Heart Disease

Code

Vitor Juliani, Rodrigo Soares

Hide

```
# ==== Libraries import ====
library(ggplot2)
library(class)
library(caret)
library(e1071)
library(randomForest)
library(factoextra)
```

Hide

```
# ==== Test and Train ====
buildTestAndTrain <- function(dataframe, seed_value, percent) {

  # Getting dataframe class
  classes <- as.factor(dataframe[, ncol(dataframe)])

  # Selecting data for test and training
  set.seed(seed_value)

  sample_size <- floor(percent * nrow(dataframe))
  train_index <-
    sample(seq_len(nrow(dataframe)), size = sample_size)

  # Preparing test object and training
  train_without_column <- dataframe[train_index,-ncol(dataframe)]
  train <- dataframe[train_index, ]
  test <- dataframe[-train_index,-ncol(dataframe)]

  # Selecting the test class column and train class column
  trainClass <- classes[train_index]
  testClass <- classes[-train_index]

  return (
    list(
      "train_without_column" = train_without_column,
      "train" = train,
      "test" = test,
      "trainClass" = trainClass,
      "testClass" = testClass
    )
  )
}
```

Hide

```
# ==== KNN classification ====
knnClassification <- function(train, test, testClass, trainClass, k) {

    # KNN predict
    knn_res <- knn(train, test, trainClass, k)

    # Accuracy of KNN
    cf_knn <- confusionMatrix(knn_res, testClass)

    # Plot expected
    plot(testClass, main = "Expected plot")

    # Plot predict
    plot(knn_res, main = "Predict knn plot")

    return(cf_knn$overall)

}
```

Hide

```
# ==== SVM classification ====
svmClassification <- function(train, test, testClass) {

  # Build model
  svm_classifier <- svm(
    formula = Class ~ .,
    data = train,
    type = 'C-classification',
    kernel = 'linear'
  )

  # SVM predict
  svm_res <- predict(svm_classifier, newdata = test)

  # Accuracy of SVM
  cf_svm <- confusionMatrix(svm_res, testClass)

  # Plot expected
  plot(testClass, main = "Expected plot")

  # Plot predict
  plot(svm_res, main = "Predict svm plot")

  return(cf_svm$overall)

}
```

Hide

```
# ==== RF classification ====
rfClassification <- function(train, test, testClass) {

  # Build model
  rf_classifier <- randomForest(
    formula = Class ~ .,
    data = train,
    ntree = 500,
    importance = TRUE
  )

  # RF predict
  rf_res = predict(rf_classifier, newdata = test)

  # Accuracy of RF
  cf_rf <- confusionMatrix(as.factor(rf_res), testClass)

  # Plot expected
  plot(testClass, main = "Expected plot")

  # Plot predict
  plot(rf_res, main = "Predict rf plot")

  return(cf_rf$overall)

}
```

Hide

```
# ==== Build dataset ====
heartDiseaseDataframe <-
  read.csv( "./dataset/processed.cleveland.data", fileEncoding = "UTF-8", sep = ",", header =
FALSE )

# Setting the name of the columns
colnames(heartDiseaseDataframe) <- c( "Age", "Sex", "ChestPainType", "RestBloodPressure", "Se
rumCholestoral",
                                      "FastingBloodSugar", "ResElectrocardiographic", "MaxHea
rtRate",
                                      "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels",
"Thal", "Class" )
```

Hide

```
# ==== Dataset preview ====

  # Header of dataset
  head(heartDiseaseDataframe)
```

	...	...	ChestPainType	RestBloodPressure	SerumCholestoral	FastingBloodSugar
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	63	1	1	145	233	1
2	67	1	4	160	286	0
3	67	1	4	120	229	0

...	...	ChestPainType	RestBloodPressure	SerumCholestoral	FastingBloodSugar
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
4 37	1	3	130	250	0
5 41	0	2	130	204	0
6 56	1	2	120	236	0

6 rows | 1-7 of 14 columns

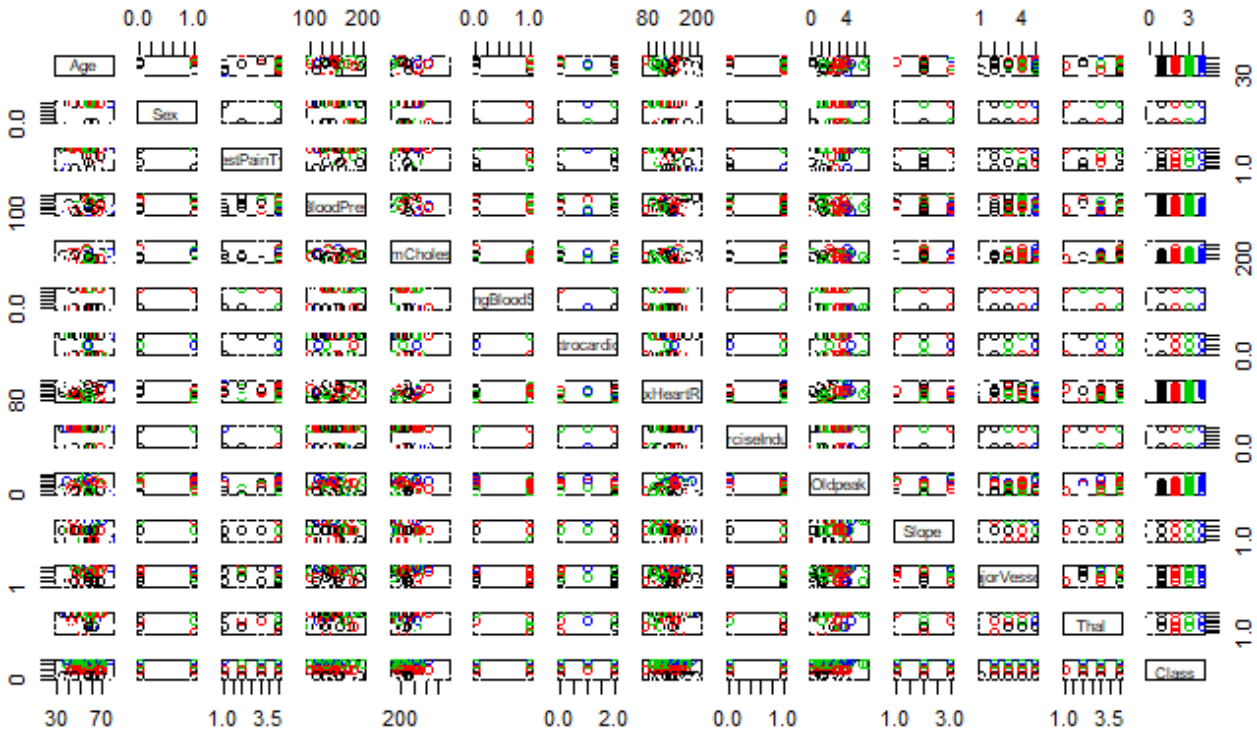
Hide

```
# Dataset structure
str(heartDiseaseDataframe)
```

```
'data.frame':   303 obs. of  14 variables:
 $ Age                    : num  63 67 67 37 41 56 62 57 63 53 ...
 $ Sex                    : num  1 1 1 1 0 1 0 0 1 1 ...
 $ ChestPainType          : num  1 4 4 3 2 2 4 4 4 4 ...
 $ RestBloodPressure      : num  145 160 120 130 130 120 140 120 130 140 ...
 $ SerumCholestoral       : num  233 286 229 250 204 236 268 354 254 203 ...
 $ FastingBloodSugar      : num  1 0 0 0 0 0 0 0 0 1 ...
 $ ResElectrocardiographic: num  2 2 2 0 2 0 2 0 2 2 ...
 $ MaxHeartRate           : num  150 108 129 187 172 178 160 163 147 155 ...
 $ ExerciseInduced        : num  0 1 1 0 0 0 0 1 0 1 ...
 $ Oldpeak                : num  2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...
 $ Slope                  : num  3 2 2 3 1 1 3 1 2 3 ...
 $ MajorVessels           : Factor w/ 5 levels "?","0.0","1.0",..: 2 5 4 2 2 2 4 2 3 2 ...
 $ Thal                   : Factor w/ 4 levels "?","3.0","6.0",..: 3 2 4 2 2 2 2 2 4 4 ...
 $ Class                  : int  0 2 1 0 0 0 3 0 2 1 ...
```

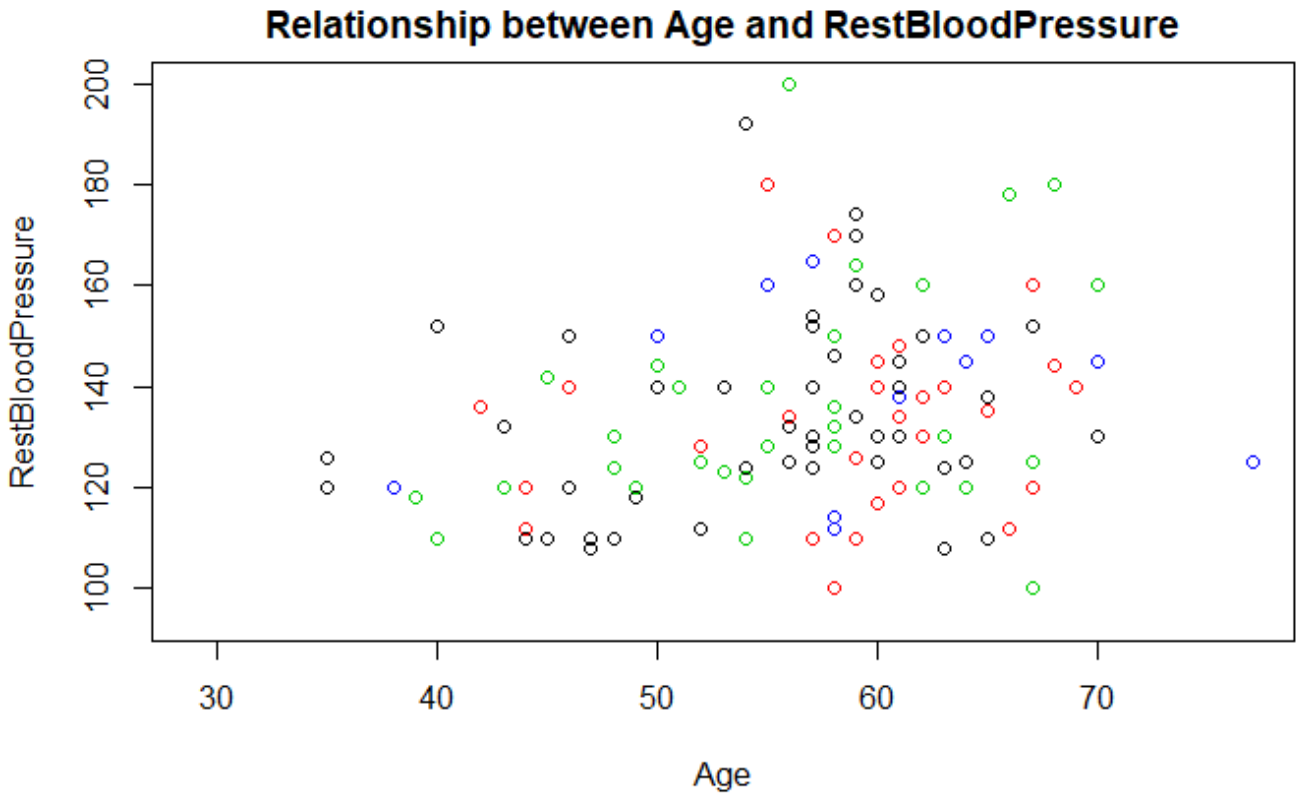
Hide

```
# ==== PLOTS ====

  # General plot
  plot(heartDiseaseDataframe, col = heartDiseaseDataframe$Class)
```



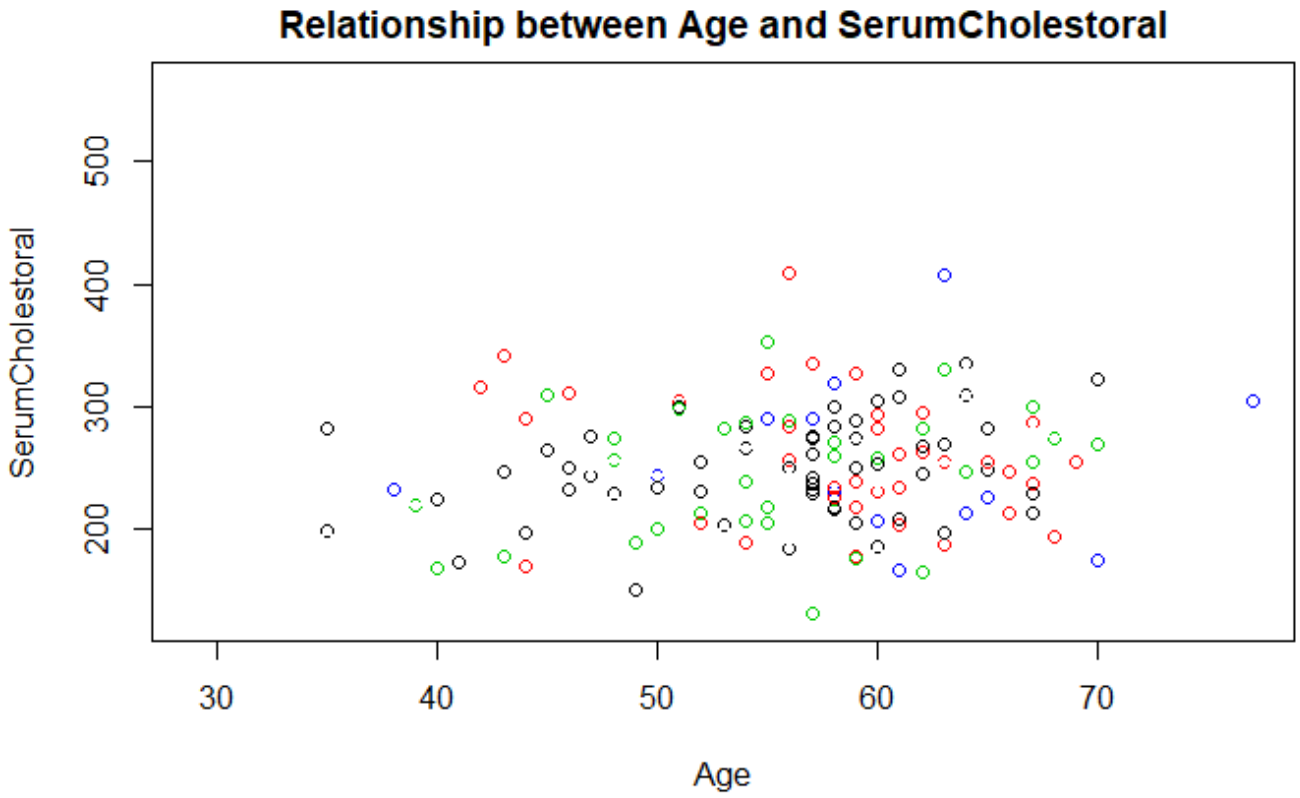
Hide

```
# Relationship between specific columns plot

  # Age x RestBloodPressure
  plot(
    heartDiseaseDataframe$Age,
    heartDiseaseDataframe$RestBloodPressure,
    col = heartDiseaseDataframe$Class,
    main = "Relationship between Age and RestBloodPressure",
    ylab = "RestBloodPressure",
    xlab = "Age"
  )
```



