Programming in R

General Homework

Sofía Gianelli Nan, Gabriela Levenfeld Sabau and Miguel Díaz-Plaza Cabrera

2023-11-05

Contents

Take a dataset

The dataset has been extracted from Kaggle: https://www.kaggle.com/datasets/iamsouravbanerjee/heart-attack-prediction-dataset/

This dataset provides a comprehensive array of features relevant to heart health and lifestyle choices, encompassing patient-specific details such as age, gender, cholesterol levels, blood pressure, heart rate, and indicators like diabetes, family history, smoking habits, obesity, and alcohol consumption. Additionally, lifestyle factors like exercise hours, dietary habits, stress levels, and sedentary hours are included. Medical aspects comprising previous heart problems, medication usage, and triglyceride levels are considered. Socioeconomic aspects such as income and geographical attributes like country, continent, and hemisphere are incorporated. The dataset, consisting of 8763 records from patients around the globe, culminates in a crucial binary classification feature denoting the presence or absence of a heart attack risk, providing a comprehensive resource for predictive analysis and research in cardiovascular health.

- `Patient ID` - Unique identifier for each patient
- `Age` - Age of the patient
- `Sex` - Gender of the patient (Male/Female)
- `Cholesterol` - Cholesterol levels of the patient
- `Blood Pressure` - Blood pressure of the patient (systolic/diastolic)
- `Heart Rate` - Heart rate of the patient
- `Diabetes` - Whether the patient has diabetes (Yes/No)
- `Family History` - Family history of heart-related problems (1: Yes, 0: No)
- `Smoking` - Smoking status of the patient (1: Smoker, 0: Non-smoker)
- `Obesity` - Obesity status of the patient (1: Obese, 0: Not obese)
- `Alcohol Consumption` - Level of alcohol consumption by the patient (None/Light/Moderate/Heavy)
- `Exercise Hours Per Week` - Number of exercise hours per week
- `Diet` - Dietary habits of the patient (Healthy/Average/Unhealthy)
- `Previous Heart Problems` - Previous heart problems of the patient (1: Yes, 0: No)
- `Medication Use` - Medication usage by the patient (1: Yes, 0: No)
- `Stress Level` - Stress level reported by the patient (1-10)
- `Sedentary Hours Per Day` - Hours of sedentary activity per day
- `Income` - Income level of the patient
- `BMI` - Body Mass Index (BMI) of the patient
- `Triglycerides` - Triglyceride levels of the patient
- `Physical Activity Days Per Week` - Days of physical activity per week
- `Sleep Hours Per Day` - Hours of sleep per day
- `Country` - Country of the patient
- `Continent` - Continent where the patient resides
- `Hemisphere` - Hemisphere where the patient resides
- `Heart Attack Risk` - Presence of heart attack risk (1: Yes, 0: No)

Before starting the project, we need to load some libraries in order to used them.

```
library(readxl)
library(skimr)
library(dplyr)
library(knitr)
library(DescTools)
library(e1071)
library(ggplot2)
library(gmodels)
library(reshape2)
library(tidyr)
library(corrplot)
library(vcd)
library(ggmosaic)
```

The first step is to import the dataset:

```
data <- read_excel("heart_attack_prediction_dataset.xlsx")
```

```
# For knowing the data base
summary(data)
```

```
##   Patient_ID             Age            Sex             Cholesterol
##  Length:8763        Min.   :18.00   Length:8763        Min.   :120.0
##  Class :character   1st Qu.:35.00   Class :character   1st Qu.:192.0
##  Mode  :character   Median :54.00   Mode  :character   Median :259.0
##                     Mean   :53.71                      Mean   :259.9
##                     3rd Qu.:72.00                      3rd Qu.:330.0
##                     Max.   :90.00                      Max.   :400.0
##  Blood_Pressure       Heart_Rate        Diabetes      Family_History
##  Length:8763        Min.   : 40.00   Min.   :0.0000   Min.   :0.000
##  Class :character   1st Qu.: 57.00   1st Qu.:0.0000   1st Qu.:0.000
##  Mode  :character   Median : 75.00   Median :1.0000   Median :0.000
##                     Mean   : 75.02   Mean   :0.6523   Mean   :0.493
##                     3rd Qu.: 93.00   3rd Qu.:1.0000   3rd Qu.:1.000
##                     Max.   :110.00   Max.   :1.0000   Max.   :1.000
##    Smoking          Obesity       Alcohol_Consumption Exercise_Hours_Per_Week
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000      Min.   : 0.002442
##  1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.:0.0000      1st Qu.: 4.981579
##  Median :1.0000   Median :1.0000   Median :1.0000      Median :10.069559
##  Mean   :0.8968   Mean   :0.5014   Mean   :0.5981      Mean   :10.014284
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000      3rd Qu.:15.050018
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000      Max.   :19.998709
##      Diet           Previous_Heart_Problems Medication_Use    Stress_Level
##  Length:8763        Min.   :0.0000          Min.   :0.0000   Min.   : 1.00
##  Class :character   1st Qu.:0.0000          1st Qu.:0.0000   1st Qu.: 3.00
##  Mode  :character   Median :0.0000          Median :0.0000   Median : 5.00
##                     Mean   :0.4958          Mean   :0.4983   Mean   : 5.47
##                     3rd Qu.:1.0000          3rd Qu.:1.0000   3rd Qu.: 8.00
##                     Max.   :1.0000          Max.   :1.0000   Max.   :10.00
##  Sedentary_Hours_Per_Day     Income             BMI        Triglycerides
##  Min.   : 0.001263       Min.   : 20062   Min.   :18.00   Min.   : 30.0
##  1st Qu.: 2.998794       1st Qu.: 88310   1st Qu.:23.42   1st Qu.:225.5
##  Median : 5.933622       Median :157866   Median :28.77   Median :417.0
##  Mean   : 5.993690       Mean   :158263   Mean   :28.89   Mean   :417.7
##  3rd Qu.: 9.019124       3rd Qu.:227749   3rd Qu.:34.32   3rd Qu.:612.0
##  Max.   :11.999313       Max.   :299954   Max.   :40.00   Max.   :800.0
##  Physical_Activity_Days_Per_Week Sleep_Hours_Per_Day   Country
##  Min.   :0.00                    Min.   : 4.000      Length:8763
##  1st Qu.:2.00                    1st Qu.: 5.000      Class :character
##  Median :3.00                    Median : 7.000      Mode  :character
##  Mean   :3.49                    Mean   : 7.024
##  3rd Qu.:5.00                    3rd Qu.: 9.000
##  Max.   :7.00                    Max.   :10.000
##   Continent          Hemisphere        Heart_Attack_Risk
##  Length:8763        Length:8763        Min.   :0.0000
##  Class :character   Class :character   1st Qu.:0.0000
##  Mode  :character   Mode  :character   Median :0.0000
##                                        Mean   :0.3582
##                                        3rd Qu.:1.0000
##                                        Max.   :1.0000
```

Looking for missing (NA) values in the dataset. As the result is 0, then it means that there are no missing values (NA).

```
# We check if there are any missing values (NA's) or not
sum(is.na(data)==TRUE)
```

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

In order to obtain a more exhaustive analysis of the dataset, we are going to apply an Exploratory Data Analysis (EDA). For that, we are going to use the functions `skim()` and `str()` included in the `library(skimr)`.

```
skim_without_charts(data)
```

Table 1: Data summary

Name	data
Number of rows	8763
Number of columns	26
Column type frequency:	
character	7
numeric	19
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Patient_ID	0	1	6	7	0	8763	0
Sex	0	1	4	6	0	2	0

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Blood_Pressure	0	1	5	7	0	3915	0
Diet	0	1	7	9	0	3	0
Country	0	1	5	14	0	20	0
Continent	0	1	4	13	0	6	0
Hemisphere	0	1	19	19	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Age	0	1	53.71	21.25	18	35.00	54.00	72.00	90
Cholesterol	0	1	259.88	80.86	120	192.00	259.00	330.00	400
Heart_Rate	0	1	75.02	20.55	40	57.00	75.00	93.00	110
Diabetes	0	1	0.65	0.48	0	0.00	1.00	1.00	1
Family_History	0	1	0.49	0.50	0	0.00	0.00	1.00	1
Smoking	0	1	0.90	0.30	0	1.00	1.00	1.00	1
Obesity	0	1	0.50	0.50	0	0.00	1.00	1.00	1
Alcohol_Consumption	0	1	0.60	0.49	0	0.00	1.00	1.00	1
Exercise_Hours_Per_Week	0	1	10.01	5.78	0	4.98	10.07	15.05	20
Previous_Heart_Problems	0	1	0.50	0.50	0	0.00	0.00	1.00	1
Medication_Use	0	1	0.50	0.50	0	0.00	0.00	1.00	1
Stress_Level	0	1	5.47	2.86	1	3.00	5.00	8.00	10
Sedentary_Hours_Per_Day	0	1	5.99	3.47	0	3.00	5.93	9.02	12
Income	0	1	158263.18	80575.19	20062	88310.00	157866.00	227749.00	299954
BMI	0	1	28.89	6.32	18	23.42	28.77	34.32	40
Triglycerides	0	1	417.68	223.75	30	225.50	417.00	612.00	800
Physical_Activity_Days_Per_Week	0	1	3.49	2.28	0	2.00	3.00	5.00	7
Sleep_Hours_Per_Day	0	1	7.02	1.99	4	5.00	7.00	9.00	10
Heart_Attack_Risk	0	1	0.36	0.48	0	0.00	0.00	1.00	1

```
str(data)
```

```
## tibble [8,763 x 26] (S3: tbl_df/tbl/data.frame)
##  $ Patient_ID                     : chr [1:8763] "BMW7812" "CZE1114" "BNI9906" "JLN3497" ...
##  $ Age                            : num [1:8763] 67 21 21 84 66 54 90 84 20 43 ...
##  $ Sex                            : chr [1:8763] "Male" "Male" "Female" "Male" ...
##  $ Cholesterol                    : num [1:8763] 208 389 324 383 318 297 358 220 145 248 ...
##  $ Blood_Pressure                 : chr [1:8763] "158/88" "165/93" "174/99" "163/100" ...
##  $ Heart_Rate                     : num [1:8763] 72 98 72 73 93 48 84 107 68 55 ...
##  $ Diabetes                       : num [1:8763] 0 1 1 1 1 1 0 0 1 0 ...
##  $ Family_History                 : num [1:8763] 0 1 0 1 1 1 0 0 0 1 ...
##  $ Smoking                        : num [1:8763] 1 1 0 1 1 1 1 1 1 1 ...
##  $ Obesity                        : num [1:8763] 0 1 0 0 1 0 0 1 1 1 ...
##  $ Alcohol_Consumption            : num [1:8763] 0 1 0 1 0 1 1 1 0 1 ...
##  $ Exercise_Hours_Per_Week        : num [1:8763] 4.17 1.81 2.08 9.83 5.8 ...
##  $ Diet                           : chr [1:8763] "Average" "Unhealthy" "Healthy" "Average" ...
##  $ Previous_Heart_Problems        : num [1:8763] 0 1 1 1 1 1 0 0 0 0 ...
##  $ Medication_Use                 : num [1:8763] 0 0 1 0 0 1 0 1 0 0 ...
##  $ Stress_Level                   : num [1:8763] 9 1 9 9 6 2 7 4 5 4 ...
##  $ Sedentary_Hours_Per_Day        : num [1:8763] 6.62 4.96 9.46 7.65 1.51 ...
##  $ Income                         : num [1:8763] 261404 285768 235282 125640 160555 ...
##  $ BMI                            : num [1:8763] 31.3 27.2 28.2 36.5 21.8 ...
##  $ Triglycerides                  : num [1:8763] 286 235 587 378 231 795 284 370 790 232 ...
##  $ Physical_Activity_Days_Per_Week: num [1:8763] 0 1 4 3 1 5 4 6 7 7 ...
##  $ Sleep_Hours_Per_Day            : num [1:8763] 6 7 4 4 5 10 10 7 4 7 ...
##  $ Country                        : chr [1:8763] "Argentina" "Canada" "France" "Canada" ...
##  $ Continent                      : chr [1:8763] "South America" "North America" "Europe" "North Amer
##  $ Hemisphere                     : chr [1:8763] "Southern Hemisphere" "Northern Hemisphere" "Norther
##  $ Heart_Attack_Risk              : num [1:8763] 0 0 0 0 0 1 1 1 0 0 ...
```

Part 1: Make a basic descriptive study

(i) Frequency tables for at least two of the continuous variables.

The first frequency table is Sleep Hours Per Day.

```
# Create a frequency table for Sleep_Hours_Per_Day
freq_table1 <- data %>%
  group_by(Sleep_Hours_Per_Day) %>%
  summarize(freq = n())

# Displaying the first frequency table
kable(freq_table1, align = c("l", "c"),
      col.names = c("Sleep Hours Per Day", "Frequency"),
      caption = "Sleep Hours Per Day Frequency Table")
```

Table 4: Sleep Hours Per Day Frequency Table

Sleep Hours Per Day	Frequency
4	1181
5	1263
6	1276
7	1270
8	1288
9	1192
10	1293

If we want to know how many hours per day sleeps women and men we do the following frequency table:

```
# Create a frequency table for Sleep_Hours_Per_Day by sex
freq_table2 <- data %>%
  group_by(Sex) %>%
  summarize("Average Sleep Hours Per Day" = round(mean(Sleep_Hours_Per_Day), 2))

kable(freq_table2, align = c("l", "c"), caption = "Average Sleep Hours Per Day by Sex")
```

Table 5: Average Sleep Hours Per Day by Sex

Sex	Average Sleep Hours Per Day
Female	7.04
Male	7.02

Now we can create a frequency table with the level of cholesterol by age

```
# Define a interval
breaks <- c(0,20,30,40,50,60,70,80,90,100)
data <- data %>%
  mutate(interval1 = cut(Age, breaks = breaks, right = FALSE))
```

```r
# Create a frequency table for Cholesterol by age
freq_table3 <- data %>%
  group_by(Age = interval1) %>%
  summarise("Average Cholesterol" = round(mean(Cholesterol)))

kable(freq_table3, align = c("l", "c"), caption = "Average Cholesterol by Age")
```

Table 6: Average Cholesterol by Age

Age	Average Cholesterol
[0,20)	254
[20,30)	261
[30,40)	263
[40,50)	259
[50,60)	260
[60,70)	259
[70,80)	263
[80,90)	256
[90,100)	255

If we want to know the stress level by country

```r
# Create a frequency table for Stress level by country
freq_table4 <- data %>%
  group_by(Country) %>%
  summarise("Average Stress Level" = round(mean(Stress_Level),1))
kable(freq_table4, align = c("l", "c"), caption = "Average Strees Level by Country")
```

Table 7: Average Strees Level by Country

Country	Average Stress Level
Argentina	5.7
Australia	5.4
Brazil	5.3
Canada	5.5
China	5.5
Colombia	5.6
France	5.4
Germany	5.4
India	5.2
Italy	5.4
Japan	5.5
New Zealand	5.3
Nigeria	5.6
South Africa	5.6
South Korea	5.6
Spain	5.2
Thailand	5.5
United Kingdom	5.6
United States	5.5

Country	Average Stress Level
Vietnam	5.3

(ii) Calculate measures of centrality, variability, and shape (skewness and kurtosis). Interpret results.

To calculate measures of centrality we are going to calculate the mean, median and mode of two variables. Then, to calculate measures of variability we are going to find the standard deviation, variance and range of the same two variables. Finally, we will compute the skewness and kurtosis.

- **Sedentary Hours Per Day**

This is a continuous variable.

```r
# Mean, median and mode

mean_value1 <- round(mean(data$Sedentary_Hours_Per_Day),1)
median_value1 <- round(median(data$Sedentary_Hours_Per_Day),1)

measures_of_centrality1 <- data.frame(
  Statistic = c("Mean of Sedentary Hours Per Day", "Median of Sedentary Hours Per Day"),
  Value = c(mean_value1, median_value1)
)

kable(measures_of_centrality1,
      align = c("l", "c"),
      col.names = c("Statistic", "Value"),
      caption = "Measures of Centrality")
```

Table 8: Measures of Centrality

Statistic	Value
Mean of Sedentary Hours Per Day	6.0
Median of Sedentary Hours Per Day	5.9

The average of sedentary hours per day is 6.0 and the median is 5.9, almost the same, this means that the distribution is almost symmetric.

```r
# Variance, standard deviation and range

variance_value1 <- round(var(data$Sedentary_Hours_Per_Day),1)
std_deviation_value1 <- round(sd(data$Sedentary_Hours_Per_Day),1)
range_value1 <- round(max(data$Sedentary_Hours_Per_Day)
                      - min(data$Sedentary_Hours_Per_Day),1)


measures_of_variability1 <- data.frame(
  Statistic = c("Variance of Sedentary Hours Per Day",
                "Standard Deviation of Sedentary Hours Per Day",
                "Range of Sedentary Hours Per Day"),
  Value = c(variance_value1,std_deviation_value1, range_value1)
)

kable(measures_of_variability1,
      align = c("l", "c"),
      col.names = c("Statistic", "Value"),
      caption = "Measures of Variability")
```

Table 9: Measures of Variability

Statistic	Value
Variance of Sedentary Hours Per Day	12.0
Standard Deviation of Sedentary Hours Per Day	3.5
Range of Sedentary Hours Per Day	12.0

The variance suggests that the data points are spread out a bit from the mean. The standard deviation is a measure of how much individual data points deviate from the mean, in this case the variability is small.

The range indicates the difference between the maximum and the minimum, this difference is 12 hours.

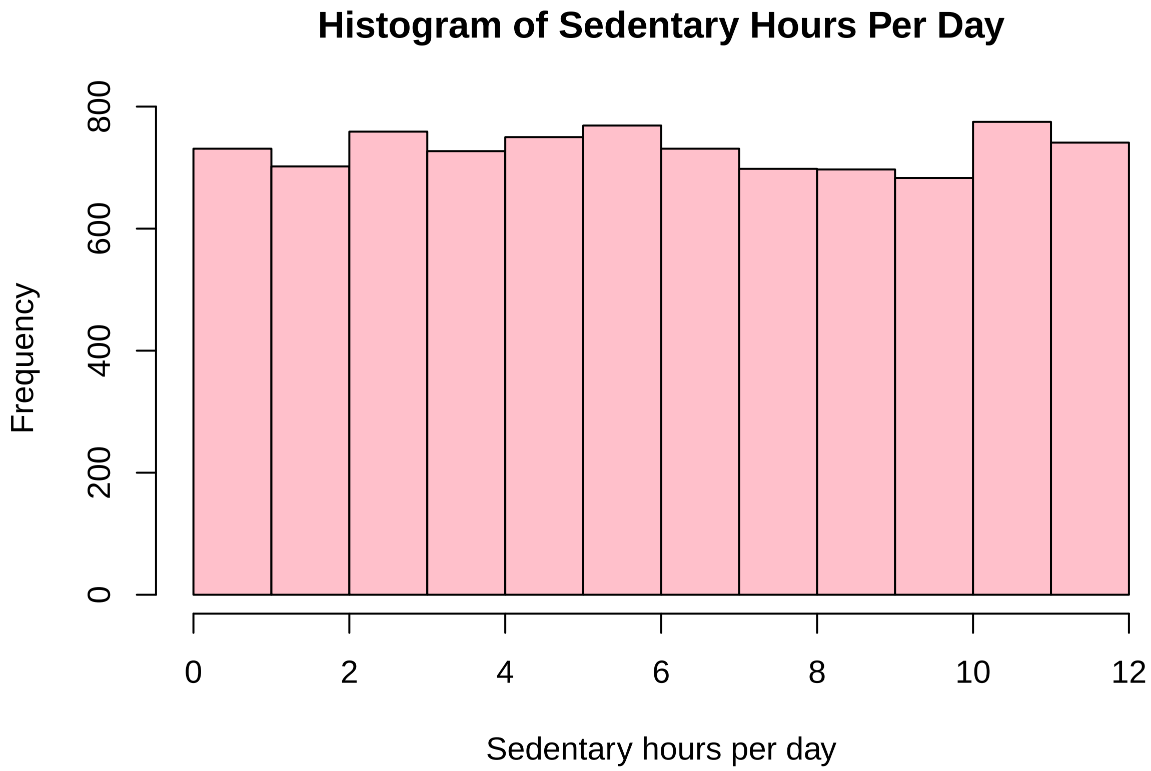
```
# Skewness and kurtosis
skewness_value1 <- round(skewness(data$Sedentary_Hours_Per_Day),1)
kurtosis_value1 <- round(kurtosis(data$Sedentary_Hours_Per_Day),1)

measures_of_shape1 <- data.frame(
  Statistic = c("Skewness", "Kurtosis"),
  Value = c(skewness_value1, kurtosis_value1)
)

kable(measures_of_shape1,
      align = c("l", "c"),
      col.names = c("Statistic", "Value"),
      caption = "Measures of Shape")
```

Table 10: Measures of Shape

Statistic	Value
Skewness	0.0
Kurtosis	-1.2

The skewness is zero, this means that this distribution is approximately symmetrical. And the negative kurtosis means that the data has fewer extreme values and fewer outliers. In this case, we can affirm that the values of sedentary hours is concentrated near the mean with hardly any extreme values.

```
hist(data$Sedentary_Hours_Per_Day,
     main = paste("Histogram of Sedentary Hours Per Day"),
     xlab = "Sedentary hours per day", col = "pink")
```



The histogram shows that the distribution is almost symmetric.

- **Heart rate**

This variable only takes integer values.

```
#Mean, median and mode

mean_heartRate <- round(mean(data$Heart_Rate),1)
median_heartRate <- round(median(data$Heart_Rate),1)
mode_heartRate <- round(Mode(data$Heart_Rate),1)

centrality_heartRate <- data.frame(
  Statistic = c("Mean of Heart Rate", "Median of Heart Rate", "Mode of Heart Rate"),
  Value = c(mean_heartRate, median_heartRate, mode_heartRate)
)

kable(centrality_heartRate,
      align = c("l", "c"),
      col.names = c("Statistic", "Value"),
      caption = "Measures of Centrality")
```

Table 11: Measures of Centrality

Statistic	Value
Mean of Heart Rate	75
Median of Heart Rate	75
Mode of Heart Rate	94

```r
#Variance, standard deviation and range

variance_heartRate <- round(var(data$Heart_Rate),1)
std_deviation_heartRate <- round(sd(data$Heart_Rate),1)
range_heartRate <- round(max(data$Heart_Rate) - min(data$Heart_Rate),1)


variability_heartRate <- data.frame(
  Statistic = c("Variance of Heart Rate"
                , "Standard Deviation of Heart Rate"
                , "Range of Heart Rate"),
  Value = c(variance_heartRate,
            std_deviation_heartRate,
            range_heartRate)
)

kable(variability_heartRate,
      align = c("l", "c"),
      col.names = c("Statistic", "Value"),
      caption = "Measures of Variability")
```

Table 12: Measures of Variability

Statistic	Value
Variance of Heart Rate	422.3
Standard Deviation of Heart Rate	20.6
Range of Heart Rate	70.0

This variable has a large variance, so the data is considerably spread around the mean. Also, we can conclude that most of heart rate values are typically around 20.6 beats per minute far from the mean heart rate. Finally, the difference between the maximum heart rate and the minimum is 70.0 beats per minute.

