Data Storytelling Statistical Report

Tan Thien Nguyen

Table of Contents

1. Introduction

World Health Organization (WHO) has estimated that heart disease caused 18 million deaths to occur worldwide in 2019, which made heart disease the top 1 cause of global death. The early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high-risk patients and, in turn, reduce complications. This research aims to investigate the most relevant factors of heart disease as well as predict the overall risk using logistic regression on data collected from an ongoing heart study in Framingham, Massachusetts, United States.

2. Rationale and Research Question

2.1. Rationale

Cardiovascular diseases (CVDs) are the leading cause of death globally. WHO estimated that 18 million people died from CVDs in 2019, representing 32% of all global deaths. Since 1990, the most significant increase in deaths has been from heart disease, rising by more than 12 million to 18 million deaths in 2019 (WHO, 2022).

Most cardiovascular diseases can be prevented by addressing behavioral risk factors such as tobacco use, unhealthy diet and obesity, physical inactivity, and harmful use of alcohol. It is crucial to detect the cardiovascular disease as early as possible to begin management with counseling and medicines (WHO, 2022).

2.2. Personal Motivation

Seven years ago, my father had a heart attack due to myocardial infarction when I worked in Phnom Penh, Cambodia, 1500 km away from my hometown, Hanoi, Vietnam. Luckily, my father was taken to the emergency room in time. He got better after a few months but has been taking medicine and treatment for the past seven years till now. From that moment, I wanted some days I could make something to help predict the risk of heart disease earlier so that people at high risk are able to be aware of this and change their lifestyle before too late.

2.3. Research Question

RQ1: "What are the most significant factors of heart disease?"

RQ2: "How accurately can Logistic Regression model predict the risk of heart disease using medical and behavioral data?"

3. Data Presentation

3.1. Import Libraries

```r
# Install and import necessary libraries

# install.packages("ggcorrplot")

library(ggplot2)
library(tidyverse)
library(ggcorrplot)
```

3.2. Data collection

The dataset is publically available on the Kaggle website. It is collected from an ongoing cardiovascular study on residents of the town of Framingham, Massachusetts. The study, which aimed to unravel the underlying causes of heart disease, started in 1948 with 5,209 men and women between the ages of 30 and 62 from the town of Framingham, Massachusetts. The classification goal is to predict whether the patient has a 10-year risk of future coronary heart disease (CHD). New attributes were added over the years to form a good dataset (Framingham Heart Study, 2022).

3.3. Dataset description

The Dataset includes 4238 records and 15 attributes. Each attribute is a potential risk factor. There are both demographic, behavioral, and medical risk factors (Kaggle, 2022).

Demographic Variables:

- sex: Male or female (Nominal: 1 means "Male", 0 means "Female")
- age: Age of the patient (Continuous - Although the recorded ages have been truncated to whole numbers, the concept of age is continuous)

Behavioral Variables:

- currentSmoker: whether or not the patient is a current smoker (Nominal: 1 means "Yes", 0 means "No")
- cigsPerDay: the number of cigarettes that the person smoked on average in one day.(Continous)

History Medical Variables:

- BPMeds: whether or not the patient was on blood pressure medication (Nominal)
- preStroke: whether or not the patient had previously had a stroke (Nominal)
- preHyp: whether or not the patient was hypertensive (Nominal)
- diabetes: whether or not the patient had diabetes (Nominal)

Current Medical Variables:

- chol: total cholesterol level (Continuous)
- systolicBP: systolic blood pressure (Continuous)
- diastolicBP: diastolic blood pressure (Continuous)
- BMI: Body Mass Index (Continuous)
- heartRate: heart rate (Continuous - In medical research, variables such as heart rate though in fact discrete, yet are considered continuous because of large number of possible values.)
- glucose: glucose level (Continuous)

Predict variable (desired target)

- heartDisease: 10 year risk of heart disease (binary: "1" = "Yes", "0" = "No")

3.4. Data Summary

3.4.1. Loading dataset

```
# Get data from csv files

raw_heart_disease <- read.csv("heart_disease.csv", header = TRUE)
head(raw_heart_disease, n = 10)

##    sex age currentSmoker cigsPerDay BPMeds preStroke preHyp diabetes chol
## 1    1  39             0          0      0         0      0        0  195
## 2    0  46             0          0      0         0      0        0  250
## 3    1  48             1         20      0         0      0        0  245
## 4    0  61             1         30      0         0      1        0  225
## 5    0  46             1         23      0         0      0        0  285
## 6    0  43             0          0      0         0      1        0  228
## 7    0  63             0          0      0         0      0        0  205
## 8    0  45             1         20      0         0      0        0  313
## 9    1  52             0          0      0         0      1        0  260
## 10   1  43             1         30      0         0      1        0  225
##    systolicBP diastolicBP   BMI heartRate glucose heartDisease
## 1       106.0          70 26.97        80      77            0
## 2       121.0          81 28.73        95      76            0
## 3       127.5          80 25.34        75      70            0
## 4       150.0          95 28.58        65     103            1
## 5       130.0          84 23.10        85      85            0
## 6       180.0         110 30.30        77      99            0
## 7       138.0          71 33.11        60      85            1
## 8       100.0          71 21.68        79      78            0
## 9       141.5          89 26.36        76      79            0
## 10      162.0         107 23.61        93      88            0
```

3.4.2. Dataset Summary and Structure

```
# Dataset structure

str(raw_heart_disease)

## 'data.frame':    4238 obs. of  15 variables:
##  $ sex          : int  1 0 1 0 0 0 0 0 1 1 ...
##  $ age          : int  39 46 48 61 46 43 63 45 52 43 ...
##  $ currentSmoker: int  0 0 1 1 1 0 0 1 0 1 ...
##  $ cigsPerDay   : int  0 0 20 30 23 0 0 20 0 30 ...
##  $ BPMeds       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ preStroke    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ preHyp       : int  0 0 0 1 0 1 0 0 1 1 ...
##  $ diabetes     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ chol         : int  195 250 245 225 285 228 205 313 260 225 ...
##  $ systolicBP   : num  106 121 128 150 130 ...
##  $ diastolicBP  : num  70 81 80 95 84 110 71 71 89 107 ...
```

```
##  $ BMI         : num  27 28.7 25.3 28.6 23.1 ...
##  $ heartRate   : int  80 95 75 65 85 77 60 79 76 93 ...
##  $ glucose     : int  77 76 70 103 85 99 85 78 79 88 ...
##  $ heartDisease : int  0 0 0 1 0 0 1 0 0 0 ...
```

```
# Dataset summary

summary(raw_heart_disease)

##       sex              age         currentSmoker      cigsPerDay
##  Min.   :0.0000   Min.   :32.00   Min.   :0.0000   Min.   : 0.000
##  1st Qu.:0.0000   1st Qu.:42.00   1st Qu.:0.0000   1st Qu.: 0.000
##  Median :0.0000   Median :49.00   Median :0.0000   Median : 0.000
##  Mean   :0.4292   Mean   :49.58   Mean   :0.4941   Mean   : 9.003
##  3rd Qu.:1.0000   3rd Qu.:56.00   3rd Qu.:1.0000   3rd Qu.:20.000
##  Max.   :1.0000   Max.   :70.00   Max.   :1.0000   Max.   :70.000
##                                                    NA's   :29
##      BPMeds          preStroke           preHyp          diabetes
##  Min.   :0.00000   Min.   :0.000000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median :0.00000   Median :0.000000   Median :0.0000   Median :0.00000
##  Mean   :0.02963   Mean   :0.005899   Mean   :0.3105   Mean   :0.02572
##  3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:1.0000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.000000   Max.   :1.0000   Max.   :1.00000
##  NA's   :53
##       chol          systolicBP     diastolicBP          BMI
##  Min.   :107.0   Min.   : 83.5   Min.   : 48.00   Min.   :15.54
##  1st Qu.:206.0   1st Qu.:117.0   1st Qu.: 75.00   1st Qu.:23.07
##  Median :234.0   Median :128.0   Median : 82.00   Median :25.40
##  Mean   :236.7   Mean   :132.4   Mean   : 82.89   Mean   :25.80
##  3rd Qu.:263.0   3rd Qu.:144.0   3rd Qu.: 89.88   3rd Qu.:28.04
##  Max.   :696.0   Max.   :295.0   Max.   :142.50   Max.   :56.80
##  NA's   :50                                       NA's   :19
##    heartRate          glucose         heartDisease
##  Min.   : 44.00   Min.   : 40.00   Min.   :0.000
##  1st Qu.: 68.00   1st Qu.: 71.00   1st Qu.:0.000
##  Median : 75.00   Median : 78.00   Median :0.000
##  Mean   : 75.88   Mean   : 81.97   Mean   :0.152
##  3rd Qu.: 83.00   3rd Qu.: 87.00   3rd Qu.:0.000
##  Max.   :143.00   Max.   :394.00   Max.   :1.000
##  NA's   :1        NA's   :388
```

From the Dataset structure and Dataset Summary, we can see that there are NA values in the dataset that need to be handled.

We also need to transform character variables to factor before EDA and modeling phases.

4. Exploratory Data Analysis

4.1. Create and Transform Variables for EDA

4.1.1. Create new variables

```
# Add ageGroup attributes
# Transform binominal variables (sex, currentSmoker, BPMeds, preStroke, preHy
p, diabetes, heartDisease) to meaningful strings

df_heart_eda <- raw_heart_disease %>%
  mutate(sex = ifelse(sex == 1, "Male","Female")) %>%
  mutate(currentSmoker = ifelse(currentSmoker == 1, "Yes","No")) %>%
  mutate(BPMeds = ifelse(BPMeds == 1, "Yes","No")) %>%
  mutate(preStroke = ifelse(preStroke == 1, "Yes","No")) %>%
  mutate(preHyp = ifelse(preHyp == 1, "Yes","No")) %>%
  mutate(diabetes = ifelse(diabetes ==1, "Yes","No")) %>%
  mutate(heartDisease = ifelse(heartDisease == 1, "Yes","No")) %>%
  mutate(ageGroup = ifelse(age %in% 30:39, "30-39", ifelse(age %in% 40:49, "4
0-49", ifelse(age %in% 50:59, "50-59", "60+"))))
```

4.1.2. Tranform variables to factor

```
# Transform nominal variables to factor

df_heart_eda <- df_heart_eda %>%
  mutate(sex = as.factor(sex)) %>%
  mutate(currentSmoker = as.factor(currentSmoker)) %>%
  mutate(BPMeds = as.factor(BPMeds)) %>%
  mutate(preStroke = as.factor(preStroke)) %>%
  mutate(preHyp = as.factor(preHyp)) %>%
  mutate(diabetes = as.factor(diabetes)) %>%
  mutate(heartDisease = as.factor(heartDisease)) %>%
  mutate(ageGroup = as.factor(ageGroup))

# Verify dataset structure for eda

str(df_heart_eda)

## 'data.frame':    4238 obs. of  16 variables:
##  $ sex          : Factor w/ 2 levels "Female","Male": 2 1 2 1 1 1 1 1 2 2
...
##  $ age          : int  39 46 48 61 46 43 63 45 52 43 ...
##  $ currentSmoker: Factor w/ 2 levels "No","Yes": 1 1 2 2 2 1 1 2 1 2 ...
##  $ cigsPerDay   : int  0 0 20 30 23 0 0 20 0 30 ...
##  $ BPMeds       : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ preStroke    : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ preHyp       : Factor w/ 2 levels "No","Yes": 1 1 1 2 1 2 1 1 2 2 ...
##  $ diabetes     : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ chol         : int  195 250 245 225 285 228 205 313 260 225 ...
```

```
##  $ systolicBP   : num  106 121 128 150 130 ...
##  $ diastolicBP  : num  70 81 80 95 84 110 71 71 89 107 ...
##  $ BMI          : num  27 28.7 25.3 28.6 23.1 ...
##  $ heartRate    : int  80 95 75 65 85 77 60 79 76 93 ...
##  $ glucose      : int  77 76 70 103 85 99 85 78 79 88 ...
##  $ heartDisease : Factor w/ 2 levels "No","Yes": 1 1 1 2 1 1 2 1 1 1 ...
##  $ ageGroup     : Factor w/ 4 levels "30-39","40-49",..: 1 2 2 4 2 2 4 2 3
2 ...
```

4.2. Data Quality

4.2.1. Duplicated Values

```
# Check duplicated values

nrow(df_heart_eda) - nrow(df_heart_eda %>% distinct())

## [1] 0
```

There is no duplicated values in the dataset.