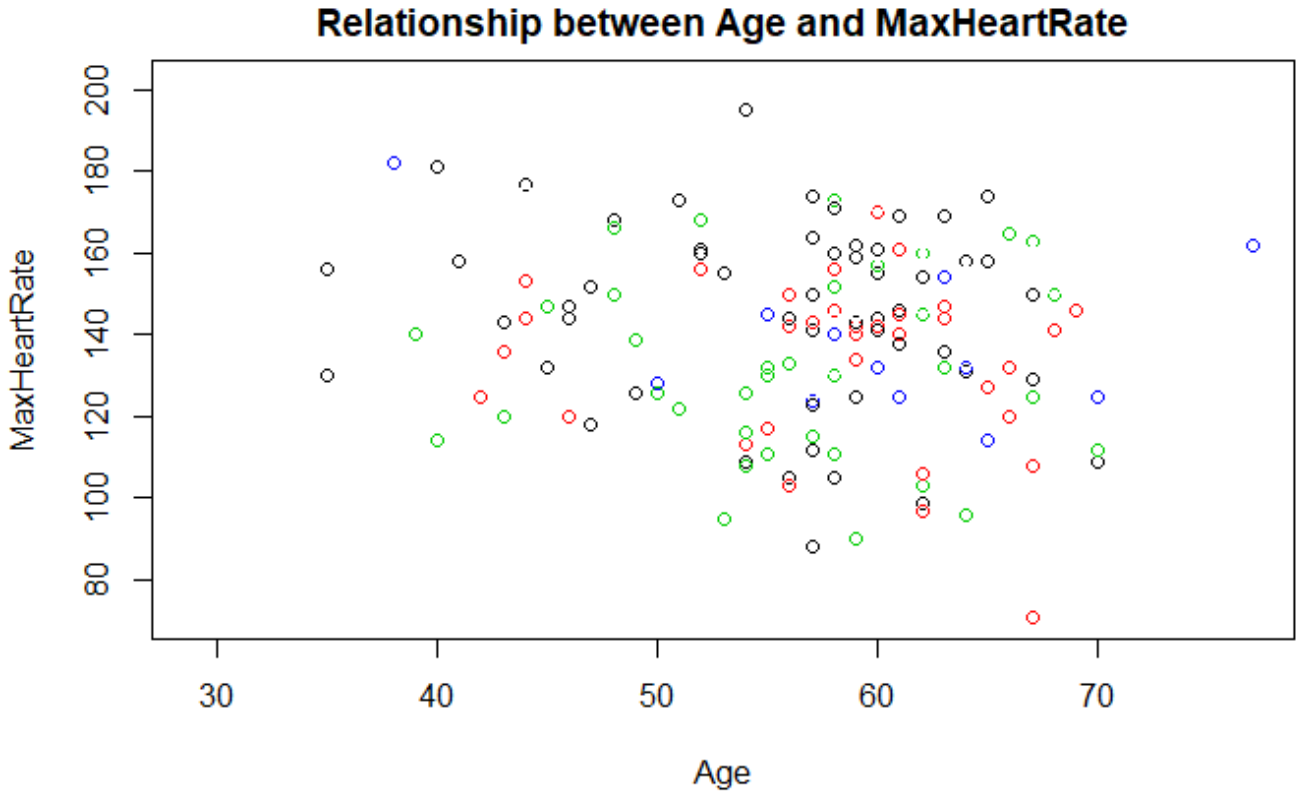
Hide

```
# Age x SerumCholestoral
plot(
  heartDiseaseDataframe$Age,
  heartDiseaseDataframe$SerumCholestoral,
  col = heartDiseaseDataframe$Class,
  main = "Relationship between Age and SerumCholestoral",
  ylab = "SerumCholestoral",
  xlab = "Age"
)
```



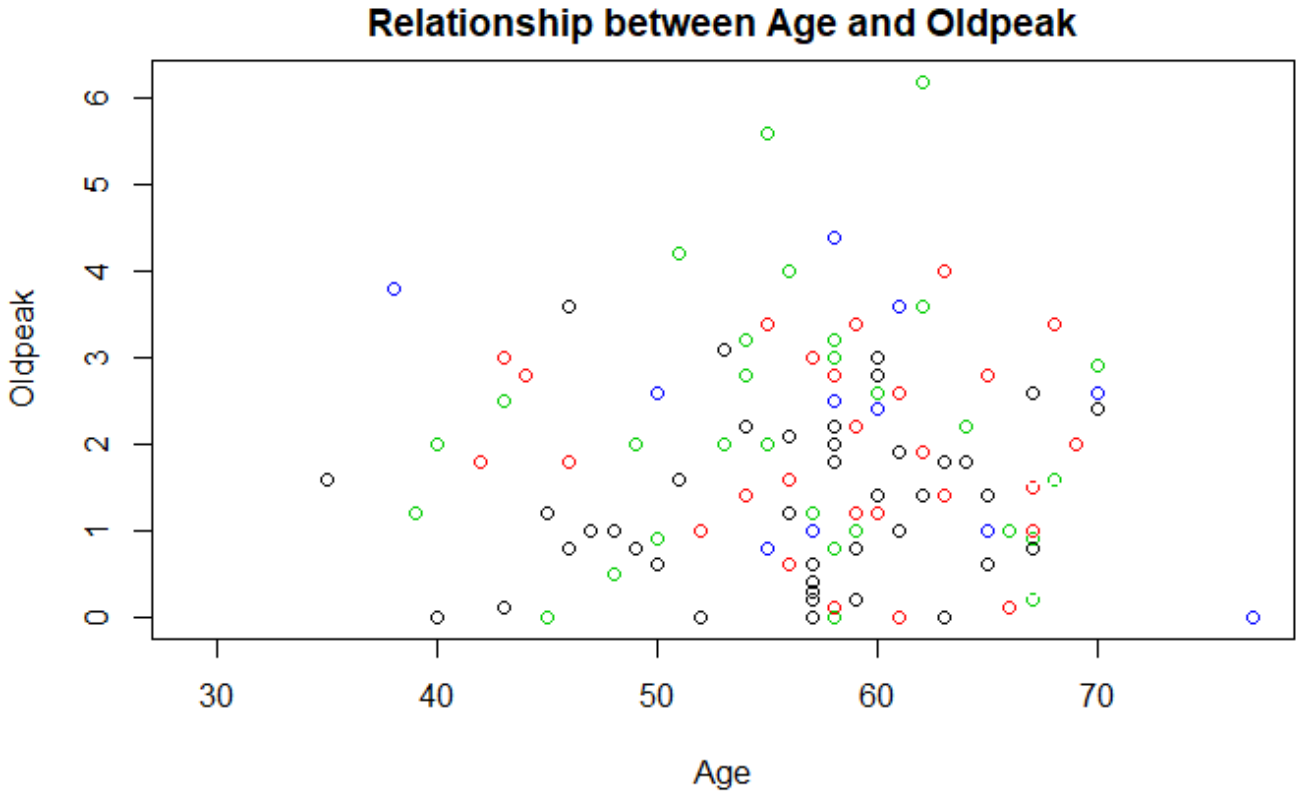
Hide

```
# Age x MaxHeartRate
plot(
  heartDiseaseDataframe$Age,
  heartDiseaseDataframe$MaxHeartRate,
  col = heartDiseaseDataframe$Class,
  main = "Relationship between Age and MaxHeartRate",
  ylab = "MaxHeartRate",
  xlab = "Age"
)
```



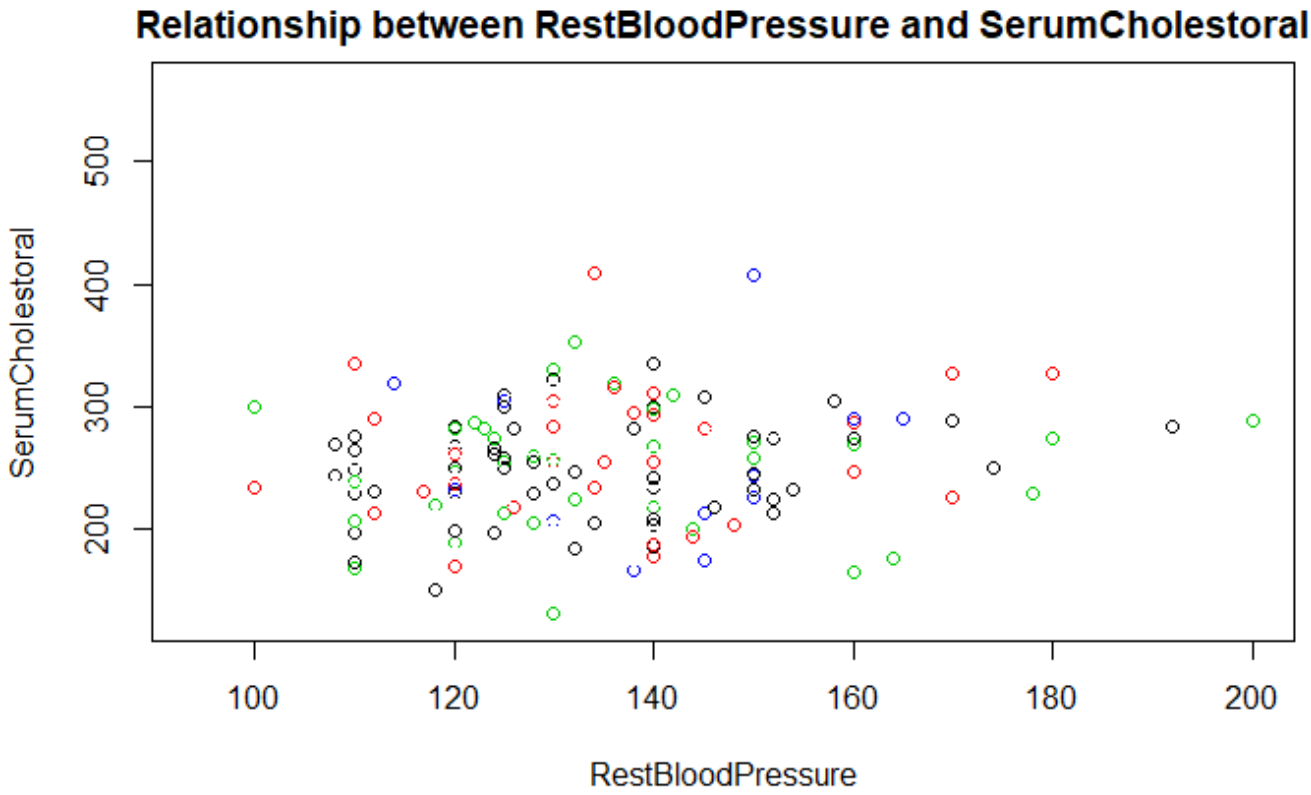
Hide

```
# Age x Oldpeak
plot(
  heartDiseaseDataframe$Age,
  heartDiseaseDataframe$Oldpeak,
  col = heartDiseaseDataframe$Class,
  main = "Relationship between Age and Oldpeak",
  ylab = "Oldpeak",
  xlab = "Age"
)
```



