="none",
                 preProcOptions=list(pcaComp=5),
                 verboseIter = TRUE,
                 allowParallel = TRUE)

# Logistic PCA Regression with 5 PCA Factors
  PCR_model <-
      train(
      as.factor(heartdesease)~.,
      data=training,
      method="glm",
      trControl = fitControlPCA,
      preProcess="pca",
      weights = NULL
    )
```

```
## Fitting parameter = none on full training set
```

```
  # Out-of-Sample predictions
  PCR_predictions <-
    predict(
      PCR_model,
      newdata = test,  type="prob")

  # Out-of-sample forecast error:
  confusionMatrix(
    as.factor(as.numeric(PCR_predictions[,2]>0.5)),
    as.factor(test$heartdesease))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 28  6
##          1  8 49
##
##                Accuracy : 0.8462
##                  95% CI : (0.7554, 0.9133)
##     No Information Rate : 0.6044
##     P-Value [Acc > NIR] : 4.859e-07
##
##                   Kappa : 0.6752
##  Mcnemar's Test P-Value : 0.7893
##
##             Sensitivity : 0.7778
##             Specificity : 0.8909
##          Pos Pred Value : 0.8235
##          Neg Pred Value : 0.8596
##              Prevalence : 0.3956
##          Detection Rate : 0.3077
##    Detection Prevalence : 0.3736
##       Balanced Accuracy : 0.8343
##
##        'Positive' Class : 0
##
```

Logistic, Ridge-, LASSO- and Elastic Net Regression ——————————

```
#  set Parameters --------------------------------------
  set.seed(1234)
  # Initialize Parameters
  Elastic_NetParameter <- data.frame(alpha=NaN, lambda=NaN)
  LASSO_Parameter      <- data.frame(alpha=NaN, lambda=NaN)
  RidgeReg_Parameter   <- data.frame(alpha=NaN, lambda=NaN)

  # Set Grids for hyperparameters
  Ridge_Grid   <-
    expand.grid(alpha  = 0,
                lambda = exp(seq(-9.21034,9.21034,length.out = 100)))
  LASSO_Grid   <-
    expand.grid(alpha  = 1,
                lambda = exp(seq(-9.21034,9.21034,length.out = 100)))
  Elastic_Grid <-
```

```
    expand.grid(alpha  = seq(0,1,length.out = 100),
                lambda = exp(seq(-9.21034,9.21034,length.out = 100)))

  # Set Training Parameter: Cross Validation
  fitControl <- trainControl(method="cv",
                             number=5,
                             verboseIter = TRUE,
                             allowParallel = TRUE)



# Logistic Ridge regression ----------------------------------------------------------
  RidgeReg_model <-
    train(as.factor(heartdesease)~.,
          data=training,
          method="glmnet",
          trControl  = fitControl,
          tuneGrid   = Ridge_Grid,
          preProcess = NULL,
          weights    = NULL)
```

```
## + Fold1: alpha=0, lambda=10000
## - Fold1: alpha=0, lambda=10000
## + Fold2: alpha=0, lambda=10000
## - Fold2: alpha=0, lambda=10000
## + Fold3: alpha=0, lambda=10000
## - Fold3: alpha=0, lambda=10000
## + Fold4: alpha=0, lambda=10000
## - Fold4: alpha=0, lambda=10000
## + Fold5: alpha=0, lambda=10000
## - Fold5: alpha=0, lambda=10000
## Aggregating results
## Selecting tuning parameters
## Fitting alpha = 0, lambda = 0.0673 on full training set
```

```
  # Parameter of the best Ridge-Model (in cross-valiation)
  RidgeReg_Parameter <- RidgeReg_model$bestTune
  RidgeReg_Parameter
```

```
##    alpha     lambda
## 36     0 0.06734151
```

```
  # Ridge Regression Prediction
  # Out-of-sample-Sample predictions
  RidgeReg_predictions <-
    predict(RidgeReg_model, test, type="prob")
  # Out-of-sample forecast error:
    # Out-of-sample forecast error:
  confusionMatrix(
    as.factor(as.numeric(RidgeReg_predictions[,2]>0.5)),
    as.factor(test$heartdesease))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 26  5
##          1 10 50
##
##                Accuracy : 0.8352
##                  95% CI : (0.7427, 0.9047)
##     No Information Rate : 0.6044
##     P-Value [Acc > NIR] : 1.684e-06
##
##                   Kappa : 0.6468
##  Mcnemar's Test P-Value : 0.3017
##
##             Sensitivity : 0.7222
##             Specificity : 0.9091
##          Pos Pred Value : 0.8387
##          Neg Pred Value : 0.8333
##              Prevalence : 0.3956
##          Detection Rate : 0.2857
##    Detection Prevalence : 0.3407
##       Balanced Accuracy : 0.8157
##
##        'Positive' Class : 0
##
```