4.2.2. NA Values

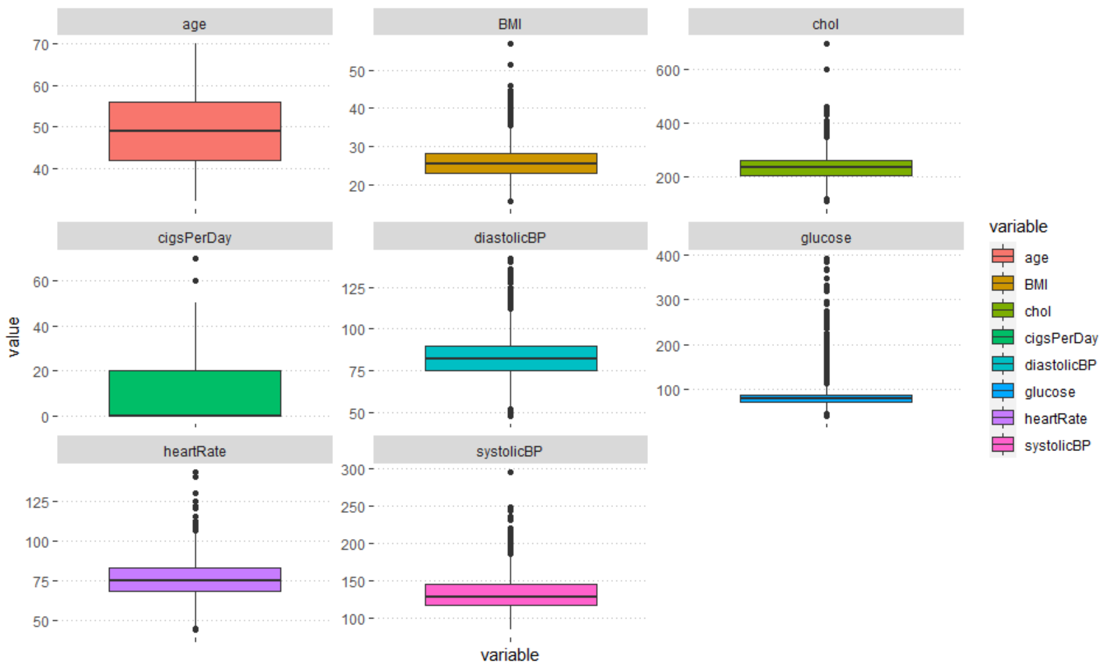
```
# Check null values in attributes

colSums(is.na(df_heart_eda))

##           sex           age currentSmoker    cigsPerDay        BPMeds
##             0             0             0            29            53
##     preStroke        preHyp      diabetes          chol    systolicBP
##             0             0             0            50             0
##   diastolicBP           BMI     heartRate       glucose  heartDisease
##             0            19             1           388             0
##      ageGroup
##             0
```

The attributes containing NA values are:

- BPMeds: whether or not the patient was on blood pressure medication in history (Nominal)
- cigsPerDay: the number of cigarettes that the person smoked on average in one day (Continous)
- chol: total cholesterol level (Continuous)
- BMI: Body Mass Index (Continuous)
- heartRate: heart rate (Continuous)
- glucose: glucose level (Continuous)

4.2.3. Outliers

```
# Check outliers of continous variables

df_heart_eda %>%
  select(age, cigsPerDay, chol, systolicBP, diastolicBP, BMI, heartRate, gluc
ose) %>%
  pivot_longer(c("age", "cigsPerDay", "chol","systolicBP", "diastolicBP", "BM
I", "heartRate", "glucose")
               ,names_to = 'variable', values_to = 'value') %>%
  ggplot(aes(x=variable,y=value, fill = variable)) + geom_boxplot() +
  facet_wrap(facets = ~variable, scales = "free") +
  theme(
    panel.grid.major.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.y = element_blank(),
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank(),
    panel.background = element_blank(),
    axis.text.x = element_blank(),
    )
```

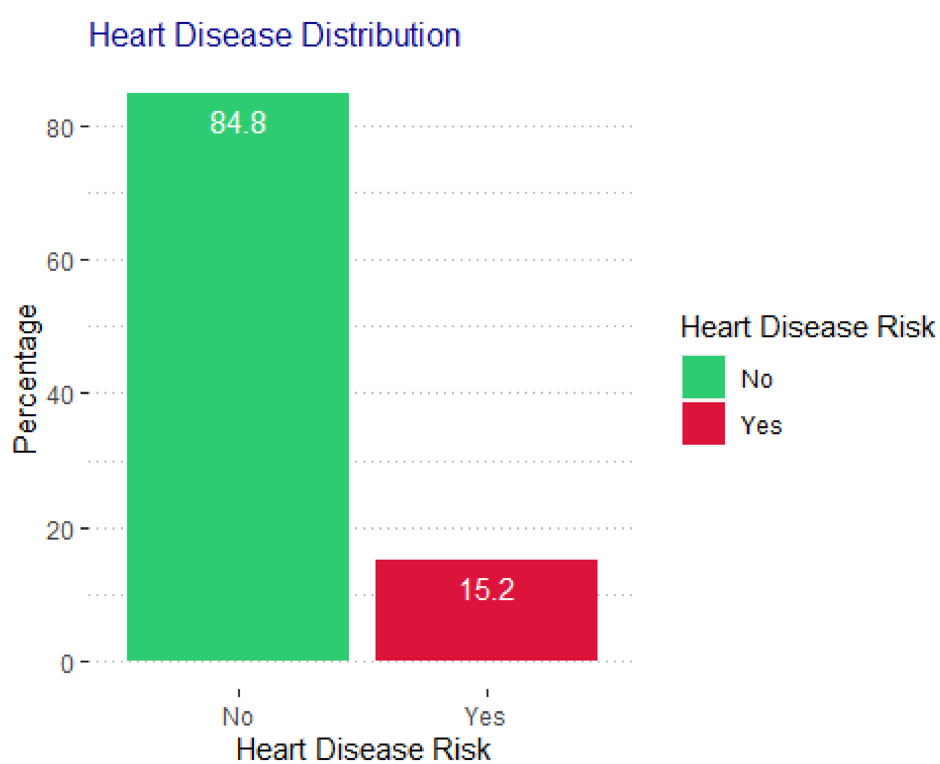


Except for age, the remaining continuous variables have a lot of outliers. This outlier examination is critical. It will help us define which method should be used to handle NA values in part 5. Data Preprocessing.

4.3. Variable Distribution

4.3.1. Heart Disease Risk Distribution

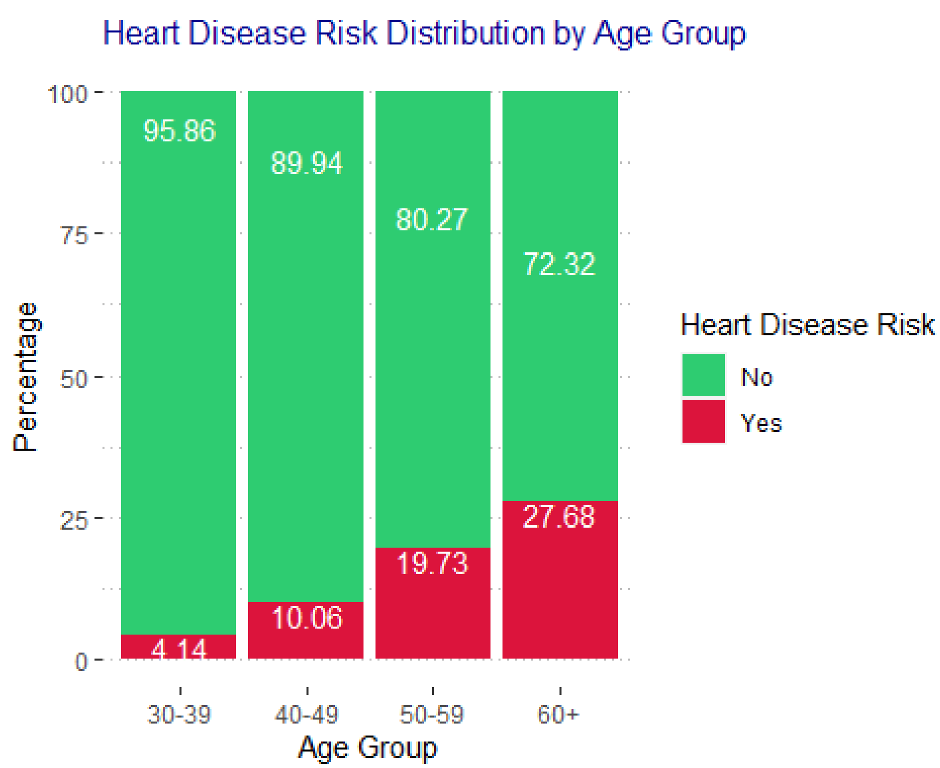
```
# Heart Disease Risk Distribution

df_heart_eda %>%
  group_by(heartDisease) %>%
  summarise(n = n()) %>%
  mutate(freq = round(n*100/sum(n),2)) %>%
  ggplot(aes(heartDisease, freq, fill = heartDisease)) + geom_col() +
  geom_text(aes(label=freq), vjust=1.6, color="white", size=4) +
  scale_fill_manual(values=c("#2ECC71", "#DC143C")) +
  labs(title="Heart Disease Distribution",x="Heart Disease Risk", y = "Percen
tage", fill = "Heart Disease Risk") +
  theme(
    panel.grid.major.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank(),
    panel.background = element_blank(),
    plot.title = element_text(color="darkblue", size=12)
    )
```



We can see that the dataset is imbalanced. This affects the method we will use to replace the NA values in part 5. Data Preprocessing.

