DASC5420 PROJECT

SONA SHAUKATH

2023-03-24

```
#loading packages
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(class)
#reading the csv file
heart <- read.csv("/Users/sonashaukath/Downloads/heart_2020_cleaned.csv", header = TRUE, ",")
x<-sample(1:nrow(heart), 10000)
heart<- heart[x, ]
```

```
#Expolatory data analysis
summary(heart)
```

```
##  HeartDisease            BMI          Smoking          AlcoholDrinking
##  Length:10000       Min.   :12.13   Length:10000       Length:10000
##  Class :character   1st Qu.:24.03   Class :character   Class :character
##  Mode  :character   Median :27.34   Mode  :character   Mode  :character
##                     Mean   :28.35
##                     3rd Qu.:31.57
##                     Max.   :79.83
##     Stroke          PhysicalHealth   MentalHealth    DiffWalking
##  Length:10000       Min.   : 0.00   Min.   : 0.000   Length:10000
##  Class :character   1st Qu.: 0.00   1st Qu.: 0.000   Class :character
##  Mode  :character   Median : 0.00   Median : 0.000   Mode  :character
##                     Mean   : 3.34   Mean   : 3.904
##                     3rd Qu.: 2.00   3rd Qu.: 3.000
##                     Max.   :30.00   Max.   :30.000
##      Sex            AgeCategory            Race             Diabetic
##  Length:10000       Length:10000       Length:10000       Length:10000
```

```
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  PhysicalActivity    GenHealth           SleepTime         Asthma
##  Length:10000       Length:10000       Min.   : 1.000   Length:10000
##  Class :character   Class :character   1st Qu.: 6.000   Class :character
##  Mode  :character   Mode  :character   Median : 7.000   Mode  :character
##                                        Mean   : 7.081
##                                        3rd Qu.: 8.000
##                                        Max.   :22.000
##  KidneyDisease       SkinCancer
##  Length:10000       Length:10000
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

```
head(heart)
```

```
##        HeartDisease   BMI Smoking AlcoholDrinking Stroke PhysicalHealth
## 220279           No 28.59      No              No     No             10
## 73904            No 32.87     Yes              No     No              0
## 179702           No 31.19     Yes              No     No              0
## 28238            No 25.83      No              No     No              0
## 126802           No 22.60      No             Yes     No              0
## 319500           No 23.96     Yes              No     No              0
##        MentalHealth DiffWalking    Sex AgeCategory     Race Diabetic
## 220279            0          No   Male       60-64    White       No
## 73904             0          No   Male       50-54    White      Yes
## 179702            0         Yes   Male       30-34    White       No
## 28238             0          No   Male       50-54    White       No
## 126802            0          No Female       18-24    White       No
## 319500            0          No   Male       40-44 Hispanic      Yes
##        PhysicalActivity GenHealth SleepTime Asthma KidneyDisease SkinCancer
## 220279              Yes Very good         7     No            No         No
## 73904                No      Good         7     No            No         No
## 179702              Yes      Fair         4     No            No         No
## 28238               Yes      Good         7     No            No         No
## 126802              Yes      Good         7     No            No         No
## 319500              Yes Very good         6     No           Yes         No
```

```
#omitting the null values
heart <- na.omit(heart)
#checking the number of null values
sum(is.na(heart))
```

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