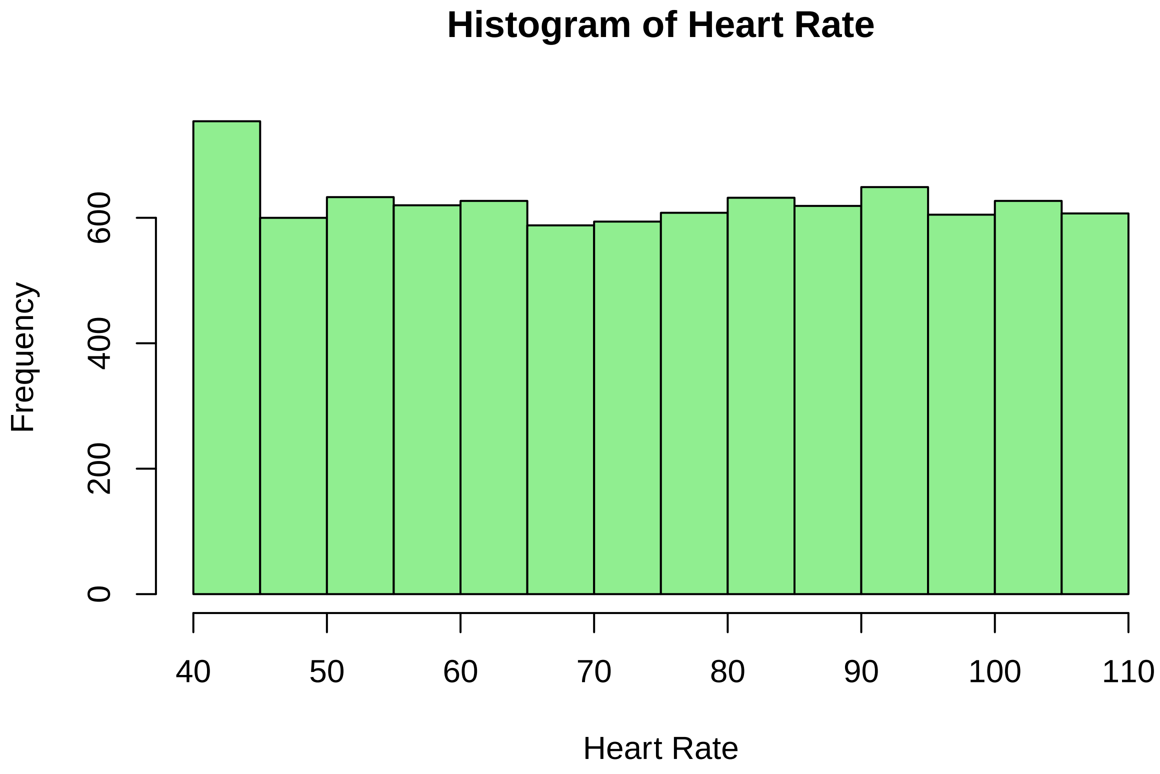
```r
# Skewness and kurtosis
skewness_heartRate <- round(skewness(data$Heart_Rate),1)
kurtosis_heartRate <- round(kurtosis(data$Heart_Rate),1)

shape_heartRate <- data.frame(
  Statistic = c("Skewness", "Kurtosis"),
  Value = c(skewness_heartRate, kurtosis_heartRate)
)

kable(shape_heartRate,
      align = c("l", "c"),
      col.names = c("Statistic", "Value"),
      caption = "Measures of Shape")
```

Table 13: Measures of Shape

Statistic	Value
Skewness	0.0
Kurtosis	-1.2

The skewness and kurtosis is the same as the previous variable, so we can assume the same affirmation as before.

```
hist(data$Heart_Rate,
     main = paste("Histogram of Heart Rate"),
     xlab = "Heart Rate", col = "lightgreen")
```



This histogram appears to be symmetric despite the value of 40 beats per minute.

(iii) Take one of the categorical variables and create groups based on it.

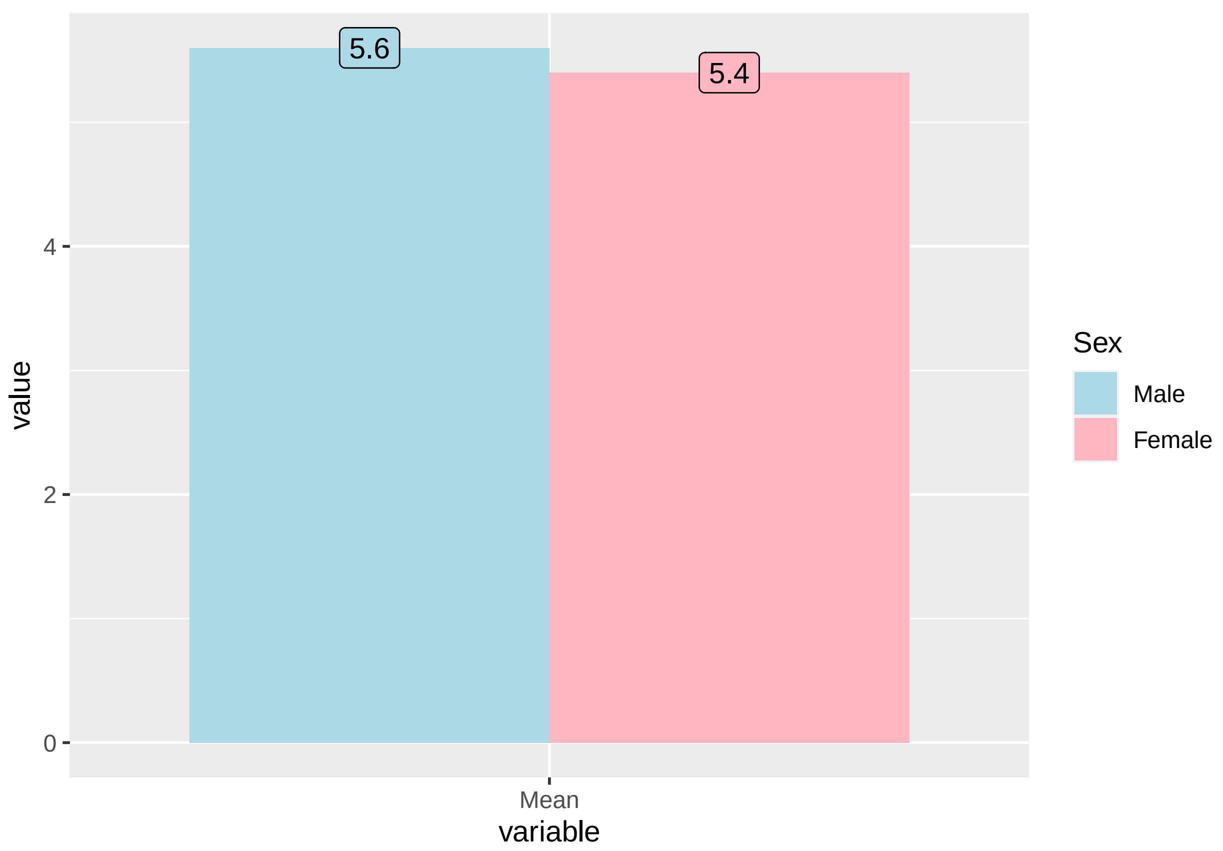
We decide to compare different variables by sex. The first variable to analyze by sex is the **Stress Level**.

```
summary_sex <- data %>%
  group_by(Sex) %>%
  summarise(
    Mean = round(mean(Stress_Level),1),
    Median = round(median(Stress_Level),1),
    SD = round(sd(Stress_Level),1),
    Min = round(min(Stress_Level),1),
    Max = round(max(Stress_Level),1)
  )

kable(summary_sex, align = c("l", "c", "c", "c", "c", "c"),
      col.names = c("Sex", "Mean"
                    , "Median" , "Standard Deviation",
                    "Minimum", "Maximum"),
      caption = "Stress Level by Sex")
```

Table 14: Stress Level by Sex

Sex	Mean	Median	Standard Deviation	Minimum	Maximum
Female	5.6	6	2.9	1	10
Male	5.4	5	2.9	1	10

```
data_sex <- data %>%
  group_by(Sex) %>%
  summarise(
    Mean = round(mean(Stress_Level), 1)
  )
sex <- melt(data_sex, id.vars = "Sex", measure.vars = "Mean")

# Define custom colors and labels
my_colors <- c("#ADD8E6", "#FFB6C1")
my_labels <- c("Male", "Female")

ggplot(sex, aes(x = variable, y = value, fill = Sex)) +
  geom_bar(position = "dodge", stat = "identity") +
  geom_label(aes(y = value, label = round(value, 2),
                accuracy = 0.1),
             position = position_dodge(width = 0.9),
             show.legend = FALSE) +
  scale_fill_manual(values = my_colors,
                    labels = my_labels) +
  theme(legend.position = "right")
```



As we can see, stress level by sex, present data quite similar. The mean of each group is 5.6 (male) and 5.4 (female). Overall, these summary provide insights into the stress levels of the two gender groups. Both groups present similar results in each parameters, suggesting that, on average, their stress levels are close.

Furthermore, we are going to make a p-test to determine if the differences are significant or if there are any patterns or relationships between gender and stress levels in the dataset we have chosen.

```r
# Create a vector for male and another for female
Stress_Male <- data %>%
  filter(Sex == "Male") %>%
  select(Stress_Level) %>%
  unlist()

Stress_Female <- data %>%
  filter(Sex == "Female") %>%
  select(Stress_Level) %>%
  unlist()

# Perform an independent two-sample t-test
t_test_result_1 <- t.test(x=Stress_Male,
                          y=Stress_Female,
                          alternative="two.sided",
                          var.equal = TRUE,
                          paired = FALSE)

# Print the t-test result
print(t_test_result_1)
```

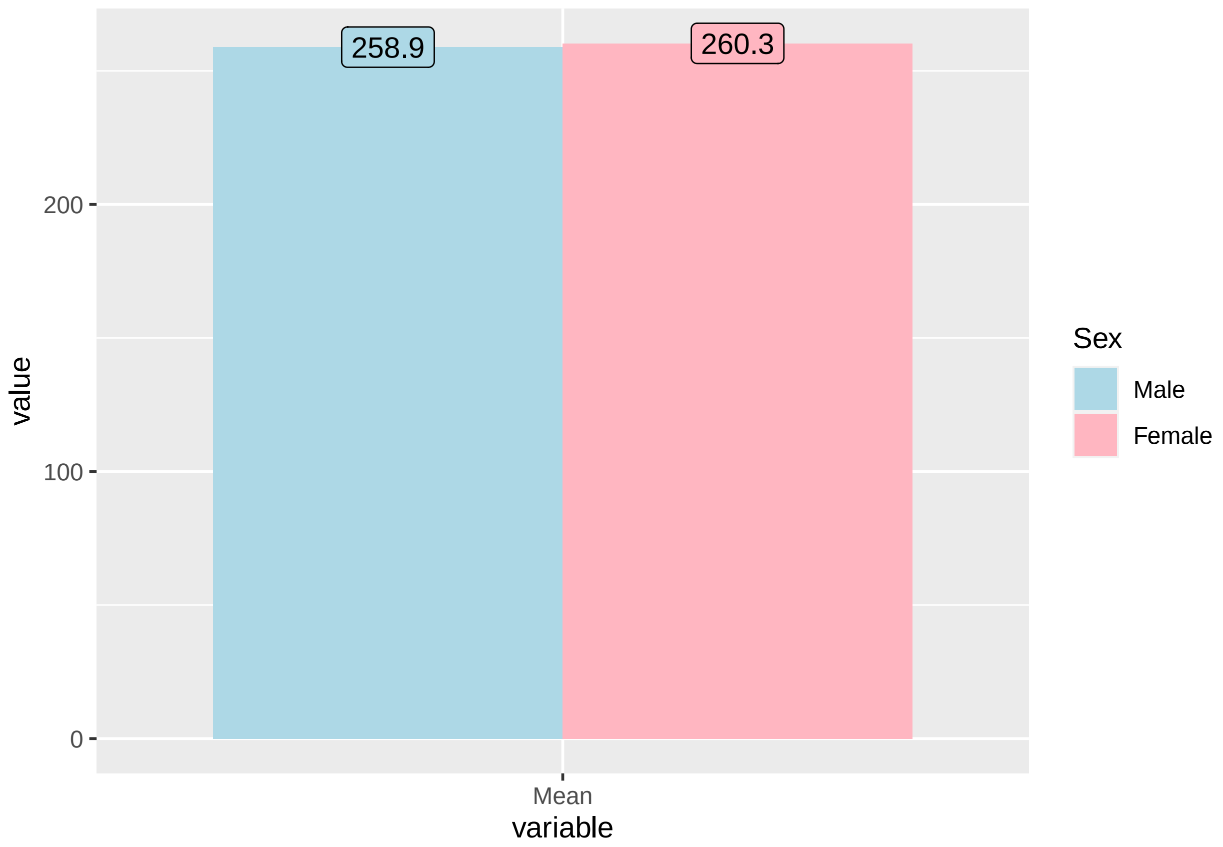
```
##
##  Two Sample t-test
##
## data:  Stress_Male and Stress_Female
## t = -2.0442, df = 8761, p-value = 0.04096
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.266231322 -0.005585097
## sample estimates:
## mean of x mean of y
##  5.428571  5.564480
```

Formulating a hypothesis test with $H_0 : \mu_{male} = \mu_{female}$ and an alternative hypothesis $H_1 : \mu_{male} \neq \mu_{female}$ we can see if this difference is signficative. The p-value of this test is $= 0.04096$ which is $< \alpha$ so we have enough evidence to reject H_0 and conclude that the means calculated before are signficative.

Then, we compare the leveles of **Cholesterol** between female and male.

```
data_sex <- data %>%
  group_by(Sex) %>%
  summarise(
    Mean = round(mean(Cholesterol), 1)
  )
sex <- melt(data_sex, id.vars = "Sex", measure.vars = "Mean")

# Define custom colors and labels
my_colors <- c("#ADD8E6", "#FFB6C1")
my_labels <- c("Male", "Female")

ggplot(sex, aes(x = variable, y = value, fill = Sex)) +
  geom_bar(position = "dodge", stat = "identity") +
  geom_label(aes(y = value,
                 label = round(value, 2), accuracy = 0.1),
             position = position_dodge(width = 0.9),
             show.legend = FALSE) +
  scale_fill_manual(values = my_colors, labels = my_labels) +
  theme(legend.position = "right")
```

```
## Warning in geom_label(aes(y = value, label = round(value, 2), accuracy = 0.1), :
## Ignoring unknown aesthetics: accuracy
```



```r
# Create a vector for male and another for female
Cholesterol_Male <- data %>%
  filter(Sex == "Male") %>%
  select(Cholesterol) %>%
  unlist()

Cholesterol_Female <- data %>%
  filter(Sex == "Female") %>%
  select(Cholesterol) %>%
  unlist()

# Perform an independent two-sample t-test
t_test_result_2 <- t.test(x=Cholesterol_Male,
                          y=Cholesterol_Female,
                          alternative="two.sided",
                          var.equal = TRUE,
                          paired = FALSE)

# Print the t-test result
print(t_test_result_2)
```

```
##
##  Two Sample t-test
##
## data:  Cholesterol_Male and Cholesterol_Female
## t = 0.71266, df = 8761, p-value = 0.4761
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.345914  5.026082
## sample estimates:
## mean of x mean of y
##  260.2828  258.9427
```

If we formulate a new hypothesis test with $H_0 : \mu_{male} = \mu_{female}$ and an alternative hypothesis $H_1 : \mu_{male} \neq \mu_{female}$ we can see if this difference is signficative. The p-value of this test is $= 0.4761$ which is $> \alpha$ so we have enough evidence to not reject H_0 and conclude that the means calculated before aren't signficative.

Let's see what happens with some other variables from the dataset.

```r
# Find the totals
total_smokers <- sum(data$Smoking)
total_diabetes <- sum(data$Diabetes)
total_familyhistory <- sum(data$Family_History)
total_obesity <- sum(data$Obesity)
total_alcohol <- sum(data$Alcohol_Consumption)
total_previousheartrisk <- sum(data$Heart_Attack_Risk)
total_medicate <- sum(data$Medication_Use)

# Separate female and male
female_smokers <- sum(data$Smoking[data$Sex == "Female"])
female_diabetes <- sum(data$Diabetes[data$Sex == "Female"])
female_familyhistory <- sum(data$Family_History[data$Sex == "Female"])
female_obesity <- sum(data$Obesity[data$Sex == "Female"])
female_alcohol <- sum(data$Alcohol_Consumption[data$Sex == "Female"])
female_previousheartrisk <- sum(data$Heart_Attack_Risk[data$Sex == "Female"])
female_medicate <- sum(data$Medication_Use[data$Sex == "Female"])

male_smokers <- sum(data$Smoking[data$Sex == "Male"])
male_diabetes <- sum(data$Diabetes[data$Sex == "Male"])
male_familyhistory <- sum(data$Family_History[data$Sex == "Male"])
male_obesity <- sum(data$Obesity[data$Sex == "Male"])
male_alcohol <- sum(data$Alcohol_Consumption[data$Sex == "Male"])
male_previousheartrisk <- sum(data$Heart_Attack_Risk[data$Sex == "Male"])
male_medicate<- sum(data$Medication_Use[data$Sex == "Male"])

total_female <- sum(data$Sex == "Female")
total_male <- sum(data$Sex == "Male")

# Create a data frame to display the percentages
percentages <- data.frame(
  Gender = c("Female", "Male"),
  Smoker_Percentage = c(round(female_smokers / total_female * 100, 1),
                        round(male_smokers / total_male * 100, 1)),
  Diabetes_Percentage = c(round(female_diabetes / total_female * 100, 1),
                          round(male_diabetes / total_male * 100, 1)),
  Familyhistory_Percentage = c(round(female_familyhistory / total_female * 100, 1),
                               round(male_familyhistory / total_male * 100, 1)),
  Obesity_Percentage = c(round(female_obesity / total_female * 100, 1),
                         round(male_obesity / total_male * 100, 1)),
  Alcohol_Percentage = c(round(female_alcohol / total_female * 100, 1),
                         round(male_alcohol / total_male * 100, 1)),
  Previousheartrisk_Percentage =
    c(round(female_previousheartrisk / total_female * 100, 1),

                                   round(male_previousheartrisk / total_male * 100, 1)),
  Medicate_Percentage = c(round(female_medicate / total_female * 100, 1),
                          round(male_medicate / total_male * 100, 1)))

# Print the table
kable(percentages,
      align = "c",
      col.names = c("Gender", "Smoker (%)",
                    "Diabetes (%)", "Family History (%)",
                    "Obesity (%)", "Alcohol (%)",
                    "Previous Heart Risk (%)", "Medicated (%)"))
```

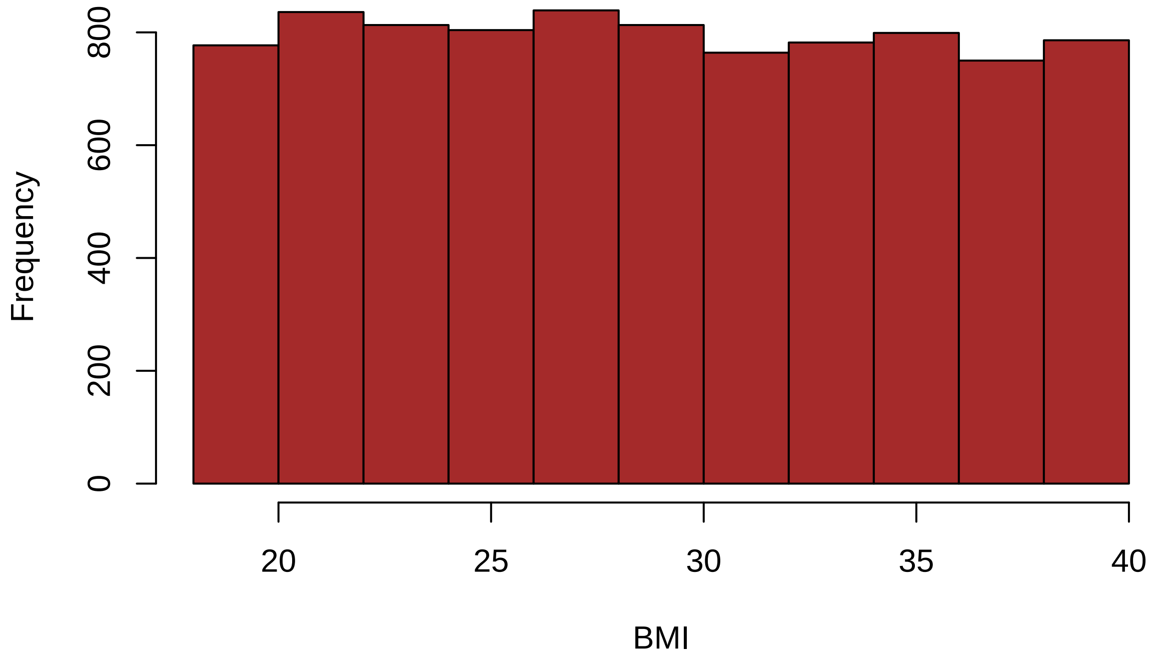
Gender	Smoker (%)	Diabetes (%)	Family History (%)	Obesity (%)	Alcohol (%)	Previous Heart Risk (%)	Medicated (%)
Female	65.9	65.0	49.1	50.0	59.7	35.6	50.4
Male	100.0	65.3	49.4	50.2	59.9	35.9	49.6

With the previous table, we can compare the quantity of female and male which smokes, have diabetes, have family history of heart attack, have obesity, consume alcohol, had previous heart risk or are medicated. In general, the male have worst health condition than the female, they have more proportion of smokers, diabetes and obesity disease, family history with heart conditions, consumption of alcohol, and previous heart risk. Additionally, the female takes more medication than the male.

(iv) For the continuous variables: make histograms, density plots, normal probability plots (QQ), box plots and other ones as you may consider. Discuss the normality of data based on graphs.