4.3.2. Heart Disease Risk Distribution by Age Group

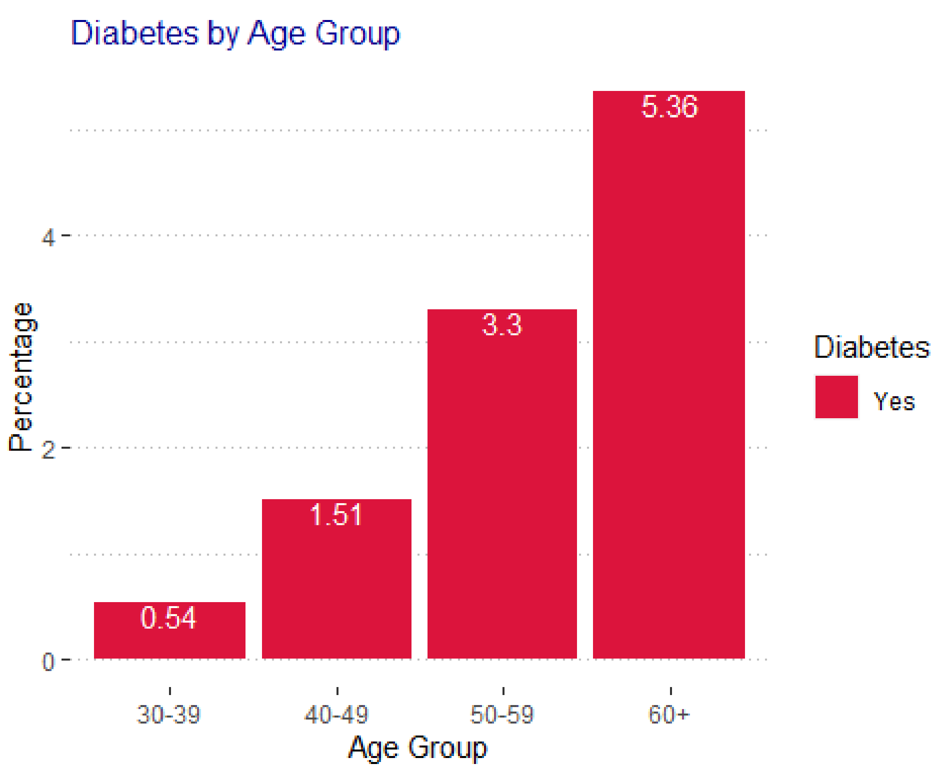
```
# Heart Disease Risk Distribution by Age Group

df_heart_eda %>%
  group_by(ageGroup, heartDisease) %>%
  summarise(n = n()) %>%
  mutate(freq = round(n*100/sum(n),2)) %>%
  ggplot(aes(ageGroup, freq, fill = heartDisease)) + geom_col() +
  geom_text(aes(label=freq), vjust=1, color="white", size=4) +
  labs(title="Heart Disease Risk Distribution by Age Group",x="Age Group", y
= "Percentage"
       , color = "Heart Disease Risk", fill = "Heart Disease Risk") +
  scale_fill_manual(values=c("#2ECC71", "#DC143C")) +
  theme(
    panel.grid.major.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank(),
    panel.background = element_blank(),
    plot.title = element_text(color="darkblue", size=12)
    )
```



The chart reveals that the older the patients are, the higher risk of Heart Disease they get.

4.3.3. Diabetes by Age Group

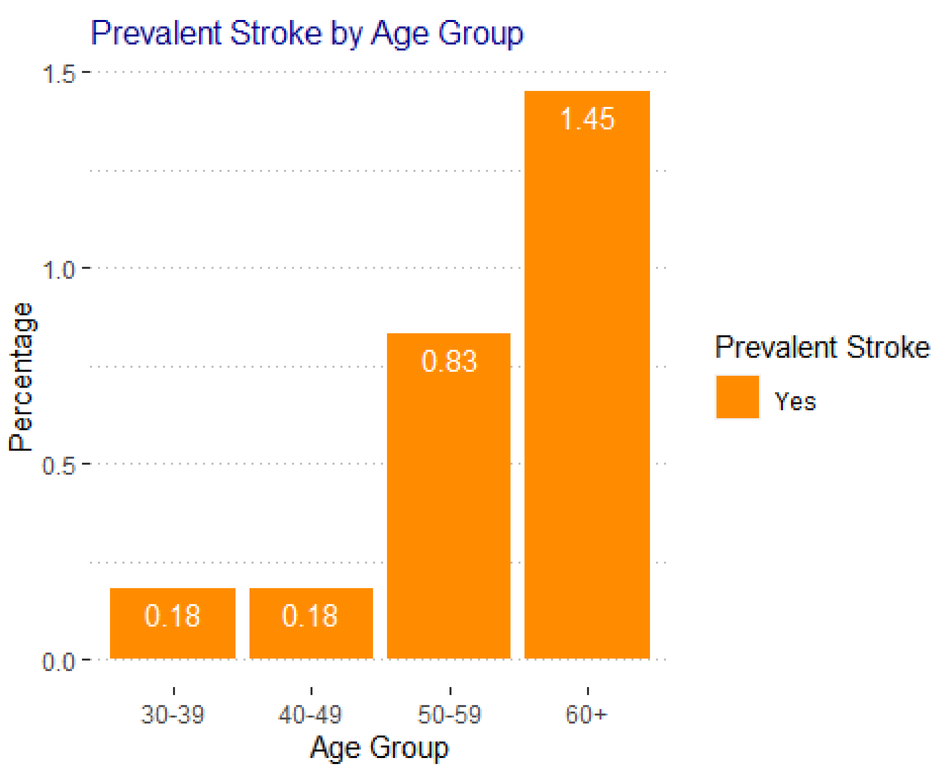
```
# Diabetes by Age Group

df_heart_eda %>%
  group_by(ageGroup, diabetes) %>%
  summarise(n = n()) %>%
  mutate(freq = round(n*100/sum(n),2)) %>%
  filter(diabetes == "Yes") %>%
  ggplot(aes(ageGroup, freq, fill = diabetes)) + geom_col() +
  geom_text(aes(label=freq), vjust=1, color="white", size=4) +
  labs(title="Diabetes by Age Group",x="Age Group", y = "Percentage", color =
"Diabetes", fill = "Diabetes") +
  scale_fill_manual(values=c("#DC143C")) +
  theme(
    panel.grid.major.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank(),
    panel.background = element_blank(),
    plot.title = element_text(color="darkblue", size=12)
    )
```



A higher percentage of Diabetes is also seen in the older age group.

4.3.4. Prevalent Stroke by Age Group

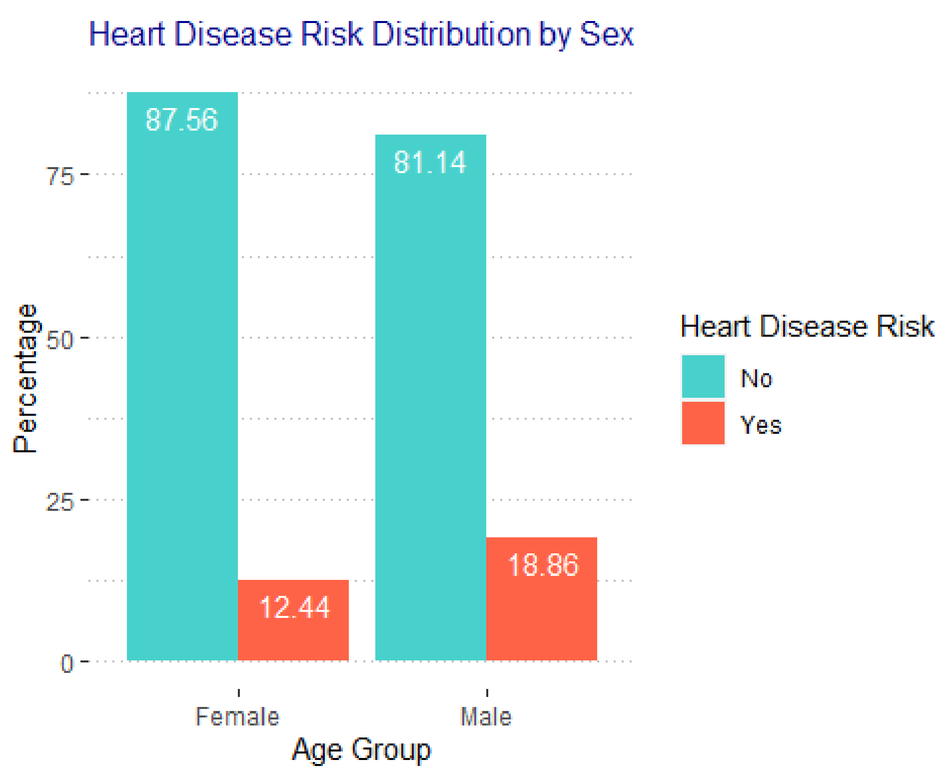
```
# Prevalent Stroke by Age Group

df_heart_eda %>%
  group_by(ageGroup, preStroke) %>%
  summarise(n = n()) %>%
  mutate(freq = round(n*100/sum(n),2)) %>%
  filter(preStroke == "Yes") %>%
  ggplot(aes(ageGroup, freq, fill = preStroke)) + geom_col() +
  geom_text(aes(label=freq), vjust=1.5, color="white", size=4) +
  labs(title="Prevalent Stroke by Age Group",x="Age Group", y = "Percentage"
       , color = "Prevalent Stroke", fill = "Prevalent Stroke") +
  scale_fill_manual(values=c( "#FF8C00")) +
  theme(
    panel.grid.major.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank(),
    panel.background = element_blank(),
    plot.title = element_text(color="darkblue", size=12)
    )
```



As the same as Diabetes, A higher percentage of Prevalent Stroke is also seen in the older age group. The risks of Heart Disease, Diabetes, and Stroke are severe and dangerous to the old. People should take care of their health when they get older.

4.3.5. Heart Disease Risk Distribution by Sex

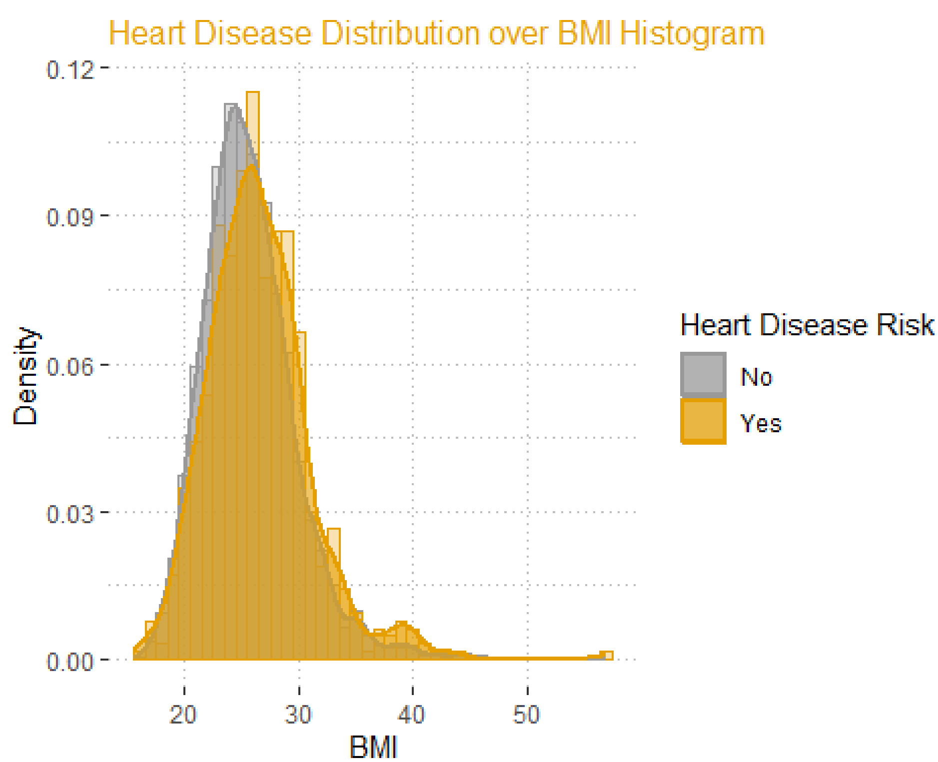
```
# Heart Disease Risk Distribution by Sex

df_heart_eda %>%
  group_by(sex, heartDisease) %>%
  summarise(n = n()) %>%
  mutate(freq = round(n*100/sum(n),2)) %>%
  ggplot(aes(sex, freq, fill = heartDisease)) + geom_bar(stat="identity", pos
ition=position_dodge()) +
  geom_text(aes(label=freq), vjust=1.5, color="white", position = position_do
dge(0.9), size=4) +
  labs(title="Heart Disease Risk Distribution by Sex",x="Age Group", y = "Per
centage"
       , color = "Heart Disease Risk", fill = "Heart Disease Risk") +
  scale_fill_manual(values=c("#48D1CC", "#FF6347")) +
  theme(
    panel.grid.major.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank(),
    panel.background = element_blank(),
    plot.title = element_text(color="darkblue", size=12)
    )
```



The percentage of Heart Disease risk is slightly higher in the Male group than the number in the Female group.