```r
#Converting the response variable to binary variable
heart$HeartDisease <- ifelse(heart$HeartDisease == "Yes", 1, 0)
#Converting Categorical variables to binary variables
heart$Smoking <- ifelse(heart$Smoking == "Yes", 1, 0)
heart$AlcoholDrinking  <- ifelse(heart$AlcoholDrinking == "Yes", 1, 0)
heart$Stroke  <- ifelse(heart$Stroke == "Yes", 1, 0)
heart$DiffWalking  <- ifelse(heart$DiffWalking == "Yes", 1, 0)
heart$Sex <- ifelse(heart$Sex == "Male", 1, 0)
heart$PhysicalActivity  <- ifelse(heart$PhysicalActivity == "Yes", 1, 0)
heart$Asthma  <- ifelse(heart$Asthma == "Yes", 1, 0)
heart$KidneyDisease <- ifelse(heart$KidneyDisease == "Yes", 1, 0)
heart$SkinCancer  <- ifelse(heart$SkinCancer == "Yes", 1, 0)
```

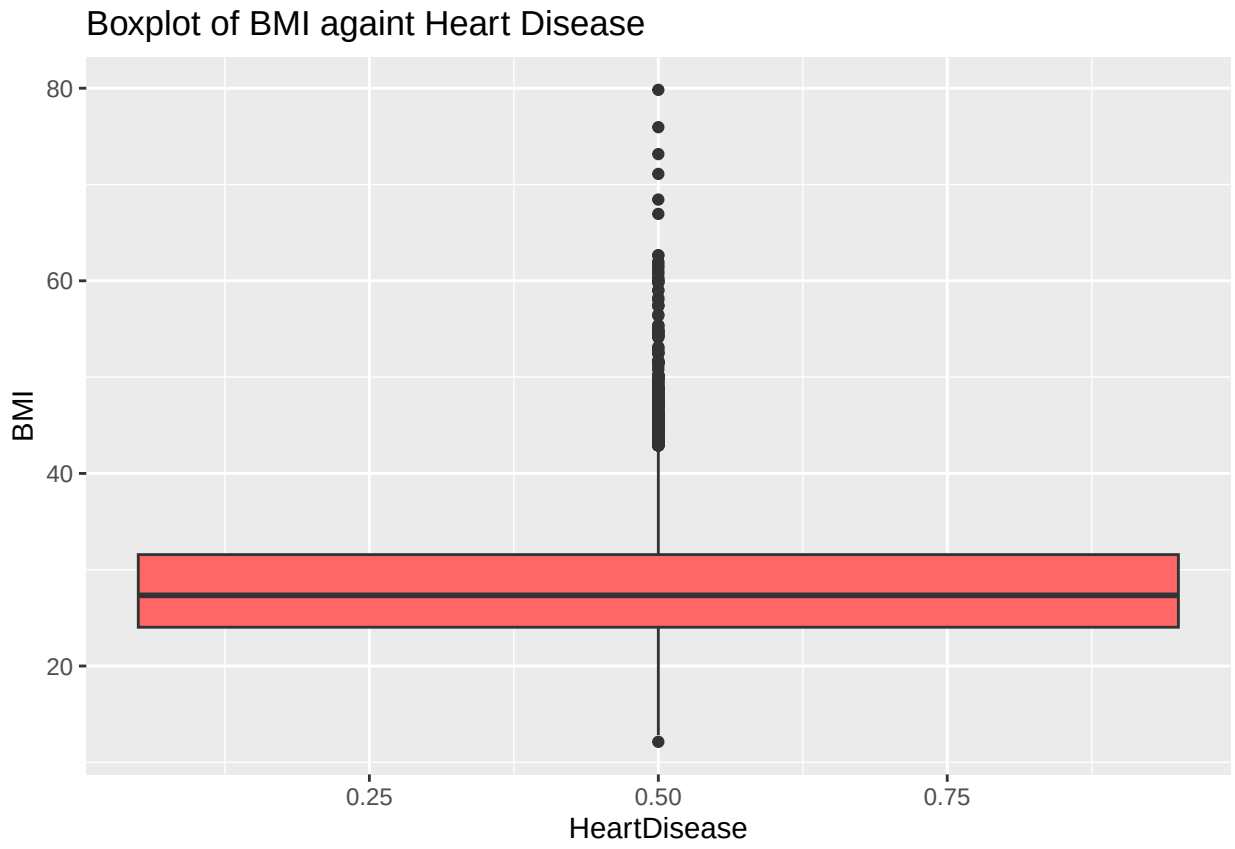
```r
#Converting GenHealth variable to numeric values
heart$GenHealth <- factor(heart$GenHealth, levels = c("Poor", "Fair", "Good", "Very good", "Excellent"))
heart$GenHealth  <- as.integer(heart$GenHealth) - 1
#Converting Sleeptime variable to numeric values
heart$SleepTime <- ifelse(heart$SleepTime >= median(heart$SleepTime), 1, 0)
#Converting Diabetes variable to numeric values
heart$Diabetic <- factor(heart$Diabetic, levels = c("No", "No, borderline diabetes", "Yes (during pregna
heart$Diabetic <- as.integer(heart$Diabetic) - 1
```

```r
#getting unique values of each column
for (col_n in names(heart)) {
  unq_val <- length(unique(heart[[col_n]]))
  print(paste("Feature '", col_n, "' has '", unq_val, "' unique values", sep = ""))
}
```

```
## [1] "Feature 'HeartDisease' has '2' unique values"
## [1] "Feature 'BMI' has '1458' unique values"
## [1] "Feature 'Smoking' has '2' unique values"
## [1] "Feature 'AlcoholDrinking' has '2' unique values"
## [1] "Feature 'Stroke' has '2' unique values"
## [1] "Feature 'PhysicalHealth' has '29' unique values"
## [1] "Feature 'MentalHealth' has '31' unique values"
## [1] "Feature 'DiffWalking' has '2' unique values"
## [1] "Feature 'Sex' has '2' unique values"
## [1] "Feature 'AgeCategory' has '13' unique values"
## [1] "Feature 'Race' has '6' unique values"
## [1] "Feature 'Diabetic' has '4' unique values"
## [1] "Feature 'PhysicalActivity' has '2' unique values"
## [1] "Feature 'GenHealth' has '5' unique values"
## [1] "Feature 'SleepTime' has '2' unique values"
## [1] "Feature 'Asthma' has '2' unique values"
## [1] "Feature 'KidneyDisease' has '2' unique values"
## [1] "Feature 'SkinCancer' has '2' unique values"
```