```
# Logistic LASSO regression ----------------------------------------------------------
  LASSO_model <- train(
    as.factor(heartdesease)~.,
    data=training,
    method="glmnet",
    trControl=fitControl,
    tuneGrid=LASSO_Grid,
    preProcess=NULL,weights=NULL
  )
```

```
## + Fold1: alpha=1, lambda=10000
## - Fold1: alpha=1, lambda=10000
## + Fold2: alpha=1, lambda=10000
## - Fold2: alpha=1, lambda=10000
## + Fold3: alpha=1, lambda=10000
## - Fold3: alpha=1, lambda=10000
## + Fold4: alpha=1, lambda=10000
## - Fold4: alpha=1, lambda=10000
## + Fold5: alpha=1, lambda=10000
## - Fold5: alpha=1, lambda=10000
## Aggregating results
## Selecting tuning parameters
## Fitting alpha = 1, lambda = 0.00413 on full training set
```

```
  # Parameter of the best LASSO-Model (in cross-valiation)
  LASSO_Parameter <- LASSO_model$bestTune
  LASSO_Parameter
```

```
##    alpha      lambda
## 21     1 0.004132013
```

```
  # Out-of-sample-Sample predictions
  LASSO_predictions <-
    predict(LASSO_model, test, type="prob")
  # Out-of-sample forecast error:
   confusionMatrix(
    as.factor(as.numeric(LASSO_predictions[,2]>0.5)),
    as.factor(test$heartdesease))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 27  5
##          1  9 50
##
##                Accuracy : 0.8462
##                  95% CI : (0.7554, 0.9133)
##     No Information Rate : 0.6044
##     P-Value [Acc > NIR] : 4.859e-07
##
##                   Kappa : 0.672
##  Mcnemar's Test P-Value : 0.4227
##
##             Sensitivity : 0.7500
##             Specificity : 0.9091
##          Pos Pred Value : 0.8437
##          Neg Pred Value : 0.8475
##              Prevalence : 0.3956
##          Detection Rate : 0.2967
##    Detection Prevalence : 0.3516
##       Balanced Accuracy : 0.8295
##
##        'Positive' Class : 0
##
```