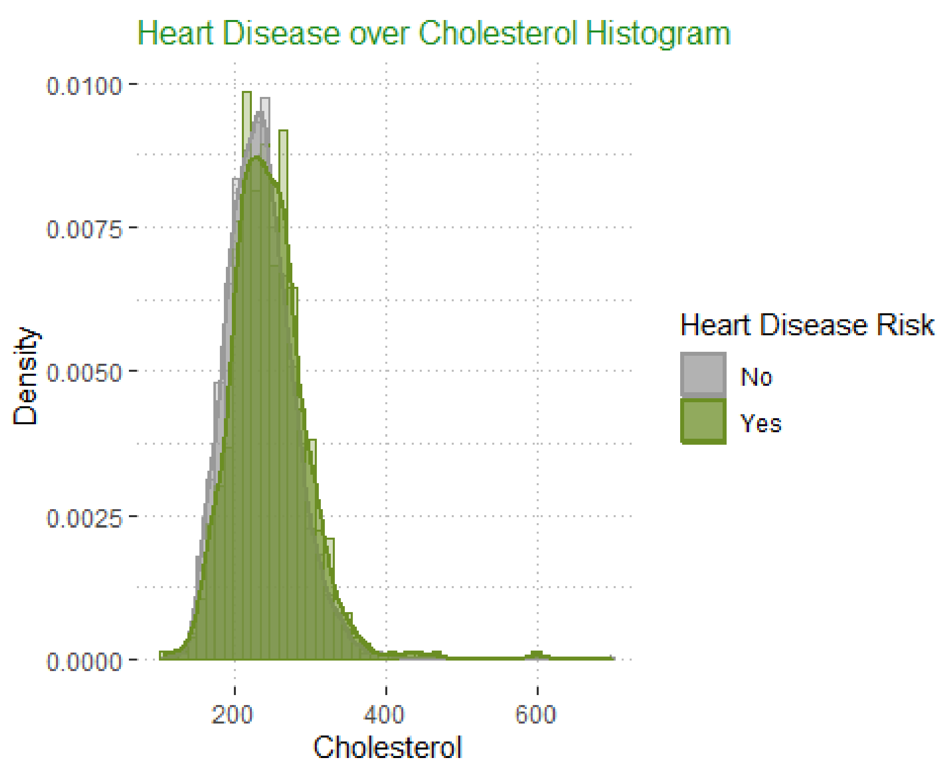
4.3.6. Heart Disease Distribution over BMI

```
# Heart Disease Distribution over BMI Histogram

df_heart_eda %>%
  filter(! is.na(BMI)) %>%
  ggplot(aes(x=BMI, fill=heartDisease, color = heartDisease)) +
  geom_histogram(aes(y=..density..), position="identity", binwidth = 1, alpha
= 0.3) +
  geom_density(lwd = 0.75, alpha = 0.6) +
  scale_color_manual(values=c("#999999", "#E69F00")) +
  scale_fill_manual(values=c("#999999", "#E69F00")) +
  labs(title="Heart Disease Distribution over BMI Histogram",x="BMI", y = "De
nsity"
       , color = "Heart Disease Risk", fill = "Heart Disease Risk") +
  theme(
    panel.grid.major.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.major.x = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.x = element_blank(),
    panel.background = element_blank(),
    plot.title = element_text(color="#E69F00", size=12)
    )
```



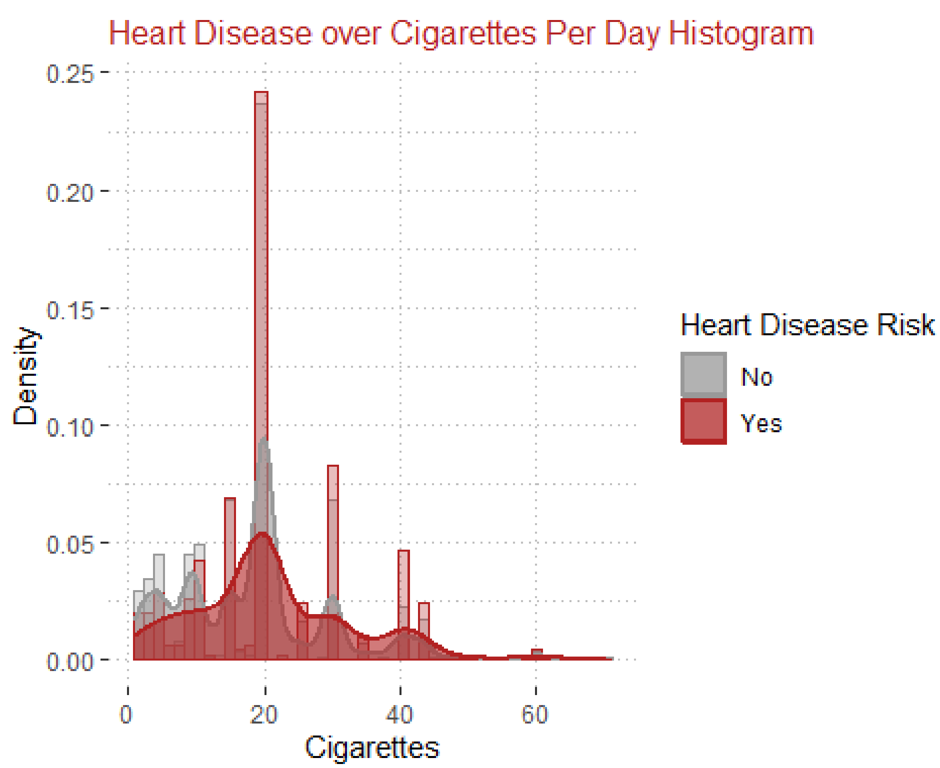
4.3.7. Heart Disease over Cholesterol

```
# Heart Disease over Cholesterol

df_heart_eda %>%
  filter(! is.na(chol)) %>%
  ggplot(aes(x=chol, fill=heartDisease, color = heartDisease)) +
  geom_histogram(aes(y=..density..), position="identity", binwidth = 12, alph
a = 0.3) +
  geom_density(lwd = 0.75, alpha = 0.6) +
  scale_color_manual(values=c("#999999", "#6B8E23")) +
  scale_fill_manual(values=c("#999999", "#6B8E23")) +
  labs(title="Heart Disease over Cholesterol Histogram",x="Cholesterol", y =
"Density"
       , color = "Heart Disease Risk", fill = "Heart Disease Risk") +
  theme(
    panel.grid.major.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.major.x = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.x = element_blank(),
    panel.background = element_blank(),
    plot.title = element_text(color="#228B22", size=12)
    )
```



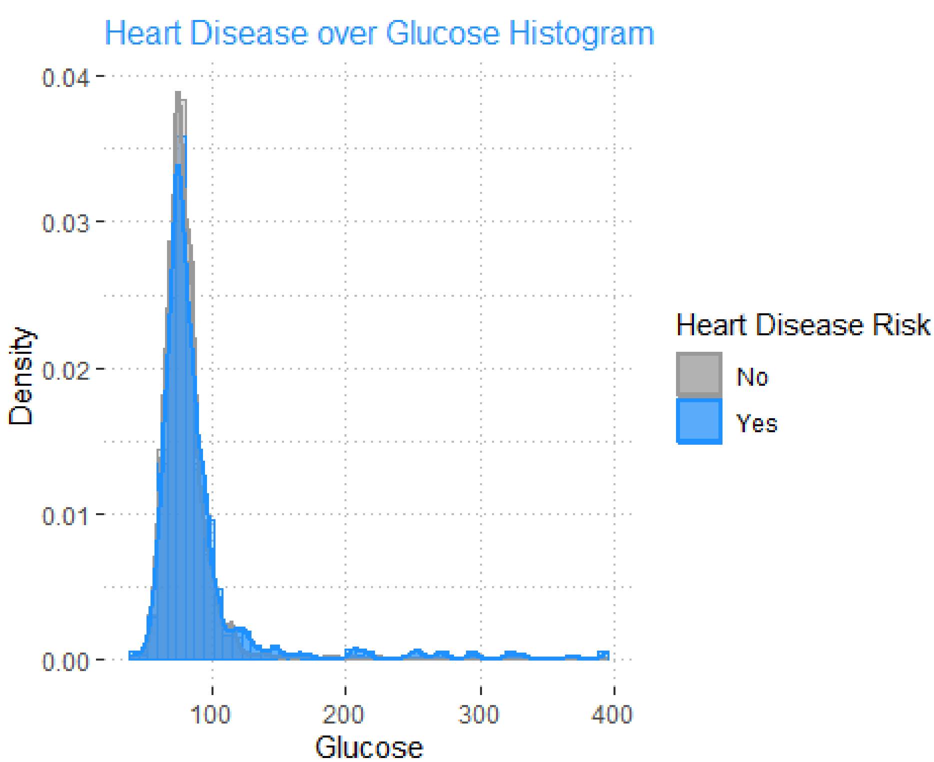
4.3.8. Heart Disease over Cigarettes Per Day

```
# Heart Disease over Cigarettes Per Day

df_heart_eda %>%
  filter(! is.na(cigsPerDay)  & cigsPerDay > 0) %>%
  ggplot(aes(x=cigsPerDay, fill=heartDisease, color = heartDisease)) +
  geom_histogram(aes(y=..density..), position="identity", binwidth = 1.5, alp
ha = 0.3) +
  geom_density(lwd = 0.75, alpha = 0.6) +
  scale_color_manual(values=c("#999999", "#B22222")) +
  scale_fill_manual(values=c("#999999", "#B22222")) +
  labs(title="Heart Disease over Cigarettes Per Day Histogram",x="Cigarettes"
, y = "Density"
       , color = "Heart Disease Risk", fill = "Heart Disease Risk") +
  theme(
    panel.grid.major.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.major.x = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.x = element_blank(),
    panel.background = element_blank(),
    plot.title = element_text(color="#B22222", size=12)
    )
```