```r
# Creating a boxplot of BMI to check for outliers
ggplot(heart, aes(x = HeartDisease , y = BMI)) +
  geom_boxplot(fill = "#FF6666") +
  labs(y = "BMI", x = "HeartDisease" ) +
  ggtitle("Boxplot of BMI againt Heart Disease")
```

```
## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
```



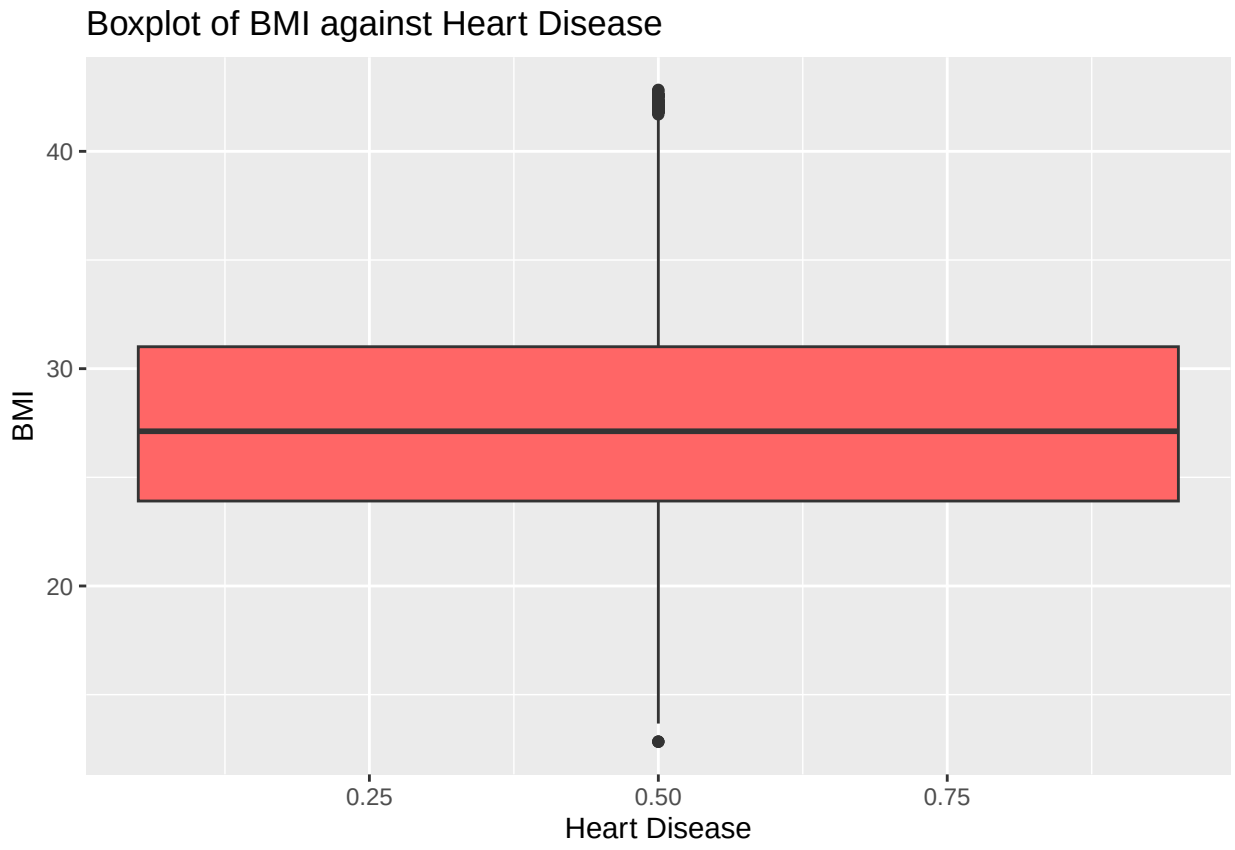

```r
# Identifying any outliers using the "identify_outliers" function from my previous response
identify_outliers <- function(x) {
  q1 <- quantile(x, 0.25)
  q3 <- quantile(x, 0.75)
  iqr <- q3 - q1
  upper_fence <- q3 + 1.5*iqr
  lower_fence <- q1 - 1.5*iqr
  outlier_indices <- which(x < lower_fence | x > upper_fence)
  return(outlier_indices)
}

# Applying the function to the BMI variable
outliers <- identify_outliers(heart$BMI)

# Print the indices of any outliers identified
if (length(outliers) > 0) {
  cat("Outliers identified in BMI. \n")

  # Remove the outliers from the dataset
  heart <- heart[-outliers,]
  cat("Outliers removed from the dataset.\n")
} else {
```