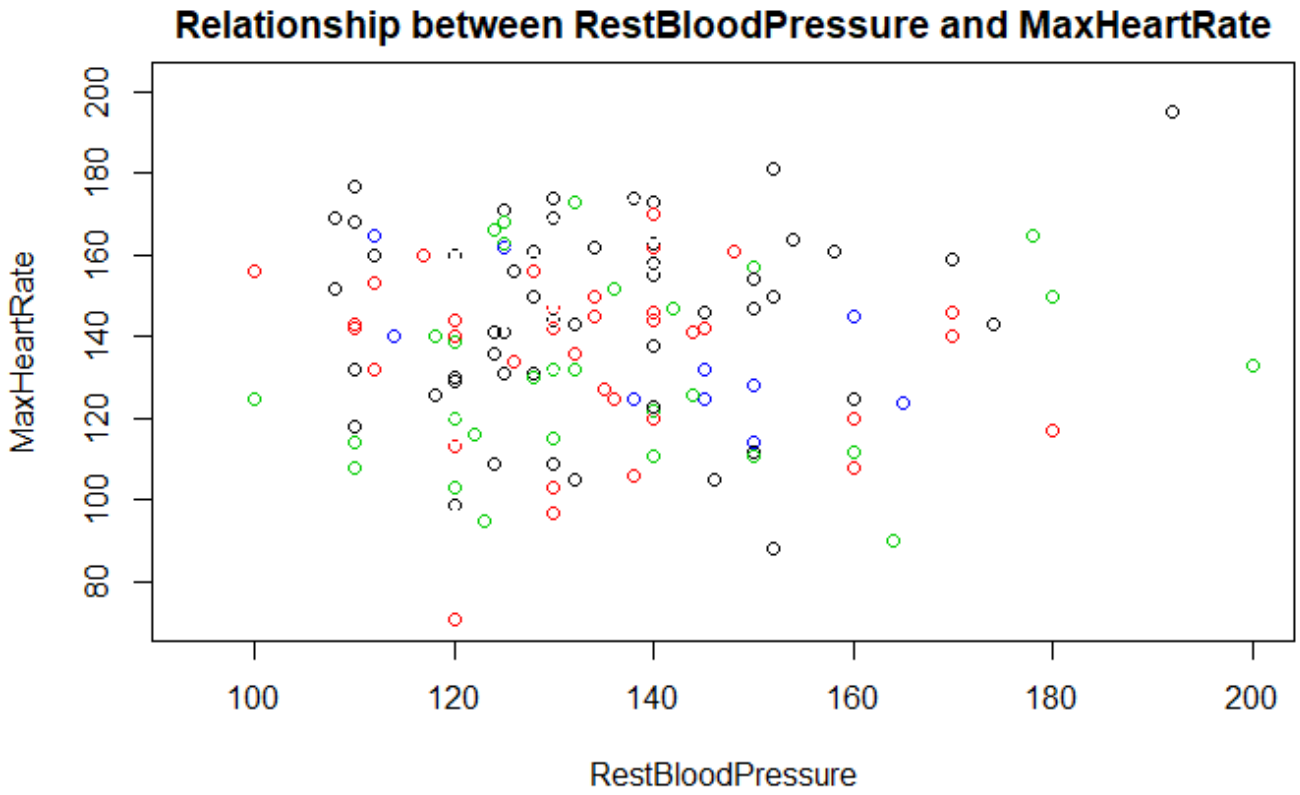
Hide

```
# Age x SerumCholestoral
plot(
  heartDiseaseDataframe$RestBloodPressure,
  heartDiseaseDataframe$SerumCholestoral,
  col = heartDiseaseDataframe$Class,
  main = "Relationship between RestBloodPressure and SerumCholestoral",
  ylab = "SerumCholestoral",
  xlab = "RestBloodPressure"
)
```



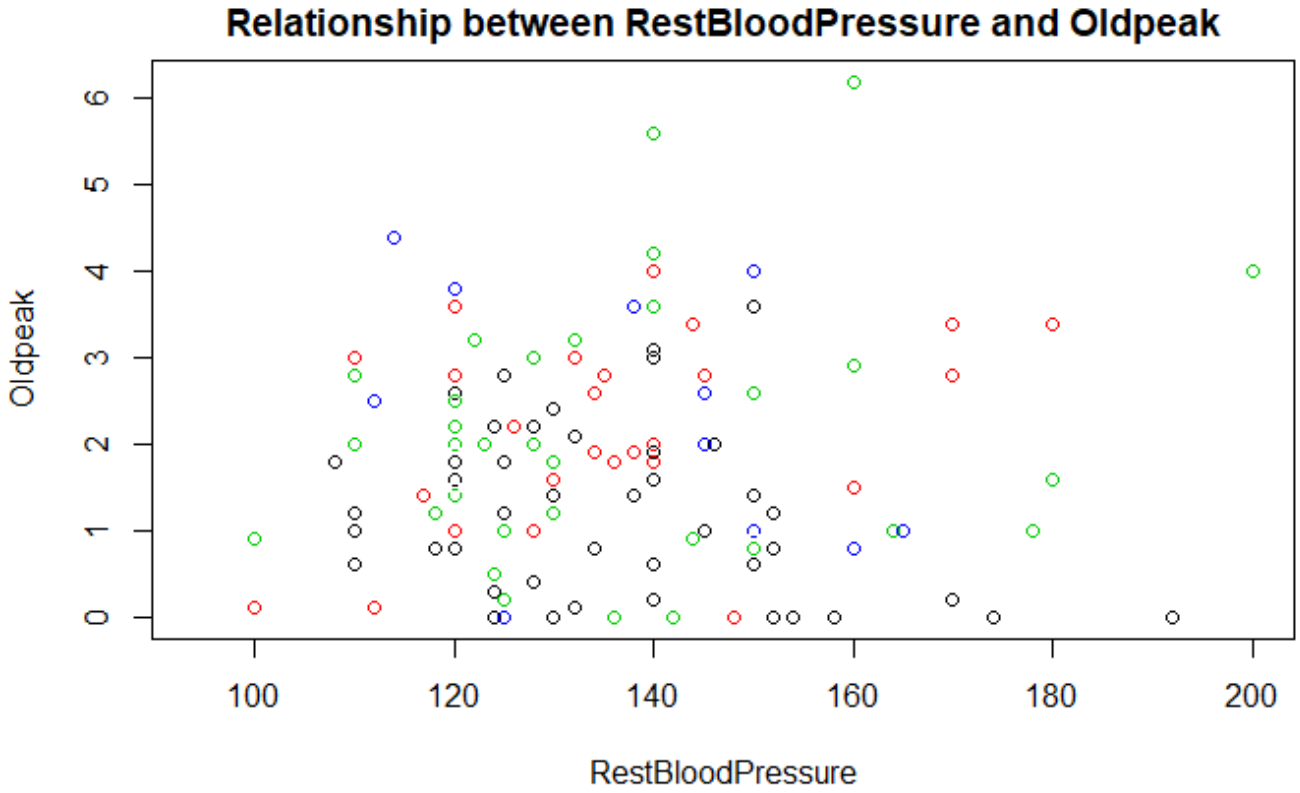
Hide

```
# RestBloodPressure x MaxHeartRate
plot(
  heartDiseaseDataframe$RestBloodPressure,
  heartDiseaseDataframe$MaxHeartRate,
  col = heartDiseaseDataframe$Class,
  main = "Relationship between RestBloodPressure and MaxHeartRate",
  ylab = "MaxHeartRate",
  xlab = "RestBloodPressure"
)
```



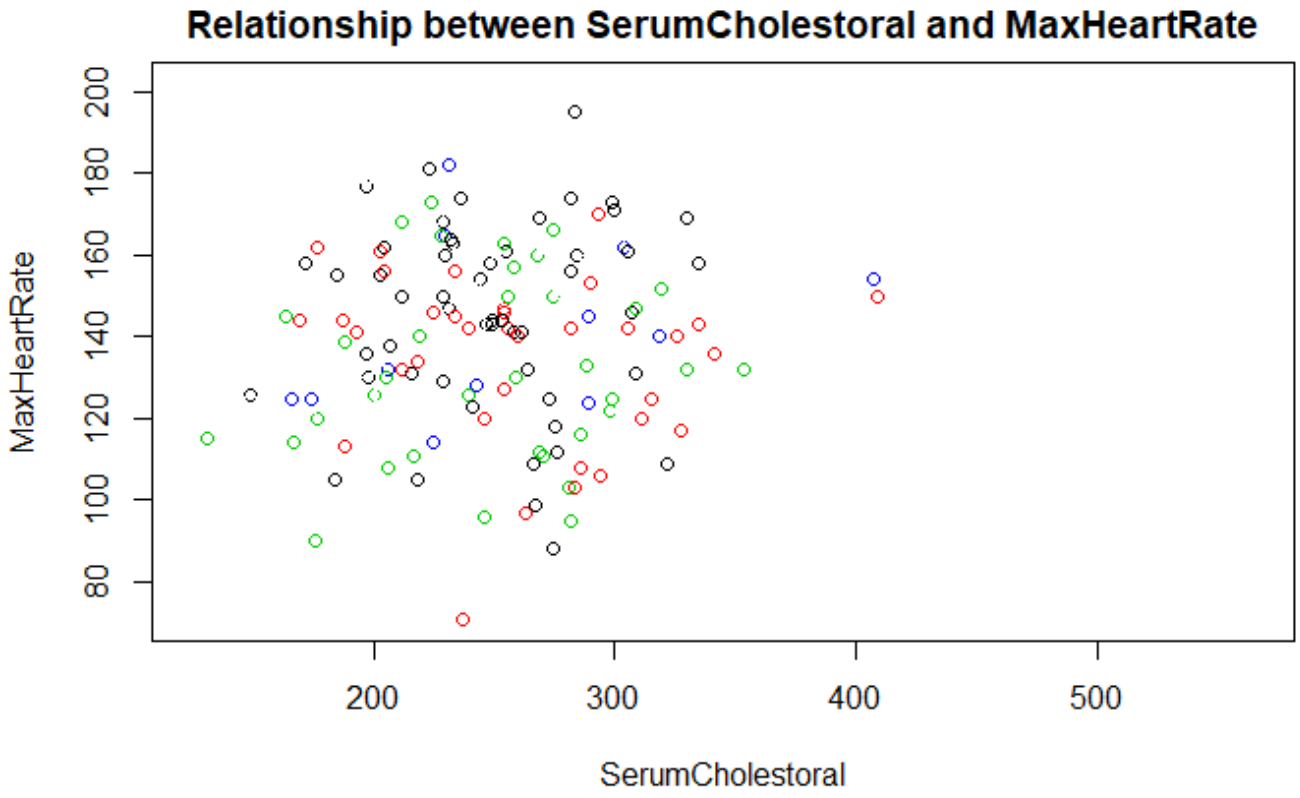
Hide

```
# RestBloodPressure x Oldpeak
plot(
  heartDiseaseDataframe$RestBloodPressure,
  heartDiseaseDataframe$Oldpeak,
  col = heartDiseaseDataframe$Class,
  main = "Relationship between RestBloodPressure and Oldpeak",
  ylab = "Oldpeak",
  xlab = "RestBloodPressure"
)
```



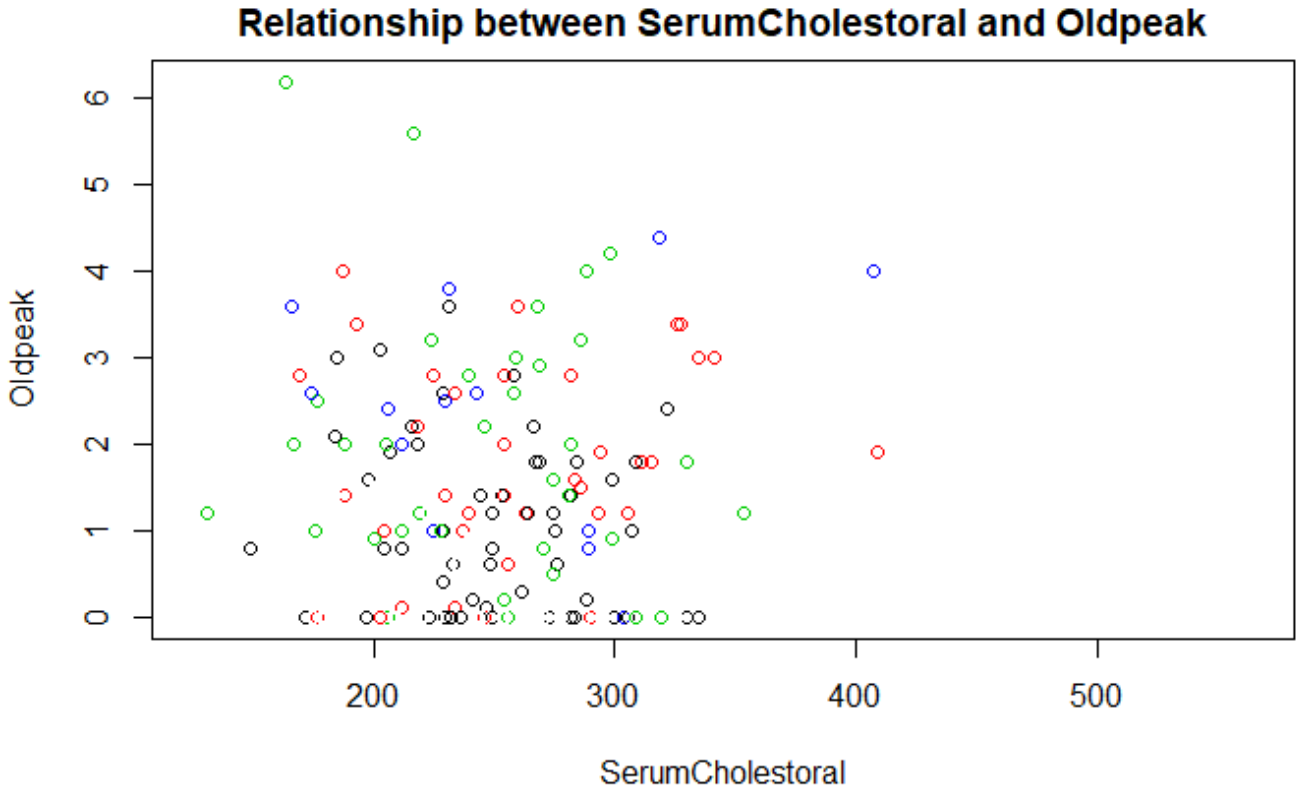
Hide

```
# SerumCholestoral x MaxHeartRate
plot(
  heartDiseaseDataframe$SerumCholestoral,
  heartDiseaseDataframe$MaxHeartRate,
  col = heartDiseaseDataframe$Class,
  main = "Relationship between SerumCholestoral and MaxHeartRate",
  ylab = "MaxHeartRate",
  xlab = "SerumCholestoral"
)
```