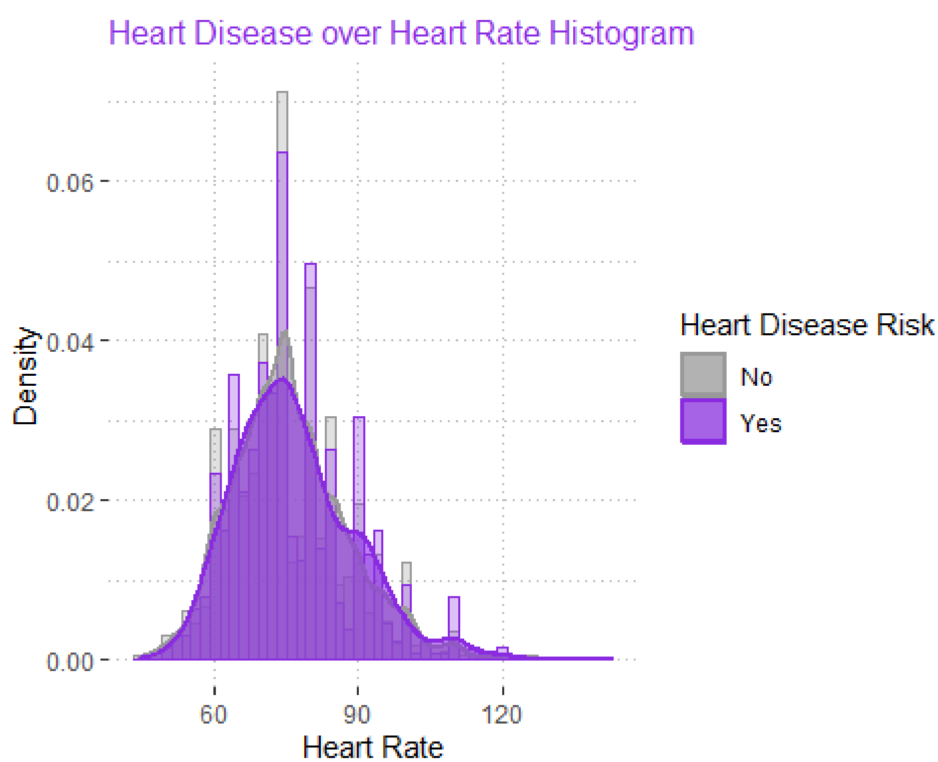
4.3.9. Heart Disease over Glucose level

```
# Heart Disease over Glucose

df_heart_eda %>%
  filter(! is.na(glucose)) %>%
  ggplot(aes(x=glucose, fill=heartDisease, color = heartDisease)) +
  geom_histogram(aes(y=..density..), position="identity", binwidth = 7, alpha
= 0.3) +
  geom_density(lwd = 0.75, alpha = 0.6) +
  scale_color_manual(values=c("#999999", "#1E90FF")) +
  scale_fill_manual(values=c("#999999", "#1E90FF")) +
  labs(title="Heart Disease over Glucose Histogram",x="Glucose", y = "Density
"
       , color = "Heart Disease Risk", fill = "Heart Disease Risk") +
  theme(
    panel.grid.major.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.major.x = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.x = element_blank(),
    panel.background = element_blank(),
    plot.title = element_text(color="#1E90FF", size=12)
    )
```



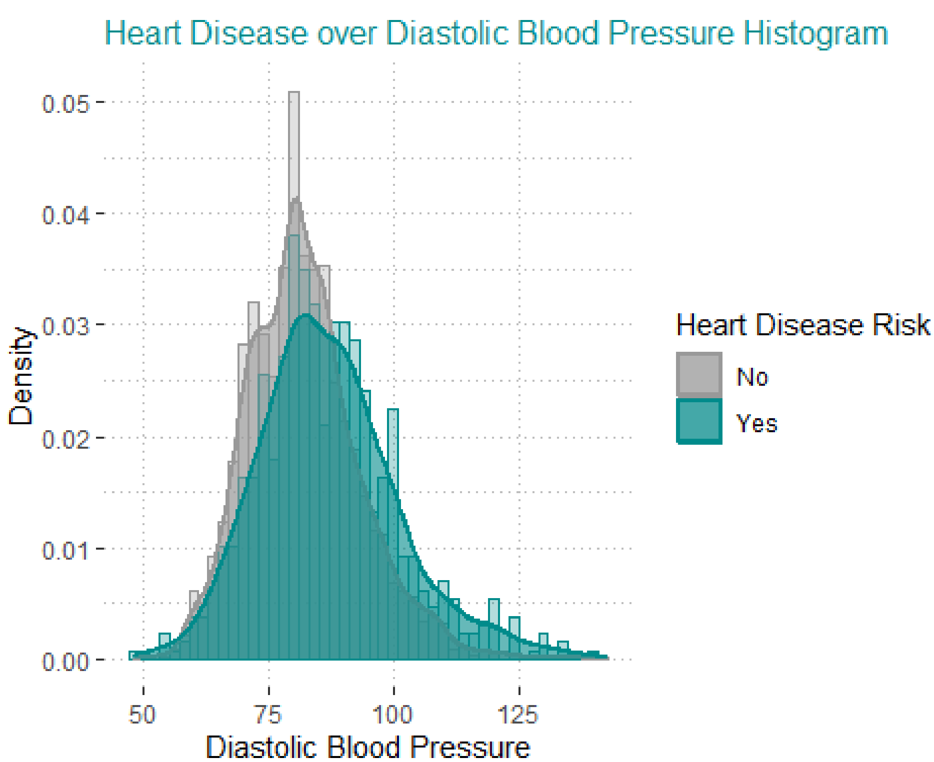
4.3.10. Heart Disease over Heart Rate

```
# Heart Disease over Heart Rate

df_heart_eda %>%
  filter(! is.na(heartRate)) %>%
  ggplot(aes(x=heartRate, fill=heartDisease, color = heartDisease)) +
  geom_histogram(aes(y=..density..), position="identity", binwidth = 2, alpha
= 0.3) +
  geom_density(lwd = 0.75, alpha = 0.6) +
  scale_color_manual(values=c("#999999", "#8A2BE2")) +
  scale_fill_manual(values=c("#999999", "#8A2BE2")) +
  labs(title="Heart Disease over Heart Rate Histogram",x="Heart Rate", y = "D
ensity"
       , color = "Heart Disease Risk", fill = "Heart Disease Risk") +
  theme(
    panel.grid.major.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.major.x = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.x = element_blank(),
    panel.background = element_blank(),
    plot.title = element_text(color="#8A2BE2", size=12)
    )
```



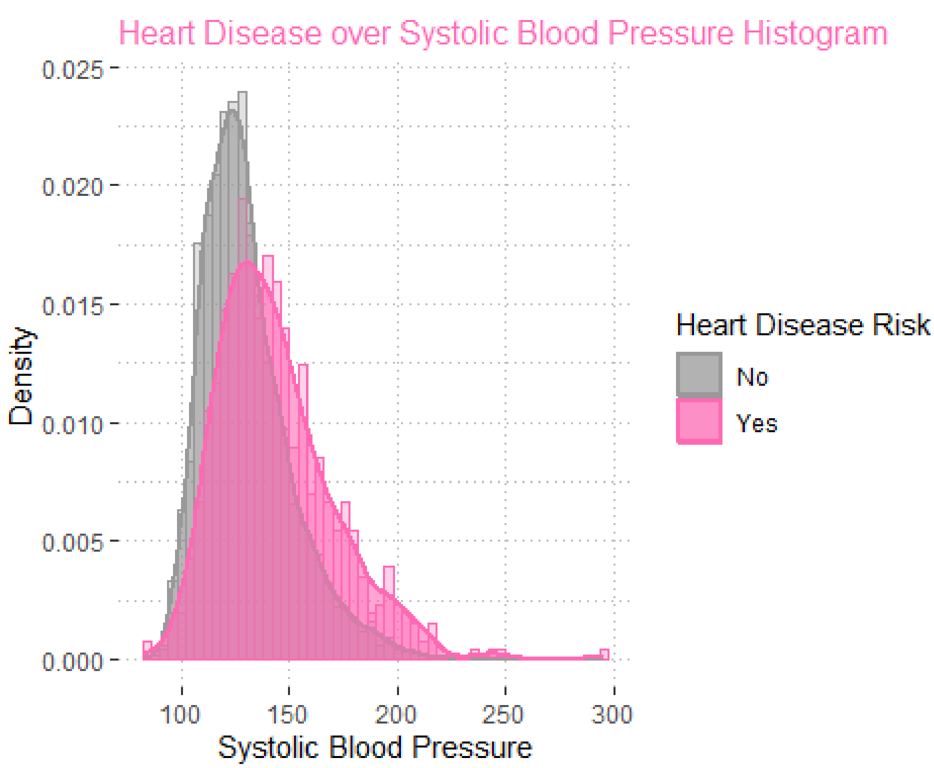
4.3.11. Heart Disease over Diastolic Blood Pressure

```
# Heart Disease over Diastolic Blood Pressure

df_heart_eda %>%
  ggplot(aes(x=diastolicBP, fill=heartDisease, color = heartDisease)) +
  geom_histogram(aes(y=..density..), position="identity", binwidth = 2, alpha
= 0.3) +
  geom_density(lwd = 0.75, alpha = 0.6) +
  scale_color_manual(values=c("#999999", "#008B8B")) +
  scale_fill_manual(values=c("#999999", "#008B8B")) +
  labs(title="Heart Disease over Diastolic Blood Pressure Histogram",x="Diast
olic Blood Pressure"
       , y = "Density", color = "Heart Disease Risk", fill = "Heart Disease R
isk") +
  theme(
    panel.grid.major.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.major.x = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.x = element_blank(),
    panel.background = element_blank(),
    plot.title = element_text(color="#008B8B", size=12)
    )
```



4.3.12. Heart Disease over Systolic Blood Pressure

```
# Heart Disease over Systolic Blood Pressure

df_heart_eda %>%
  ggplot(aes(x=systolicBP, fill=heartDisease, color = heartDisease)) +
  geom_histogram(aes(y=..density..), position="identity", binwidth = 4, alpha
= 0.3) +
  geom_density(lwd = 0.75, alpha = 0.6) +
  scale_color_manual(values=c("#999999", "#FF69B4")) +
  scale_fill_manual(values=c("#999999", "#FF69B4")) +
  labs(title="Heart Disease over Systolic Blood Pressure Histogram",x="Systol
ic Blood Pressure"
       , y = "Density", color = "Heart Disease Risk", fill = "Heart Disease R
isk") +
  theme(
    panel.grid.major.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.major.x = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.x = element_blank(),
    panel.background = element_blank(),
    plot.title = element_text(color="#FF69B4", size=12)
    )
```



The histograms show that the patients, who have a high level of Cholesterol, a high number of Cigarettes per day, and a high level of Diastolic Blood Pressure and Systolic Blood Pressure, are more sensitive to Heart Disease than those who do not.

One more important reason why we should use a histogram with a density curve to visualize the distribution of Heart Disease Risk (Yes or No) over independent variables is that we will see whether there is a bias in any attribute in the dataset, e.g., Heart Disease is only seen in Male.

In the next sections, we will do statistical analysis to examine what we have seen from EDA. We also apply Logistic Regression to build a model for Heart Disease prediction. To do these, we need to do Data Preprocessing to get data ready.