```
  cat("No outliers identified in BMI.\n")
}
```

```
## Outliers identified in BMI.
## Outliers removed from the dataset.
```

```
# Creating a boxplot of BMI to check if the outliers are removed
ggplot(heart, aes(x = HeartDisease , y = BMI)) +
  geom_boxplot(fill = "#FF6666") +
  labs(y = "BMI", x = "Heart Disease") +
  ggtitle("Boxplot of BMI against Heart Disease")
```

```
## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
```

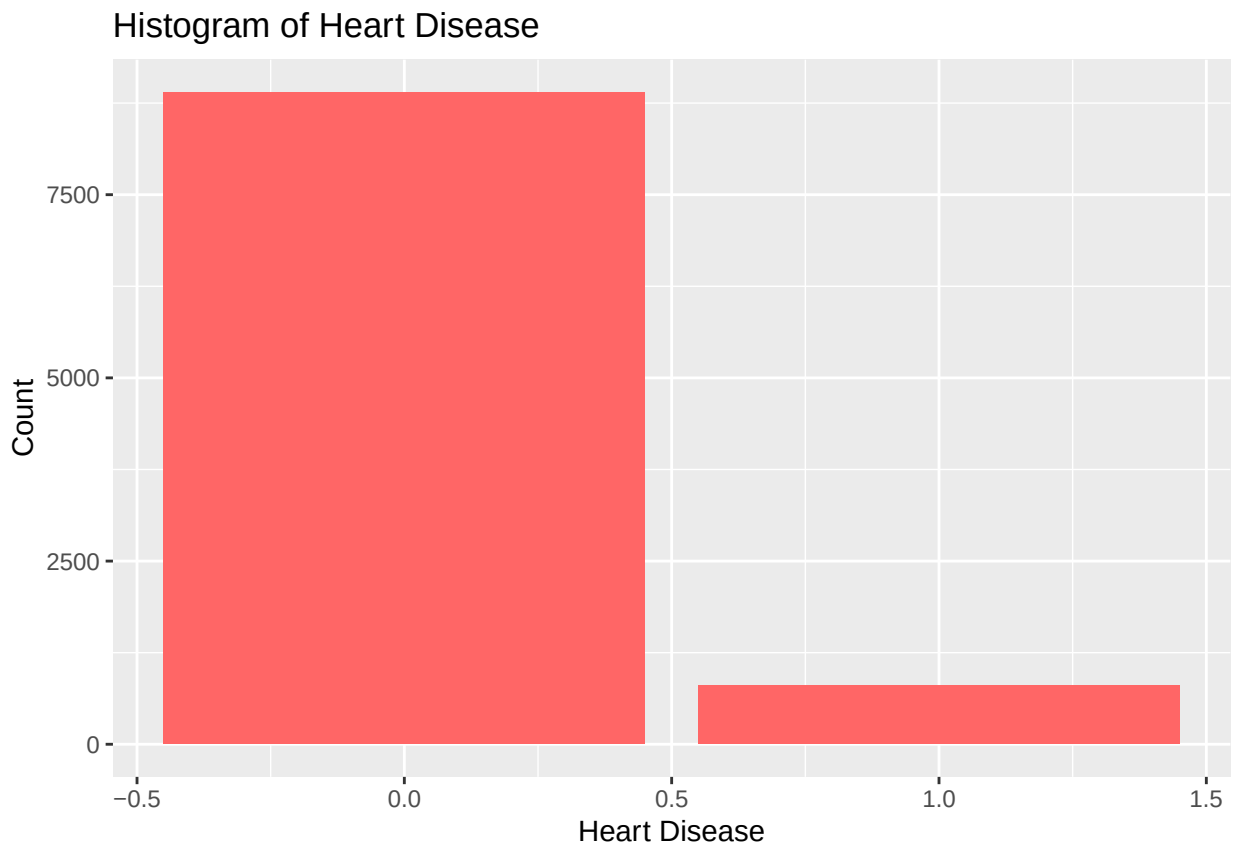