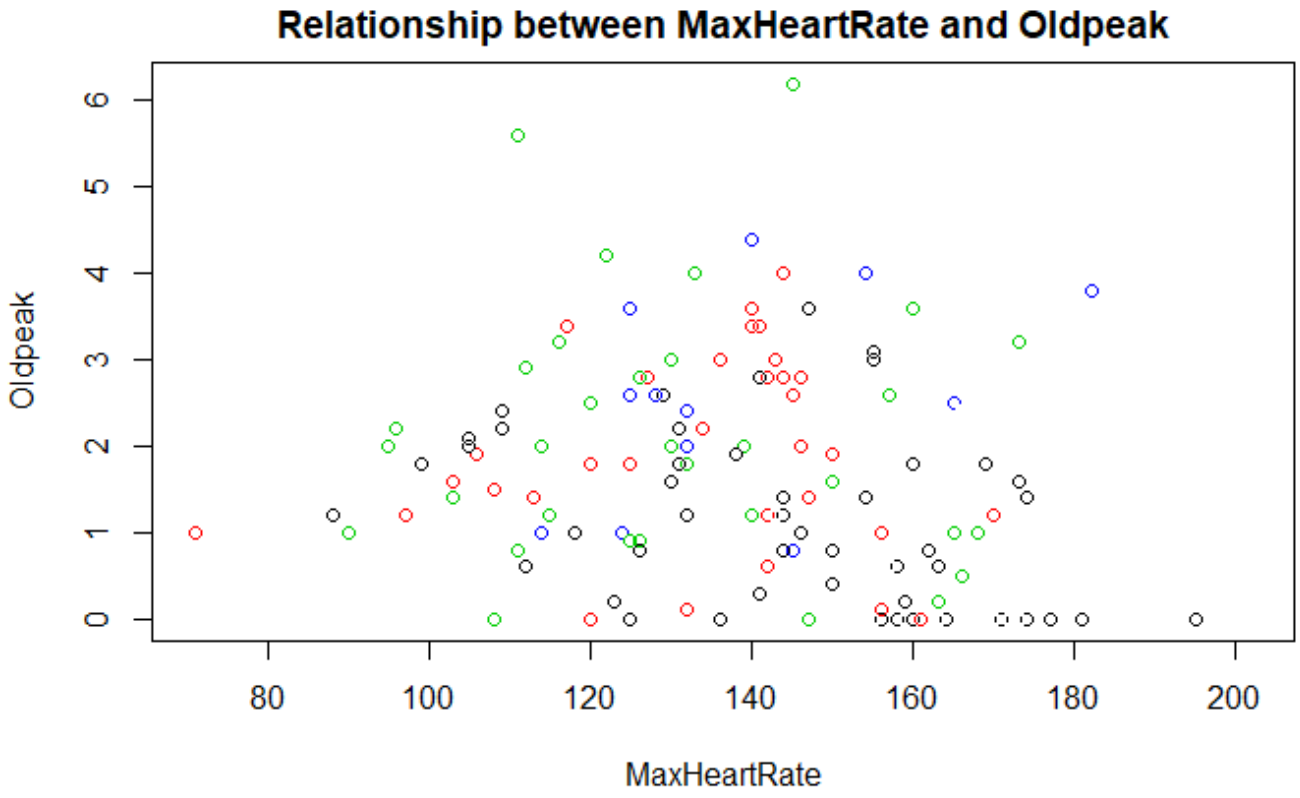
Hide

```
# SerumCholestoral x Oldpeak
plot(
  heartDiseaseDataframe$SerumCholestoral,
  heartDiseaseDataframe$Oldpeak,
  col = heartDiseaseDataframe$Class,
  main = "Relationship between SerumCholestoral and Oldpeak",
  ylab = "Oldpeak",
  xlab = "SerumCholestoral"
)
```



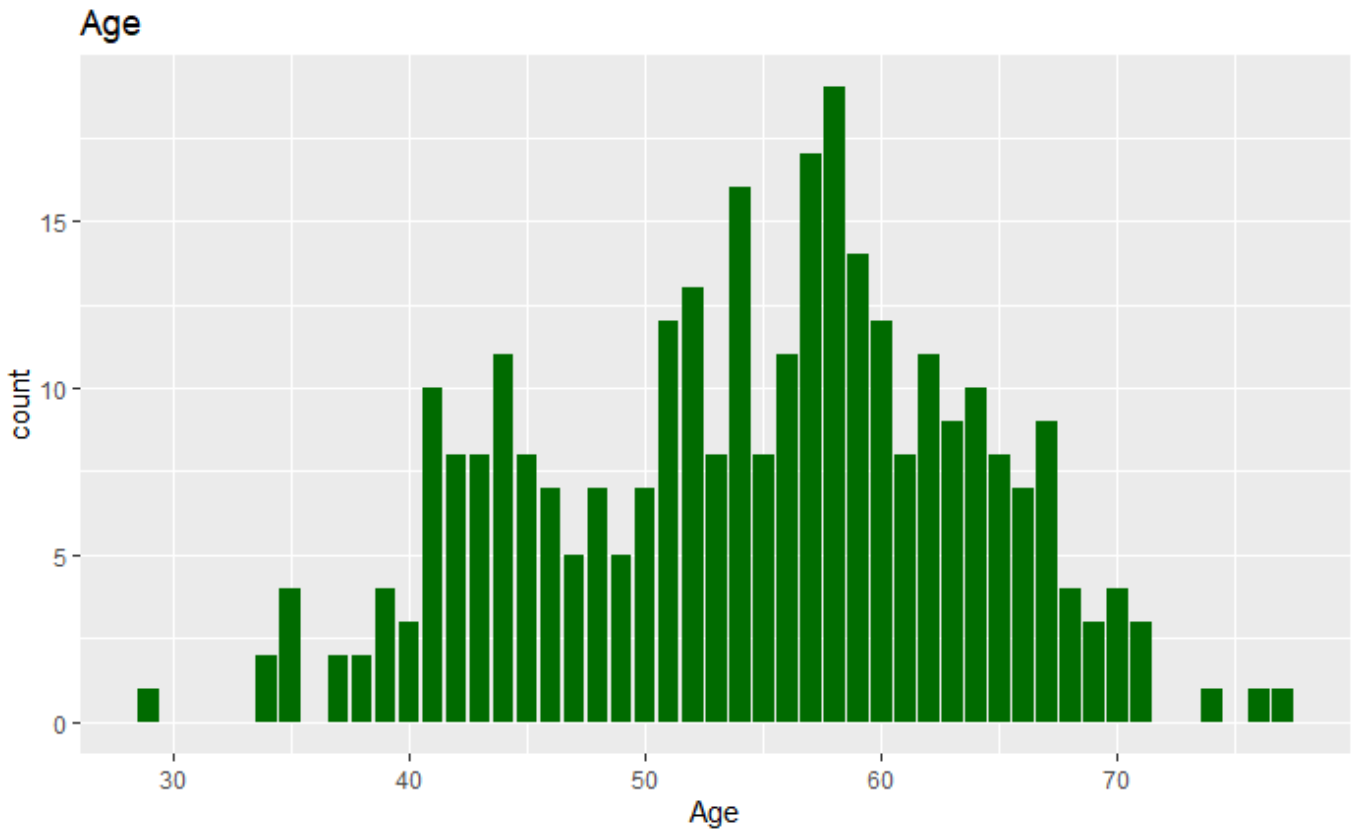
Hide

```
  # MaxHeartRate x Oldpeak
  plot(
    heartDiseaseDataframe$MaxHeartRate,
    heartDiseaseDataframe$Oldpeak,
    col = heartDiseaseDataframe$Class,
    main = "Relationship between MaxHeartRate and Oldpeak",
    ylab = "Oldpeak",
    xlab = "MaxHeartRate"
)
```



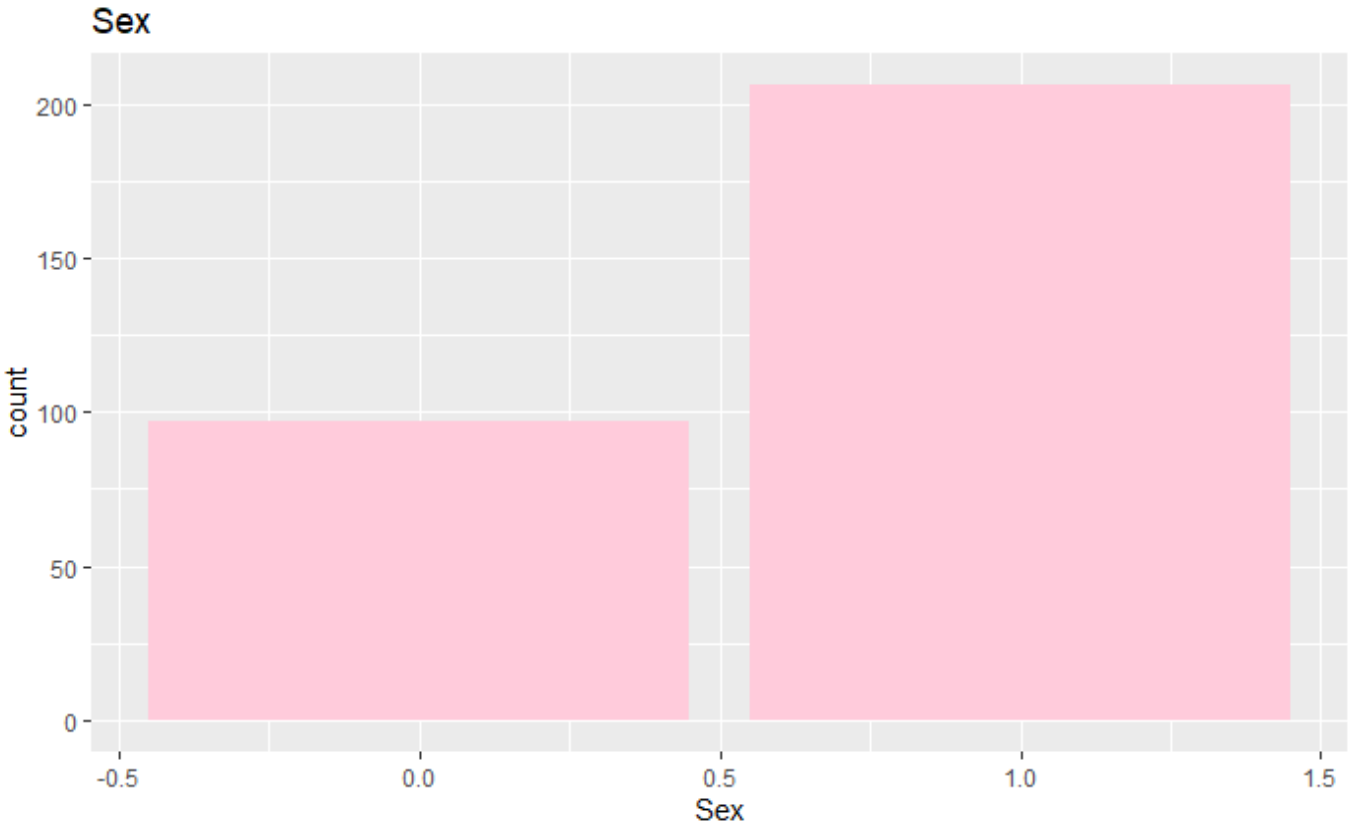
Hide

```
# Dataset plots

  # Age
  ggplot(heartDiseaseDataframe) +
    geom_bar(aes(Age), fill = "#006b00") +
    ggtitle("Age")
```