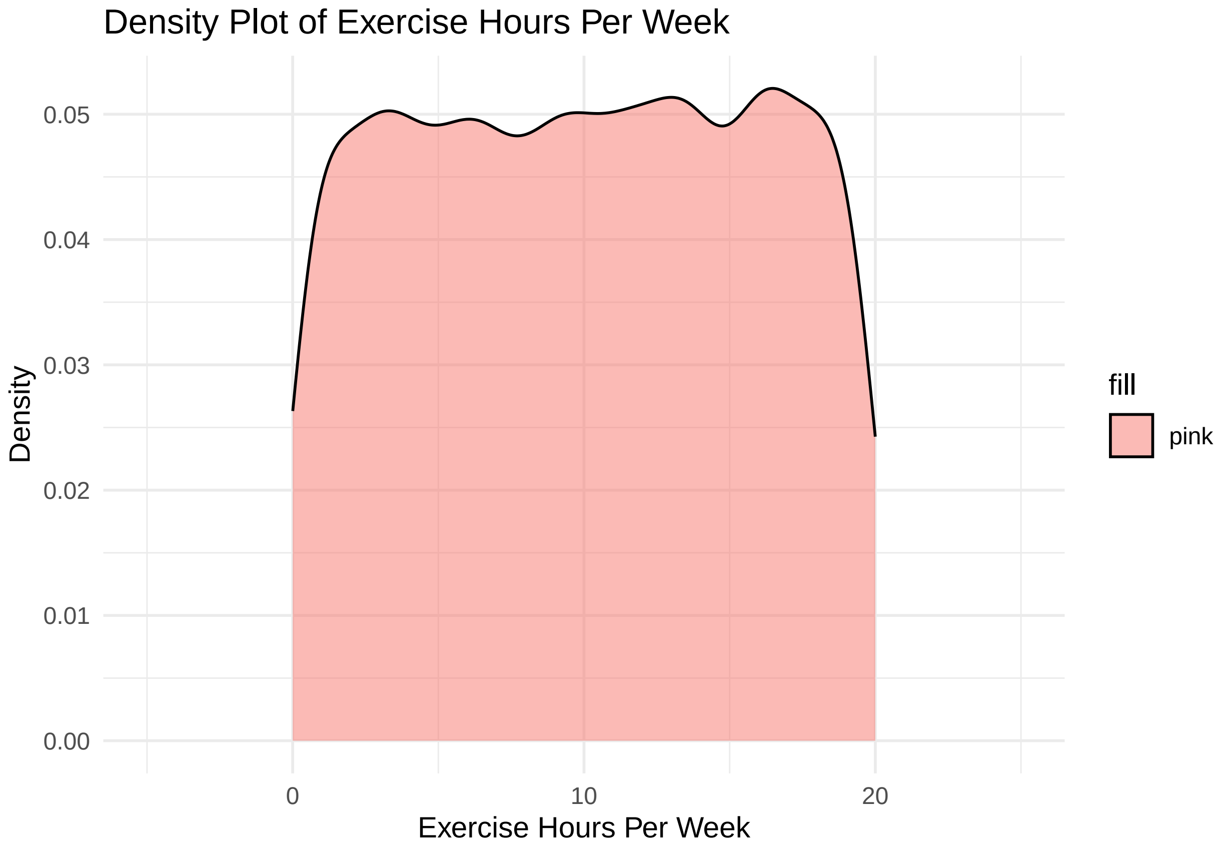
- **Histogram**

```
# Histogram
hist(data$BMI,col="brown",main="",xlab="BMI")
```



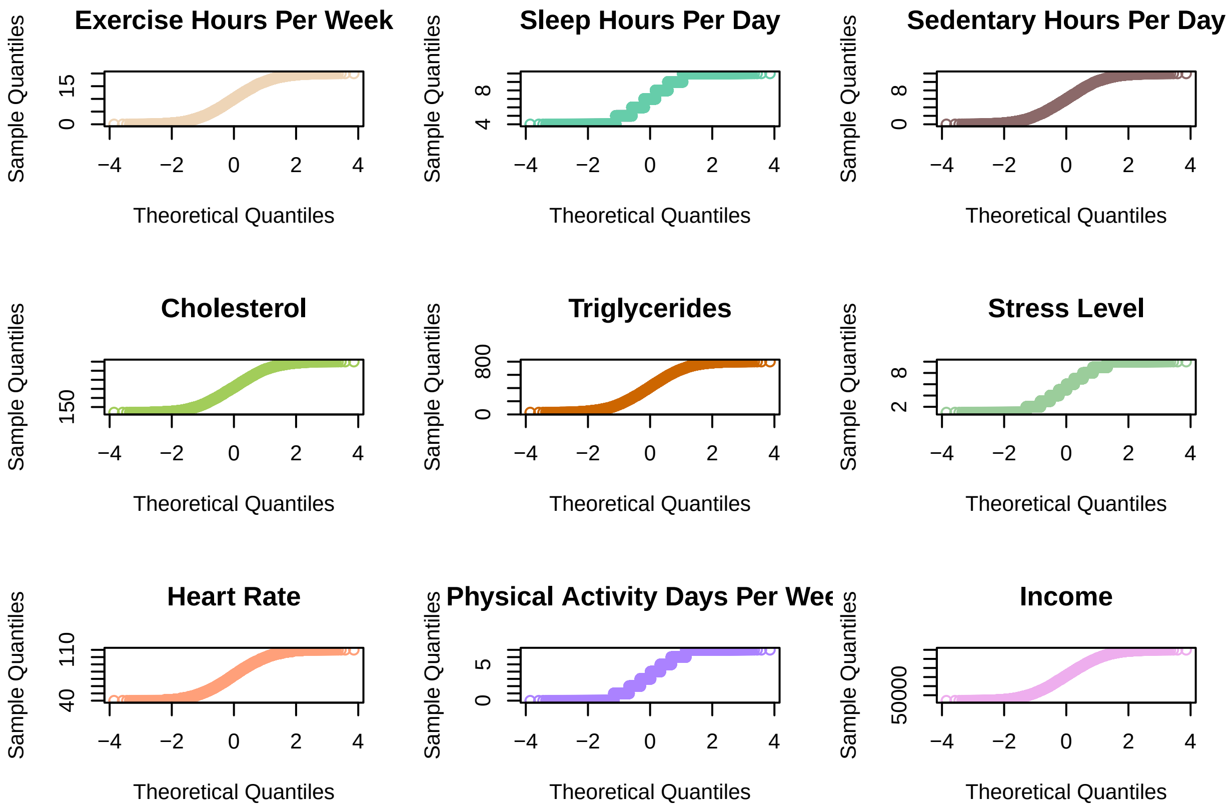
- **Density Plot**

```
#Density Plot
ggplot(data, aes(x = Exercise_Hours_Per_Week)) +
  geom_density(aes(fill = "pink"), alpha = 0.5) +
  labs(title = "Density Plot of Exercise Hours Per Week",
       x = "Exercise Hours Per Week",
       y = "Density") +
  theme_minimal() +
coord_cartesian(xlim = c(-5, 25))
```



- **Normal probability plots (QQ)**

```
#QQ plots
par(mfrow=c(3, 3))  # Split the graphic area in 1 row and 3 columns
qqnorm(data$Exercise_Hours_Per_Week, main="Exercise Hours Per Week", col = "#EED5B7")
qqnorm(data$Sleep_Hours_Per_Day, main="Sleep Hours Per Day", col = "#66CDAA")
qqnorm(data$Sedentary_Hours_Per_Day, main="Sedentary Hours Per Day", col = "#8B6969")
qqnorm(data$Cholesterol, main="Cholesterol", col = "#A2CD5A")
qqnorm(data$Triglycerides, main="Triglycerides", col = "#CD6600")
qqnorm(data$Stress_Level, main="Stress Level", col = "#9BCD9B")
qqnorm(data$Heart_Rate, main="Heart Rate", col = "#FFA07A")
qqnorm(data$Physical_Activity_Days_Per_Week,
       main="Physical Activity Days Per Week",
       col = "#AB82FF")
qqnorm(data$Income, main="Income", col = "#EEAEEE")
```

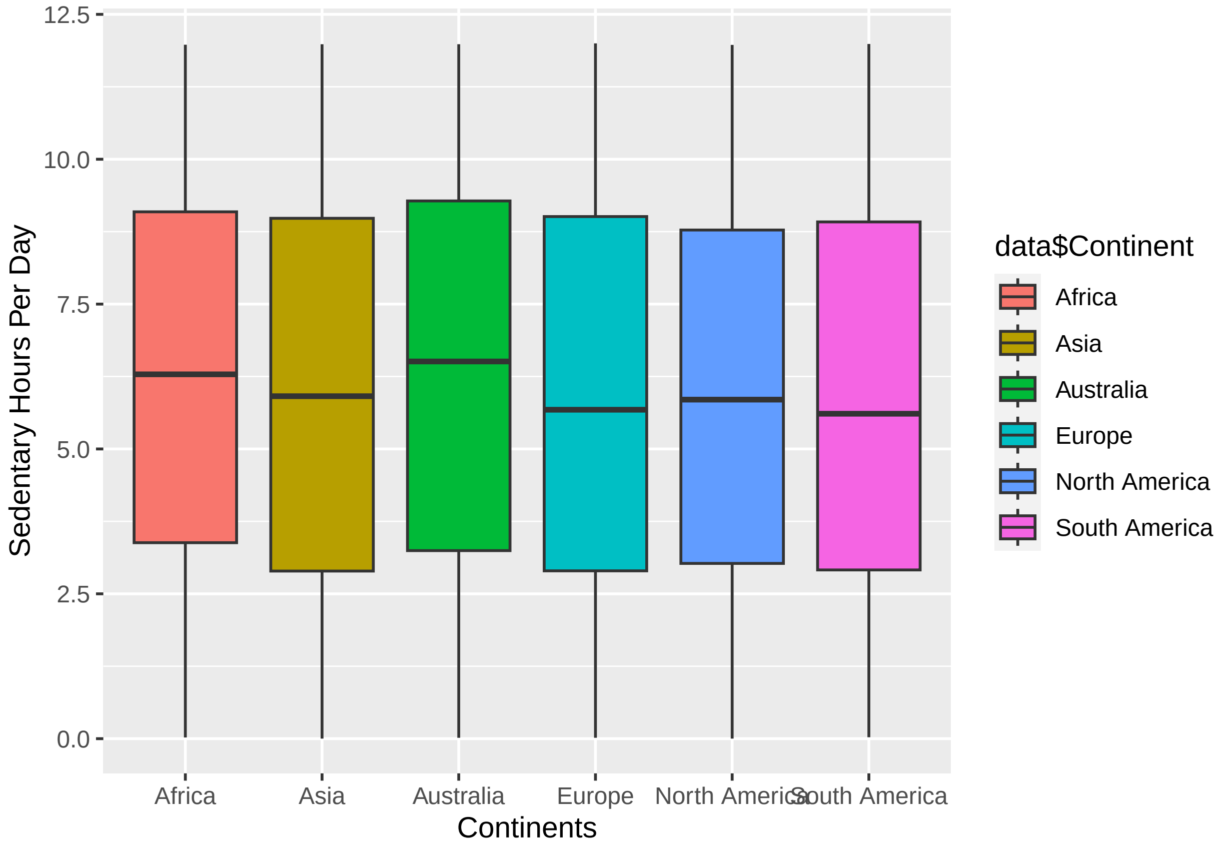


```
par(mfrow=c(3, 3))
```

Looking at the graphs, we can conclude that Sleep Hours Per Day, Stress Level and Physical Activity Days Per Week are discrete variables. Meanwhile, Exercise Hours Per Week, Sedentary Hours Per Day, Cholesterol, Triglycerides, Heart Rate and Income are continuous variables.

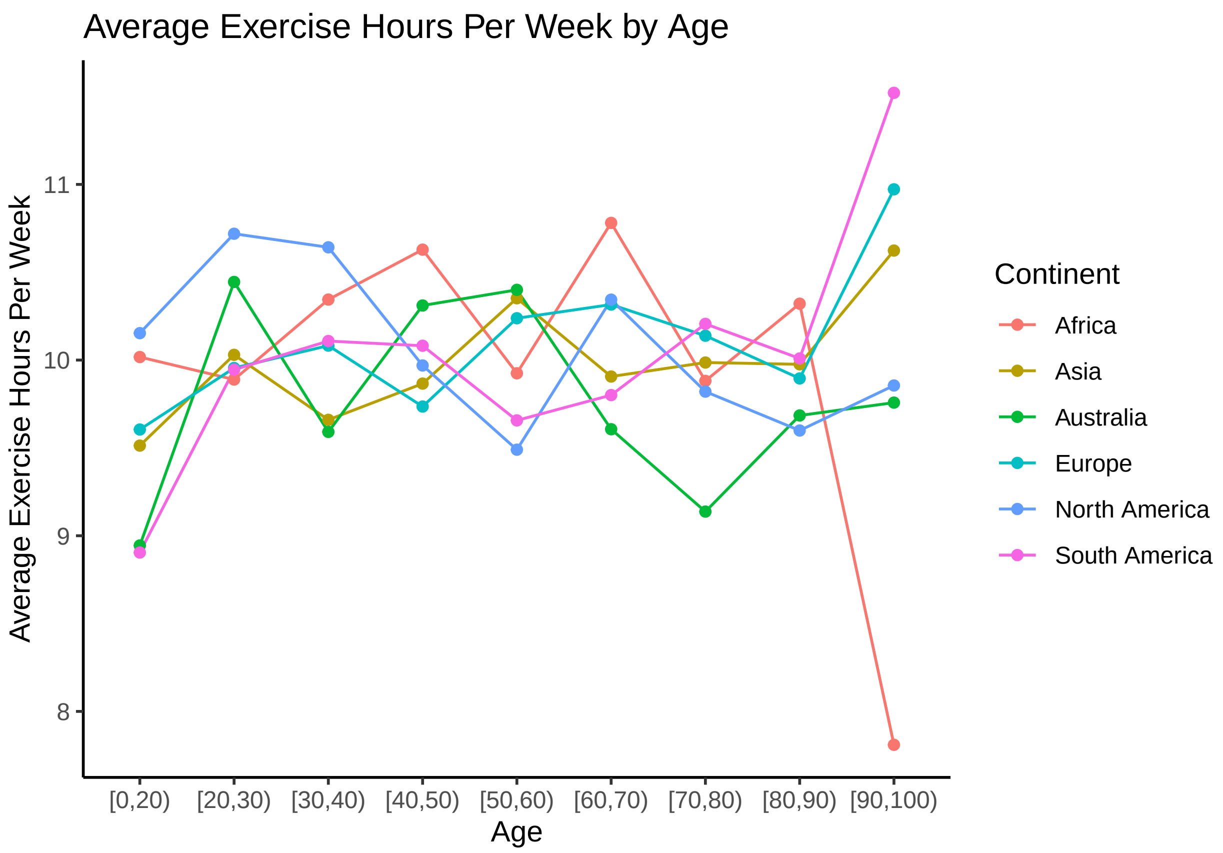
- **Box plots**

```
#Box plots
qplot(data$Continent,
      data$Sedentary_Hours_Per_Day,
      geom="boxplot",
      fill = data$Continent,
      xlab = "Continents",
      ylab = "Sedentary Hours Per Day")
```



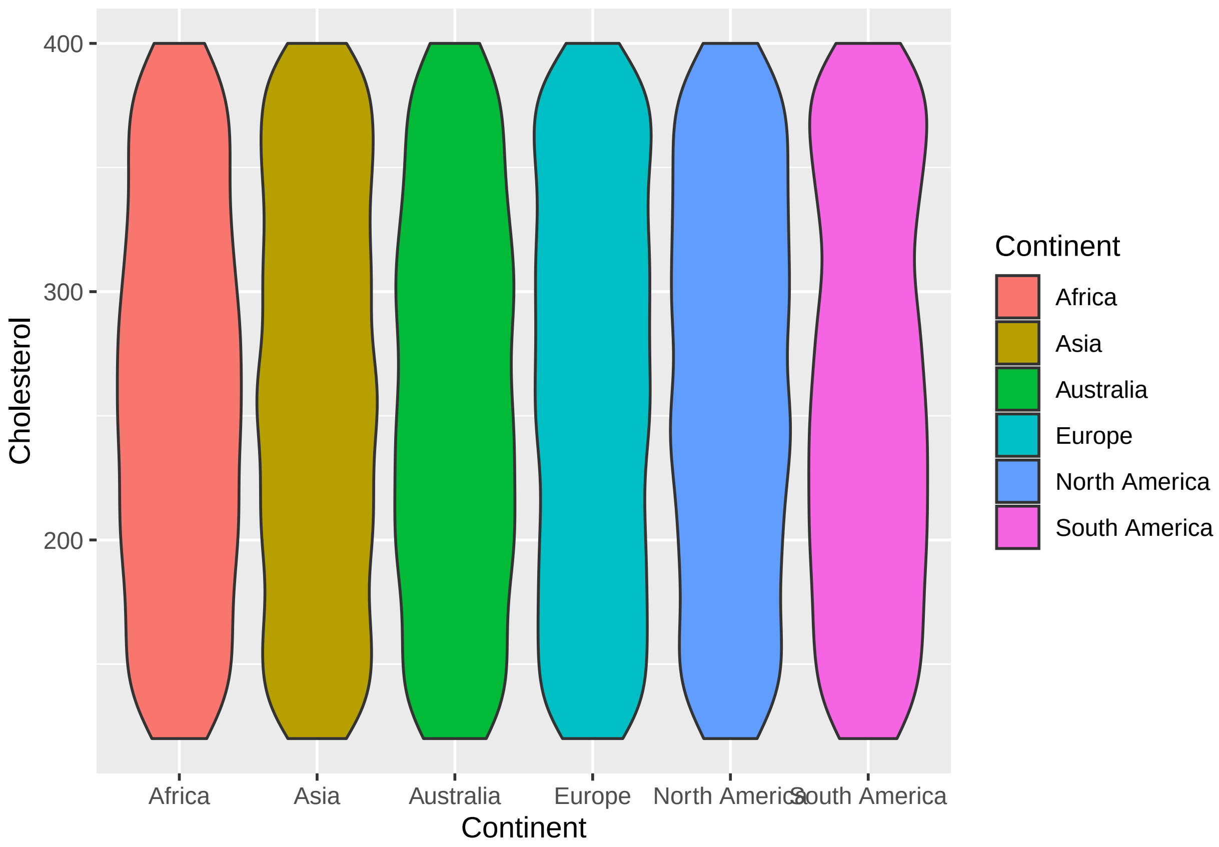
- **Joint Plot**

```
#Joint plot
data %>%
  group_by(Continent, interval1) %>%
  summarize(MeanExerciseHrsPerWeek = mean(Exercise_Hours_Per_Week)) %>%
  ggplot(aes(x = interval1,
             y = MeanExerciseHrsPerWeek,
             color = Continent,
             group = Continent)) +
  geom_line() +
  geom_point() +
  labs(title = "Average Exercise Hours Per Week by Age",
       x = "Age",
       y = "Average Exercise Hours Per Week") +
  theme_classic()
```



- **Violin Plot**

```
#Violin plot
qplot(Continent,
      Cholesterol,
      data = data,
      geom="violin",
      fill = Continent) +coord_cartesian(ylim = c(120,400))
```

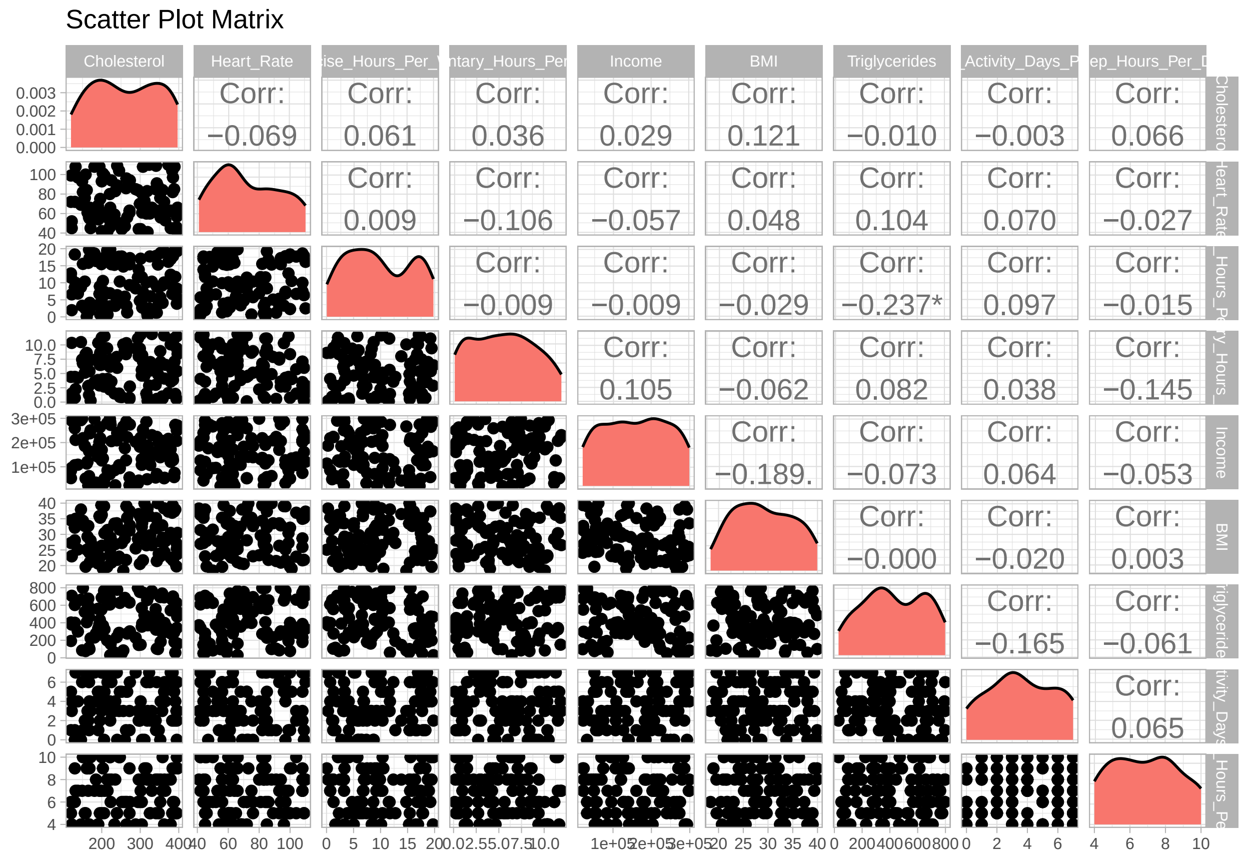


```r
#ylim is the range of variable cholesterol
```

- **Study normality based on graphs**

We made it with the package GGally.

```r
# Reduced data, random sample
data_red<-data[sample(nrow(data),100),]
# We draw the scatter plot of the continuous variables to check normality
library(GGally)
ggpairs(data_red[,c(4,6,12,17,18,19,20,21,22)],aes(fill="pink"),
        lower = list(continuous = "points"),
        title = "Scatter Plot Matrix") +
  theme_light(base_size=8)
```

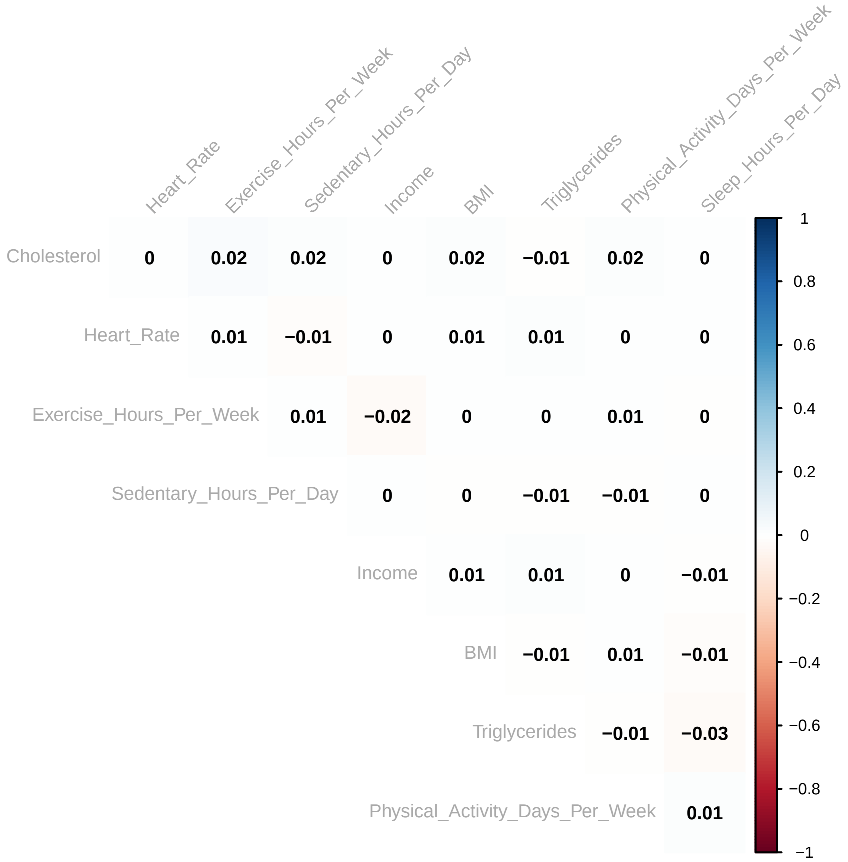


Every variable seems normal. If any variable wasn't normal (present assymetry), we should apply logarithms.