```
# Logistic Elastic Net regression ----------------------------------------------------------
  Elastic_Net_model <- train(
    as.factor(heartdesease)~.,
    data=training,
    method="glmnet",
    trControl=fitControl,preProcess=NULL,weights=NULL
  )
```

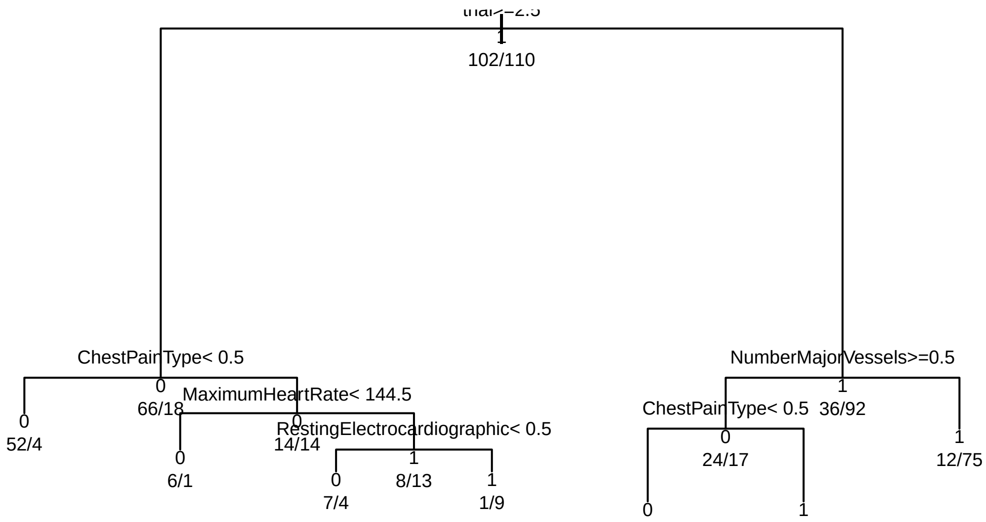
```
## + Fold1: alpha=0.10, lambda=0.04376
## - Fold1: alpha=0.10, lambda=0.04376
## + Fold1: alpha=0.55, lambda=0.04376
## - Fold1: alpha=0.55, lambda=0.04376
## + Fold1: alpha=1.00, lambda=0.04376
## - Fold1: alpha=1.00, lambda=0.04376
## + Fold2: alpha=0.10, lambda=0.04376
## - Fold2: alpha=0.10, lambda=0.04376
## + Fold2: alpha=0.55, lambda=0.04376
## - Fold2: alpha=0.55, lambda=0.04376
## + Fold2: alpha=1.00, lambda=0.04376
## - Fold2: alpha=1.00, lambda=0.04376
## + Fold3: alpha=0.10, lambda=0.04376
## - Fold3: alpha=0.10, lambda=0.04376
## + Fold3: alpha=0.55, lambda=0.04376
## - Fold3: alpha=0.55, lambda=0.04376
## + Fold3: alpha=1.00, lambda=0.04376
## - Fold3: alpha=1.00, lambda=0.04376
## + Fold4: alpha=0.10, lambda=0.04376
## - Fold4: alpha=0.10, lambda=0.04376
## + Fold4: alpha=0.55, lambda=0.04376
## - Fold4: alpha=0.55, lambda=0.04376
## + Fold4: alpha=1.00, lambda=0.04376
## - Fold4: alpha=1.00, lambda=0.04376
## + Fold5: alpha=0.10, lambda=0.04376
## - Fold5: alpha=0.10, lambda=0.04376
## + Fold5: alpha=0.55, lambda=0.04376
## - Fold5: alpha=0.55, lambda=0.04376
## + Fold5: alpha=1.00, lambda=0.04376
## - Fold5: alpha=1.00, lambda=0.04376
## Aggregating results
## Selecting tuning parameters
## Fitting alpha = 0.1, lambda = 0.00438 on full training set
```

```r
# Parameter of the best Elastic-Net-Model (in cross-valiation)
Elastic_NetParameter <- Elastic_Net_model$bestTune