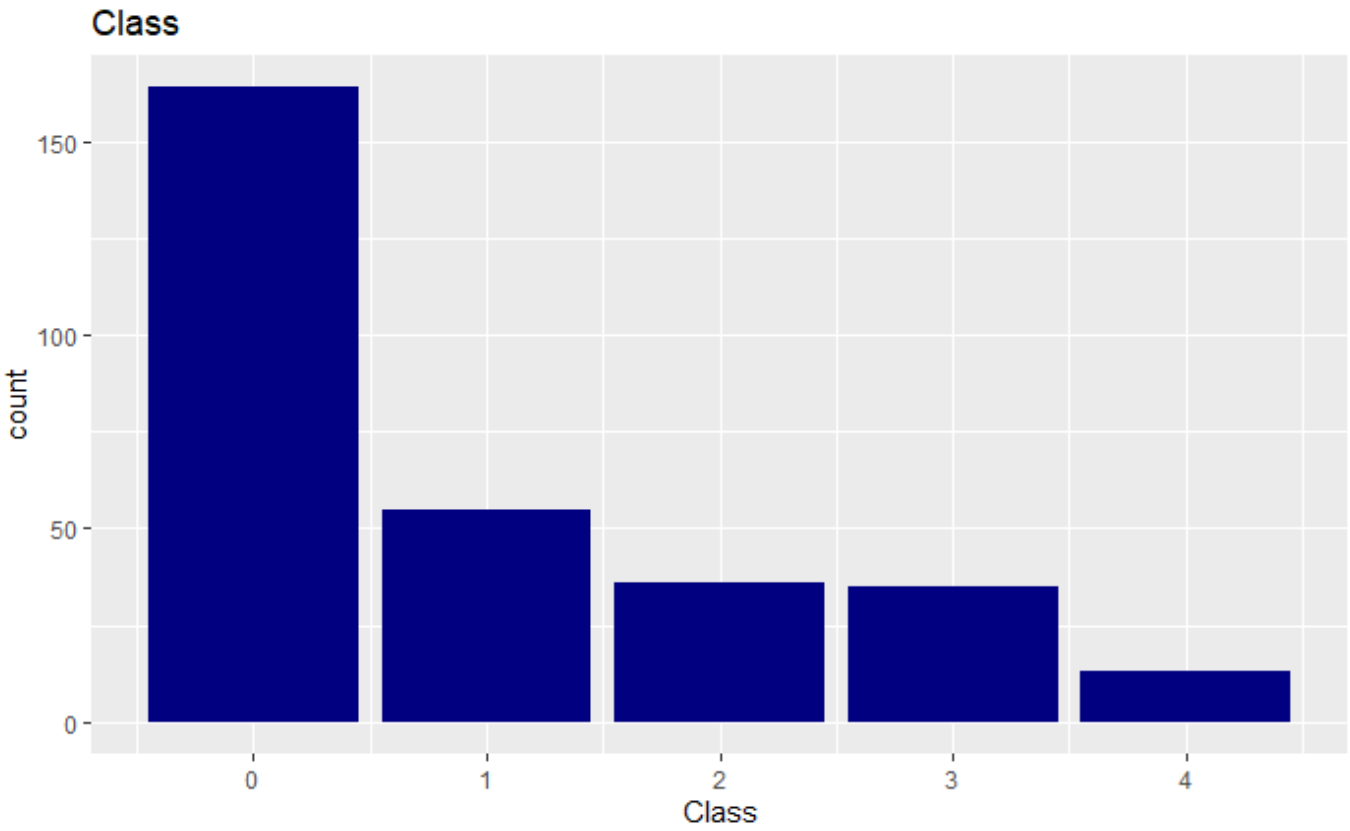
Hide

```
# Sex
ggplot(heartDiseaseDataframe) +
  geom_bar(aes(Sex), fill = "#ffcbdb") +
  ggtitle("Sex")
```



Hide

```
# Class
ggplot(heartDiseaseDataframe) +
  geom_bar(aes(Class), fill = "#000080") +
  ggtitle("Class")
```

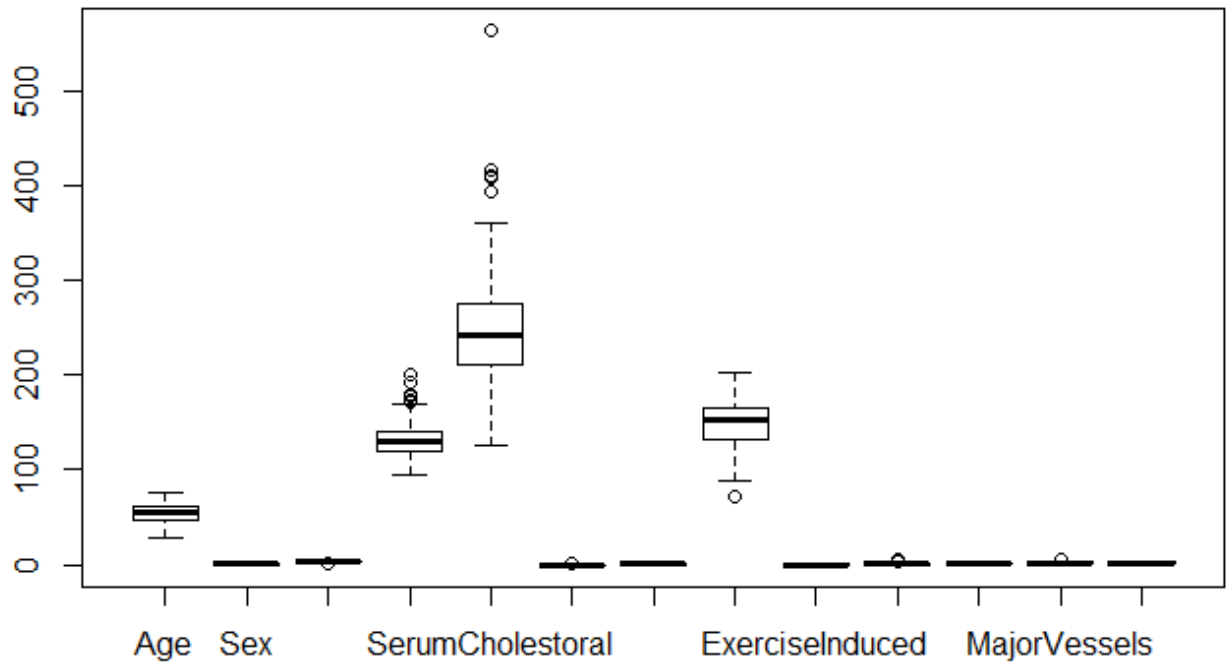


Hide

```
# === Handling data ====

  # Change "?" to NA
  heartDiseaseDataframe[heartDiseaseDataframe == "?"] <- NA

  # Removing NA values
  heartDiseaseDataframe <- na.omit(heartDiseaseDataframe)

  # Handling table columns

    # Class
    heartDiseaseDataframe$Class <- ifelse(test = heartDiseaseDataframe$Class >= 1, yes = "Unh
ealthy", no = "Healthy")
    heartDiseaseDataframe$Class <- as.factor(heartDiseaseDataframe$Class)

    # Other columns
    heartDiseaseDataframe[, c(1:9, 11:13)] <- sapply(heartDiseaseDataframe[, c(1:9, 11:13)],
 as.integer)
```

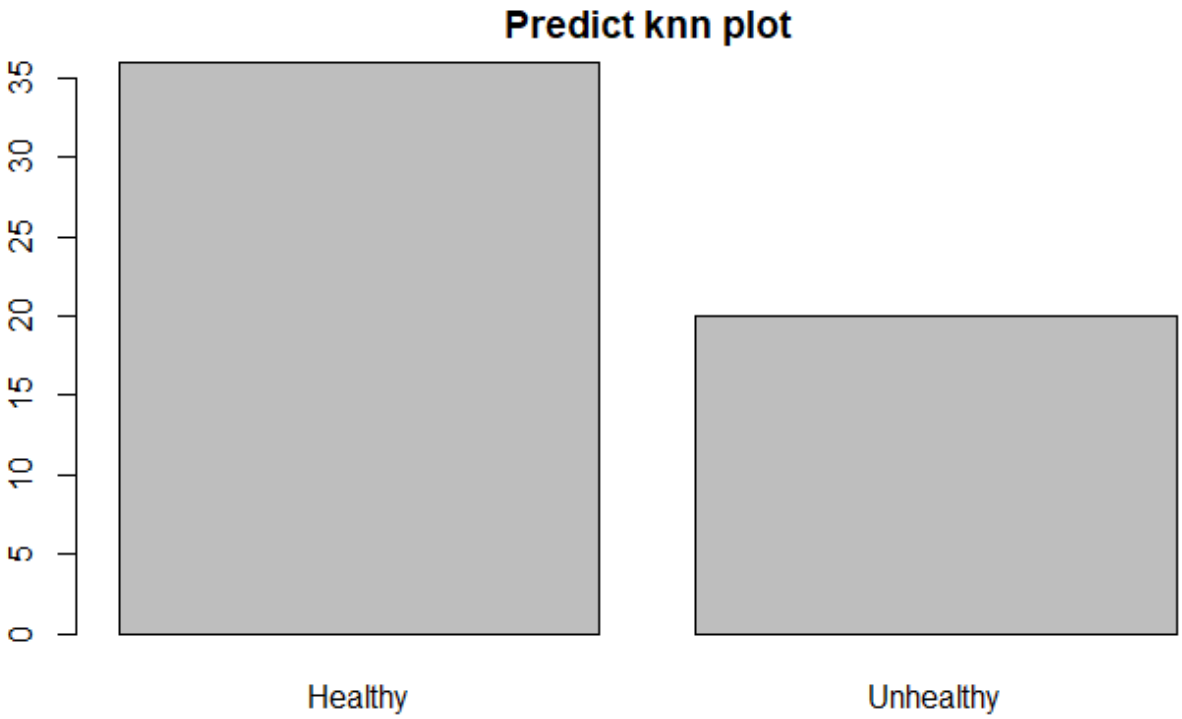
Hide

```
# ==== Boxplot with all data ====
boxplot(heartDiseaseDataframe[, -ncol(heartDiseaseDataframe)])
```



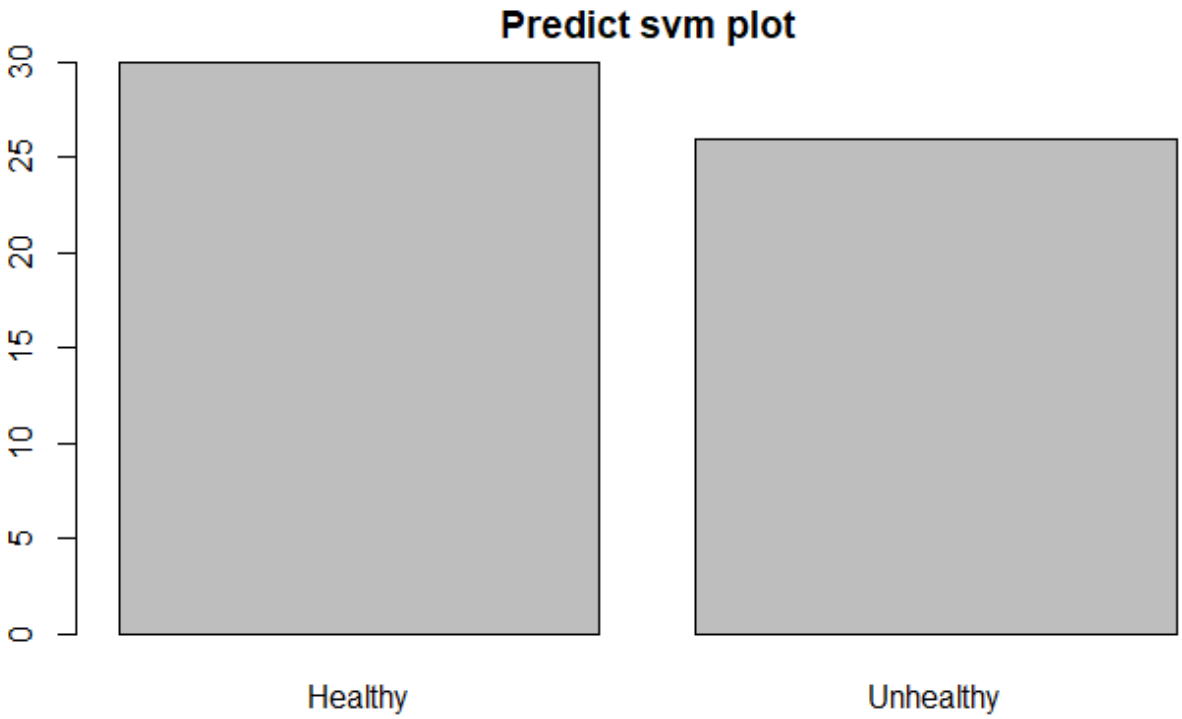
Hide

```
# ==== Data classification with all datas ====

  # Create datframe result of classifications #
  first_data <- data.frame()

  # Separate test and train model
  first_model <- buildTestAndTrain(heartDiseaseDataframe, 123, 0.8)

  # KNN classification
  Knn <- knnClassification(
    first_model$train_without_column,
    first_model$test,
    first_model$testClass,
    first_model$trainClass,
    9
  )
```



Hide

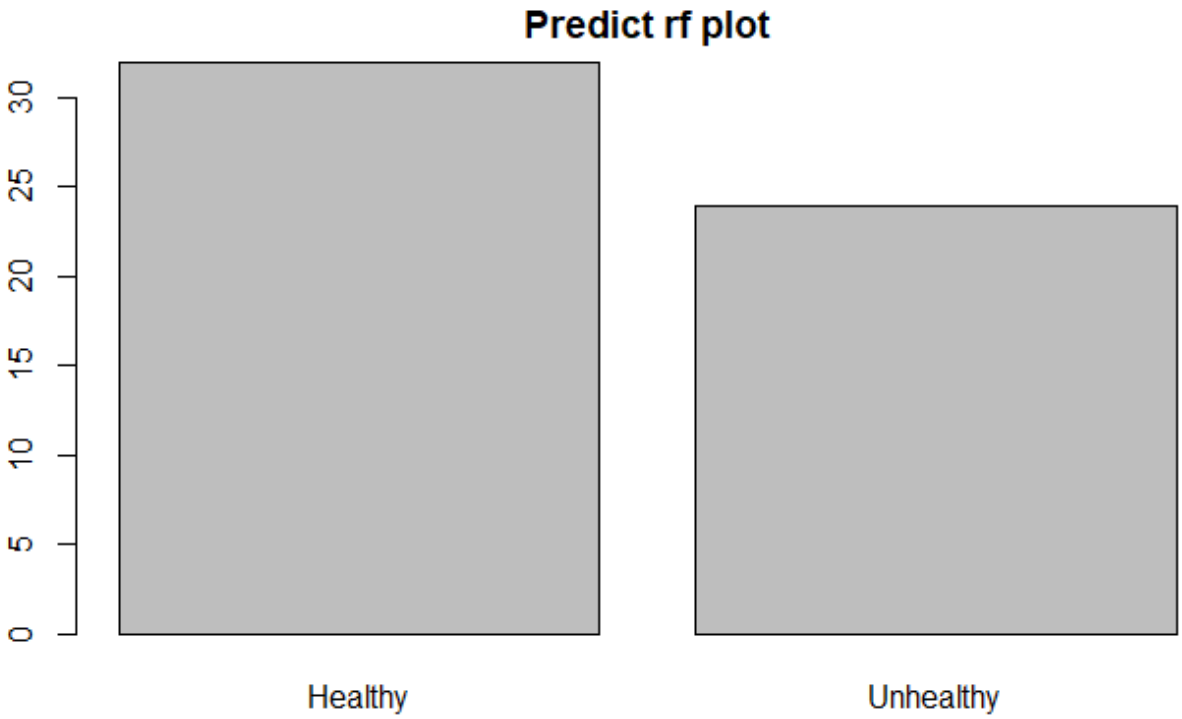
```
first_data <- rbind(first_data, Knn)

# SVM classification
Svm <- svmClassification(first_model$train,
                         first_model$test,
                         first_model$testClass)
```



