Heart Failure

20125091 - Đặng Trường Duy

Contents

1 About Dataset

1.1 Context

Cardiovascular diseases (CVDs) are the number 1 cause of death globally, taking an estimated 17.9 million lives each year, which accounts for 31% of all deaths worldwide. Four out of 5CVD deaths are due to heart attacks and strokes, and one-third of these deaths occur prematurely in people under 70 years of age. Heart failure is a common event caused by CVDs and this dataset contains 11 features that can be used to predict a possible heart disease.

People with cardiovascular disease or who are at high cardiovascular risk (due to the presence of one or more risk factors such as hypertension, diabetes, hyperlipidemia or already established disease) need early detection and management wherein a machine learning model can be of great help.

This dataset was created by combining different datasets already available independently but not combined before. In this dataset, 5 heart datasets are combined over 11 common features which makes it the largest heart disease dataset available so far for research purposes. The five datasets used for its curation are:

- Cleveland: 303 observations
- Hungarian: 294 observations
- Switzerland: 123 observations
- Long Beach VA: 200 observations
- Stalog (Heart) Data Set: 270 observations

Total: 1190 observations

Duplicated: 272 observations

Final dataset: 918 observations

Every dataset used can be found under the Index of heart disease datasets from UCI Machine Learning Repository

Reference: ***Heart Failure Prediction Dataset | Kaggle***

1.2 Attributes Information

Detail of all the columns (attributes) of the dataset:

1. Age: age of the patient [years]
2. Sex: sex of the patient [M: Male, F: Female]
3. ChestPainType: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]
4. RestingBP: resting blood pressure [mm Hg]
5. Cholesterol: serum cholesterol [mm/dl]
6. FastingBS: fasting blood sugar [1: if FastingBS $>$ 120 mg/dl, 0: otherwise]
7. RestingECG: resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of $>$ 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]
8. MaxHR: maximum heart rate achieved [Numeric value between 60 and 202]
9. ExerciseAngina: exercise-induced angina [Y: Yes, N: No]
10. Oldpeak: oldpeak = ST [Numeric value measured in depression]
11. ST_Slope: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]
12. HeartDisease: conclusion of the patient [1: heart disease, 0: Normal]

2 Descriptive Statistics

2.1 Set Up Working Space

Load some important libraries.

```r
library(ggplot2)
library(GGally)
library(dplyr)
library(extrafont)
library(moments)
```

Set up working directory.

```r
setwd('/home/dui/Windows/CS/APCS/Sophomore/Semester 3/STAT452/Final Project/20125091')
```

Get working directory.

```r
getwd()
[1] "/home/dui/Windows/CS/APCS/Sophomore/Semester 3/STAT452/Final Project/20125091"
```

Read dataset from **'heart.csv'** into **data** variable. Then, backup the **data** into **'heart.rda'**.

```r
heart <- read.csv('./heart.csv', header = TRUE)
save(heart, file = './heart.rda')
```

Attach the data to R.

```r
attach(heart)
```

Get the number of columns (attributes) of the dataset and number of rows (instances) of the dataset.

```r
ncol(heart)
[1] 12
nrow(heart)
[1] 918
```

Encode categorical data to enumerate value with ***factor*** function.

```r
heart$Sex <- factor(heart$Sex, levels = c('M', 'F'))
levels(heart$Sex) <- c('Male', 'Female')
heart$ChestPainType <- factor(heart$ChestPainType, levels = c('TA', 'ATA', 'NAP', 'ASY'))
heart$RestingECG <- factor(heart$RestingECG, levels = c('Normal', 'ST', 'LVH'))
heart$FastingBS <- factor(heart$FastingBS)
levels(heart$FastingBS) <- c('Normal', 'Fasting Blood Sugar')
heart$ExerciseAngina <- factor(heart$ExerciseAngina, levels = c('Y', 'N'))
heart$ST_Slope <- factor(heart$ST_Slope, levels = c('Up', 'Flat', 'Down'))
heart$HeartDisease <- factor(heart$HeartDisease)
levels(heart$HeartDisease) <- c('Normal', 'Heart Disease')
```

Take a look of the data by using ***str*** function of R.

```
str(heart)
'data.frame':   918 obs. of  12 variables:
 $ Age           : int  40 49 37 48 54 39 45 54 37 48 ...
 $ Sex           : Factor w/ 2 levels "Male","Female": 1 2 1 2 1 1 2 1 1 2 ...
 $ ChestPainType : Factor w/ 4 levels "TA","ATA","NAP",..: 2 3 2 4 3 3 2 2 4 2 ...
 $ RestingBP     : int  140 160 130 138 150 120 130 110 140 120 ...
 $ Cholesterol   : int  289 180 283 214 195 339 237 208 207 284 ...
 $ FastingBS     : Factor w/ 2 levels "Normal","Fasting Blood Sugar": 1 1 1 1 1 1 1 1 1 1 ...
 $ RestingECG    : Factor w/ 3 levels "Normal","ST",..: 1 1 2 1 1 1 1 1 1 1 ...
 $ MaxHR         : int  172 156 98 108 122 170 170 142 130 120 ...
 $ ExerciseAngina: Factor w/ 2 levels "Y","N": 2 2 2 1 2 2 2 2 1 2 ...
 $ Oldpeak       : num  0 1 0 1.5 0 0 0 0 1.5 0 ...
 $ ST_Slope      : Factor w/ 3 levels "Up","Flat","Down": 1 2 1 2 1 1 1 1 2 1 ...
 $ HeartDisease  : Factor w/ 2 levels "Normal","Heart Disease": 1 2 1 2 1 1 1 1 2 1 ...
```

From ***summary*** function of R. The output of the ***summary*** function include basic information of quantitative and qualitative data of the dataframe.