5. Data Preprocessing

5.1. Data Cleaning

In this section, we will handle missing values before modeling. A new data frame, df_heart_clean, will be created for processing not to affect the raw data if we need to verify anything.

```
# Create new data frame

df_heart_clean <- raw_heart_disease
```

5.1.1. Verify NA Values

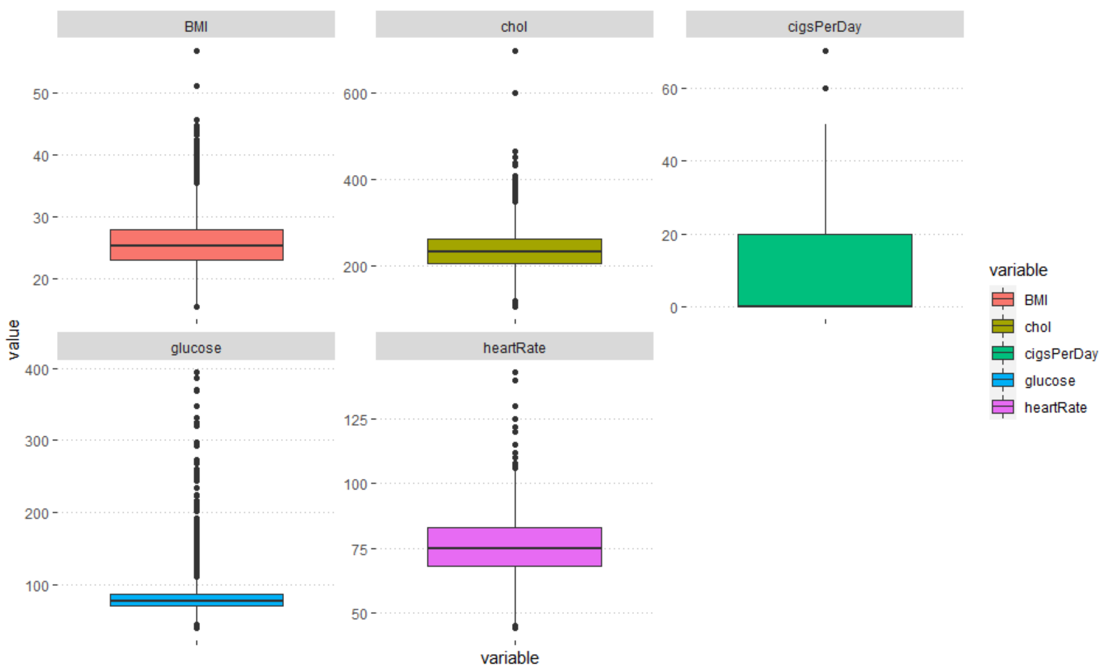
```
# Verify null values again

colSums(is.na(df_heart_clean))

##           sex           age currentSmoker    cigsPerDay        BPMeds
##             0             0             0            29            53
##     preStroke        preHyp      diabetes          chol    systolicBP
##             0             0             0            50             0
##   diastolicBP           BMI     heartRate       glucose  heartDisease
##             0            19             1           388             0
```

The attributes containing NA values are:

- BPMeds: whether or not the patient was on blood pressure medication in history (Nominal)
- cigsPerDay: the number of cigarettes that the person smoked on average in one day (Continous)
- chol: total cholesterol level (Continuous)
- BMI: Body Mass Index (Continuous)
- heartRate: heart rate (Continuous)
- glucose: glucose level (Continuous)

5.1.2. Verify Outliers

```
# Check outliers of continous variables

df_heart_clean %>%
  select(cigsPerDay, chol, BMI, heartRate, glucose) %>%
  pivot_longer(c("cigsPerDay", "chol", "BMI", "heartRate", "glucose"),names_t
o = 'variable', values_to = 'value') %>%
  ggplot(aes(x=variable,y=value, fill = variable)) + geom_boxplot() + facet_w
rap(facets = ~variable, scales = "free") + theme(
    panel.grid.major.y = element_line(colour = "gray", linetype = "dotted"),
    panel.grid.minor.y = element_blank(),
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank(),
    panel.background = element_blank(),
    axis.text.x = element_blank(),
    )
```



Regarding BPMeds, it is a nominal variable. We will replace NA values with the mode.

CigsPerDay, Chol, BMI, HeartRate, and Glucose are continuous variables. We can replace missing values using mean or median.

We will replace missing CigsPerDay values with the mean. However, Chol, BMI, HeartRate, and Glucose have a lot of outliers. Because the mean is sensitive to outliers, we will replace these missing values with the median.

From the EDA part, we know that the dataset is imbalanced. So, we will define the mean and median values based on the proportion of Heart Disease.

5.1.3. Mode Function

```
# Get mode function

getmode <- function(v) {
 uniqv <- unique(v)
 uniqv[which.max(tabulate(match(v, uniqv)))]
}
```

5.1.4. Define mean and median

```
# Define mean and median values

meanCigsPerDay0 <- mean(df_heart_clean$cigsPerDay[df_heart_clean$heartDisease
==0], na.rm = TRUE)
meanCigsPerDay1 <- mean(df_heart_clean$cigsPerDay[df_heart_clean$heartDisease
==1], na.rm = TRUE)

medianChol0 <- median(df_heart_clean$chol[df_heart_clean$heartDisease==0], na
.rm = TRUE)
medianChol1 <- median(df_heart_clean$chol[df_heart_clean$heartDisease==1], na
.rm = TRUE)

medianBMI0 <- median(df_heart_clean$BMI[df_heart_clean$heartDisease==0], na.r
m = TRUE)
medianBMI1 <- median(df_heart_clean$BMI[df_heart_clean$heartDisease==1], na.r
m = TRUE)

medianHeartRate0 <- median(df_heart_clean$heartRate[df_heart_clean$heartDisea
se==0], na.rm = TRUE)
medianHeartRate1 <- median(df_heart_clean$heartRate[df_heart_clean$heartDisea
se==1], na.rm = TRUE)

medianGlucose0 <- median(df_heart_clean$glucose[df_heart_clean$heartDisease==
0], na.rm = TRUE)
medianGlucose1 <- median(df_heart_clean$glucose[df_heart_clean$heartDisease==
1], na.rm = TRUE)
```

5.1.5. Handle NA Values

```
# Replace NA values by mean and median values

df_heart_clean <- df_heart_clean %>%
  mutate(BPMeds = ifelse(is.na(BPMeds), getmode(BPMeds), BPMeds)) %>%
  mutate(cigsPerDay = ifelse(is.na(cigsPerDay), ifelse(heartDisease == 0, mea
nCigsPerDay0, meanCigsPerDay1), cigsPerDay)) %>%
  mutate(chol = ifelse(is.na(chol), ifelse(heartDisease == 0, medianChol0, me
dianChol1), chol)) %>%
  mutate(BMI = ifelse(is.na(BMI), ifelse(heartDisease == 0, medianBMI0, media
nBMI1), BMI)) %>%
```

```
  mutate(heartRate = ifelse(is.na(heartRate), ifelse(heartDisease == 0, media
nHeartRate0, medianHeartRate1), heartRate)) %>%
  mutate(glucose = ifelse(is.na(glucose), ifelse(heartDisease == 0, medianGlu
cose0, medianGlucose1), glucose))

# Verify NA values again

colSums(is.na(df_heart_clean))

##          sex          age currentSmoker    cigsPerDay       BPMeds
##            0            0             0             0            0
##    preStroke       preHyp      diabetes          chol  systolicBP
##            0            0             0             0            0
##  diastolicBP          BMI     heartRate       glucose heartDisease
##            0            0             0             0            0
```

There is no NA values so far. We have a clean dataset for statistical analysis and modeling.

5.2. Correlation Matrix

This part will create the Correlation Heat Map to see the correlation between variables. We created the Correlation Matrix before Feature Engineering because the Correlation matrix is only applied for numeric values.

```
# Correlation Matrix

corrmatrix = cor(df_heart_clean)
corrmatrix

##                       sex         age currentSmoker  cigsPerDay       BPMe
ds
## sex           1.000000000 -0.02897864    0.19759647  0.31679720 -0.051544
97
## age          -0.028978639  1.00000000   -0.21374795 -0.19236540  0.120954
92
## currentSmoker  0.197596474 -0.21374795    1.00000000  0.76687191 -0.048358
46
## cigsPerDay    0.316797198 -0.19236540    0.76687191  1.00000000 -0.045646
26
## BPMeds       -0.051544968  0.12095492   -0.04835846 -0.04564626  1.000000
00
## preStroke    -0.004546327  0.05765482   -0.03298779 -0.03269916  0.114608
66
## preHyp        0.005313349  0.30719408   -0.10325974 -0.06589065  0.258696
68
## diabetes      0.015707987  0.10125769   -0.04429512 -0.03704810  0.051394
33
## chol         -0.069415283  0.25985130   -0.04654549 -0.02632617  0.078686
75
## systolicBP   -0.035989265  0.39430154   -0.13023012 -0.08844549  0.251502
93
## diastolicBP   0.057933469  0.20610399   -0.10774649 -0.05631854  0.192355
```

```
53
## BMI            0.081465097  0.13559390   -0.16732202 -0.09230960  0.099780
30
## heartRate     -0.116621089 -0.01284772    0.06233048  0.07487699  0.015142
26
## glucose        0.010001489  0.11790575   -0.05509233 -0.05623196  0.049282
50
## heartDisease   0.088427567  0.22525610    0.01945627  0.05799354  0.086417
14
##                  preStroke       preHyp     diabetes          chol  systo
licBP
## sex           -0.0045463266  0.005313349  0.015707987 -0.0694152831 -0.035
98927
## age            0.0576548158  0.307194077  0.101257689  0.2598512988  0.394
30154
## currentSmoker -0.0329877865 -0.103259740 -0.044295121 -0.0465454926 -0.130
23012
## cigsPerDay    -0.0326991572 -0.065890654 -0.037048100 -0.0263261733 -0.088
44549
## BPMeds         0.1146086635  0.258696683  0.051394329  0.0786867546  0.251
50293
## preStroke      1.0000000000  0.074829673  0.006949243  0.0001303577  0.057
00872
## preHyp         0.0748296728  1.000000000  0.077808409  0.1631838410  0.696
75477
## diabetes       0.0069492431  0.077808409  1.000000000  0.0403670802  0.111
28343
## chol           0.0001303577  0.163183841  0.040367080  1.0000000000  0.207
56702
## systolicBP     0.0570087220  0.696754768  0.111283433  0.2075670212  1.000
00000
## diastolicBP    0.0451902439  0.615751424  0.050329234  0.1639835294  0.784
00209
## BMI            0.0253798233  0.300815610  0.086519178  0.1145414809  0.325
55526
## heartRate     -0.0176742429  0.147196391  0.048996225  0.0904202659  0.182
14270
## glucose        0.0186832127  0.083720955  0.606582110  0.0454277635  0.135
31227
## heartDisease   0.0618099461  0.177602731  0.097316513  0.0825393306  0.216
42904
##               diastolicBP         BMI   heartRate     glucose heartDisease
## sex            0.05793347  0.08146510 -0.11662109  0.01000149   0.08842757
## age            0.20610399  0.13559390 -0.01284772  0.11790575   0.22525610
## currentSmoker -0.10774649 -0.16732202  0.06233048 -0.05509233   0.01945627
## cigsPerDay    -0.05631854 -0.09230960  0.07487699 -0.05623196   0.05799354
## BPMeds         0.19235553  0.09978030  0.01514226  0.04928250   0.08641714
## preStroke      0.04519024  0.02537982 -0.01767424  0.01868321   0.06180995
## preHyp         0.61575142  0.30081561  0.14719639  0.08372096   0.17760273
## diabetes       0.05032923  0.08651918  0.04899623  0.60658211   0.09731651
## chol           0.16398353  0.11454148  0.09042027  0.04542776   0.08253933
```

```
## systolicBP     0.78400209  0.32555526  0.18214270  0.13531227   0.21642904
## diastolicBP    1.00000000  0.37678107  0.18125699  0.05919728   0.14529910
## BMI            0.37678107  1.00000000  0.06746915  0.08248497   0.07528314
## heartRate      0.18125699  0.06746915  1.00000000  0.08745701   0.02285676
## glucose        0.05919728  0.08248497  0.08745701  1.00000000   0.12250507
## heartDisease   0.14529910  0.07528314  0.02285676  0.12250507   1.00000000
```