Hide

```
first_data <- rbind(first_data, Svm)

# RF classification
Rf <- rfClassification(first_model$train,
                       first_model$test,
                       first_model$testClass)
```


Expected plot
0
5
10
15
20
25
30
Healthy
Unhealthy


Predict rf plot
0
5
10
15
20
25
30
Healthy
Unhealthy

Hide

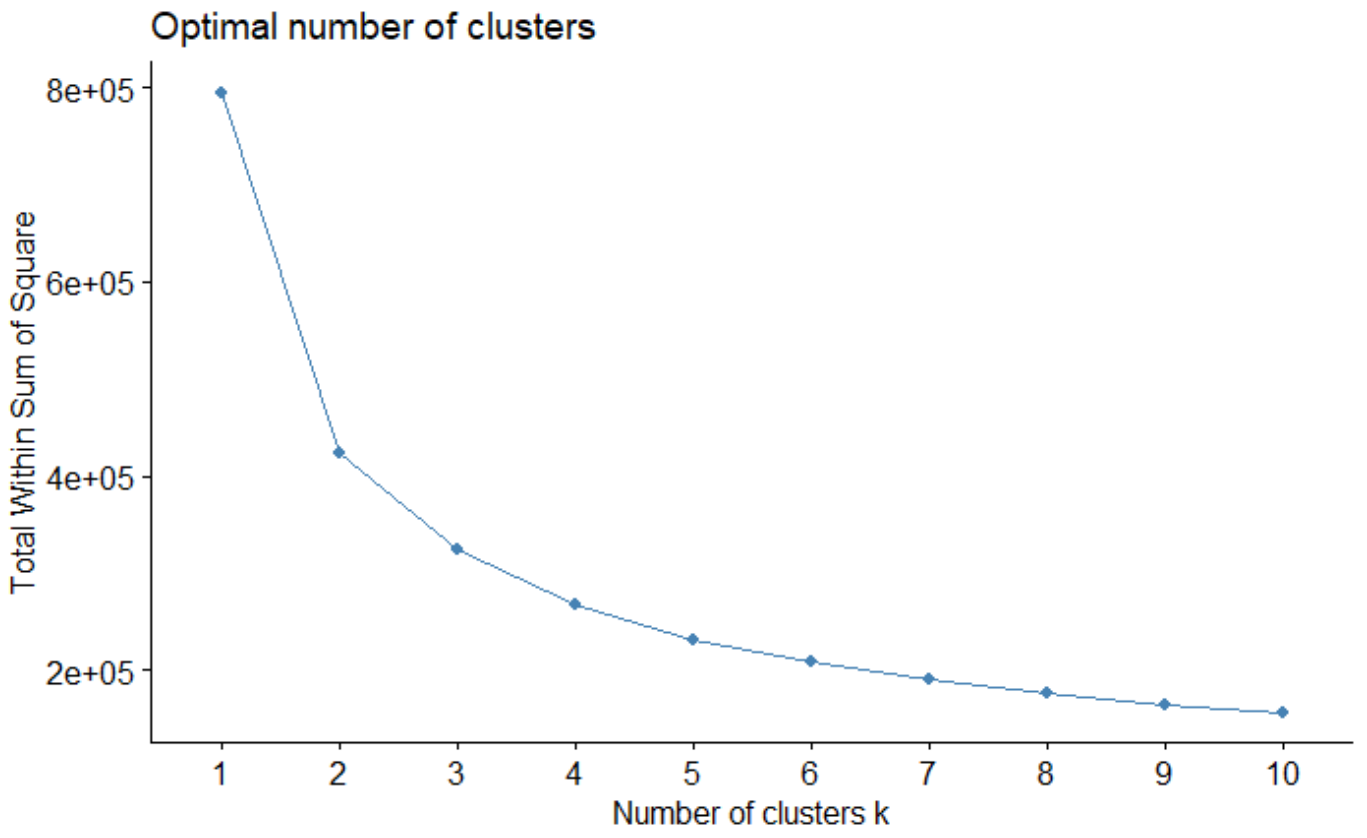
```
  first_data <- rbind(first_data, Rf)

  # renaming first data columns
  colnames(first_data) <- c( "Accuracy", "Kappa", "AccuracyLower", "AccuracyUpper", "Accuracy
Null", "AccuracyPValue",
                             "McnemarPValue" )

  first_data
```

Accuracy <dbl>	Kappa <dbl>	AccuracyLower <dbl>	AccuracyUpper <dbl>	AccuracyNull <dbl>	AccuracyPValue <dbl>	Mcner
0.6071429	0.2450980	0.4675369	0.7350087	0.5714286	3.449153e-01	0
0.8214286	0.6464646	0.6960284	0.9108990	0.5714286	6.758181e-05	0
0.8214286	0.6500000	0.6960284	0.9108990	0.5714286	6.758181e-05	0

3 rows

Hide

```
  # Cluster
  cluster_data <- heartDiseaseDataframe[, -ncol(heartDiseaseDataframe)]

  # Elbow method to see number of necessary clusters
  fviz_nbclust(cluster_data, kmeans, method = "wss")
```



Hide

```
# Calculate cluster by grouping
cluster_res_2 <- kmeans(cluster_data, 2)
cluster_res_19 <- kmeans(cluster_data, 19)

# Cluster Analysis
result_2 <- as.data.frame(table(heartDiseaseDataframe$Class, cluster_res_2$cluster))
result_19 <- as.data.frame(table(heartDiseaseDataframe$Class, cluster_res_19$cluster))

result_2
```

Var1 <fctr>	Var2 <fctr>	Freq <int>
Healthy	1	55
Unhealthy	1	65
Healthy	2	99
Unhealthy	2	59

4 rows

Hide

```
result_19
```

Var1 <fctr>	Var2 <fctr>	Freq <int>
Healthy	1	18
Unhealthy	1	1
Healthy	2	7
Unhealthy	2	7
Healthy	3	3
Unhealthy	3	3
Healthy	4	5
Unhealthy	4	9
Healthy	5	14
Unhealthy	5	0

1-10 of 38 rows

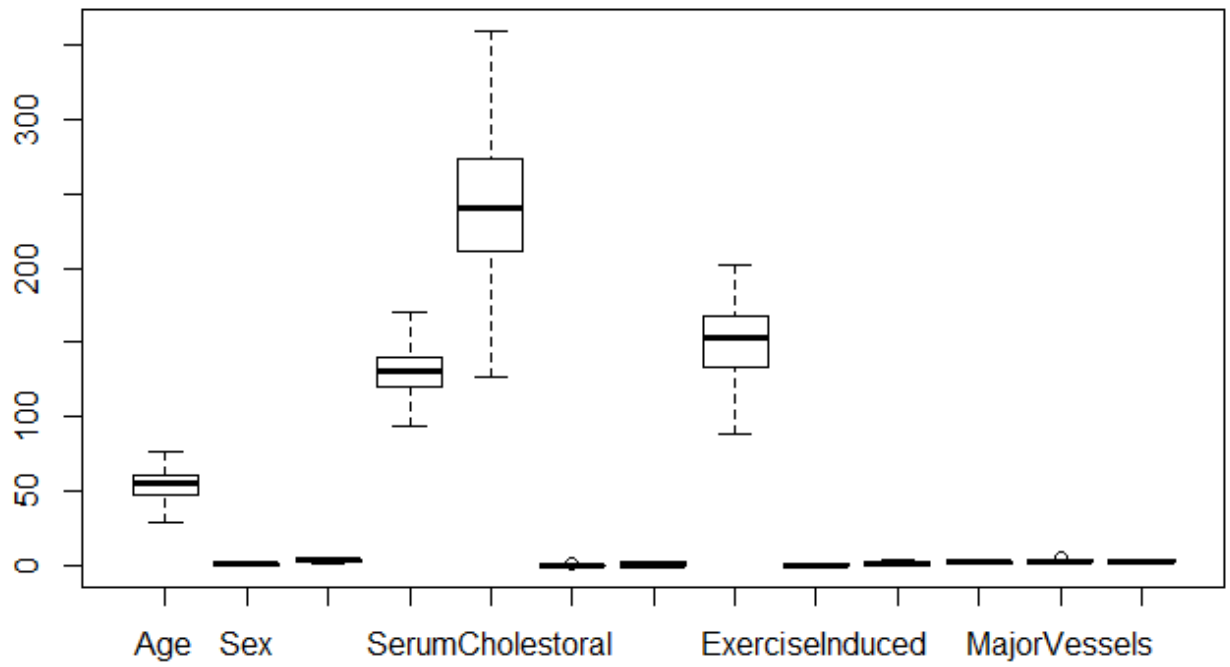
Previous **1** 2 3 4 Next

Hide

```
# ==== Handling data 2 ====

  # Removing outliers
  outlier_values <- boxplot.stats(heartDiseaseDataframe[, 1])$out
  heartDiseaseDataframe <- heartDiseaseDataframe[!(heartDiseaseDataframe[, 1] %in% outlier_va
lues),]

  outlier_values <- boxplot.stats(heartDiseaseDataframe[, 4])$out
  heartDiseaseDataframe <- heartDiseaseDataframe[!(heartDiseaseDataframe[, 4] %in% outlier_va
lues),]

  outlier_values <- boxplot.stats(heartDiseaseDataframe[, 5])$out
  heartDiseaseDataframe <- heartDiseaseDataframe[!(heartDiseaseDataframe[, 5] %in% outlier_va
lues),]

  outlier_values <- boxplot.stats(heartDiseaseDataframe[, 8])$out
  heartDiseaseDataframe <- heartDiseaseDataframe[!(heartDiseaseDataframe[, 8] %in% outlier_va
lues),]

  outlier_values <- boxplot.stats(heartDiseaseDataframe[, 10])$out
  heartDiseaseDataframe <- heartDiseaseDataframe[!(heartDiseaseDataframe[, 10] %in% outlier_v
alues),]
```

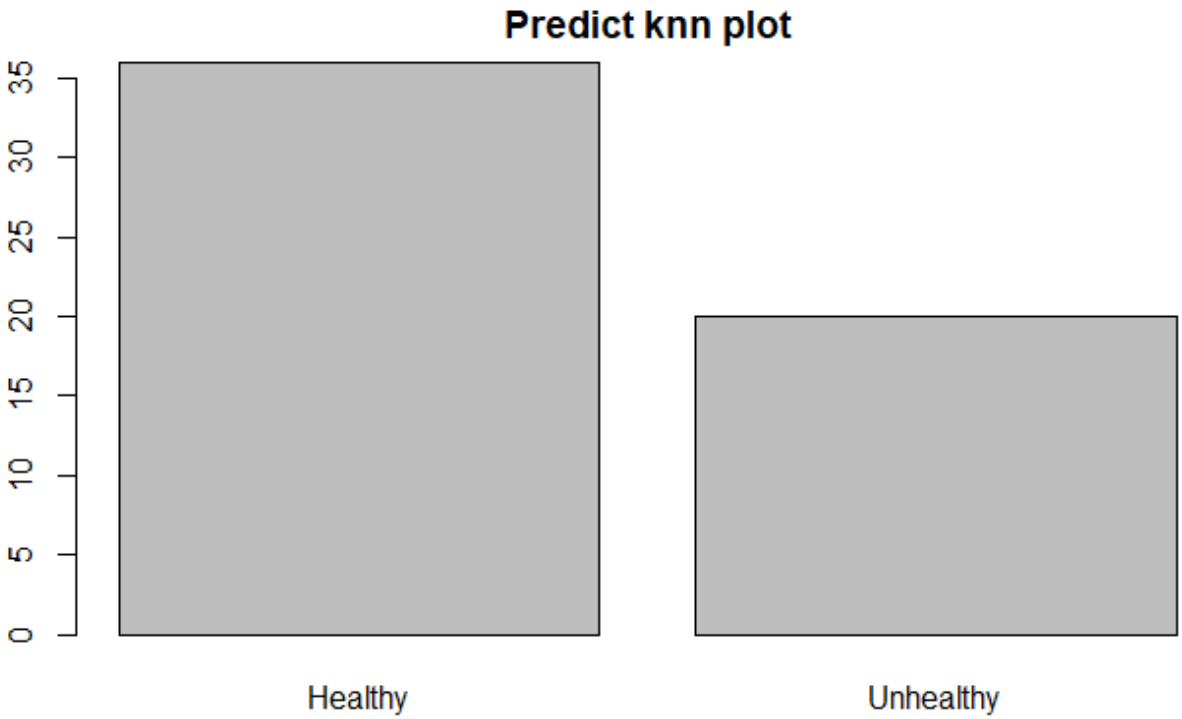
Hide

```
# ==== Boxplot without outliers ====
boxplot(heartDiseaseDataframe[, -ncol(heartDiseaseDataframe)])
```