- **Correlation matrix**

With the package `library(corrplot)`.

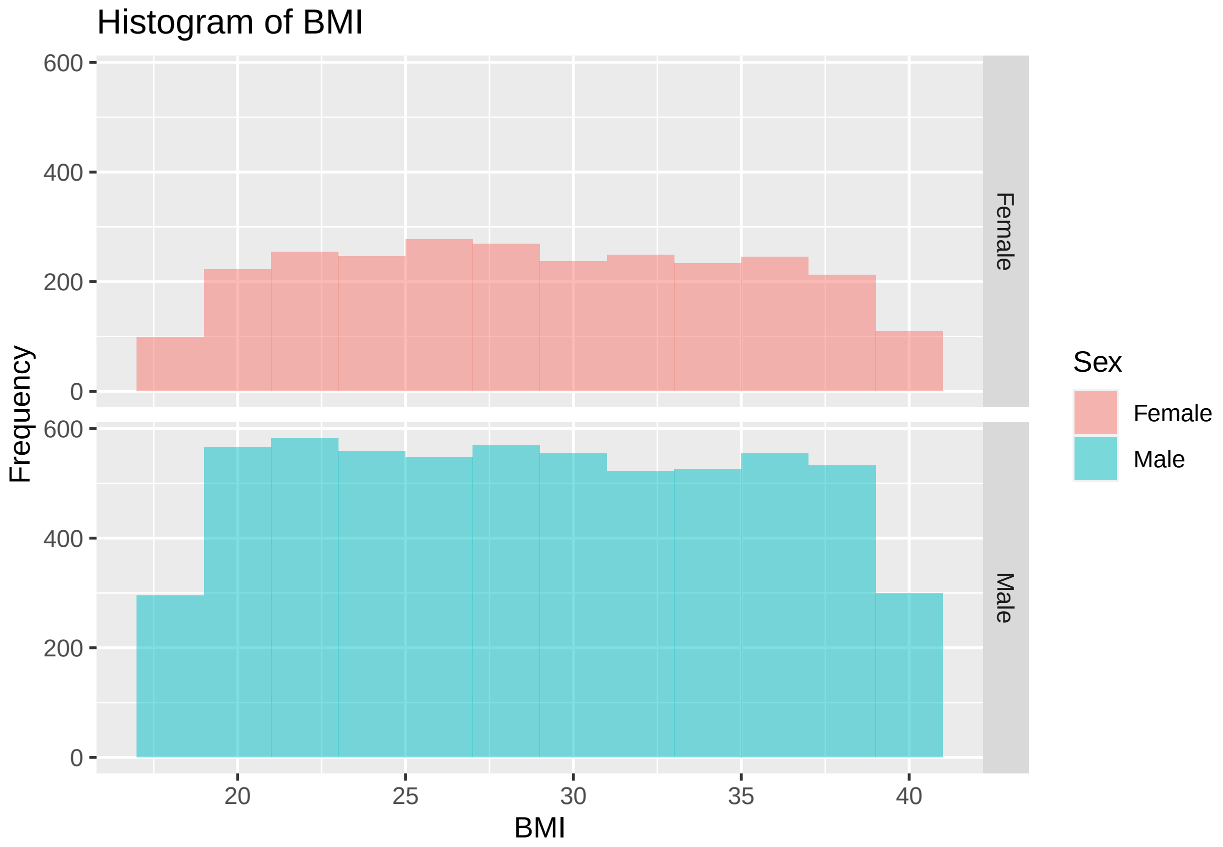
```
# Compute the correlation matrix
correlation_matrix=cor(data[,c(4,6,12,17,18,19,20,21,22)])
# We represent it
par(cex = 0.6) #adjust the size of the graph before you create it
corrplot(correlation_matrix,
         type = "upper",
         tl.col = "darkgrey",
         tl.srt = 45,
         method = "color",
         addCoef.col = "black",
         addCoefasPercent = FALSE, diag = FALSE)
```



(v) Then, repeat the previous plots for each group studied in (iii) and compare the results among them.

- **Histograms by Sex**

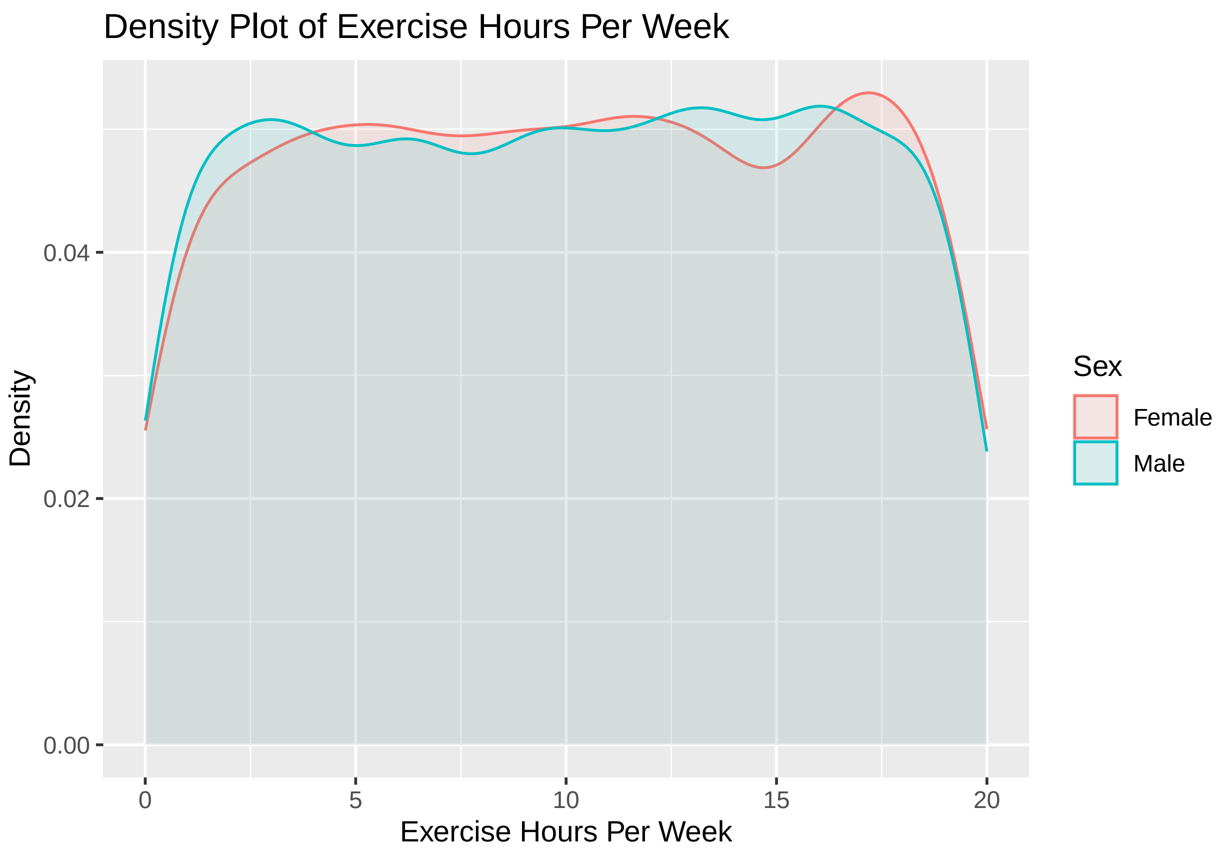
```r
# Histograms
ggplot(data, aes(x = BMI)) +
  geom_histogram(binwidth = 2, aes(fill = Sex), alpha = 0.5) +
  labs(title = "Histogram of BMI",
       x = "BMI",
       y = "Frequency") +
  facet_grid(Sex ~ .)
```



The body mass index (BMI) values are generally higher for males than for females. The major BMI for females is below 300, whereas for males, it typically exceeds 500

- **Density Plots by Sex**

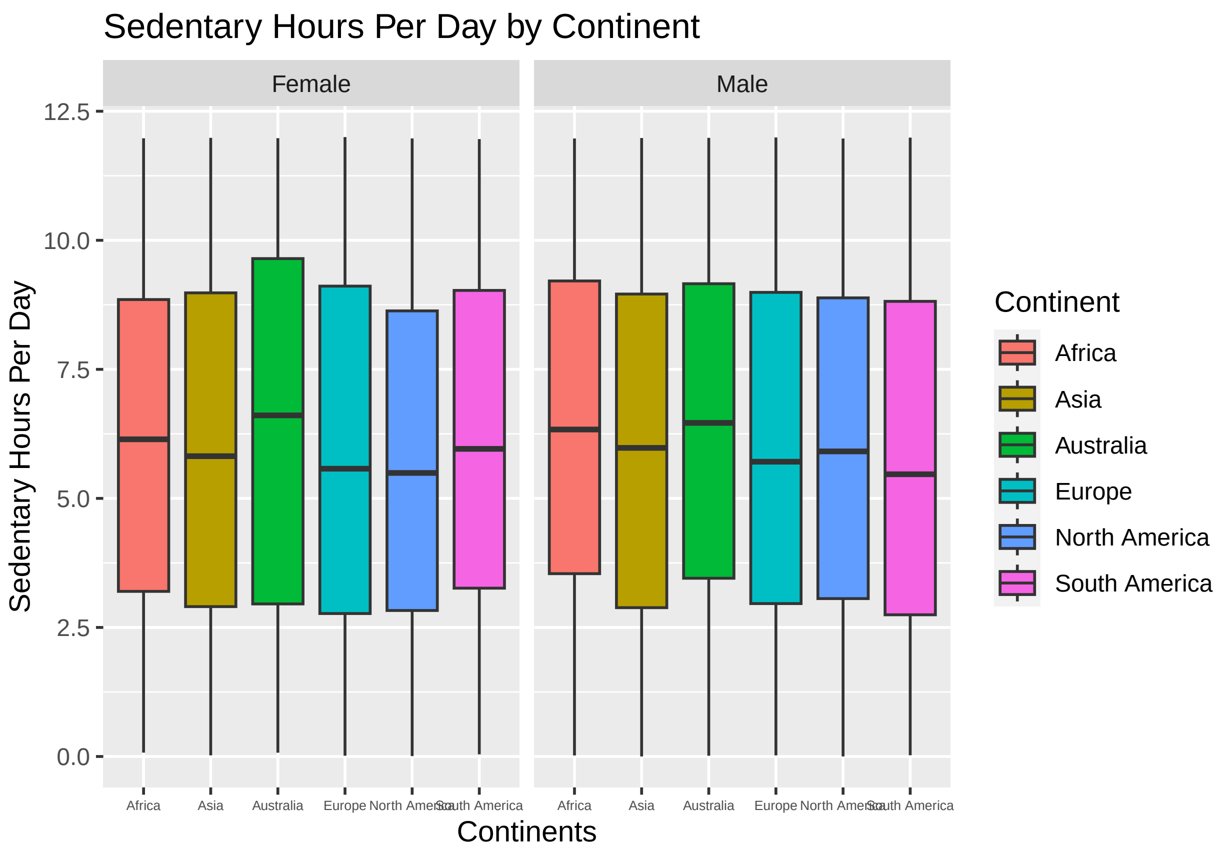
```
#Density Plots
ggplot(data, aes(x = Exercise_Hours_Per_Week)) +
  geom_density(aes(group = Sex, colour = Sex, fill = Sex), alpha = 0.1) +
  labs(title = "Density Plot of Exercise Hours Per Week",
       x = "Exercise Hours Per Week",
       y = "Density")
```



Observing the graph above, we can conclude that both density are similar to a normal distribution, and there aren't significative difference between female and male.

- **Box plots by Sex**

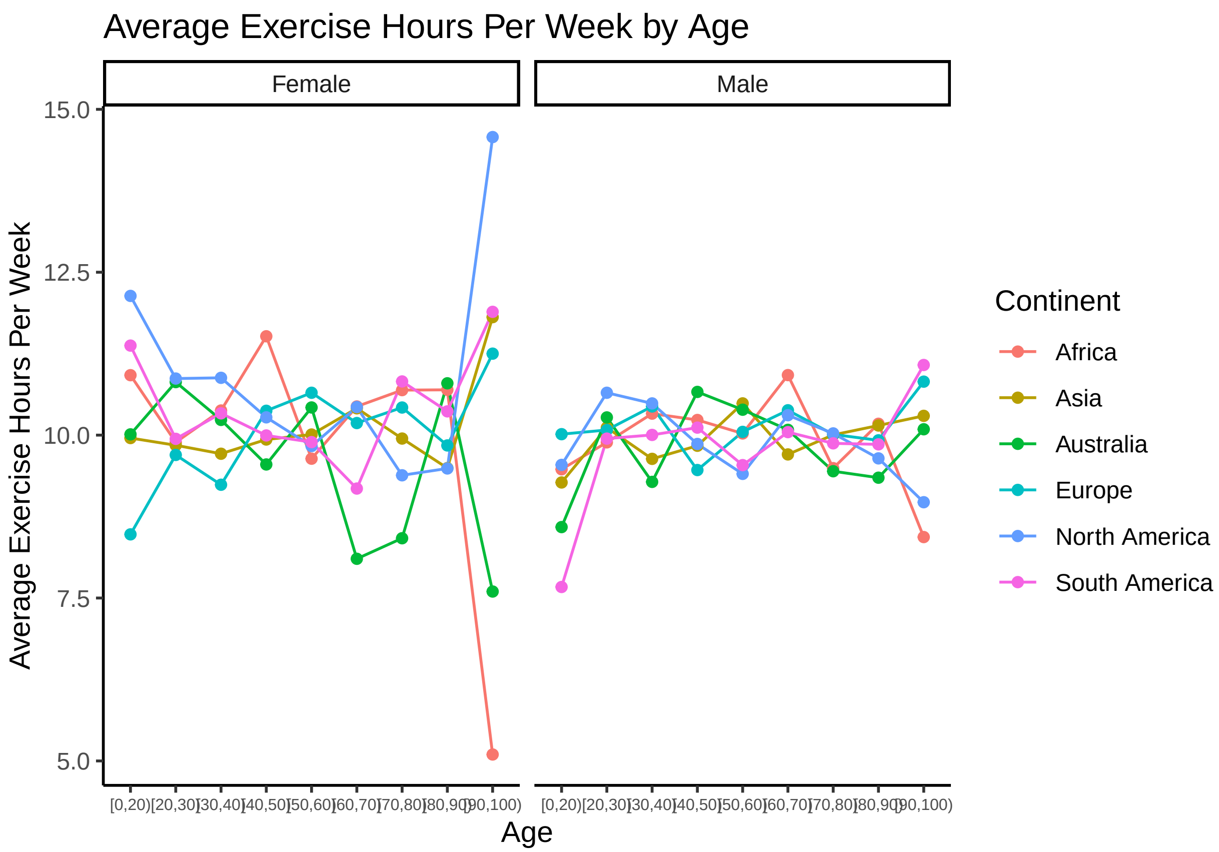
```
#Box plots
Box_plot2 <- ggplot(rbind(data[data$Sex == "Female", ], data[data$Sex == "Male", ]),
                    aes(x = Continent,
                        y = Sedentary_Hours_Per_Day,
                        fill = Continent)) +
  geom_boxplot() +
  labs(title = "Sedentary Hours Per Day by Continent",
       x = "Continents",
       y = "Sedentary Hours Per Day") +
  facet_wrap(~Sex)

Box_plot2 <- Box_plot2 + theme(axis.text.x = element_text(size = 5))
print(Box_plot2)
```



In the next conclusion we will focus on the difference between female and male in each continent. In Africa, Europe and North America the mean of sedentary hours per day is slightly greater for male than for female. Meanwhile, in Australia and South America the mean of sedentary hours per day is minor for male than for female. Visually, there doesn't appear to be a significant difference between females and males in Asia.

- **Joint Plot by Sex**

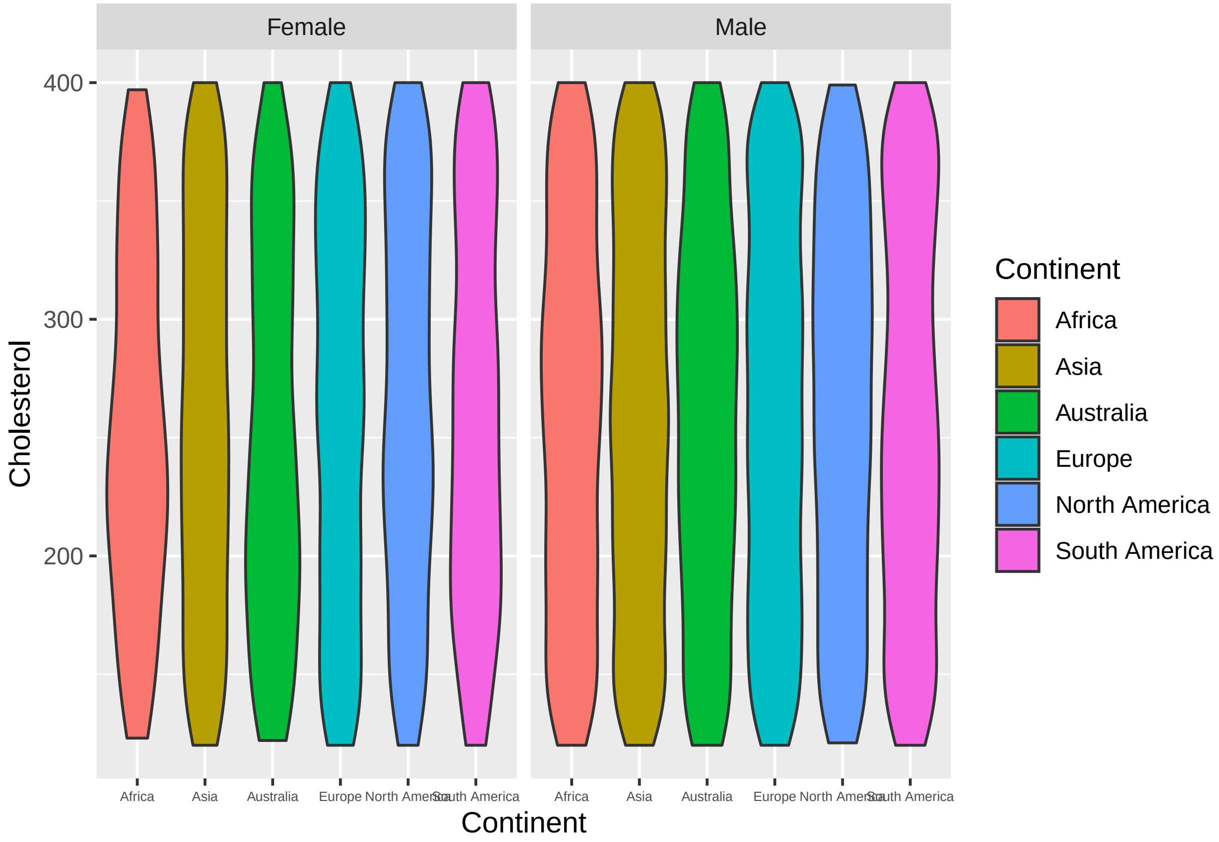
```
#Joint plot
joint_data <- data %>%
  group_by(Sex, Continent, interval1) %>%
  summarize(MeanExerciseHrsPerWeek = mean(Exercise_Hours_Per_Week))

joint_plot <- ggplot(joint_data, aes(x = interval1,
                                     y = MeanExerciseHrsPerWeek,
                                     color = Continent,
                                     group = Continent)) +
  geom_line() +
  geom_point() +
  labs(title = "Average Exercise Hours Per Week by Age",
       x = "Age",
       y = "Average Exercise Hours Per Week") +
  facet_wrap(~Sex) +
  theme_classic()
joint_plot <- joint_plot + theme(axis.text.x = element_text(size = 6))
print(joint_plot)
```



The initial observation from the graph above is that there is more dispersion between continents within each age interval for females compared to males. Additionally, there is more dispersion among age intervals for females than for males.

- **Violin Plot by Sex**

```
#Violin plots
violin_plots <- qplot(Continent, Cholesterol,
                      data = data,
                      geom = "violin",
                      fill = Continent) +
  coord_cartesian(ylim = c(120, 400)) +
  facet_wrap(~Sex)
violin_plots <- violin_plots + theme(axis.text.x = element_text(size = 5))
print(violin_plots)
```



This graph is not so precise so we can conclude that there is not significative difference for the variable cholesterol between female and male in each continent.

(vi) Take a categorical variable and show the frequency table. Take two categorical variables and show the descriptive contingency table. Make mosaic plots and explain the results.

We'll take Diet as a categorical variable

```
freq_table5 <- data %>%
  group_by(Diet) %>%
  summarise(Frequency = n())
kable(freq_table5, align = c("l", "c"), col.names = c("Diet", "Frequency"))
```

Diet	Frequency
Average	2912
Healthy	2960
Unhealthy	2891

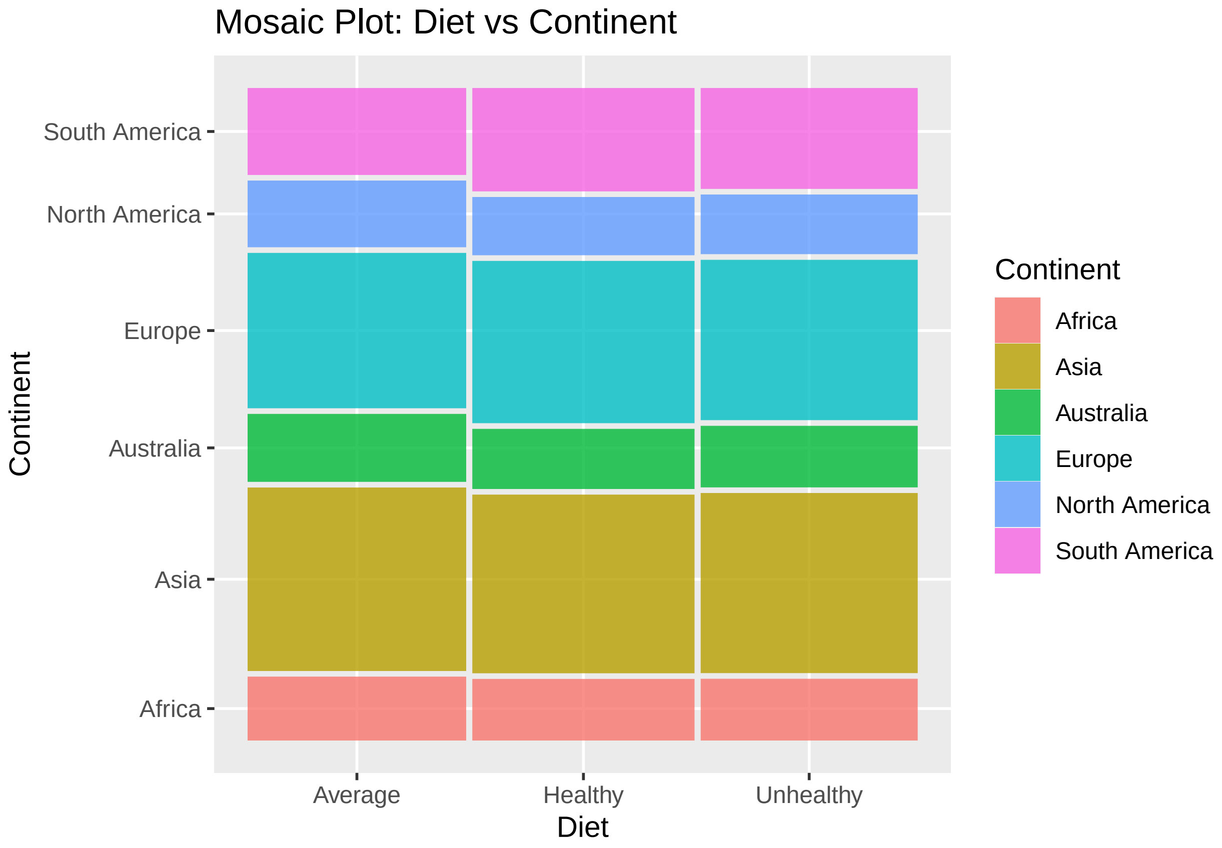
Now we are going to show the **contingency table** between Diet and Continent.

```
contingency_table <- xtabs(~ Continent + Diet, data)
contingency_table <- addmargins(contingency_table)
rownames(contingency_table)[nrow(contingency_table)] <- "Total"
colnames(contingency_table)[ncol(contingency_table)] <- "Total"
kable(contingency_table, align = "c")
```

	Average	Healthy	Unhealthy	Total
Africa	297	291	285	873
Asia	858	849	836	2543
Australia	316	284	284	884
Europe	726	771	744	2241
North America	310	275	275	860
South America	405	490	467	1362
Total	2912	2960	2891	8763

- **Mosaic plot**

```
#Mosaic Plot for Diet and Continent
ggplot(data = data) +
  geom_mosaic(aes(x = product(Continent, Diet), fill=Continent)) +
  labs(title="Mosaic Plot: Diet vs Continent")
```



Note that Asia and Europe are the continent with more observations. Furthermore, we can comment the diet inside each continent: Inside South America and Europe the major part of the observation follows a healthy diet. In North America, Australia, Africa and Asia the majority follows an average diet.