5.3. Feature Engineering

This section will transform nominal variables into factors to fit the dataset into the Logistic Regression model.

```
# Transform nominal variables to factor ones

df_heart_model <- df_heart_clean %>%
  mutate(sex = as.factor(sex)) %>%
  mutate(currentSmoker = as.factor(currentSmoker)) %>%
  mutate(BPMeds = as.factor(BPMeds)) %>%
  mutate(preStroke = as.factor(preStroke)) %>%
  mutate(preHyp = as.factor(preHyp)) %>%
  mutate(diabetes = as.factor(diabetes)) %>%
  mutate(heartDisease = as.factor(heartDisease))

# Dataset structure for modeling
str(df_heart_model)

## 'data.frame':    4238 obs. of  15 variables:
##  $ sex          : Factor w/ 2 levels "0","1": 2 1 2 1 1 1 1 1 2 2 ...
##  $ age          : int  39 46 48 61 46 43 63 45 52 43 ...
##  $ currentSmoker: Factor w/ 2 levels "0","1": 1 1 2 2 2 1 1 2 1 2 ...
##  $ cigsPerDay   : num  0 0 20 30 23 0 0 20 0 30 ...
##  $ BPMeds       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ preStroke    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ preHyp       : Factor w/ 2 levels "0","1": 1 1 1 2 1 2 1 1 2 2 ...
##  $ diabetes     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ chol         : int  195 250 245 225 285 228 205 313 260 225 ...
##  $ systolicBP   : num  106 121 128 150 130 ...
##  $ diastolicBP  : num  70 81 80 95 84 110 71 71 89 107 ...
##  $ BMI          : num  27 28.7 25.3 28.6 23.1 ...
##  $ heartRate    : num  80 95 75 65 85 77 60 79 76 93 ...
##  $ glucose      : num  77 76 70 103 85 99 85 78 79 88 ...
##  $ heartDisease : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 2 1 1 1 ...
```

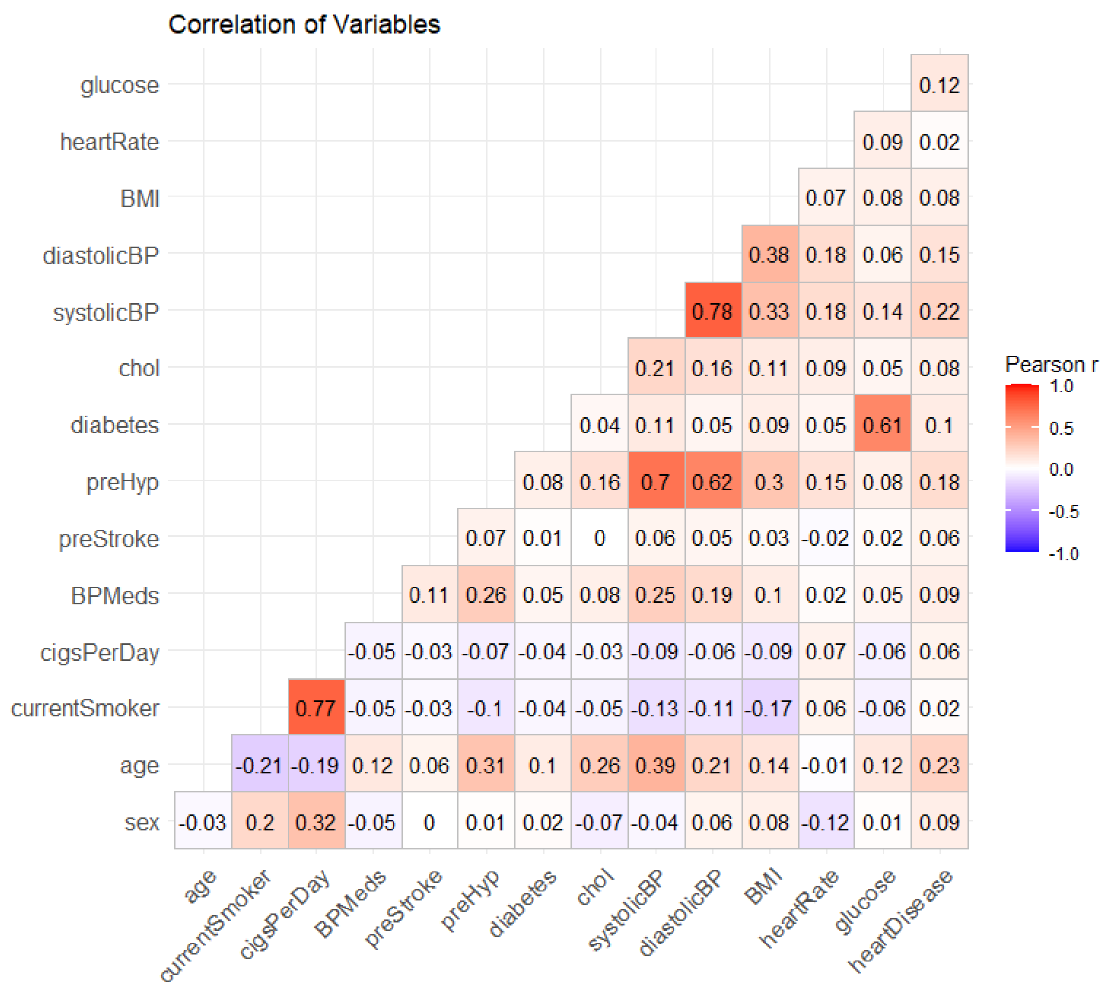
6. Inferential Statistics

6.1. Correlation Coefficient

The correlation coefficient between variables will be illustrated here. The correlation coefficient Pearson r is the value used to determine the extent to which paired scores are in the same or opposite position within their own distribution. Pearson r varies from +1 to -1:

- +1: perfect correlation and positive (> 0.5 or < -0.5 : strong correlation)
- -1: perfect correlation and negative
- 0: no correlation

```r
# Visualize the correlation using ggcorrplot library

ggcorrplot(corr = cormatrix,
           type = "lower",
           lab = T,
           legend.title = "Pearson r",
           title = 'Correlation of Variables')
```



The correlation heat map shows that currentSmoker and cigsPerDay (of course), systolicBP and diastolicBP, preHyp and systolicBP, diastolicBP, diabetes, and glucose have strong correlations.

6.2. Data Modeling

Logistic regression is a type of regression analysis in statistics used for the prediction of the outcome of a categorical dependent variable from a set of predictor or independent variables. The predictor variables can be continuous or categorical. In logistic regression, the dependent variable is always binary. Logistic regression is mainly used for prediction and also calculating the probability of success (Andy et al., 2012).

6.2.1. Logistic Regression Model

```
# Apply logistic model to the dataset.
# glm : generalized linear model.
# heartDisease~.  : we want too use all independent varibles to predict heart
Disease
# data = df_heart_model : data we will use for the model
# parameter family = "binomial" : makes glm() function do Logistic Regression

logistic_model <- glm(heartDisease~., data = df_heart_model, family = "binomi
al")

# Show the model detail
summary(logistic_model)

##
## Call:
## glm(formula = heartDisease ~ ., family = "binomial", data = df_heart_model
)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9914  -0.5967  -0.4320  -0.2923   2.8078
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -8.194890   0.644318 -12.719  < 2e-16 ***
## sex1            0.504529   0.100395   5.025 5.02e-07 ***
## age             0.062267   0.006173  10.087  < 2e-16 ***
## currentSmoker1  0.014129   0.144135   0.098  0.92191
## cigsPerDay      0.021301   0.005708   3.732  0.00019 ***
## BPMeds1         0.242477   0.220145   1.101  0.27071
## preStroke1      0.965574   0.441434   2.187  0.02872 *
## preHyp1         0.230533   0.128510   1.794  0.07283 .
## diabetes1       0.172939   0.294714   0.587  0.55734
## chol            0.001882   0.001024   1.837  0.06617 .
## systolicBP      0.014190   0.003532   4.017 5.89e-05 ***
## diastolicBP    -0.003055   0.005966  -0.512  0.60864
## BMI             0.004175   0.011734   0.356  0.72195
## heartRate      -0.001502   0.003882  -0.387  0.69885
## glucose         0.006889   0.002148   3.207  0.00134 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3611.5  on 4237  degrees of freedom
## Residual deviance: 3208.5  on 4223  degrees of freedom
## AIC: 3238.5
##
## Number of Fisher Scoring iterations: 5
```

Since p-value < 0.05 for sex, age, cigsPerDay, preStroke, systolicBP and glucose, these independent variables are significant.

We will re-run the model by including only significant variables.

6.2.2 Logistic Regression Model for Significant variables only

```
# Apply logistic model for only significant independent variables.
logistic_model_signif <- glm(heartDisease~. -currentSmoker -BPMeds -preHyp -d
iabetes -chol -diastolicBP -BMI -heartRate
                                  , data = df_heart_model
                                  , family = "binomial")

# Present the model detail
summary(logistic_model_signif)
```

```
##
## Call:
## glm(formula = heartDisease ~ . - currentSmoker - BPMeds - preHyp -
##     diabetes - chol - diastolicBP - BMI - heartRate, family = "binomial",
##     data = df_heart_model)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0645  -0.5905  -0.4359  -0.3000   2.7943
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.466921   0.389498 -21.738  < 2e-16 ***
## sex1         0.483907   0.097236   4.977 6.47e-07 ***
## age          0.064687   0.005927  10.913  < 2e-16 ***
## cigsPerDay   0.021522   0.003856   5.582 2.38e-08 ***
## preStroke1   1.045205   0.436254   2.396   0.0166 *
## systolicBP   0.017057   0.002001   8.523  < 2e-16 ***
## glucose      0.007691   0.001630   4.717 2.39e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3611.5  on 4237  degrees of freedom
```

```
## Residual deviance: 3217.7  on 4231  degrees of freedom
## AIC: 3231.7
##
## Number of Fisher Scoring iterations: 5
```

6.2.3. Logistic Regression statistical model

We now determined the significant variables affecting heart disease from the above model. They are sex, age, the number of cigarettes per day, prevalent stroke, systolic blood pressure, and glucose. Now, our logistic regression statistical model looks like this:

logit(p)=log(p/(1−p))=β0 + β1*sex + β2*age + β3*cigsPerDay + β4*preStroke + β5*systolicBP + β6*glucose

p is binomial proportion, and βi are regression coefficients.