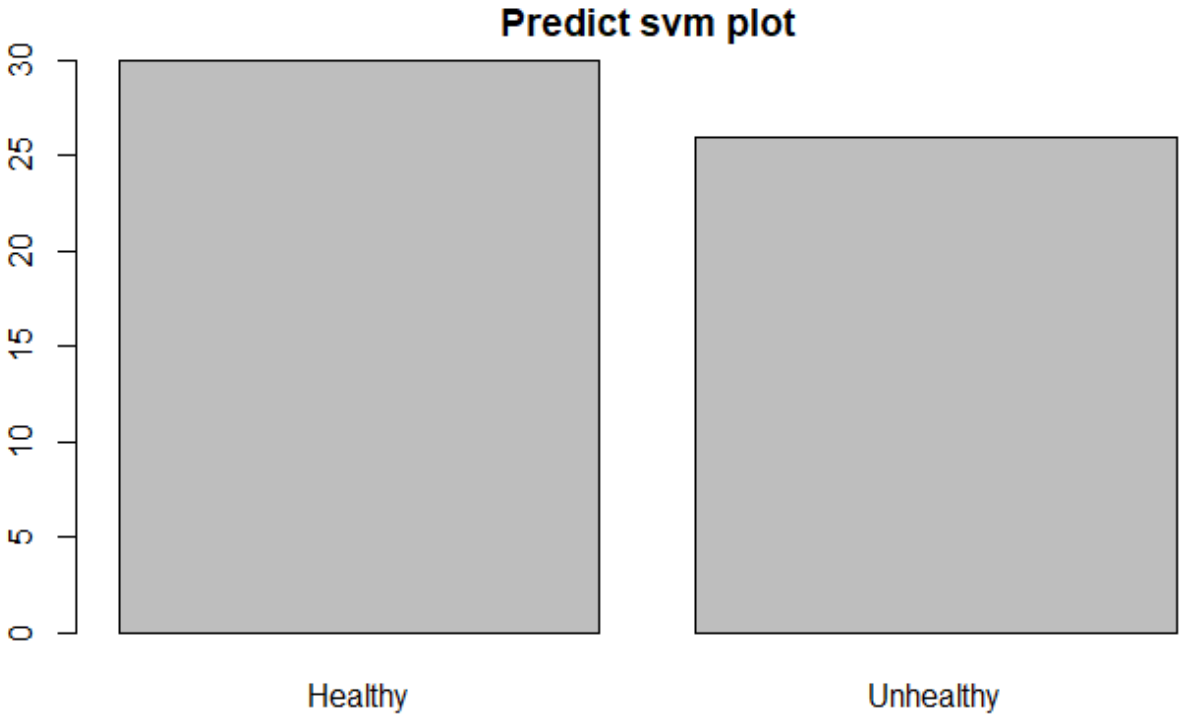
Hide

```
# ==== Data classification without outliers ====

  # Create datframe result of classifications without dataframe outliers
  second_data <- data.frame()

  # Separate test and train model
  second_model <- buildTestAndTrain(heartDiseaseDataframe, 123, 0.8)

  # KNN classification
  Knn <- knnClassification(
    second_model$train_without_column,
    second_model$test,
    second_model$testClass,
    second_model$trainClass,
    9
  )
```



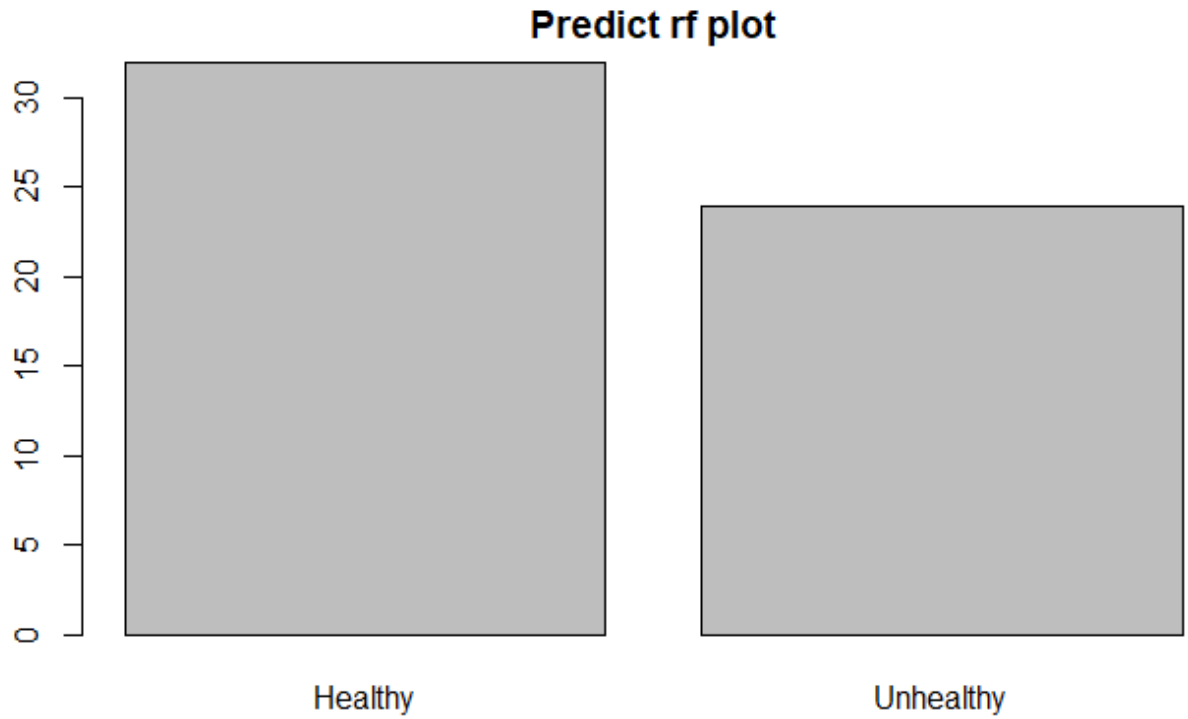
Hide

```
second_data <- rbind(second_data, Knn)

# SVM classification
Svm <- svmClassification(second_model$train,
                  second_model$test,
                  second_model$testClass)
```



Hide

```
second_data <- rbind(second_data, Svm)

# RF classification
Rf <- rfClassification(second_model$train,
                  second_model$test,
                  second_model$testClass)
```



Hide

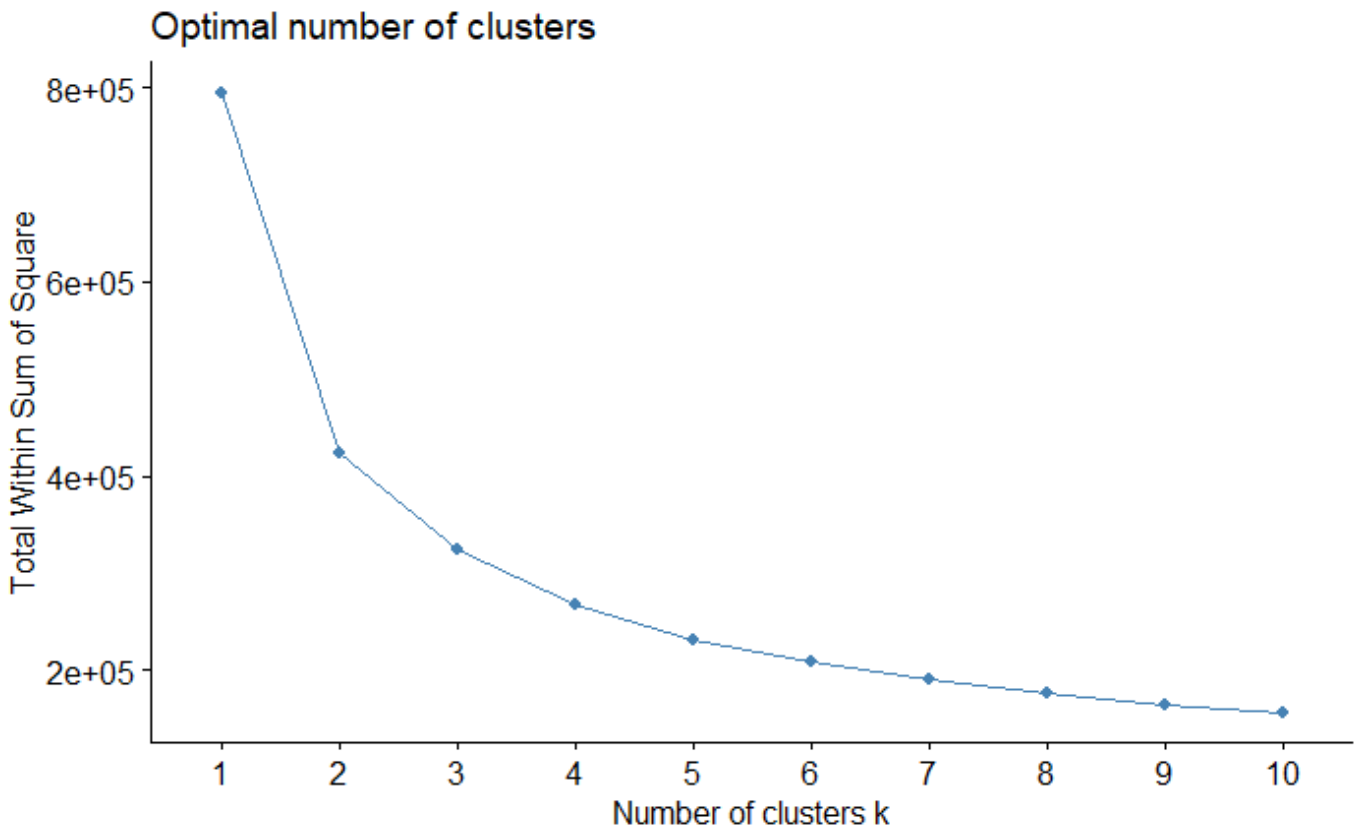
```
  second_data <- rbind(second_data, Rf)

  # Renaming dataset columns
  colnames(second_data) <- c( "Accuracy", "Kappa", "AccuracyLower", "AccuracyUpper", "Accurac
yNull", "AccuracyPValue",
                              "McnemarPValue" )

  second_data
```

Accuracy <dbl>	Kappa <dbl>	AccuracyLower <dbl>	AccuracyUpper <dbl>	AccuracyNull <dbl>	AccuracyPValue <dbl>	Mcner
0.6071429	0.2450980	0.4675369	0.7350087	0.5714286	3.449153e-01	0
0.8214286	0.6464646	0.6960284	0.9108990	0.5714286	6.758181e-05	0
0.8214286	0.6500000	0.6960284	0.9108990	0.5714286	6.758181e-05	0

3 rows

Hide

```
  # Cluster
  cluster_data <- heartDiseaseDataframe[, -ncol(heartDiseaseDataframe)]

  # Elbow method to see number of necessary clusters
  fviz_nbclust(cluster_data, kmeans, method = "wss")
```



Hide

```
  # Calculate cluster by grouping
  cluster_res_2 <- kmeans(cluster_data, 2)
  cluster_res_11 <- kmeans(cluster_data, 19)

  # Cluster Analysis
  result_2_outlier <- as.data.frame(table(heartDiseaseDataframe$Class, cluster_res_2$cluste
r))
  result_11_outlier <- as.data.frame(table(heartDiseaseDataframe$Class, cluster_res_11$cluste
r))

  result_2_outlier
```

Var1 <fctr>	Var2 <fctr>	Freq <int>
Healthy	1	55
Unhealthy	1	65
Healthy	2	99
Unhealthy	2	59

4 rows

Hide

```
  result_11_outlier
```

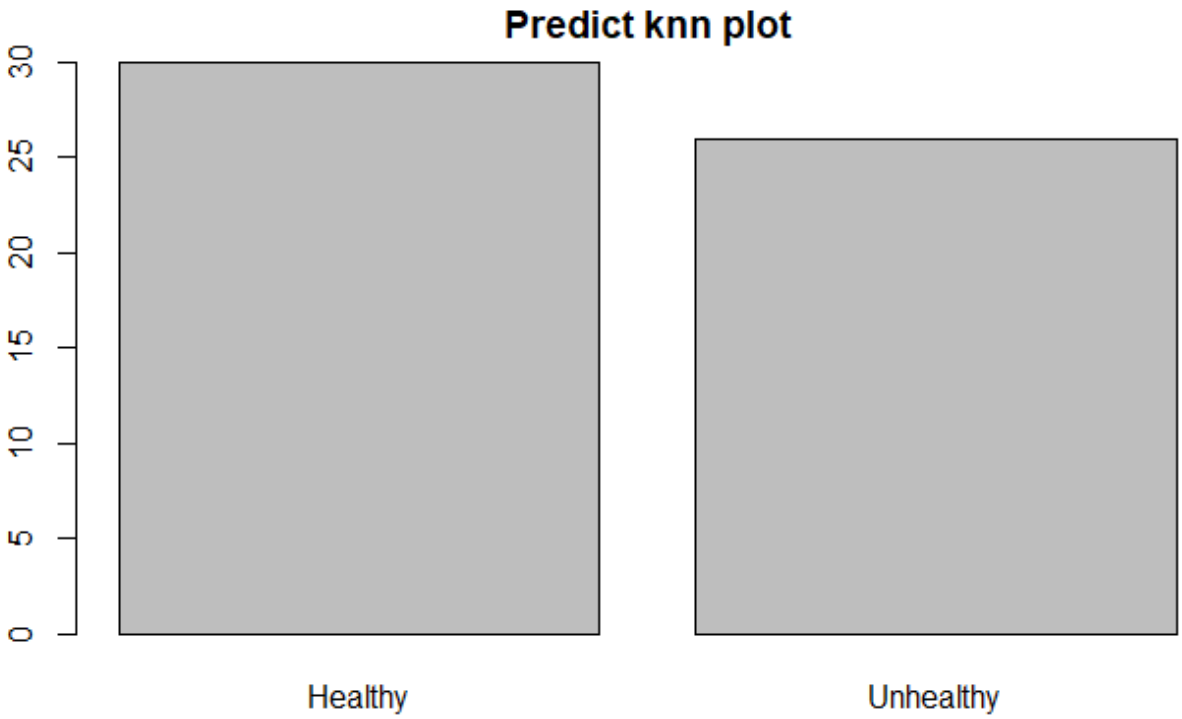
Var1 <fctr>	Var2 <fctr>	Freq <int>
Healthy	1	18
Unhealthy	1	1
Healthy	2	7
Unhealthy	2	7
Healthy	3	3
Unhealthy	3	3
Healthy	4	5
Unhealthy	4	9
Healthy	5	14
Unhealthy	5	0