```
#scaling the continuous variable
heart$BMI <- scale(heart$BMI)
summary(heart$BMI)
```

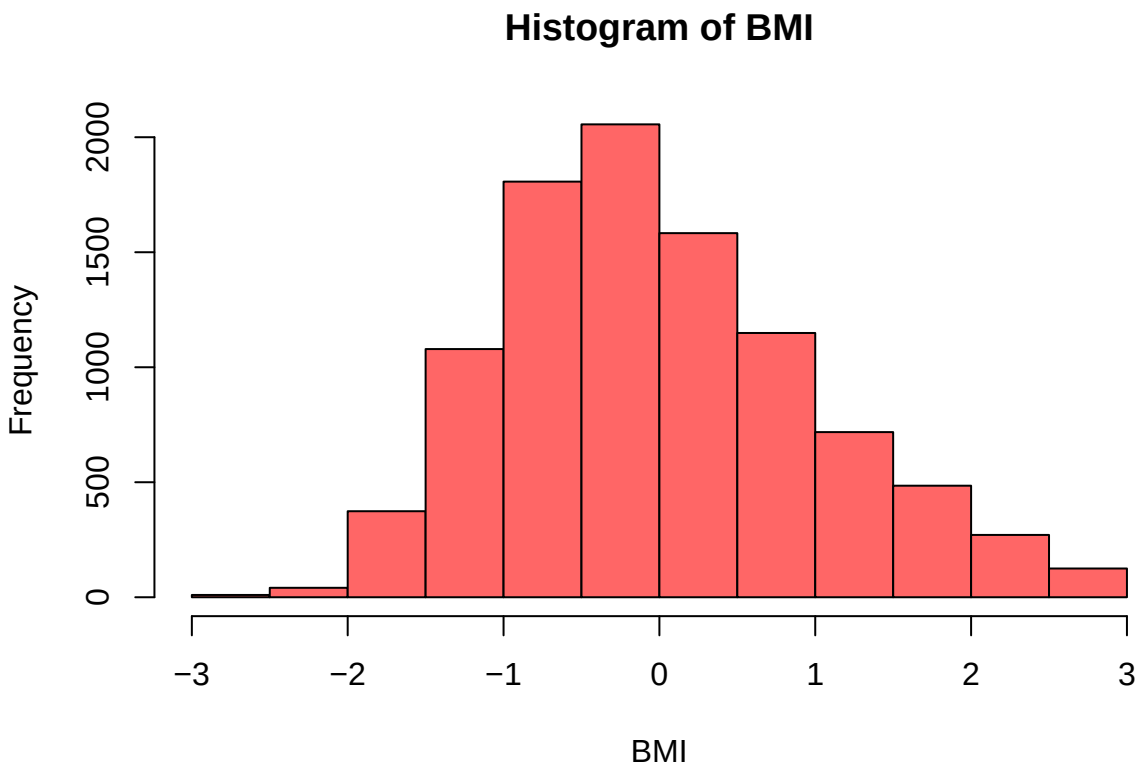
```
##        V1
##  Min.   :-2.8173
##  1st Qu.:-0.7228
##  Median :-0.1160
```

```
##  Mean   : 0.0000
##  3rd Qu.: 0.6193
##  Max.   : 2.8536
```

```r
#Data Visualisation
# Histogram for checking the balance of the outcome variable HeartDisease for classification
ggplot(heart, aes(x = HeartDisease)) +
  geom_bar(position = position_dodge(preserve = "single"), fill = "#FF6666") +
  labs(title = "Histogram of Heart Disease", x = "Heart Disease", y = "Count")
```

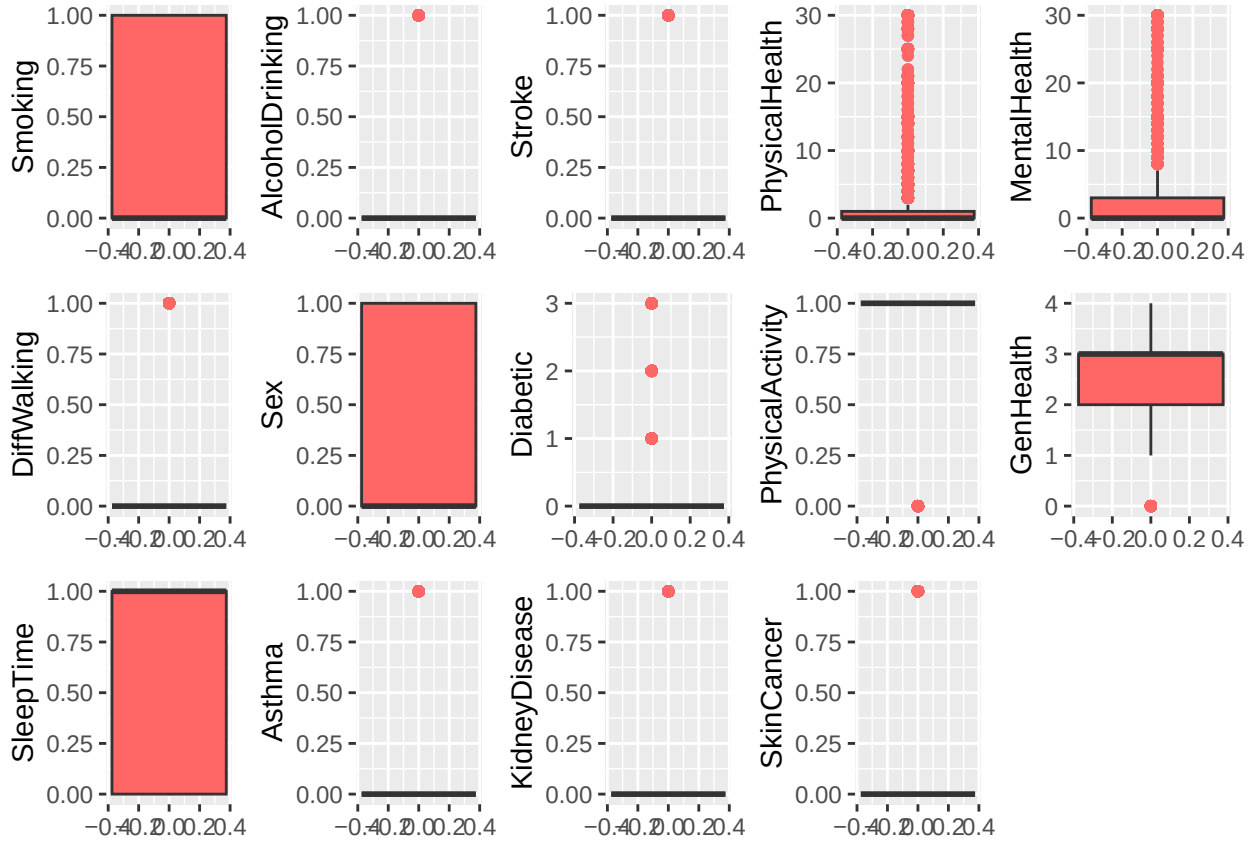