# Out-of-sample-Sample predictions
Elastic_Net_predictions <-
  predict(Elastic_Net_model, test, type="prob")
# Out-of-sample forecast error:
confusionMatrix(
  as.factor(as.numeric(Elastic_Net_predictions[,2]>0.5)),
  as.factor(test$heartdesease))
```

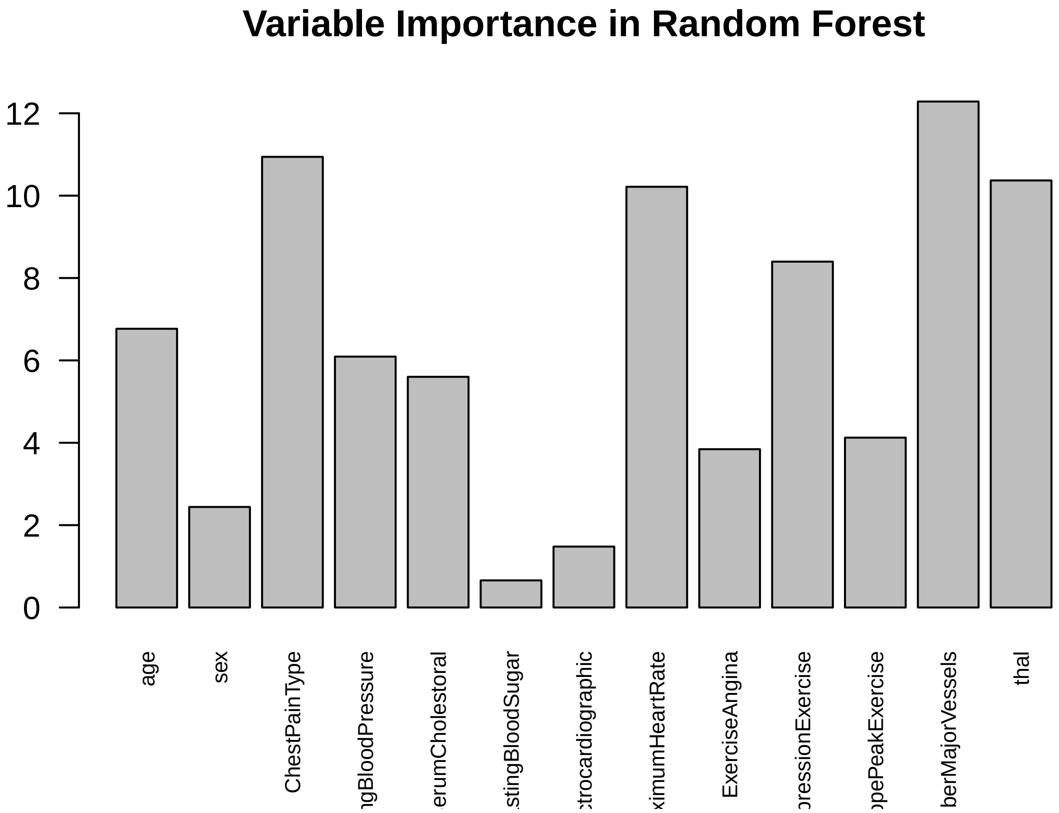
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 27  5
##          1  9 50
##
##                Accuracy : 0.8462
##                  95% CI : (0.7554, 0.9133)
##     No Information Rate : 0.6044
##     P-Value [Acc > NIR] : 4.859e-07
##
##                   Kappa : 0.672
##  Mcnemar's Test P-Value : 0.4227
##
##             Sensitivity : 0.7500
##             Specificity : 0.9091
##          Pos Pred Value : 0.8437
##          Neg Pred Value : 0.8475
##              Prevalence : 0.3956
##          Detection Rate : 0.2967
##    Detection Prevalence : 0.3516
##       Balanced Accuracy : 0.8295
##
##        'Positive' Class : 0
##
```

Random Forest

```
# Decision Tree -----------------------------------------------
  # grow tree
  tree <- rpart(as.factor(heartdesease)~.,
           data=training)

# Plot decision tree as example
  plot(tree)
  text(tree, use.n=TRUE, all=TRUE, cex=.6)
```



```
#  Random Forest -----------------------------------------------
  set.seed(1234)
  rf_model <-
    ranger (as.factor(heartdesease)~.,
           data=training,
           probability = TRUE,
           num.trees = 150,
           splitrule="gini",
           importance="impurity",
          mtry=sqrt(ncol(training)-1),
          verbose = TRUE)

barplot(rf_model$variable.importance, cex.names=0.7, horiz=FALSE, las=2,
        main="Variable Importance in Random Forest")
```



```
# Out-of-sample-Sample predictions
rf_predictions <- predict(rf_model,test)
rf_predictions <- rf_predictions$predictions
# Out-of-sample forecast error:
confusionMatrix(
  as.factor(as.numeric(rf_predictions[,2]>0.5)),
  as.factor(test$heartdesease))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 28  8
##          1  8 47
##
##                Accuracy : 0.8242
##                  95% CI : (0.7302, 0.896)
##     No Information Rate : 0.6044
##     P-Value [Acc > NIR] : 5.408e-06
##
```

```
##                   Kappa : 0.6323
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.7778
##             Specificity : 0.8545
##          Pos Pred Value : 0.7778
##          Neg Pred Value : 0.8545
##              Prevalence : 0.3956
##          Detection Rate : 0.3077
##    Detection Prevalence : 0.3956
##       Balanced Accuracy : 0.8162
##
##        'Positive' Class : 0
##
```

Conclusion:

- The random forest model achieves a similar performance compared to the regression models on the test set.
- The variable importance plot shows that the most important variables in the random forest model are:

1. Number of Major Vessels
2. MaximumHeartRate (Feature 8: maximum heart rate achieved)
3. ChestPainType (Feature 3: chest pain type (4 values))
4. DepressionExercise (Feature 10. oldpeak = ST depression induced by exercise relative to rest)

- The Ridge Regression Model achieves the best accuracy on the test set, slightly better than the PCA Regression, LASSO, ELastic Net and Random Forest