Hide

```
# ==== Handling data 3 ====

  # PCA
  dataframe.pca <- prcomp(heartDiseaseDataframe[, -ncol(heartDiseaseDataframe)], center = TRU
E, scale. = TRUE)
  newDataframe <- as.data.frame(predict(dataframe.pca, heartDiseaseDataframe))

  # Adding class column to the new dataframe
  newDataframe$Class <- heartDiseaseDataframe$Class
```

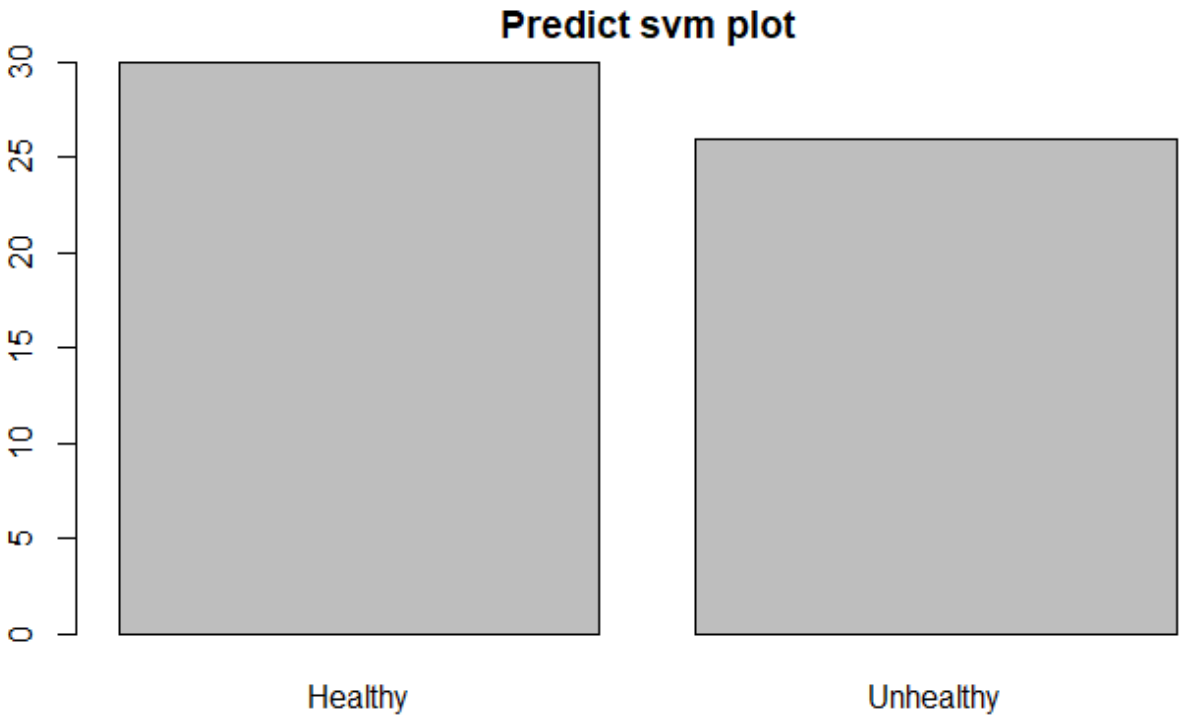
Hide

```
# ==== Data classification with PCA ====

  # Create datframe result of classifications with cleaning data pca
  third_data <- data.frame()

  # Separate test and train model
  third_model <- buildTestAndTrain(newDataframe, 123, 0.8)

  # KNN classification
  Knn <- knnClassification(
    third_model$train_without_column,
    third_model$test,
    third_model$testClass,
    third_model$trainClass,
    9
  )
```



Hide

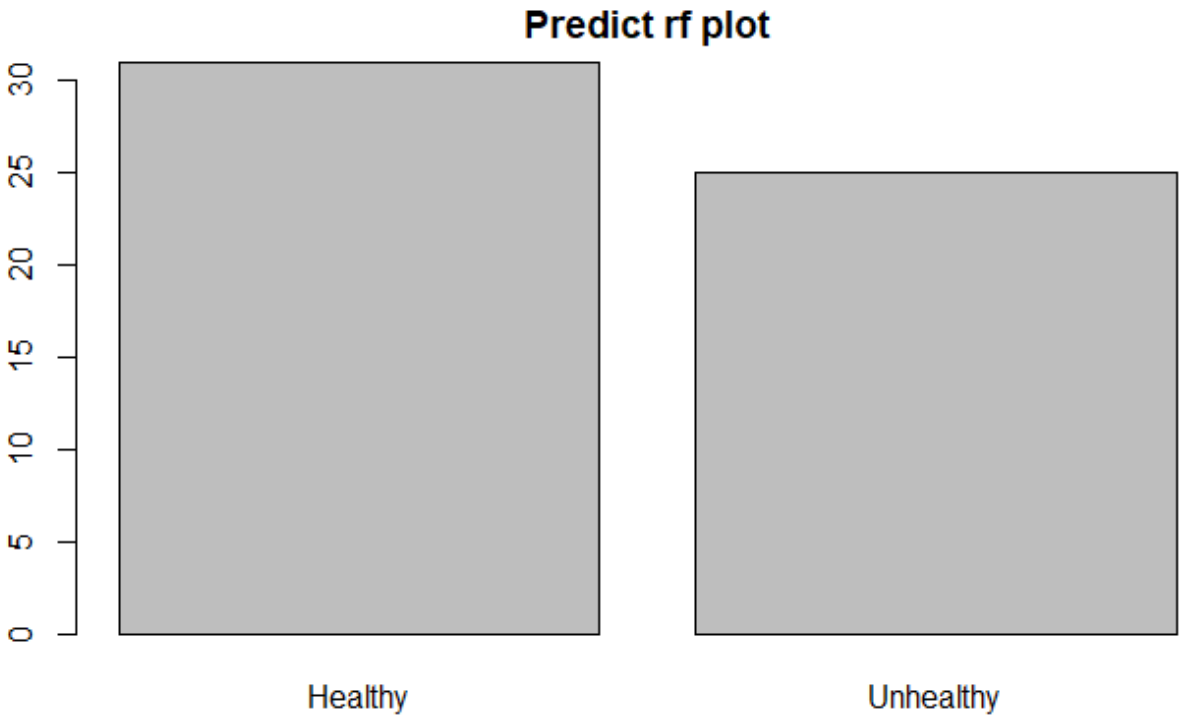
```
third_data <- rbind(third_data, Knn)

# SVM classification
Svm <- svmClassification(third_model$train, third_model$test, third_model$testClass)
```



Hide

```
third_data <- rbind(third_data, Svm)

# RF classification
Rf <- rfClassification(third_model$train, third_model$test, third_model$testClass)
```


Expected plot
0
5
10
15
20
25
30
Healthy
Unhealthy


Predict rf plot
0
5
10
15
20
25
30
Healthy
Unhealthy

Hide

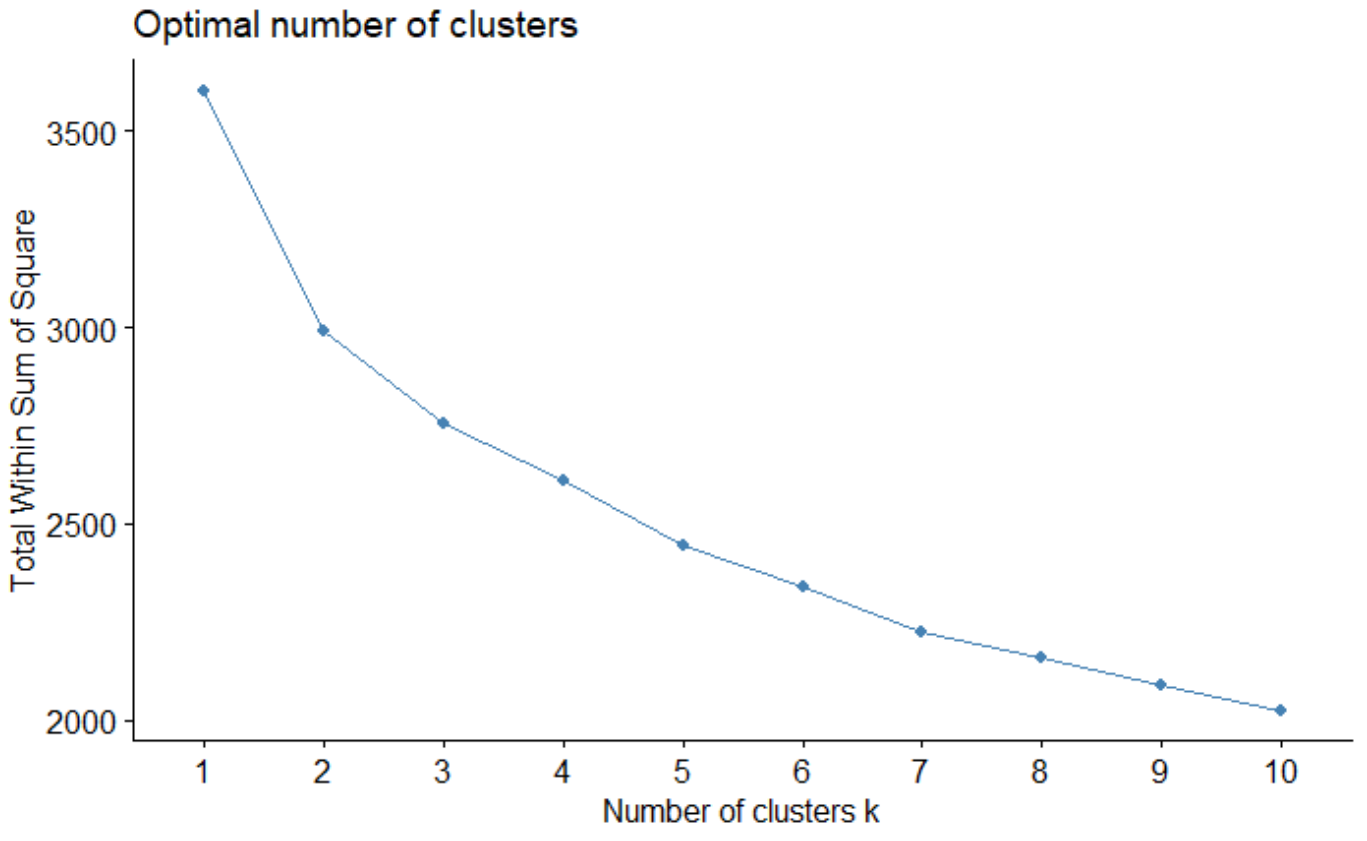
```
  third_data <- rbind(third_data, Rf)

  # renaming dataset columns
  colnames(third_data) <- c( "Accuracy", "Kappa", "AccuracyLower", "AccuracyUpper", "Accuracy
Null", "AccuracyPValue",
                             "McnemarPValue" )

  third_data
```

Accuracy <dbl>	Kappa <dbl>	AccuracyLower <dbl>	AccuracyUpper <dbl>	AccuracyNull <dbl>	AccuracyPValue <dbl>	Mcner
0.8214286	0.6464646	0.6960284	0.9108990	0.5714286	6.758181e-05	0
0.8214286	0.6464646	0.6960284	0.9108990	0.5714286	6.758181e-05	0
0.8035714	0.6130653	0.6756670	0.8976517	0.5714286	2.219097e-04	0

3 rows

Hide

```
  # Cluster
  cluster_data <- newDataframe[, -ncol(newDataframe)]

  # Elbow method to see number of necessary clusters
  fviz_nbclust(cluster_data, kmeans, method = "wss")
```



Hide

```
# Calculate cluster by grouping
cluster_res_2 <- kmeans(cluster_data, 2)
cluster_res_11 <- kmeans(cluster_data, 11)

# Cluster Analysis
result_2_pca <- as.data.frame(table(newDataframe$Class, cluster_res_2$cluster))
result_11_pca <- as.data.frame(table(newDataframe$Class, cluster_res_11$cluster))

result_2_pca
```

Var1 <fctr>	Var2 <fctr>	Freq <int>
Healthy	1	21
Unhealthy	1	93
Healthy	2	133
Unhealthy	2	31

4 rows

Hide

```
result_11_pca
```

Var1 <fctr>	Var2 <fctr>	Freq <int>
Healthy	1	13
Unhealthy	1	3
Healthy	2	7
Unhealthy	2	10
Healthy	3	13
Unhealthy	3	4
Healthy	4	4
Unhealthy	4	40
Healthy	5	28
Unhealthy	5	3

1-10 of 22 rows