Before filling the equation with numeric values from the model, we will see why log(p/(1-p)).

Logistic Regression work with **odds** rather than proportions. The odds are the ratio of something happening (e.g., heartDisease = 1) to something not happening (e.g., heartDisease = 0). Therefore, if the probability of heartDisease = 1 is p, then the odds = p(1-p) and p = odds/(odds + 1).

The problem is that the probability varies from 0 to 1, but the odds do not. If p is larger and larger, the odds will go to infinity.

So, the natural logarithm was used. The term log odds or logit appears from now on. And we have logit(p) and log(p/1-p).

The second important reason is that the histogram of the log(odds) is approximated with a normal distribution. So, the log(odds) makes things symmetrical, easier to interpret, and easier for statistics. This makes the log(odds) useful for solving certain statistics problems, specifically ones where we are trying to determine probabilities of binary dependent variables, such as yes/no or true/false.

Replacing βi by the values from the model, we have our logistic regression statistical:

log(odds of heart disease)= -8.466921 + 0.483907*sex + 0.064687*age + 0.021522*cigsPerDay + 1.045205*preStroke + 0.017057*systolicBP + 0.007691*glucose

6.2.4. Odds ratio and 95% CI

To make it easier for interpretation, we now calculate the odds ratio and Confidence Interval.

```
# The odds ratio
exp(coefficients(logistic_model_signif))
```

```
##   (Intercept)         sex1          age    cigsPerDay    preStroke1    systoli
cBP
## 0.0002103115 1.6224002792 1.0668250294 1.0217556934 2.8439821687 1.0172029
955
##      glucose
## 1.0077209011
```

```
# Confidence Interval 95% CI for the odds ratio
exp(confint(logistic_model_signif))
```

```
##                    2.5 %       97.5 %
## (Intercept) 9.703057e-05 0.0004469003
## sex1        1.341103e+00 1.9636365733
## age         1.054564e+00 1.0793629004
## cigsPerDay  1.014040e+00 1.0294908851
## preStroke1  1.189499e+00 6.6946287059
## systolicBP  1.013227e+00 1.0212110419
## glucose     1.004531e+00 1.0109950665
```

```
# Combine all in one CI table
cbind(coefficients = coef(logistic_model_signif),odds_ratio=exp(coef(logistic
_model_signif)),exp(confint(logistic_model_signif)))
```

```
##             coefficients   odds_ratio        2.5 %       97.5 %
## (Intercept) -8.466920790 0.0002103115 9.703057e-05 0.0004469003
## sex1         0.483906706 1.6224002792 1.341103e+00 1.9636365733
## age          0.064686975 1.0668250294 1.054564e+00 1.0793629004
## cigsPerDay   0.021522416 1.0217556934 1.014040e+00 1.0294908851
## preStroke1   1.045205242 2.8439821687 1.189499e+00 6.6946287059
## systolicBP   0.017056699 1.0172029955 1.013227e+00 1.0212110419
## glucose      0.007691247 1.0077209011 1.004531e+00 1.0109950665
```

6.3. Model Interpretation

In section 6.2, we implemented three parts:

- Build Logistic Regression Model with all attributes (section 6.2.1). We will call this model as Original Model
- Build Logistic Regression Model using significant attributes only (section 6.2.2). We will call this model as Significant Model
- Calculate the Odds ratio and Confidence Interval (section 6.2.4). We will call the result table as CI table.

We use the Original Model for comparison. Now we will interpret the results from the Significant Model and CI table.

6.3.1. Coefficients

The first line of the Significant Model is the call of the glm() function.

The second one gives us a summary of the deviance residuals.

Then we have the most important results we would like to look at in more detail. Coefficients.

(Intercept):

- **-8.466921** is the Intercept when other variables = 0. It means when all independent variables = 0, log(odds of heart disease) = -8.466921
- **0.389498** is the standard error of the Intercept. And the z-value, (-17.501), is the estimated intercept divided by the standard error = -8.466921/0.389498).
- **< 2e-16** : P-value of the Intercept

Now, we will focus on regression coefficient parameter values βi in the model:

sex:

- **0.483907** : holding all other features constant, the odds of getting diagnosed with heart disease for males (sex = 1) over that of females (sex = 0) is exp(0.483907) = 1.6224 with 95% CI being 1.341 and 1.964 . In other words, we can say that the odds of getting diagnosed with heart disease for males are 62.24% higher than the odds for females.

age:

- **0.064687** : holding all other features constant, we will see 6.68% increase in the odds of getting diagnosed with heart disease for a one year increase in age since exp(0.064687) = 1.0668 with 95% CI being 1.054 and 1.079.

cigsPerDay:

- **0.021522** : holding all other features constant, we will see 2.18% increase in the odds of getting diagnosed with heart disease for every extra cigarette the patient smokes since exp(0.021522) = 1.0218 with 95% CI being 1.014 and 1.029.

preStroke:

- **1.045205** : holding all other features constant, the odds of getting diagnosed with heart disease for person got prevalent stroke (preStroke = 1) over that ones didn't (preStroke = 0) is exp(1.045205) = 2.844 with 95% CI being 1.189 and 6.695. In other words, we can say that the odds for prevalent stroke person are 184.4% higher than the odds for non-prevalent stroke one.

systolicBP:

- **0.017057** : holding all other features constant, we will see 1.72% increase in the odds of getting diagnosed with heart disease for every unit increase in systolic Blood Pressure since exp(0.017057) = 1.0172 with 95% CI being 1.0132and 1.0212.

glucose:

- **0.007691** : holding all other features constant, we will see 0.77% increase in the odds of getting diagnosed with heart disease for every unit increase in glucose level since exp(0.007691) = 1.0077 with 95% CI being 1.0045 and 1.011.

The line Dispersion parameter in the Significant Model, is the default dispersion parameter used for logistic regression

6.3.2. Deviance Residual

The next part is Deviance Residual:

The values in this part can be used to compare models, compute R squared, and overall p-value.

- Null deviance 3611.5 is the value when we only have intercept in the model
- Residual deviance 3217.7 is the value when we put all the variables in the model
- The difference between Null deviance and Residual deviance reveals that the model is a good fit.
- AIC = 3231.7 is the Akaike Information Criterion, which tells us the quality of the model. The lower the AIC, the better is the model. As we see from the Original Model, AIC = 3238.5. The Significant model has AIC = 3231.7. So, the Significant model is the better one

The last line is the number of Fisher Scoring iterations, which shows us how quickly the glm() function converged on the maximum likelihood estimates for the coefficients.

6.4. Prediction

Now, we will use the model to predict heart disease.

```
# Predict probabilities of heart disease on whole dataset
pred_model_prob <- predict(logistic_model_signif,newdata = df_heart_model[,-1
5],type = "response")

# Select values with threshold = 0.5
pred_value <- ifelse(pred_model_prob>0.5,1,0)

# Create confusion matrix
confusion_matrix <- table(predicted = pred_value,actual = df_heart_model$hear
tDisease)
confusion_matrix

##          actual
## predicted    0    1
##         0 3567  591
##         1   27   53

# Evaluation metrics
accuracy <- (sum(diag(confusion_matrix))/sum(confusion_matrix))*100
sensitivity <- (confusion_matrix[2,2]/(confusion_matrix[2,2] + confusion_matr
```

```
ix[1,2])) * 100
specificity <- (confusion_matrix[1,1]/(confusion_matrix[1,1] + confusion_matr
ix[2,1])) * 100

paste("Accuracy with threshold 0.5 is",round(accuracy,2),"%")

## [1] "Accuracy with threshold 0.5 is 85.42 %"

paste("Sensitivity with threshold 0.5 is",round(sensitivity,2),"%")

## [1] "Sensitivity with threshold 0.5 is 8.23 %"

paste("Specificity with threshold 0.5 is",round(specificity,2),"%")

## [1] "Specificity with threshold 0.5 is 99.25 %"
```

Although the accuracy and specificity are so high, we can see that sensitivity is only 8.23%. It means that the model performs very poorly in predicting the risk of heart disease. Therefore, we will change the threshold to improve sensitivity

```
# Change threshold to 0.1

# Select values with threshold = 0.5
pred_value_1 <- ifelse(pred_model_prob>0.1,1,0)

# Create confusion matrix
confusion_matrix_1 <- table(predicted = pred_value_1,actual = df_heart_model$
heartDisease)
confusion_matrix_1

##            actual
## predicted    0    1
##          0 1635  114
##          1 1959  530

# Evaluation metrics
accuracy1 <- (sum(diag(confusion_matrix_1))/sum(confusion_matrix_1))*100
sensitivity1 <- (confusion_matrix_1[2,2]/(confusion_matrix_1[2,2] + confusion
_matrix_1[1,2])) * 100
specificity1 <- (confusion_matrix_1[1,1]/(confusion_matrix_1[1,1] + confusion
_matrix_1[2,1])) * 100

paste("Accuracy with threshold 0.1 is",round(accuracy1,2),"%")

## [1] "Accuracy with threshold 0.1 is 51.09 %"

paste("Sensitivity with threshold 0.1 is",round(sensitivity1,2),"%")

## [1] "Sensitivity with threshold 0.1 is 82.3 %"

paste("Specificity with threshold 0.1 is",round(specificity1,2),"%")

## [1] "Specificity with threshold 0.1 is 45.49 %"
```

Now, we have a good sensitivity value for predicting the risk of heart disease at 82.3 %

7. Conclusion and Discussion

The research implemented statistical analysis to figure out the key factors of heart disease and predict the overall risk of heart disease in 10 years using logistic regression on data collected from an ongoing Framingham Heart Study.

The Logistic Regression model has shown that Age, Sex, The number of cigarettes per day, Systolic Blood Pressure, Prevalent Stroke, and Glucose have a statistically significant correlation with the probability of heart disease. While the changes in Age, Sex, and Prevalent Stroke have a big impact on heart disease, the number of cigarettes per day, Systolic Blood Pressure, and Glucose variables have a smaller effect.

The model also performs well in predicting the risk of heart disease at the threshold of 0.1 with 82.3 % accuracy.

However, the number doesn't show a significant correlation between cholesterol level and heart disease. The reason could be that there is not enough data for evaluation because Cholesterol level is one of the most important factors affecting heart disease in the real world.

Another limitation of the research is in handling the imbalanced dataset, which needs to be improved in the further steps.

8. Bibliography

Andy F., Jeremy M., Zoe F. (2012). Discovering Statistics using R., London: SAGE Publications Ltd

David S. M., George P. M., Bruce A. C. (2021). Introduction to the Practice of Statistics. 10th ed., New York: Macmillan Learning

Framingham Heart Study. (2022). About the Framingham Heart Study, [online], available: https://www.framinghamheartstudy.org/fhs-about/ [accessed 10 Apr, 2022]

Kaggle (2022). Logistic Regression, [online], available: https://kaggle.com [accessed 10 Apr, 2022]

Medium. (2022). Logistic Regression in R, [online], available: https://medium.com [accessed 10 Apr, 2022]

Robert R. P. (2013). Understanding Statistics In the Behavioral Sciences. 10th ed., California: Jon-David Hague

WHO. (2022). Cardiovascular diseases, [online], available: https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1 [accessed 10 Apr, 2022].