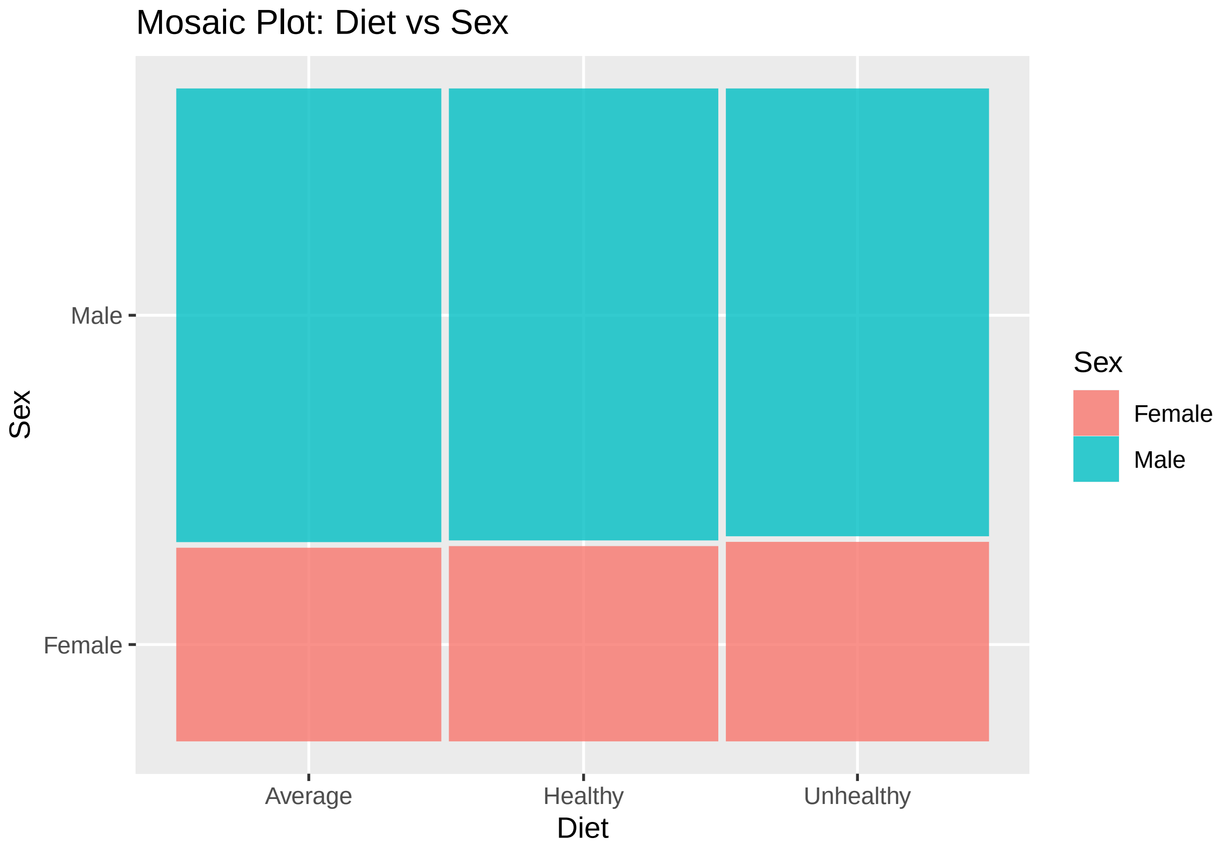
If we want to see the **contingency table** of Diet and Sex:

```
contingency_table2 <- xtabs(~ Sex + Diet, data)
contingency_table2 <- addmargins(contingency_table2)
rownames(contingency_table2)[nrow(contingency_table2)] <- "Total"
colnames(contingency_table2)[ncol(contingency_table2)] <- "Total"
kable(contingency_table2, align = "c")
```

	Average	Healthy	Unhealthy	Total
Female	870	892	890	2652
Male	2042	2068	2001	6111
Total	2912	2960	2891	8763

- **Mosaic plot**

```
#Mosaic Plot for Diet and Sex
ggplot(data = data) +
  geom_mosaic(aes(x = product(Sex, Diet), fill=Sex)) +
  labs(title="Mosaic Plot: Diet vs Sex")
```



Observing the graph we can deduce that there is more males than females. If we observe the diet for each sex we deduce that is more female following an unhealthy diet than average or healthy diet. Additionally, For the male the diet distribution is the opposite, there are more males following an average diet than healthy or unhealthy.

In conclusion, it is evident that more people are inclined to follow an average or healthy diet rather than an unhealthy one.

Part 2: Apply caret library.

The second part of the project consists on using the **Caret** library, which it is a package used for creating predictive models.

Selecting a dichotomic target variable

Firstly, it is necessary to load the library we are going to deal with and some other libraries I consider fundamental to perform the exercise.

```
library(caret)
library(fastDummies)
library(randomForest)
```

As a reminder, the structure of the dataset we are working with is shown below.

```
df <- read_excel("heart_attack_prediction_dataset.xlsx")
str(df)
```

```
## tibble [8,763 x 26] (S3: tbl_df/tbl/data.frame)
##  $ Patient_ID                     : chr [1:8763] "BMW7812" "CZE1114" "BNI9906" "JLN3497" ...
##  $ Age                            : num [1:8763] 67 21 21 84 66 54 90 84 20 43 ...
##  $ Sex                            : chr [1:8763] "Male" "Male" "Female" "Male" ...
##  $ Cholesterol                    : num [1:8763] 208 389 324 383 318 297 358 220 145 248 ...
##  $ Blood_Pressure                 : chr [1:8763] "158/88" "165/93" "174/99" "163/100" ...
##  $ Heart_Rate                     : num [1:8763] 72 98 72 73 93 48 84 107 68 55 ...
##  $ Diabetes                       : num [1:8763] 0 1 1 1 1 1 0 0 1 0 ...
##  $ Family_History                 : num [1:8763] 0 1 0 1 1 1 0 0 0 1 ...
##  $ Smoking                        : num [1:8763] 1 1 0 1 1 1 1 1 1 1 ...
##  $ Obesity                        : num [1:8763] 0 1 0 0 1 0 0 1 1 1 ...
##  $ Alcohol_Consumption            : num [1:8763] 0 1 0 1 0 1 1 1 0 1 ...
##  $ Exercise_Hours_Per_Week        : num [1:8763] 4.17 1.81 2.08 9.83 5.8 ...
##  $ Diet                           : chr [1:8763] "Average" "Unhealthy" "Healthy" "Average" ...
##  $ Previous_Heart_Problems        : num [1:8763] 0 1 1 1 1 1 0 0 0 0 ...
##  $ Medication_Use                 : num [1:8763] 0 0 1 0 0 1 0 1 0 0 ...
##  $ Stress_Level                   : num [1:8763] 9 1 9 9 6 2 7 4 5 4 ...
##  $ Sedentary_Hours_Per_Day        : num [1:8763] 6.62 4.96 9.46 7.65 1.51 ...
##  $ Income                         : num [1:8763] 261404 285768 235282 125640 160555 ...
##  $ BMI                            : num [1:8763] 31.3 27.2 28.2 36.5 21.8 ...
##  $ Triglycerides                  : num [1:8763] 286 235 587 378 231 795 284 370 790 232 ...
##  $ Physical_Activity_Days_Per_Week: num [1:8763] 0 1 4 3 1 5 4 6 7 7 ...
##  $ Sleep_Hours_Per_Day            : num [1:8763] 6 7 4 4 5 10 10 7 4 7 ...
##  $ Country                        : chr [1:8763] "Argentina" "Canada" "France" "Canada" ...
##  $ Continent                      : chr [1:8763] "South America" "North America" "Europe" "North Ame
##  $ Hemisphere                     : chr [1:8763] "Southern Hemisphere" "Northern Hemisphere" "Northe
##  $ Heart_Attack_Risk              : num [1:8763] 0 0 0 0 0 1 1 1 0 0 ...
```

With this information, we have concluded that a good categorical and dichotomic variable is the **Heart.Attack.Risk**. So this is the one parameter that we are going to predict and analyze the relationships with other variables.

Data preprocessing

In this section, we must prepare the dataset in order to use it for the machine learning model. This means handling missing data, deleting unnecessary information, encoding categorical variables and splitting the data.

1. Handling missing data.

From the begging of the project, we know that we do not have any missing value, so this is not a problem to us and we can skip this section.

2. Deleting unnecessary information.

The structure of the dataframe presented above indicates that some features (variables) can be omitted for this specific situation. The main reason is that they do not provide any relevant information. Therefore, the following parameters will be removed: Patient_ID, Income, Country, Hemisphere, Blood_Pressure, Family_History, Medication_Use and Triglycerides.

```
df<-df[,-c(1,18,23,25,5,8,15,20)]
```

3. Encoding categorical variables.

In this step, we address the encoding of categorical variables. We currently have 18 variables in the dataframe, and our goal is to convert the categorical ones (‘Sex’, ‘Diet’ and ‘Country’) into binary variables. To do so, we will apply the *One Hot Encoding* technique. This technique transforms categorical data into binary vectors, making them suitable for machine learning algorithms.

```
df <- dummy_cols(df,select_columns = c('Sex','Diet','Continent'))
df <- df[, !(names(df) %in% c('Sex','Diet', 'Continent'))]

df$Continent_North_America <- df$`Continent_North America`
df$Continent_South_America <- df$`Continent_South America`
df <- df[, !names(df) %in% c('Continent_North America', 'Continent_South America')]

df$Heart_Attack_Risk <- as.factor(df$Heart_Attack_Risk)
```

4. Standarization.

On our current dataframe, we encounter variables like *Age*, *Cholesterol*, ... with a diverse range of values. In this section, our aim is to convert all the numeric variables within a consistent range, between 0 and 1. This process enhances the performance of our machine learning model which will be more robust and efficient.

```
preProcess_range_model = preProcess(df, method='range')
df = predict(preProcess_range_model, newdata = df)
apply(df[], 2, FUN=function(x){c('min'=min(x), 'max'=max(x))})
```

```
##     Age          Cholesterol   Heart_Rate   Diabetes Smoking Obesity
## min "0.00000000" "0.000000000" "0.00000000" "0"      "0"     "0"    
## max "1.00000000" "1.000000000" "1.00000000" "1"      "1"     "1"    
##     Alcohol_Consumption Exercise_Hours_Per_Week Previous_Heart_Problems
## min "0"                 "0.0000000000"          "0"
## max "1"                 "1.0000000000"          "1"
##     Stress_Level Sedentary_Hours_Per_Day BMI
## min "0.0000000"  "0.0000000000"          "0.0000000000"
## max "1.0000000"  "1.0000000000"          "1.0000000000"
##     Physical_Activity_Days_Per_Week Sleep_Hours_Per_Day Heart_Attack_Risk
## min "0.0000000"                     "0.0000000"         "0"
## max "1.0000000"                     "1.0000000"         "1"
##     Sex_Female Sex_Male Diet_Average Diet_Healthy Diet_Unhealthy
## min "0"        "0"      "0"          "0"          "0"
## max "1"        "1"      "1"          "1"          "1"
##     Continent_Africa Continent_Asia Continent_Australia Continent_Europe
## min "0"              "0"            "0"                 "0"
## max "1"              "1"            "1"                 "1"
##     Continent_North_America Continent_South_America
## min "0"                     "0"
## max "1"                     "1"
```

5. Splitting the data.

In the final step of our data preprocessing, in order to do our machine learning model, we split the data into training (containing 80% of the dataset) and test (20% of the dataset). The training set allows the model to learn the essential insights and generalize from the data available. On the other hand, the test set allows us to prove model's accuracy and verify how well the model can make predictions on unseen data. In practice, the dataset is usually divided into three parts: train, test and validation. However since we are not working with a big amount of data we can omit the validation set.

```r
set.seed(123)

n_train <- floor(0.8*nrow(df))
n_test <- nrow(df)-n_train
Train <- sample(1:nrow(df),n_train)
Test <- (1:nrow(df))[-Train]
trainData <- df[Train,]
testData <- df[Test,]
```

After completely the data preprocessing, our dataset is now properly prepared for the application of our machine learning model.

Select predictors

Principal Component Analysis (PCA)

The purpose is to choose the variables that might be more important in order to predict if someone is going to suffer a heart attack.

However, in order to select predictors, we can reduce dimensions by computing another variables (components) that summarize the variability of the predictors. This is called Principal Component Analysis (PCA).

We can implement it in R with this two libraries: `library(FactoMineR)` and `library(factoextra)`. If our data was not already standardized, FactorMineR automatically standardizes the data.

```r
library(FactoMineR)
library(factoextra)
library(corrplot)
```

We remove the target variable (Heart_Attack_Risk) to apply PCA.

```r
df_PCA<-df[,-15]
```

```r
# PCA with FactorMineR
# This function, by default, only considers 5 dimensions
# (i.e. 5 PCs) in the results.
# That can be changed in the ncp argument
PCA<-PCA(df_PCA, graph = F)
PCA$var$cor
```

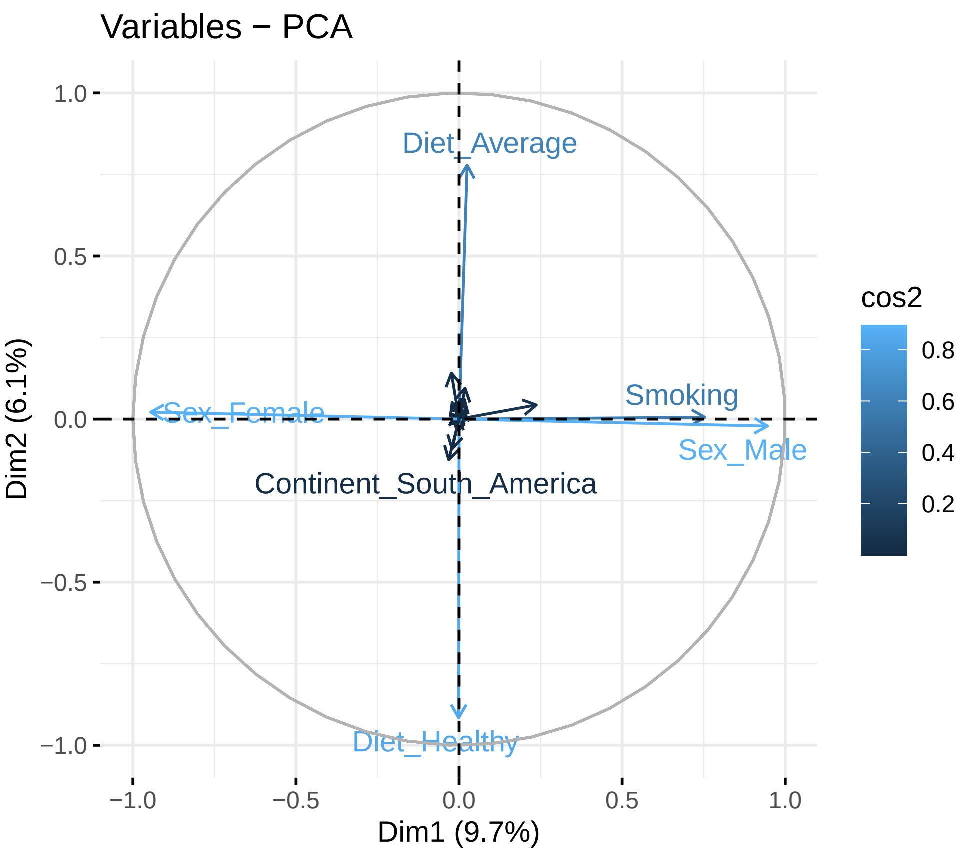
```
##                                         Dim.1        Dim.2        Dim.3
## Age                              0.2368779464  0.043435915 -0.029717937
## Cholesterol                      0.0171584919  0.021669449 -0.026433591
## Heart_Rate                      -0.0212312567  0.050918928  0.031408946
## Diabetes                         0.0026306175 -0.014911027  0.016547909
## Smoking                          0.7537880271  0.005536168 -0.013750074
## Obesity                          0.0034742353  0.010665692  0.026041591
## Alcohol_Consumption              0.0081093687 -0.017166316  0.008389587
## Exercise_Hours_Per_Week         -0.0089136076 -0.028025377  0.003984507
## Previous_Heart_Problems          0.0027325875 -0.029893310  0.052865753
## Stress_Level                    -0.0269111009 -0.013203516  0.032940030
## Sedentary_Hours_Per_Day          0.0168557832  0.016283938  0.012495355
## BMI                              0.0001826243 -0.024812865  0.025684284
## Physical_Activity_Days_Per_Week -0.0138652246  0.009906244 -0.042517588
## Sleep_Hours_Per_Day             -0.0098997093  0.013760841 -0.050136742
## Sex_Female                      -0.9455924213  0.021438555  0.012780334
## Sex_Male                         0.9455924213 -0.021438555 -0.012780334
## Diet_Average                     0.0247732544  0.778482809  0.604404132
## Diet_Healthy                    -0.0012419212 -0.915730636  0.377705582
## Diet_Unhealthy                  -0.0235693192  0.141227233 -0.985443955
## Continent_Africa                 0.0200587691  0.018821683  0.013579391
## Continent_Asia                   0.0182163974  0.094842476  0.024475097
## Continent_Australia              0.0163491660  0.060180127  0.034554423
## Continent_Europe                -0.0213330720 -0.091261021 -0.030874631
## Continent_North_America          0.0053437527  0.061276185  0.039451345
## Continent_South_America         -0.0316902319 -0.124813121 -0.065824562
##                                         Dim.4        Dim.5
## Age                             -0.0004375275  0.078957832
## Cholesterol                      0.0034954415 -0.031233567
## Heart_Rate                       0.0297107315  0.058086901
## Diabetes                         0.0039412398 -0.072083523
## Smoking                          0.0209335369  0.021056081
## Obesity                          0.0163735704  0.011295200
## Alcohol_Consumption             -0.0102076881 -0.014306788
## Exercise_Hours_Per_Week          0.0194264900 -0.005318485
## Previous_Heart_Problems          0.0615967391  0.066686094
## Stress_Level                    -0.0159275322  0.069848360
## Sedentary_Hours_Per_Day         -0.0400528307  0.065129820
## BMI                              0.0159007872  0.021156824
## Physical_Activity_Days_Per_Week  0.0151126390 -0.087459224
## Sleep_Hours_Per_Day             -0.0365238743  0.054919560
## Sex_Female                      -0.0073521485  0.015956747
## Sex_Male                         0.0073521485 -0.015956747
## Diet_Average                     0.0982329134  0.010866654
## Diet_Healthy                    -0.0858186681 -0.011153686
## Diet_Unhealthy                  -0.0120875458  0.000332968
## Continent_Africa                 0.0519142815  0.278854339
## Continent_Asia                  -0.8892327345 -0.423302292
## Continent_Australia              0.0655535552  0.284184583
## Continent_Europe                 0.7375074488 -0.641428450
## Continent_North_America          0.0690826248  0.239724057
## Continent_South_America          0.0717158148  0.639045714
```

$PC1$ explains very well `Smoking` and `Sex.Male` as they have positive coefficients near to 1.`Sex.Female` has a big importance but negatively to PC1.