```r
#Histogram of the continuous variable BMI
hist(heart$BMI, main = "Histogram of BMI", xlab = "BMI", col = "#FF6666")
```




```
#Visualising the categorical variables using a plot grid of box plots
library(ggplot2)
library(cowplot)
b0 <- ggplot(data = heart, aes(y= HeartDisease))+
  geom_boxplot(outlier.color = "#FF6666", fill = "#FF6666")
b1 <- ggplot(data = heart, aes(y= Smoking))+
  geom_boxplot(outlier.color = "#FF6666", fill = "#FF6666")
b2 <- ggplot(data = heart, aes(y = AlcoholDrinking))+
  geom_boxplot(outlier.color = "#FF6666", fill = "#FF6666")
b3 <- ggplot(data = heart, aes(y = Stroke))+
  geom_boxplot(outlier.color = "#FF6666", fill = "#1CA16O")
b4 <- ggplot(data = heart, aes(y = PhysicalHealth))+
  geom_boxplot(outlier.color = "#FF6666", fill = "#FF6666")
b5 <- ggplot(data = heart, aes(y = MentalHealth))+
  geom_boxplot(outlier.color = "#FF6666", fill = "#FF6666")
b6 <- ggplot(data = heart, aes(y = DiffWalking))+
  geom_boxplot(outlier.color = "#FF6666", fill = "#FF6666")
b7 <- ggplot(data = heart, aes(y = Sex))+
  geom_boxplot(outlier.color = "#FF6666", fill = "#FF6666")
b8 <- ggplot(data = heart, aes(y = Diabetic))+
  geom_boxplot(outlier.color = "#FF6666", fill = "#FF6666")
b9 <- ggplot(data = heart, aes(y = PhysicalActivity))+
  geom_boxplot(outlier.color = "#FF6666", fill = "#FF6666")
b10 <- ggplot(data = heart, aes(y = GenHealth))+
  geom_boxplot(outlier.color = "#FF6666", fill = "#FF6666")
b11 <- ggplot(data = heart, aes(y = SleepTime))+
```

```
  geom_boxplot(outlier.color = "#FF6666", fill = "#FF6666")
b12 <- ggplot(data = heart, aes(y = Asthma))+
  geom_boxplot(outlier.color = "#FF6666", fill = "#FF6666")
b13 <- ggplot(data = heart, aes(y = KidneyDisease))+
  geom_boxplot(outlier.color = "#FF6666", fill = "#FF6666")
b14 <- ggplot(data = heart, aes(y = SkinCancer))+
  geom_boxplot(outlier.color = "#FF6666", fill = "#FF6666")
b <- plot_grid(b1, b2, b3, b4, b5,
               b6, b7, b8, b9, b10,
               b11, b12, b13, b14,
               ncol = 5, label_fontface = "italic", rel_heights = c(1.2,1.2,1.2))
b
```




```
#using caret package to create partition and to create classification models
library(caret)
```

```
## Loading required package: lattice
```

```
library(class)
library(lattice)
#setting seed for reproductibility
set.seed(5420)
#Splitting data into Training and Testing Data sets
#70% is Train data and 30% is Test data
```

```r
index <- createDataPartition(heart$HeartDisease, p = 0.7, list = FALSE)
train_heart <- heart[index, ]
test_heart <- heart[-(index), ]
train_heart$HeartDisease <- as.factor(train_heart$HeartDisease)
test_heart$HeartDisease <- as.factor(test_heart$HeartDisease)
```

```r
#Classification models
#KNN Classifier
library("MLmetrics")
```

```
##
## Attaching package: 'MLmetrics'