Here we can see the importance of the variables to the first two PC's:

```
fviz_pca_var(PCA,col.var="cos2",repel=T)
```

```
## Warning: ggrepel: 19 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



We calculate the variability explained by the PC's.

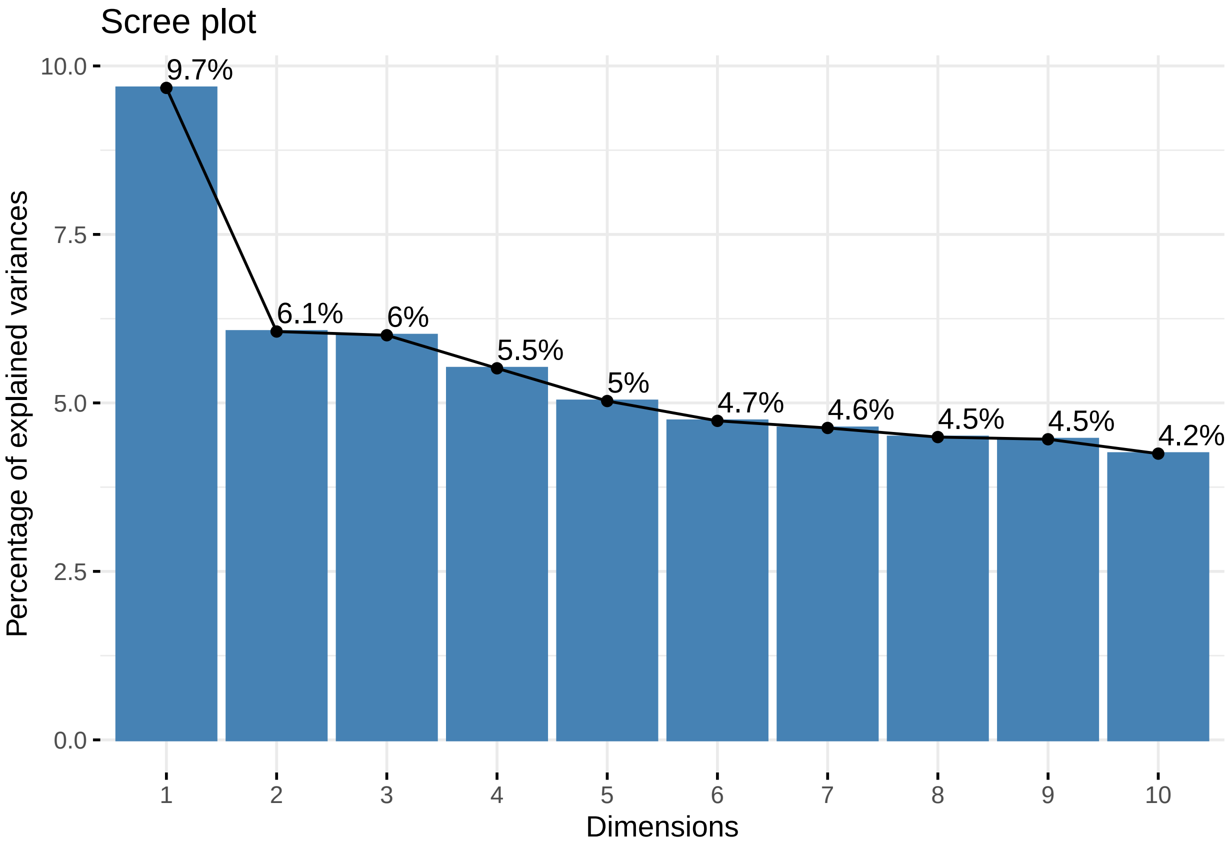
```
# Explained variance and cummulative explained variance
eig.val<-get_eigenvalue(PCA);eig.val
```

```
##           eigenvalue variance.percent cumulative.variance.percent
## Dim.1   2.418473e+00     9.673893e+00                    9.673893
## Dim.2   1.514727e+00     6.058909e+00                   15.732802
## Dim.3   1.501025e+00     6.004100e+00                   21.736902
## Dim.4   1.378395e+00     5.513579e+00                   27.250481
## Dim.5   1.256999e+00     5.027997e+00                   32.278478
## Dim.6   1.183416e+00     4.733666e+00                   37.012144
## Dim.7   1.157048e+00     4.628191e+00                   41.640335
## Dim.8   1.123228e+00     4.492911e+00                   46.133246
## Dim.9   1.115168e+00     4.460672e+00                   50.593918
## Dim.10 1.061780e+00     4.247122e+00                   54.841039
## Dim.11 1.032859e+00     4.131436e+00                   58.972476
## Dim.12 1.022153e+00     4.088611e+00                   63.061087
## Dim.13 1.012175e+00     4.048699e+00                   67.109785
## Dim.14 1.004021e+00     4.016083e+00                   71.125868
## Dim.15 1.000510e+00     4.002041e+00                   75.127909
## Dim.16 9.940498e-01     3.976199e+00                   79.104108
## Dim.17 9.828774e-01     3.931510e+00                   83.035618
## Dim.18 9.643347e-01     3.857339e+00                   86.892957
## Dim.19 9.628918e-01     3.851567e+00                   90.744524
## Dim.20 9.539723e-01     3.815889e+00                   94.560413
## Dim.21 9.460363e-01     3.784145e+00                       98.344558
## Dim.22 4.138605e-01     1.655442e+00                      100.000000
## Dim.23 1.746409e-27     6.985635e-27                      100.000000
## Dim.24 7.531285e-30     3.012514e-29                      100.000000
## Dim.25 2.190012e-30     8.760049e-30                      100.000000
```

```
mean(eig.val[,1])
```

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

```
# SCREE PLOT
fviz_eig(PCA,addlabels=TRUE)
```



As we can see, PC1 and PC2 only explained 15.73 of the total variability, which is very poor. We need 17 PC's, which is almost all of the variables, to reach more than 80 of the variability; so is not worth it to take principal components in this case.

Models

In this section, the aim is to evaluate the performance of different classification models for the binary classification task we are deal with. For each of these techniques, we will also compute the confusion matrix, a tool which will indicate us how many instances were correctly classified and how many were misclassified. Our task attempts to estimate if an individual has risk of experiencing a heart attack or not based on various independent variables such as age, cholesterol levels, and the presence of diabetes, among others.

Logistic Regression

This method is usually used for binary classification tasks such as our case. It estimates the probability that the dependent variable (*Heart_Attack_Risk*) falls into one of the two categories based on the values of the independent variables (e.g., *Age*, *Cholesterol*, *Diabetes*, etc.).

```
fit_har_lr <- glm(Heart_Attack_Risk~.,data=trainData,family=binomial)
summary(fit_har_lr)
```

```
##
## Call:
## glm(formula = Heart_Attack_Risk ~ ., family = binomial, data = trainData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0982  -0.9516  -0.9018   1.4007   1.5991
##
## Coefficients: (3 not defined because of singularities)
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    -0.401875   0.174520  -2.303   0.0213 *
## Age                             0.028141   0.095165   0.296   0.7675
## Cholesterol                     0.150951   0.086634   1.742   0.0814 .
## Heart_Rate                     -0.064497   0.084890  -0.760   0.4474
## Diabetes                        0.123857   0.052747   2.348   0.0189 *
## Smoking                        -0.060259   0.107083  -0.563   0.5736
## Obesity                        -0.073335   0.049969  -1.468   0.1422
## Alcohol_Consumption            -0.067068   0.050930  -1.317   0.1879
## Exercise_Hours_Per_Week         0.046193   0.086199   0.536   0.5920
## Previous_Heart_Problems        -0.010825   0.049995  -0.217   0.8286
## Stress_Level                   -0.046631   0.078688  -0.593   0.5534
## Sedentary_Hours_Per_Day        -0.092392   0.086333  -1.070   0.2845
## BMI                            -0.031510   0.087124  -0.362   0.7176
## Physical_Activity_Days_Per_Week -0.007573   0.076455  -0.099   0.9211
## Sleep_Hours_Per_Day            -0.090033   0.075293  -1.196   0.2318
## Sex_Female                      0.013660   0.065001   0.210   0.8336
## Sex_Male                              NA         NA      NA       NA
## Diet_Average                   -0.030425   0.061377  -0.496   0.6201
## Diet_Healthy                    0.025131   0.061415   0.409   0.6824
## Diet_Unhealthy                        NA         NA      NA       NA
## Continent_Africa               -0.006769   0.100292  -0.067   0.9462
## Continent_Asia                 -0.126962   0.078666  -1.614   0.1065
## Continent_Australia            -0.075179   0.101202  -0.743   0.4576
## Continent_Europe               -0.138437   0.080814  -1.713   0.0867 .
## Continent_North_America         0.041800   0.100994   0.414   0.6790
## Continent_South_America               NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9145.2  on 7009  degrees of freedom
## Residual deviance: 9120.4  on 6987  degrees of freedom
## AIC: 9166.4
##
## Number of Fisher Scoring iterations: 4
```

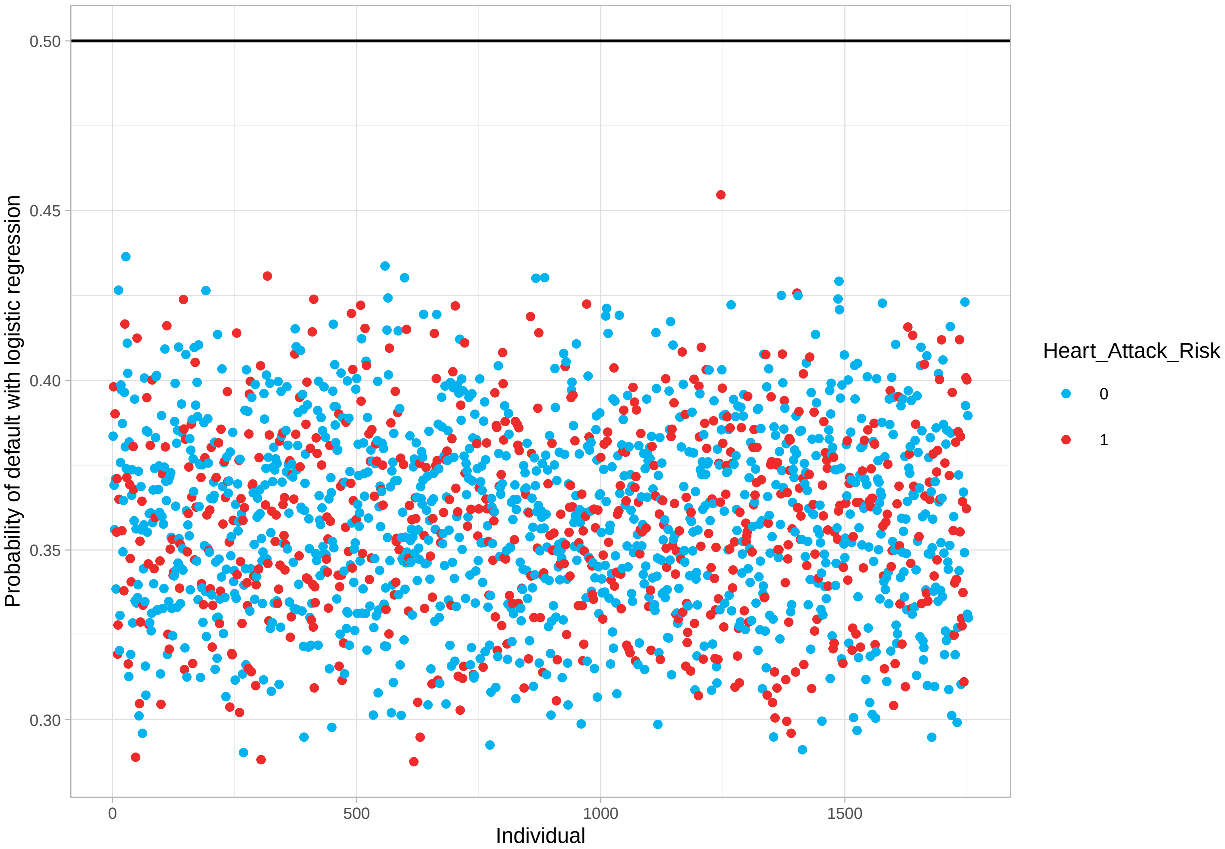
- The p-value indicate the statistical significance of each predictor variable. As we can see, just the *Diabetes* feature has a significant impact on the outcome.
- Fisher Scoring iterations, indicate the number of iterations used to estimate the model parameters.
- Value of the logistic regression coefficient is: $R^2 = \frac{\text{Residual deviance}}{\text{Null deviance}} = \frac{9120.4}{9145.2} = 0.9972882$.

Next we are going to train our data with the logistic regression model.

```
true_type <- as.matrix(testData[,15])
predict_har_lr_test <- predict.glm(fit_har_lr,newdata=testData[,-15],type="response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
ggplot(testData,aes(x=1:n_test,y=predict_har_lr_test,color=Heart_Attack_Risk)) +
  theme_light(base_size=8) +
  geom_point(size=1) +
  scale_color_manual(values=c("deepskyblue2","firebrick2")) +
  xlab("Individual") +
  ylab("Probability of default with logistic regression") +
  geom_hline(yintercept=0.5)
```



The chart shows that all probabilities of default are below 0.5, and there are no points above 0.5. This suggest that the model is predicting that all individuals belong to the compliant class which clearly indicate that our model it is overfitting.

After training our model, the next step is to evaluate its performance. To do so, we are going to obtain the confusion matrix. The goal is to compare the predictions made by the logistic regression model with the true values and quantify the errors made in the test sample.

```
predict_har_lr_bin <- character(length=nrow(testData))
predict_har_lr_bin[predict_har_lr_test>0.5] <- "Yes"
predict_har_lr_bin[predict_har_lr_test<=0.5] <- "No"
addmargins(table(predict_har_lr_bin,testData$Heart_Attack_Risk))
```

```
##
## predict_har_lr_bin    0    1  Sum
##                 No 1124  629 1753
##                Sum 1124  629 1753
```

If we analysed the confusion matrix we can conclude:

- 1) The model incorrectly classified 629 instances as “Yes” (1) when they were actually “No” (0).
- 2) And it correctly classified 1124 points.

```
predict_har_lr_bin<-as.factor(predict_har_lr_bin)
predict_har_lr_bin<-as.numeric(predict_har_lr_bin)
1-sum(predict_har_lr_bin!=testData$Heart_Attack_Risk)/n_test
```

```
## [1] 0.3588135
```

The **test error rate** is 0.3588135, which means that 35.88% of the cases in the test dataset are predicted incorrectly by the model. So this is not a model with a big accuracy.

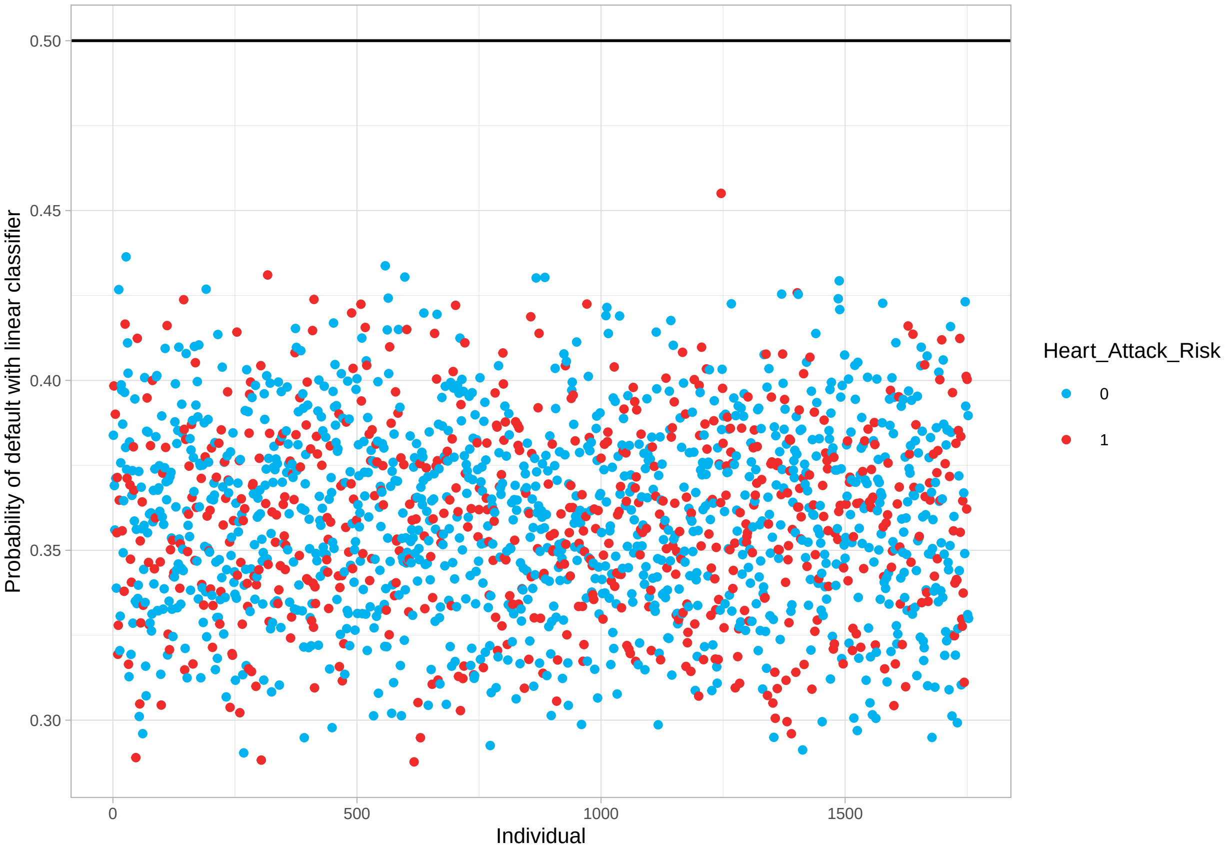
Linear Classifier

A linear classifier is a model that makes a decision to categories a set of data points to a discrete class based on a linear combination of its explanatory variables. The technique used for this purpose is called *Linear Discriminant Analysis* (LDA).

```
## Call:
## lda(Heart_Attack_Risk ~ ., data = trainData)
##
## Prior probabilities of groups:
##         0         1
## 0.6419401 0.3580599
##
## Group means:
##         Age Cholesterol Heart_Rate  Diabetes   Smoking   Obesity
## 0 0.4969475   0.4933389  0.5004667 0.6424444 0.9000000 0.5068889
## 1 0.4972112   0.5057598  0.4950825 0.6693227 0.8944223 0.4884462
##   Alcohol_Consumption Exercise_Hours_Per_Week Previous_Heart_Problems
## 0           0.6060000                0.498881               0.5022222
## 1           0.5900398                0.503113               0.4996016
##   Stress_Level Sedentary_Hours_Per_Day       BMI
## 0    0.4994321               0.5033227 0.4942827
## 1    0.4951306               0.4963311 0.4920896
##   Physical_Activity_Days_Per_Week Sleep_Hours_Per_Day Sex_Female  Sex_Male
## 0                       0.5000952           0.5057037  0.3008889 0.6991111
## 1                       0.4990324           0.4958167  0.3075697 0.6924303
##   Diet_Average Diet_Healthy Diet_Unhealthy Continent_Africa Continent_Asia
## 0    0.3444444    0.3315556      0.3240000       0.09955556      0.2991111
## 1    0.3346614    0.3402390      0.3250996       0.10597610      0.2844622
##   Continent_Australia Continent_Europe Continent_North_America
## 0           0.1013333        0.2588889              0.09377778
## 1           0.1003984        0.2442231              0.10597610
##   Continent_South_America
## 0               0.1473333
## 1               0.1589641
##
## Coefficients of linear discriminants:
##                                          LD1
## Age                              0.22734416
## Cholesterol                      1.21681337
## Heart_Rate                      -0.51850121
## Diabetes                         0.99361643
## Smoking                         -0.48844670
## Obesity                         -0.59096099
## Alcohol_Consumption             -0.54230037
## Exercise_Hours_Per_Week          0.37165057
## Previous_Heart_Problems         -0.08741687
## Stress_Level                    -0.37378847
## Sedentary_Hours_Per_Day         -0.74360394
## BMI                             -0.25346824
## Physical_Activity_Days_Per_Week -0.05882198
## Sleep_Hours_Per_Day             -0.72467781
## Sex_Female                       0.05544484
## Sex_Male                        -0.05544484
## Diet_Average                    -0.22952619
## Diet_Healthy                     0.21754624
## Diet_Unhealthy                   0.01429798
## Continent_Africa                 0.51404864
## Continent_Asia                  -0.45537177
## Continent_Australia             -0.04130174
## Continent_Europe                -0.54735759
## Continent_North_America          0.91864594
## Continent_South_America          0.57200269
```

The prior probabilities are 0.6419401 (class 0) and 0.3580599 (class 1). This information tell us that Class 0 is more prevalent in the training data, as it has a higher prior probability compared to Class 1.

```r
predict_har_lda_test <- predict(fit_har_lda,testData[,-15])
ggplot(testData,aes(x=1:n_test,
                y=predict_har_lda_test$posterior[,2],
                color=Heart_Attack_Risk)) +
  theme_light(base_size=8) +
  geom_point(size=1) +
  scale_color_manual(values=c("deepskyblue2","firebrick2")) +
  xlab("Individual") +
  ylab("Probability of default with linear classifier") +
  geom_hline(yintercept=0.5)
```



```
addmargins(table(predict_har_lda_test$class,testData$Heart_Attack_Risk))
```

```
##
##          0    1  Sum
##   0   1124  629 1753
##   1      0    0    0
##   Sum 1124  629 1753
```

```
sum(predict_har_lda_test$class!=testData$Heart_Attack_Risk)/n_test
```

```
## [1] 0.3588135
```

In the same manner as before, we follow the same structure. First, we obtain the default probabilities and then constructed the confusion matrix. The result obtained with this technique are:

- The model made 1124 correct predictions. In particular, it can be seen how the model tends to predict well those outcomes where there is no risk for the patient to suffer a heart attack (value = 0).
- On the other hand, the model predict 629 times wrong.
- The **test error rate** is 0.3588135, which means that around 35.88% of the cases in the test dataset are predicted incorrectly by the model.

Hence, it indicates that the model is making incorrect predictions for a significant portion of the test data. However present similar results to the previous model with logistic regression.

Naive Bayes classifier

This is a supervised machine learning technique also used for classification tasks. For this purpose, we will use the `naiveBayes` function with the training sample.

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##         0         1
## 0.6419401 0.3580599
##
## Conditional probabilities:
##    Age
## Y        [,1]      [,2]
##   0 0.4969475 0.2931761
##   1 0.4972112 0.2961485
##
##    Cholesterol
## Y        [,1]      [,2]
##   0 0.4933389 0.2888030
##   1 0.5057598 0.2887179
##
##    Heart_Rate
## Y        [,1]      [,2]
##   0 0.5004667 0.2941408
##   1 0.4950825 0.2949184
##
##    Diabetes
## Y        [,1]      [,2]
##   0 0.6424444 0.4793335
##   1 0.6693227 0.4705508
##
##    Smoking
## Y        [,1]      [,2]
##   0 0.9000000 0.3000333
##   1 0.8944223 0.3073576
##
##    Obesity
## Y        [,1]      [,2]
##   0 0.5068889 0.5000081
##   1 0.4884462 0.4999661
##
##    Alcohol_Consumption
## Y        [,1]      [,2]
##   0 0.6060000 0.4886891
##   1 0.5900398 0.4919240
##
##    Exercise_Hours_Per_Week
## Y       [,1]      [,2]
##   0 0.498881 0.2889334
##   1 0.503113 0.2915306
## 
##    Previous_Heart_Problems
## Y        [,1]      [,2]
##   0 0.5022222 0.5000506
##   1 0.4996016 0.5000995
## 
##    Stress_Level
## Y        [,1]      [,2]
##   0 0.4994321 0.3202673
##   1 0.4951306 0.3131457
## 
##    Sedentary_Hours_Per_Day
## Y        [,1]      [,2]
##   0 0.5033227 0.2879803
##   1 0.4963311 0.2923469
## 
##    BMI
## Y        [,1]      [,2]
##   0 0.4942827 0.2869796
##   1 0.4920896 0.2867279
## 
##    Physical_Activity_Days_Per_Week
## Y        [,1]      [,2]
##   0 0.5000952 0.3239505
##   1 0.4990324 0.3315628
## 
##    Sleep_Hours_Per_Day
## Y        [,1]      [,2]
##   0 0.5057037 0.3333030
##   1 0.4958167 0.3292465
## 
##    Sex_Female
## Y        [,1]      [,2]
##   0 0.3008889 0.4586955
##   1 0.3075697 0.4615793
## 
##    Sex_Male
## Y        [,1]      [,2]
##   0 0.6991111 0.4586955
##   1 0.6924303 0.4615793
## 
##    Diet_Average
## Y        [,1]      [,2]
##   0 0.3444444 0.4752396
##   1 0.3346614 0.4719660
## 
##    Diet_Healthy
## Y        [,1]      [,2]
##   0 0.3315556 0.4708245
##   1 0.3402390 0.4738839
## 
##    Diet_Unhealthy
## Y        [,1]      [,2]
##   0 0.3240000 0.4680520
##   1 0.3250996 0.4685054
##
##    Continent_Africa
## Y         [,1]      [,2]
##   0 0.09955556 0.2994398
##   1 0.10597610 0.3078684
##
##    Continent_Asia
## Y        [,1]      [,2]
##   0 0.2991111 0.4579195
##   1 0.2844622 0.4512478
##
##    Continent_Australia
## Y        [,1]      [,2]
##   0 0.1013333 0.3018031
##   1 0.1003984 0.3005904
##
##    Continent_Europe
## Y        [,1]      [,2]
##   0 0.2588889 0.4380731
##   1 0.2442231 0.4297112
##
##    Continent_North_America
## Y         [,1]      [,2]
##   0 0.09377778 0.2915517
##   1 0.10597610 0.3078684
##
##    Continent_South_America
## Y        [,1]      [,2]
##   0 0.1473333 0.3544773
##   1 0.1589641 0.3657155
```

As we can see the prior probabilities are 0.6419401 for no risk of having a heart attack (value = 0) and 0.3580599 for having risk for a heart attack. In addition, we have different tables for each of the predictor. Let's analyzed the first situation which is for the *Age* variable. In this case,

- Y = 0 (**not having risk for a heart attack**) the mean is 0.4969475 and the standard deviation is 0.2931761
- Y = 1 (**having risk for a heart attack**) the mean is $0.4972112 $ and the standard deviation is 0.2961485