## The following objects are masked from 'package:caret':
##
##     MAE, RMSE

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

```r
knn_model <- train(HeartDisease~., data = train_heart, method = "knn")
knn_predict <- predict(knn_model, newdata = test_heart)
knn_cm <- confusionMatrix(knn_predict, test_heart$HeartDisease)
knn_f1 <- F1_Score(knn_predict, test_heart$HeartDisease)
knn_cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2653  234
##          1   14    8
##
##                Accuracy : 0.9147
##                  95% CI : (0.904, 0.9246)
##     No Information Rate : 0.9168
##     P-Value [Acc > NIR] : 0.6714
##
##                   Kappa : 0.0474
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.99475
##             Specificity : 0.03306
##          Pos Pred Value : 0.91895
##          Neg Pred Value : 0.36364
##              Prevalence : 0.91681
##          Detection Rate : 0.91200
##    Detection Prevalence : 0.99244
##       Balanced Accuracy : 0.51390
##
```

```
##        'Positive' Class : 0
##
```

```
cat("F1 score of KNN Classifier:", knn_f1)
```

```
## F1 score of KNN Classifier: 0.9553475
```

```
#SVM Classifier
library(e1071)
svm_model <- svm(HeartDisease~., data = train_heart)
svm_predict <- predict(svm_model, newdata = test_heart)
svm_cm <- confusionMatrix(svm_predict, test_heart$HeartDisease)
svm_f1 <- F1_Score(svm_predict, test_heart$HeartDisease)
svm_cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2664  238
##          1    3    4
##
##                Accuracy : 0.9172
##                  95% CI : (0.9065, 0.9269)
##     No Information Rate : 0.9168
##     P-Value [Acc > NIR] : 0.4903
##
##                   Kappa : 0.0276
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.99888
##             Specificity : 0.01653
##          Pos Pred Value : 0.91799
##          Neg Pred Value : 0.57143
##              Prevalence : 0.91681
##          Detection Rate : 0.91578
##    Detection Prevalence : 0.99759
##       Balanced Accuracy : 0.50770
##
##        'Positive' Class : 0
##
```

```
cat("f1 score of SVM Classifier: ", svm_f1)
```

```
## f1 score of SVM Classifier:  0.9567247
```

```
#Random Forest Classifier
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
r_model <- randomForest(HeartDisease~., data = train_heart)
r_predict <- predict(r_model, newdata = test_heart)
r_cm <- confusionMatrix(r_predict, test_heart$HeartDisease)
r_f1 <- F1_Score(r_predict, test_heart$HeartDisease)
r_cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2659  235
##          1    8    7
##
##                Accuracy : 0.9165
##                  95% CI : (0.9058, 0.9263)
##     No Information Rate : 0.9168
##     P-Value [Acc > NIR] : 0.5438
##
##                   Kappa : 0.0452
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.99700
##             Specificity : 0.02893
##          Pos Pred Value : 0.91880
##          Neg Pred Value : 0.46667
##              Prevalence : 0.91681
##          Detection Rate : 0.91406
##    Detection Prevalence : 0.99484
##       Balanced Accuracy : 0.51296
##
##        'Positive' Class : 0
##
```

```
cat("F1 score of Random Forest Classifier: ", r_f1)
```

```
## F1 score of Random Forest Classifier:  0.9563028
```

```r
#Decision Tree Classifier
library(rpart)
d_model <- rpart(HeartDisease~., data = train_heart)
d_predict <- predict(d_model, newdata = test_heart, type = "class")
d_cm <- confusionMatrix(d_predict, test_heart$HeartDisease)
d_f1 <- F1_Score(d_predict, test_heart$HeartDisease)
d_cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2667  242
##          1    0    0
##
##                Accuracy : 0.9168
##                  95% CI : (0.9062, 0.9266)
##     No Information Rate : 0.9168
##     P-Value [Acc > NIR] : 0.5171
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.9168
##          Neg Pred Value :    NaN
##              Prevalence : 0.9168
##          Detection Rate : 0.9168
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 0
##
```

```r
cat("f1 score of Decision Tree: ",d_f1)
```

```
## f1 score of Decision Tree:  0.9565997
```

```r
#Naive Bayes Classifier
library(naivebayes)
```

```
## naivebayes 0.9.7 loaded
```

```r
n_model <- naiveBayes(HeartDisease~., data = train_heart)
n_predict <- predict(n_model, newdata = test_heart, type = "class")
n_cm <- confusionMatrix(n_predict, test_heart$HeartDisease)
n_f1 <- F1_Score(n_predict, test_heart$HeartDisease)
n_cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2360  121
##          1  307  121
##
##                Accuracy : 0.8529
##                  95% CI : (0.8395, 0.8656)
##     No Information Rate : 0.9168
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2852
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8849
##             Specificity : 0.5000
##          Pos Pred Value : 0.9512
##          Neg Pred Value : 0.2827
##              Prevalence : 0.9168
##          Detection Rate : 0.8113
##    Detection Prevalence : 0.8529
##       Balanced Accuracy : 0.6924
##
##        'Positive' Class : 0
##
```