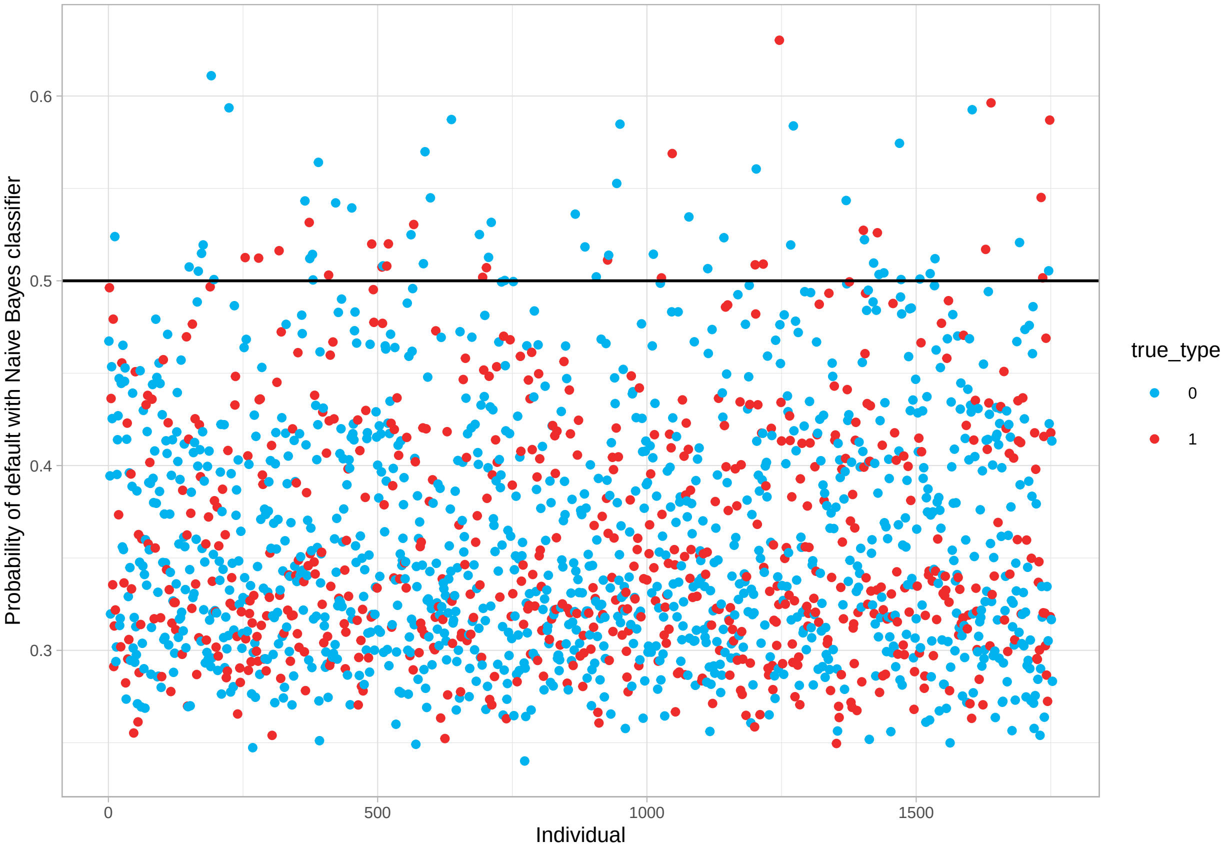
```
contrasts(trainData$Heart_Attack_Risk)
```

```
##   1
## 0 0
## 1 1
```

```
contrasts(testData$Heart_Attack_Risk)
```

```
##    1
## 0 0
## 1 1
```

```
predict_har_test <- predict(fit_har,as.matrix(testData[,-15]),type="raw")
true_type <- as.matrix(testData[,15])
ggplot(testData,aes(x=1:n_test,y=predict_har_test[,2],color=true_type)) +
  theme_light(base_size=8) +
  geom_point(size=1) +
  scale_color_manual(values=c("deepskyblue2","firebrick2")) +
  xlab("Individual") +
  ylab("Probability of default with Naive Bayes classifier") +
  geom_hline(yintercept=0.5)
```



The plot shows that most probabilities of default are below 0.5. Indeed the points are scattered and mixed, so clearly our model it is not performance well in distinguishing between the two options.

After training our model, the next step is to evaluate its performance with the confusion matrix.

```
predict_har_test_type <- predict(fit_har,as.matrix(testData[,-15]),type="class")
addmargins(table(predict_har_test_type,true_type))
```

```
##                      true_type
## predict_har_test_type    0    1  Sum
##                   0   1073  604 1677
##                   1     51   25   76
##                   Sum 1124  629 1753
```

```r
sum(predict_har_test_type!=true_type)/n_test
```

```
## [1] 0.3736452
```

The result obtained with this technique are:

- The model made 1098 correct predictions. Which is the sum of 1073 (which is the number of times the model predicts that someone is at risk of having a heart attack and is true) plus 25, the times where the model predicts that someone is at no risk of having a heart attack and is true.
- On the other hand, the model predict 655 times wrong.
- The **test error rate** is 0.3736452, which means that around 37.36% of the cases in the test dataset are predicted incorrectly by the model.

Hence, it indicates that the model is making incorrect predictions for a significant portion of the test data.

Ensemble model

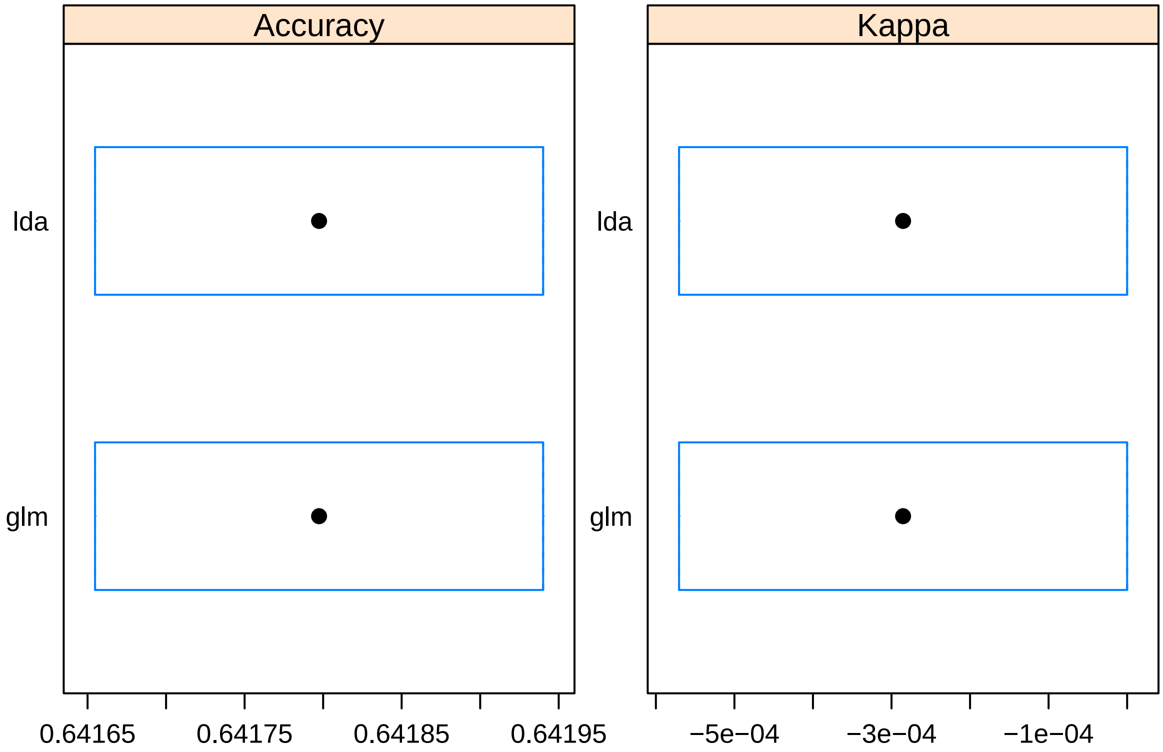
Ensemble modeling is a methodology that leverages the predictions of multiple individual models in order to enhance overall predictive accuracy. Often we can achieve better results combining the strengths of different models than with a single model.

```r
library(caretEnsemble)

trainControl2 = trainControl(
  method = 'cv',
  number = 2,
  savePredictions = 'final',
  classProbs=TRUE)


levels(trainData$Heart_Attack_Risk) <- make.names(levels(trainData$Heart_Attack_Risk))
algorithmList = c('glm', 'lda')
models = caretList(Heart_Attack_Risk~ .,
                   data=trainData,
                   trControl=trainControl2, methodList=algorithmList)
results = resamples(models)
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: glm, lda
## Number of resamples: 2
##
## Accuracy
##          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## glm 0.6416548 0.6417261 0.6417974 0.6417974 0.6418688 0.6419401    0
## lda 0.6416548 0.6417261 0.6417974 0.6417974 0.6418688 0.6419401    0
##
```

```
## Kappa
##              Min.       1st Qu.        Median          Mean       3rd Qu. Max.
## glm -0.0005704844 -0.0004278633 -0.0002852422 -0.0002852422 -0.0001426211    0
## lda -0.0005704844 -0.0004278633 -0.0002852422 -0.0002852422 -0.0001426211    0
##     NA's
## glm    0
## lda    0
```

```
# Box plots to compare models
scales = list(x=list(relation="free"), y=list(relation="free"))
bwplot(results, scales=scales)
```



From this firsts results, we can say: - **Accuracy**. Both models present similar values, so they are working similarly at making the correct predictions. - **Kappa**. Measure the agreement between the model's predictions and random chance. Both values are close to zero which might represent that the models are struggling to capture meaningful patterns in the data.

Next step is based on created the Generalized Linear Model, by adding all the models and creating a new model that theoretically will perform better.

```
# Ensemble the predictions of 'models' to form a new combined prediction based on glm
stack.glm = caretStack(models, method="glm")
print(stack.glm)
```

```
## A glm ensemble of 2 base models: glm, lda
##
```

```
## Ensemble results:
## Generalized Linear Model
##
## 7010 samples
##    2 predictor
##    2 classes: 'X0', 'X1'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 7010, 7010, 7010, 7010, 7010, 7010, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.6396606  0
```

```
# Predict on testData
har_predicteds = predict(stack.glm, newdata=testData)
table(har_predicteds)
```

```
## har_predicteds
##   X0   X1
## 1753    0
```

```
# Compute the confusion matrix
har_predicteds <- factor(har_predicteds, levels = levels(testData$Heart_Attack_Risk))

confusionMatrix(reference = testData$Heart_Attack_Risk,
                data = har_predicteds,
                mode = 'everything', positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 0 0
##          1 0 0
##
##                Accuracy : NaN
##                  95% CI : (NA, NA)
##     No Information Rate : NA
##     P-Value [Acc > NIR] : NA
##
##                   Kappa : NaN
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity :  NA
##             Specificity :  NA
##          Pos Pred Value :  NA
##          Neg Pred Value :  NA
##               Precision :  NA
##                  Recall :  NA
##                      F1 :  NA
##               Prevalence : NaN
##           Detection Rate : NaN
##     Detection Prevalence : NaN
##        Balanced Accuracy :  NA
##
##         'Positive' Class : 1
##
```

EXTRA: What will happend if we choose a non-categorical variable?

The structure of the problem remains the same. We must choose a variable that ensure these conditions, see which variables are most relevant to the problem at hand, apply different models and see which one is working better.

For this exercise we are going to choose *Cholesterol* parameter, which is a continuous variable.

1. Select predictors

The idea is to find which variables have a huge impact on predicting the *Cholesterol* variable.

```
fit_cholesterol<-lm(Cholesterol~.,data = df)
summary(fit_cholesterol)
```

```
##
## Call:
## lm(formula = Cholesterol ~ ., data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.53785 -0.24414 -0.00454  0.24775  0.54168
##
## Coefficients: (3 not defined because of singularities)
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     0.4724800  0.0212433  22.241   <2e-16 ***
## Age                            -0.0200111  0.0117042  -1.710   0.0873 .
## Heart_Rate                      0.0008951  0.0105090   0.085   0.9321
## Diabetes                       -0.0083111  0.0064782  -1.283   0.1995
## Smoking                         0.0262563  0.0132416   1.983   0.0474 *
## Obesity                        -0.0084163  0.0061706  -1.364   0.1726
## Alcohol_Consumption            -0.0046617  0.0062946  -0.741   0.4590
## Exercise_Hours_Per_Week         0.0206007  0.0106644   1.932   0.0534 .
## Previous_Heart_Problems        -0.0037271  0.0061734  -0.604   0.5460
## Stress_Level                   -0.0214667  0.0097151  -2.210   0.0272 *
## Sedentary_Hours_Per_Day         0.0194481  0.0106857   1.820   0.0688 .
## BMI                             0.0170453  0.0107372   1.588   0.1124
## Physical_Activity_Days_Per_Week  0.0143023  0.0094606   1.512   0.1306
## Sleep_Hours_Per_Day             0.0039218  0.0093122   0.421   0.6737
## Heart_Attack_Risk1              0.0117825  0.0064358   1.831   0.0672 .
## Sex_Female                      0.0038327  0.0080559   0.476   0.6343
## Sex_Male                               NA         NA      NA       NA
## Diet_Average                   -0.0022114  0.0075854  -0.292   0.7706
## Diet_Healthy                   -0.0076462  0.0075521  -1.012   0.3113
## Diet_Unhealthy                          NA         NA      NA       NA
## Continent_Africa               -0.0093987  0.0125253  -0.750   0.4530
## Continent_Asia                 -0.0004230  0.0096999  -0.044   0.9652
## Continent_Australia            -0.0156412  0.0124857  -1.253   0.2103
## Continent_Europe                0.0003280  0.0099226   0.033   0.9736
## Continent_North_America         0.0088269  0.0125845   0.701   0.4831
## Continent_South_America                NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2886 on 8740 degrees of freedom
## Multiple R-squared:  0.003987,   Adjusted R-squared:  0.00148
## F-statistic:  1.59 on 22 and 8740 DF,  p-value: 0.03916
```

We can conclude that there are two variables which highly affect when predicting the Cholesterol value of a patient:

- Smoking
- Stress_Level

Furthermore, we can noticed how variables such as *Age*, *Exercise_Hours_Per_Week*, *Sedentary_Hours_Per_Day* and *Heart_Attack_Risk* have also some influence on the *Cholesterol* variable. However for simplicity we are just going to work with the two most influence features.

2. Models and Interpretations

Regression tree First, we are going to use a regression tree to predict *Cholesterol* using the predictors *Smoking* and *Stress_Leve.* To do this, we generate a training sample and a test sample with the data we have (same steps as before).

```
##
## Regression tree:
## tree(formula = Cholesterol ~ Smoking + Stress_Level, data = trainData,
##     control = tree.control(nobs = nrow(trainData), mincut = 5,
##         minsize = 10))
## Variables actually used in tree construction:
## character(0)
## Number of terminal nodes:  1
## Residual mean deviance:  0.08341 = 584.6 / 7009
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.497800 -0.244200 -0.004929  0.000000  0.248600  0.502200
```

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
## 1) root 7010 584.6 0.4978 *
```

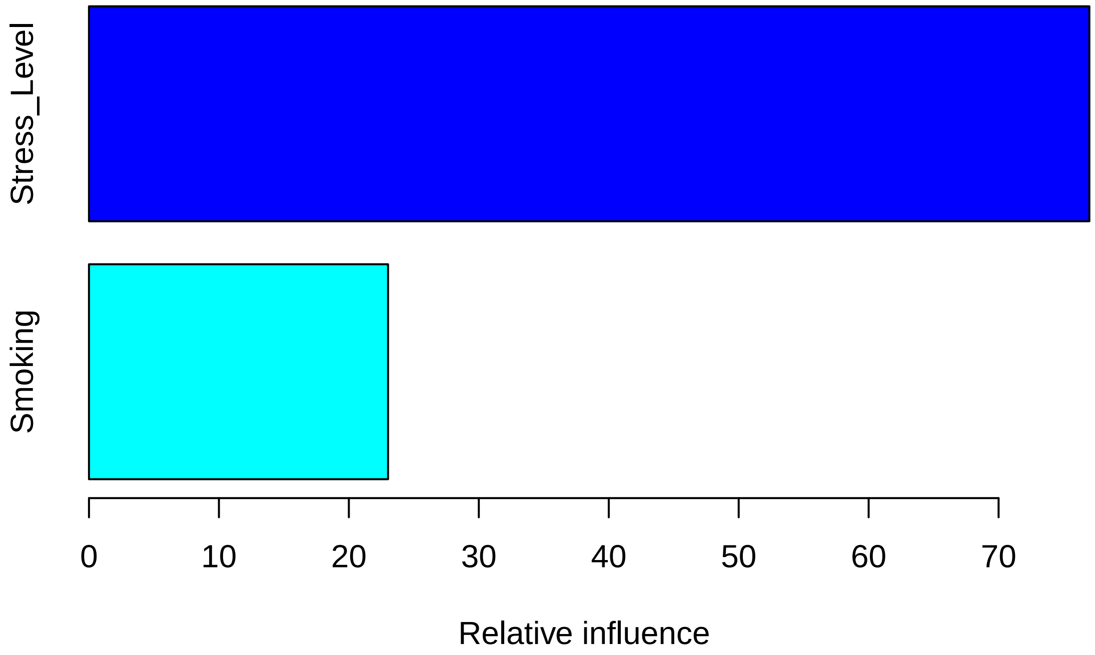
There is not enough relationship between the data so it only makes one node. Thus, we can create the plot and indeed it is not a good model choice.

Boosting For performing this technique, we must set up some options in order to work.

1. `distribution` must be `Gaussian`.
2. `n.trees` is the number of trees to use, which by default is 100.
3. `interaction.depth` is the depth of each tree, which defaults to 4.
4. `shrinkage` is the *lambda* parameter, which defaults to 0.1.
5. `bag.fraction` indicates the number of observations used to construct the trees in the algorithm, which defaults to 0.5.

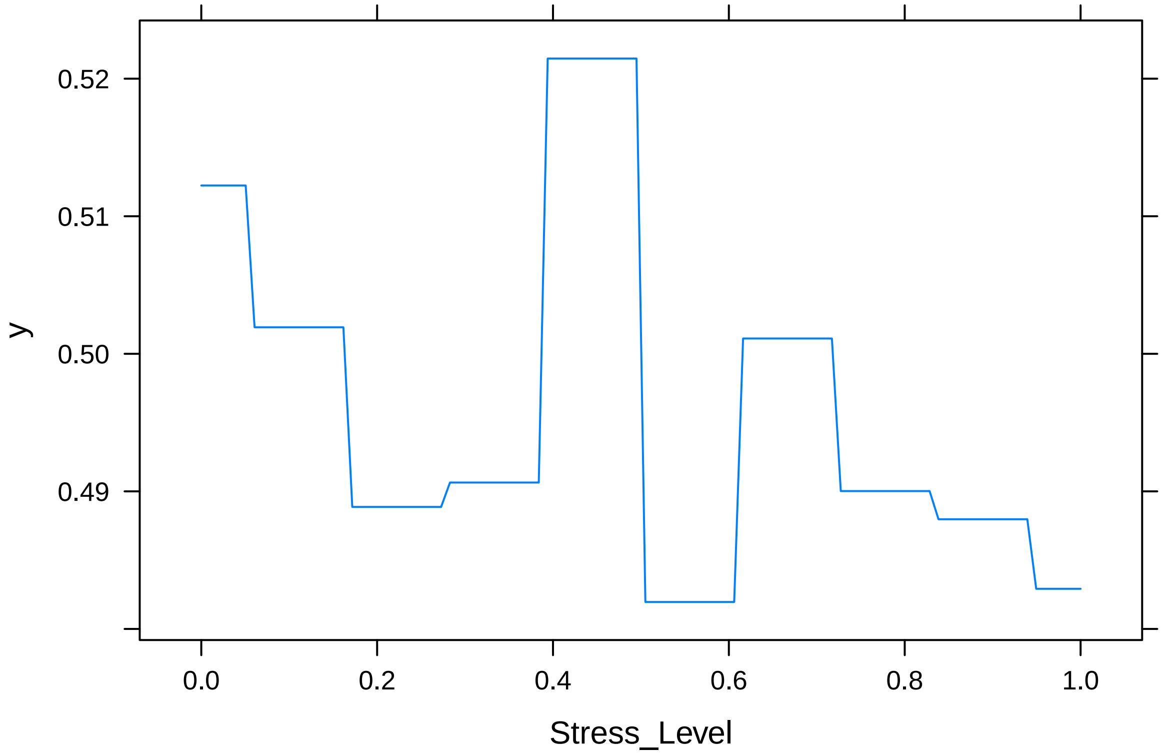
The summary function provides a measure of the relative influence of each predictor. We can also obtain a graph with the values of these influences, although with two predictors it is not very useful.

```
library(gbm)
boost.ch <- gbm(Cholesterol~Stress_Level+Smoking,data=trainData,
                    distribution="gaussian",n.trees=100,interaction.depth=4,
               shrinkage=0.1,bag.fraction=0.5)
summary(boost.ch)
```

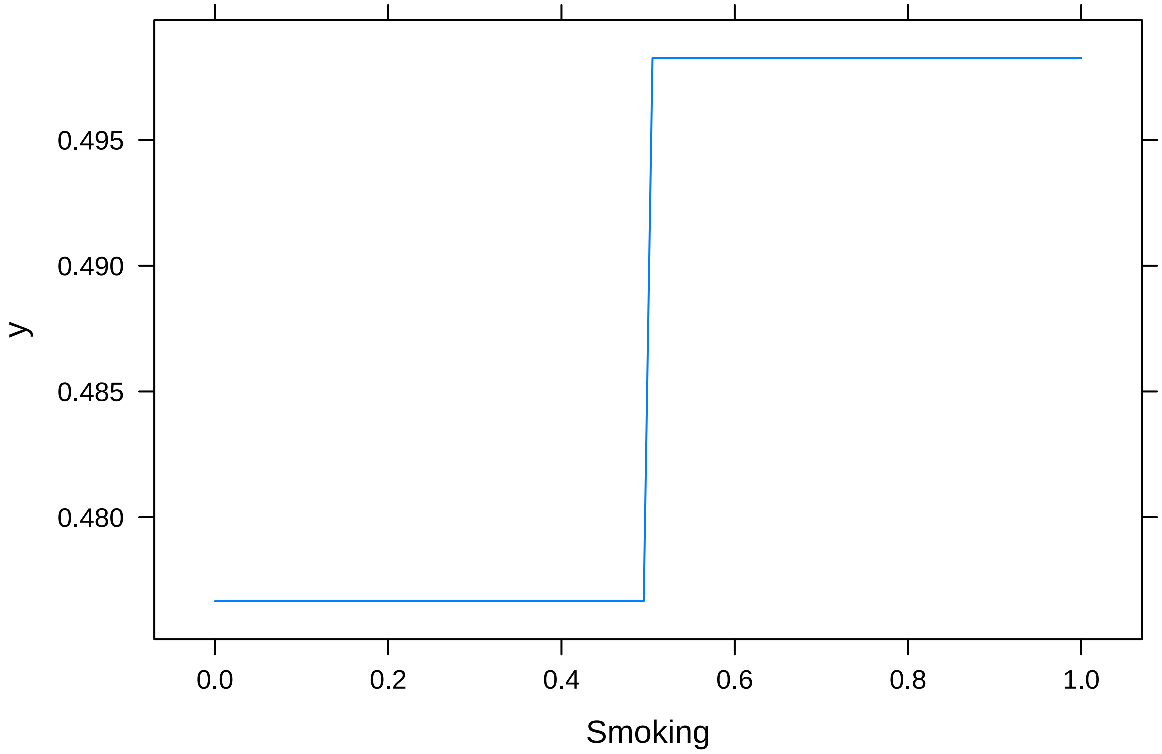


```
##                       var  rel.inf
## Stress_Level Stress_Level 76.9849
## Smoking           Smoking 23.0151
```

```
plot(boost.ch,i="Stress_Level")
```



```
plot(boost.ch,i="Smoking")
```

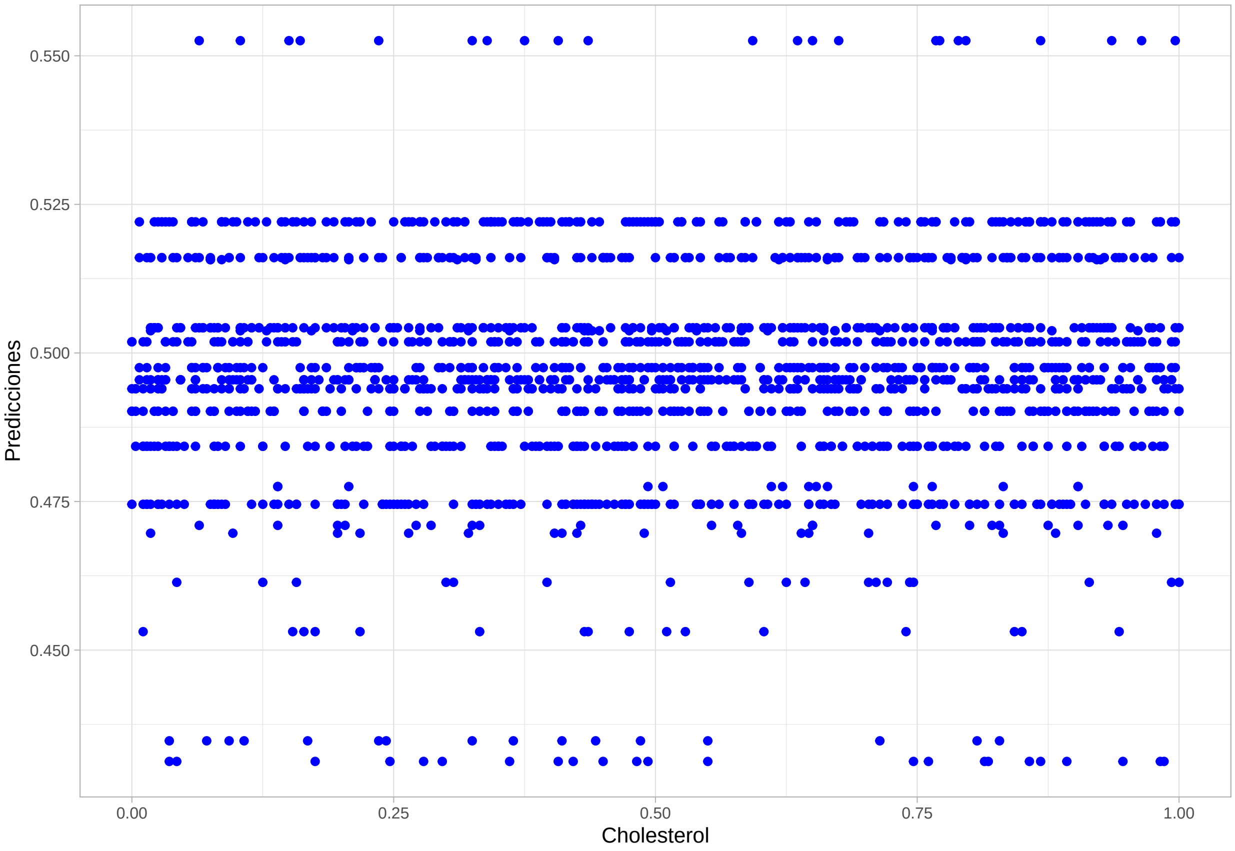


Here, we show two plots of the marginal effect of each of the predictor. This is getting by removinng the effect of the other predictor.

```
pred_ch_Test_stress_smoking <- predict(boost.ch,newdata=testData[,c(5,10)],
                                       n.trees=100)
ECMT_bo <- mean((testData$Cholesterol-pred_ch_Test_stress_smoking)^2)
ECMT_bo
```

```
## [1] 0.0834837
```

```
testData$Pred.Cholesterol<- pred_ch_Test_stress_smoking
ggplot(testData,aes(x=Cholesterol,y=Pred.Cholesterol)) +
  theme_light(base_size=8) +
  geom_point(size=1,color="blue") +
  labs(x="Cholesterol",y="Predicciones")
```



Finally, we have carried out the prediction of the test sample observations. The ECM test with this tree turns out to be 0.08347973.

K- means clustering Let's consider a different problem. We want to discover behavior patterns between two different variables in order to classify our patients in groups. This technique belongs to unsupervised classification and it is called clustering. Particularly, k-means clustering finds k groups and approximate the observations to the nearest k-mean.

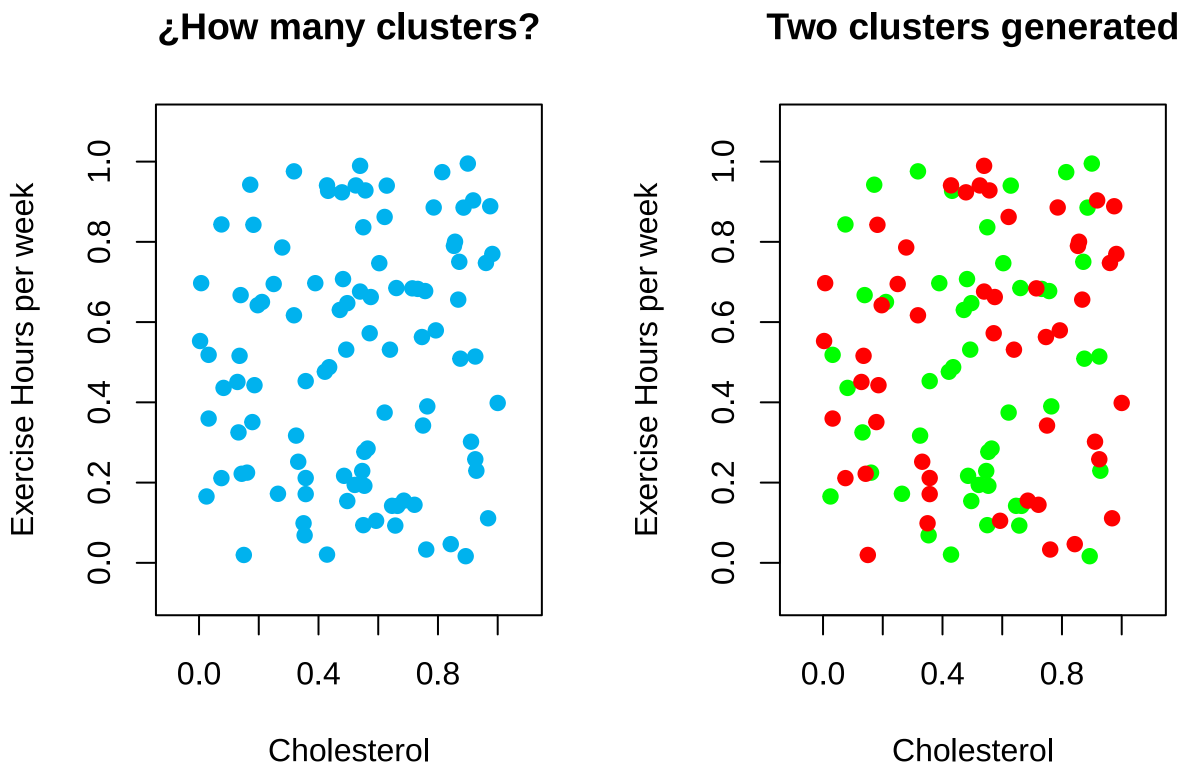
For drawing clusters in a bi dimensional space, we are going to choose two variables ($p = 2$: Cholesterol and Exercise Hours per week) and two populations ($K = 2$: Heart_Attack_risk (0 if not and 1 if there is risk)).

```
n <- 100
p <- 2
set.seed(33)

# We split the two variables into the two different populations
df_cluster<-df[,c(2,8,15)]
df_cluster_0 <- df_cluster[df_cluster$Heart_Attack_Risk == 0, c(1,2)]
df_cluster_1 <- df_cluster[df_cluster$Heart_Attack_Risk == 1, c(1,2)]
df_cluster_0<-as.matrix(df_cluster_0) #1:5624
df_cluster_1<-as.matrix(df_cluster_1) #1:3139
# We make two samples because the size is too big to appreciate the results
df_cluster_0 <- df_cluster_0[sample(1:nrow(df_cluster_0), 50), ]
df_cluster_1 <- df_cluster_1[sample(1:nrow(df_cluster_1), 50), ]

X <- matrix(NA,nrow=n,ncol=p)
```

```
X[1:50,]<-df_cluster_0
X[51:100,]<-df_cluster_1
```

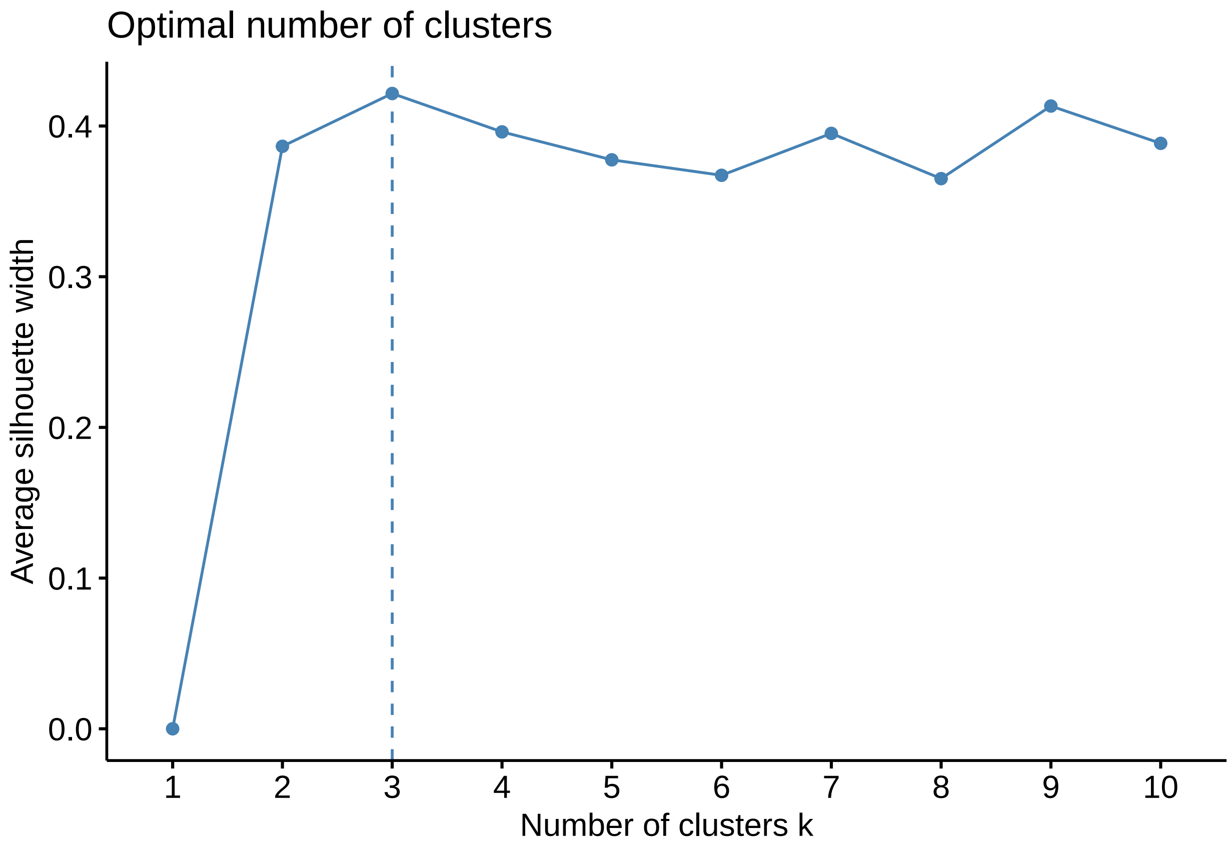
We make a graph of the data set where we see the two groups generated and where we can see that mostly of the observations are mixed between groups.

```
par(mfrow=c(1,2))
color_1 <- "green" # no risk
color_2 <- "red" # risk
colors_X <- rep("deepskyblue2",n)
plot(X[,1],X[,2],pch=19,col=colors_X,
     xlim=c(range(X[,1])[1]-0.1,range(X[,1])[2]+0.1),
     ylim=c(range(X[,2])[1]-0.1,range(X[,2])[2]+0.1),
     main="¿How many clusters?",
     xlab="Cholesterol",ylab="Exercise Hours per week")
colors_X <- c(rep(color_1,50),rep(color_2,50))
plot(X[,1],X[,2],pch=19,col=colors_X,
     xlim=c(range(X[,1])[1]-0.1,range(X[,1])[2]+0.1),
     ylim=c(range(X[,2])[1]-0.1,range(X[,2])[2]+0.1),
     main="Two clusters generated",
     xlab="Cholesterol",ylab="Exercise Hours per week")
```



We are going to use the function `kmeans` to select how many clusters K are neccesary with the average silhouette. Furthermore, we can see the silhouette obtained for all observations in the data set with the function `silhouette`:

```
library("factoextra")
par(mfrow=c(1,1))
fviz_nbclust(X,kmeans,method="silhouette",k.max=10)
```



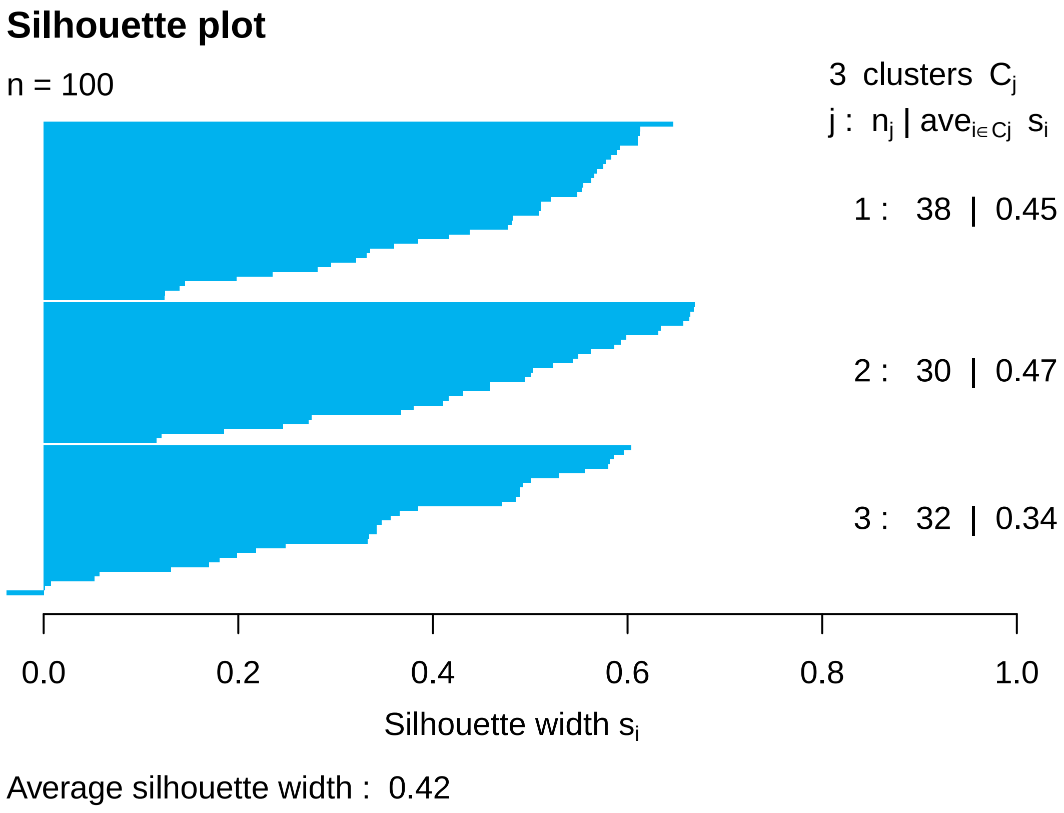
```
X_kmeans <- kmeans(X,centers=3,nstart=1000) # k=3
X_kmeans$cluster
```

```
##   [1] 2 1 1 1 2 1 3 1 2 3 2 3 2 1 2 2 1 2 2 2 3 1 3 3 3 3 3 1 3 3 1 2 2 3 2 1 1
##  [38] 1 3 1 3 2 2 2 1 2 3 1 1 3 1 2 2 3 1 2 3 1 1 1 3 3 3 1 2 3 3 1 2 2 2 1 1 3
##  [75] 3 1 3 3 2 2 1 1 3 2 1 3 1 1 3 1 1 2 1 2 3 1 3 2 1 1
```

```
X_kmeans$centers
```

```
##        [,1]      [,2]
## 1 0.6929511 0.7730844
## 2 0.6483333 0.1831785
## 3 0.1972098 0.4863753
```

```
library(cluster)
sil_X_kmeans <- silhouette(X_kmeans$cluster,
                           dist(X,method="euclidean"))
plot(sil_X_kmeans,col="deepskyblue2",
     main="Silhouette plot")
```



So, we draw the clusters obtained by k-means and compare it to the true partition. For that we need the library `pracma`:

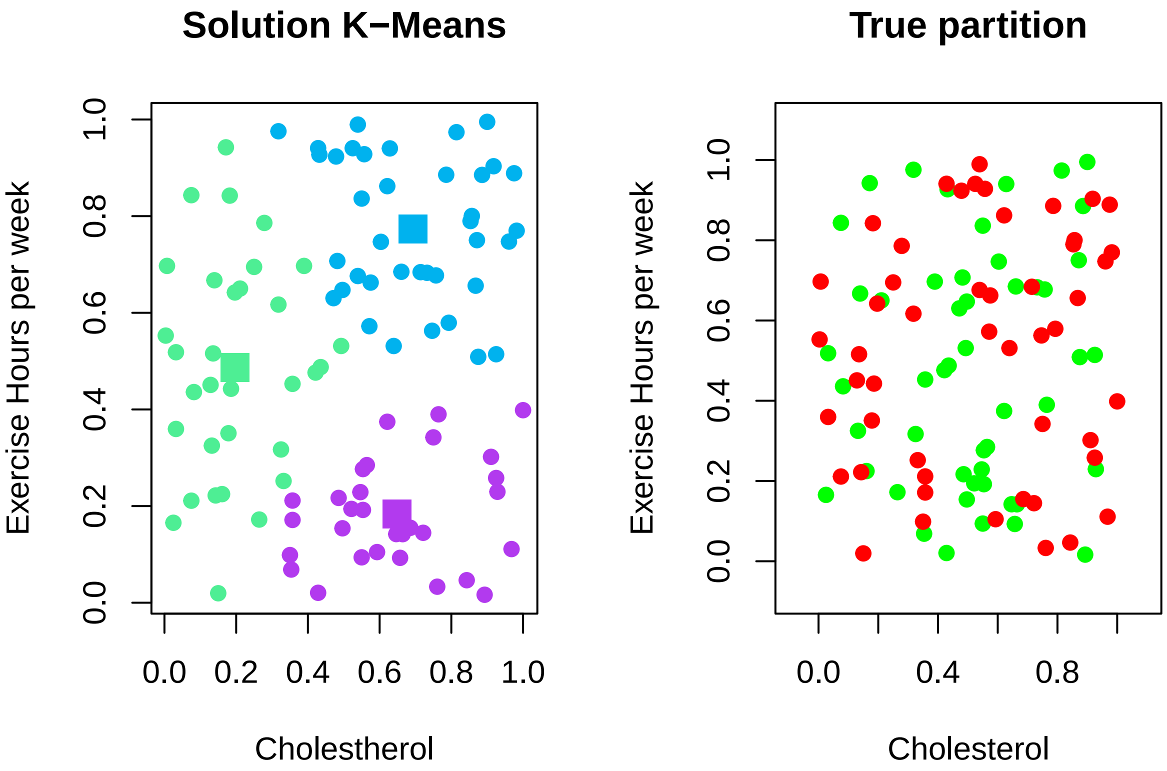
```
color_1 <- "deepskyblue2"
color_2 <- "darkorchid2"
color_3 <- "seagreen2"
library(pracma)
```

```
##
## Attaching package: 'pracma'

## The following object is masked from 'package:e1071':
##
##     sigmoid

## The following objects are masked from 'package:DescTools':
##
##     Mode, Rank
```

```
cl_current <- X_kmeans$cluster
cl_current
```

```
##   [1] 2 1 1 1 2 1 3 1 2 3 2 3 2 1 2 2 1 2 2 2 3 1 3 3 3 3 3 1 3 3 1 2 2 3 2 1 1
##  [38] 1 3 1 3 2 2 2 1 2 3 1 1 3 1 2 2 3 1 2 3 1 1 1 3 3 3 1 2 3 3 1 2 2 2 1 1 3
##  [75] 3 1 3 3 2 2 1 1 3 2 1 3 1 1 3 1 1 2 1 2 3 1 3 2 1 1
```

```r
par(mfrow=c(1,2))
run_step <- 1
while (run_step==1){
  C1_center <- colMeans(X[cl_current==1,])
  C2_center <- colMeans(X[cl_current==2,])
  C3_center <- colMeans(X[cl_current==3,])
  colors_X <- c(color_1,color_2,color_3)[cl_current]
  plot(X[,1],X[,2],pch=19,col=colors_X,
       main="Solution K-Means",
       xlab="Cholestherol",ylab="Exercise Hours per week")
  points(C1_center[1],C1_center[2],pch=15,col=color_1,cex=2)
  points(C2_center[1],C2_center[2],pch=15,col=color_2,cex=2)
  points(C3_center[1],C3_center[2],pch=15,col=color_3,cex=2)
  d_1 <- distmat(X,C1_center)
  d_2 <- distmat(X,C2_center)
  d_3 <- distmat(X,C3_center)
  distances <- cbind(d_1,d_2,d_3)
  cl_new <- apply(distances,1,which.min)
  cl_new
  cl_equal <- sum(cl_current == cl_new)
  if (cl_equal==100){
    run_step <- 0
  } else {
    cl_current <- cl_new
  }
}

# True partition
color_1 <- "green" # no risk
color_2 <- "red" # risk
colors_X <- c(rep(color_1,50),rep(color_2,50))
plot(X[,1],X[,2],pch=19,col=colors_X,
     xlim=c(range(X[,1])[1]-0.1,range(X[,1])[2]+0.1),
     ylim=c(range(X[,2])[1]-0.1,range(X[,2])[2]+0.1),
     main="True partition"
     ,xlab="Cholesterol",ylab="Exercise Hours per week")
```



In conclusion, algorithm clustering k-means is not useful for this two variables and two populations due to data does not follow any pattern. Moreover, k-means gives us 3 clusters when there are only two different populations known.

Final conclusion

It is important to highlight the relevance of the dataset in our project. The quality of the data we work with influences our ability to gain insights and develop effective models. In our particular case, the dataset was synthetically generated through artificial intelligence. This may account for some of the challenges we encountered during the project, including the difficulty in arriving at definitive conclusions and achieving good results with our machine learning models.The synthetically data has cause a limited correlation between variables which add complexity to our task of creating predictive models for specific features. Nonetheless, throughout the course of this project, we explored various machine learning models and gained valuable insights into their performance.