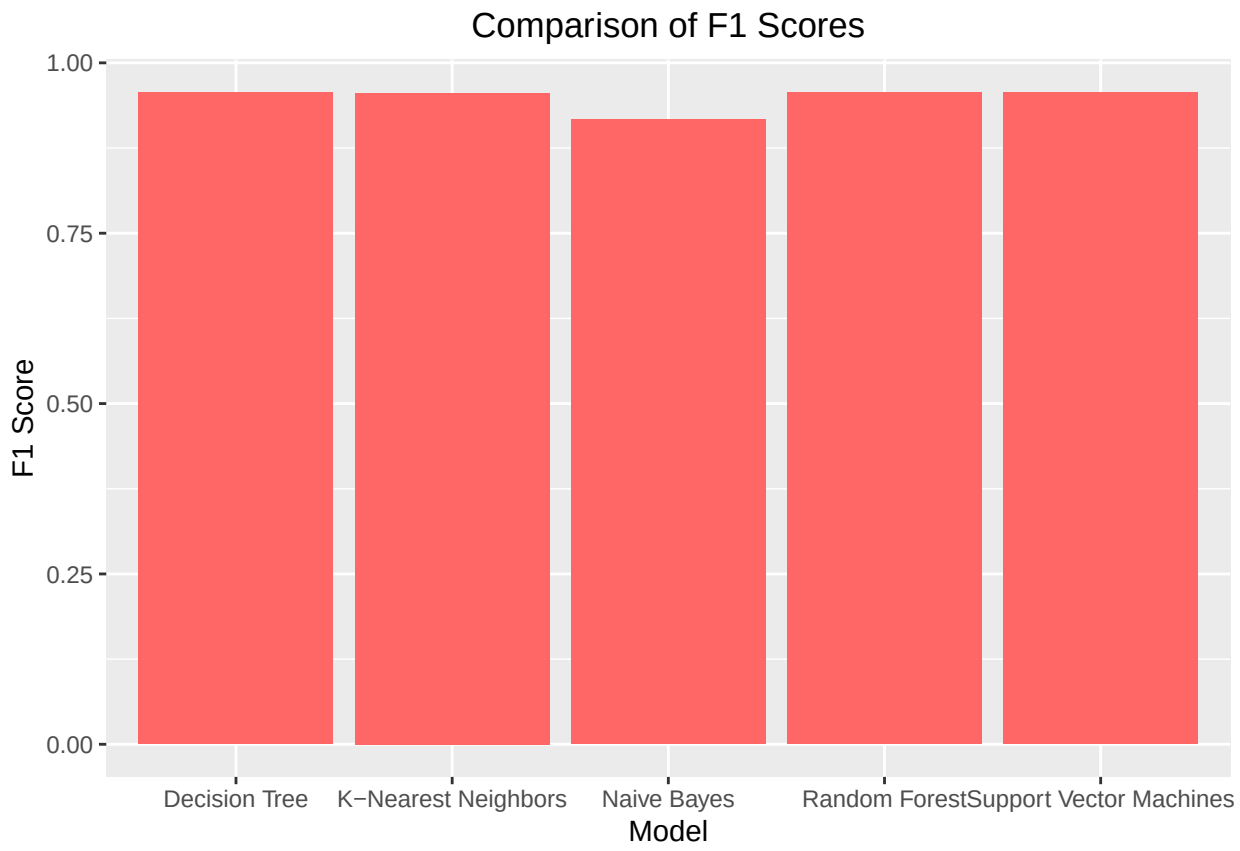
```
cat("F1 score of Naive Bayes Classifier: ",n_f1)
```

```
## F1 score of Naive Bayes Classifier:  0.9168609
```

```
#Comparing f1 scores by plotting
f1_scores <- data.frame(Model = c("K-Nearest Neighbors", "Support Vector Machines", "Random Forest", "Na
                        F1_Score = c(knn_f1,svm_f1, r_f1, n_f1, d_f1))
f1_scores
```

```
##                     Model  F1_Score
## 1     K-Nearest Neighbors 0.9553475
## 2 Support Vector Machines 0.9567247
## 3           Random Forest 0.9563028
## 4             Naive Bayes 0.9168609
## 5           Decision Tree 0.9565997
```

```
ggplot(f1_scores, aes(x = Model, y = F1_Score)) +
geom_bar(stat = "identity", fill = "#FF6666") +
ggtitle("Comparison of F1 Scores") +
xlab("Model") +
ylab("F1 Score") +
theme(plot.title = element_text(hjust = 0.5))
```


Comparison of F1 Scores
1.00
0.75
0.50
0.25
0.00
F1 Score
Decision Tree
K−Nearest Neighbors
Naive Bayes
Random Forest
Support Vector Machines
Model