For quantitative data we get: Minimum value (Min.), First quartile (1st Qu.), Median (Median), Mean (Mean), Third quartile (3rd Qu.), Maximum value (Max.).

For qualitative data we have the frequency of each category.

```
summary(heart)
      Age            Sex      ChestPainType   RestingBP      Cholesterol   
 Min.   :28.00   Male  :725   TA : 46       Min.   :  0.0   Min.   :  0.0  
 1st Qu.:47.00   Female:193   ATA:173       1st Qu.:120.0   1st Qu.:173.2  
 Median :54.00                NAP:203       Median :130.0   Median :223.0  
 Mean   :53.51                ASY:496       Mean   :132.4   Mean   :198.8  
 3rd Qu.:60.00                              3rd Qu.:140.0   3rd Qu.:267.0  
 Max.   :77.00                              Max.   :200.0   Max.   :603.0  
               FastingBS    RestingECG      MaxHR       ExerciseAngina
 Normal             :704   Normal:552   Min.   : 60.0   Y:371         
 Fasting Blood Sugar:214   ST    :178   1st Qu.:120.0   N:547         
                           LVH   :188   Median :138.0                 
                                        Mean   :136.8                 
                                        3rd Qu.:156.0                 
                                        Max.   :202.0                 
    Oldpeak        ST_Slope          HeartDisease
 Min.   :-2.6000   Up  :395   Normal       :410  
 1st Qu.: 0.0000   Flat:460   Heart Disease:508  
 Median : 0.6000   Down: 63                      
 Mean   : 0.8874                                 
 3rd Qu.: 1.5000                                 
 Max.   : 6.2000                                 
```

2.2 Basic Statistic and Analysis

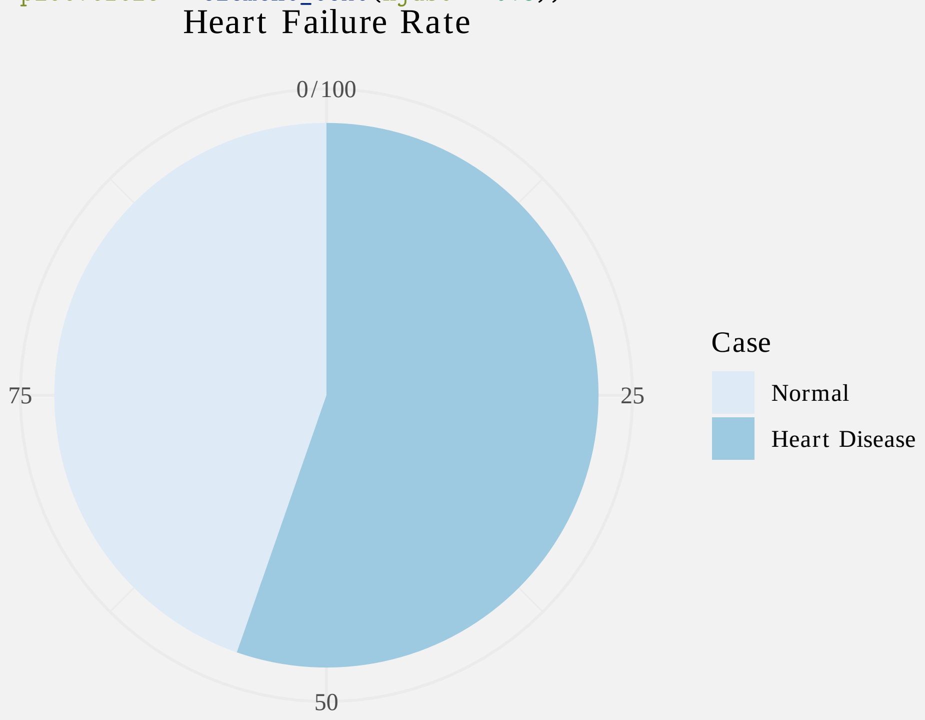
2.2.1 “HeartDisease” attribute

Compute percentages of people have a heart disease and people have not.

```
disease_rate <- heart %>%
    group_by(HeartDisease) %>%
    summarise(count = n()) %>%
    mutate(percentage = count / sum(count) * 100)
disease_rate
# A tibble: 2 x 3
  HeartDisease  count percentage
  <fct>         <int>      <dbl>
1 Normal          410       44.7
2 Heart Disease   508       55.3
```

Visualize the **disease_rate** with pie chart.

```
ggplot(disease_rate, aes(x = '', y = percentage, fill = HeartDisease)) +
    geom_bar(width = 2, stat = 'identity') +
    coord_polar('y', start = 0) + theme_minimal() +
    ggtitle('Heart Failure Rate') + xlab('') + ylab('') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```

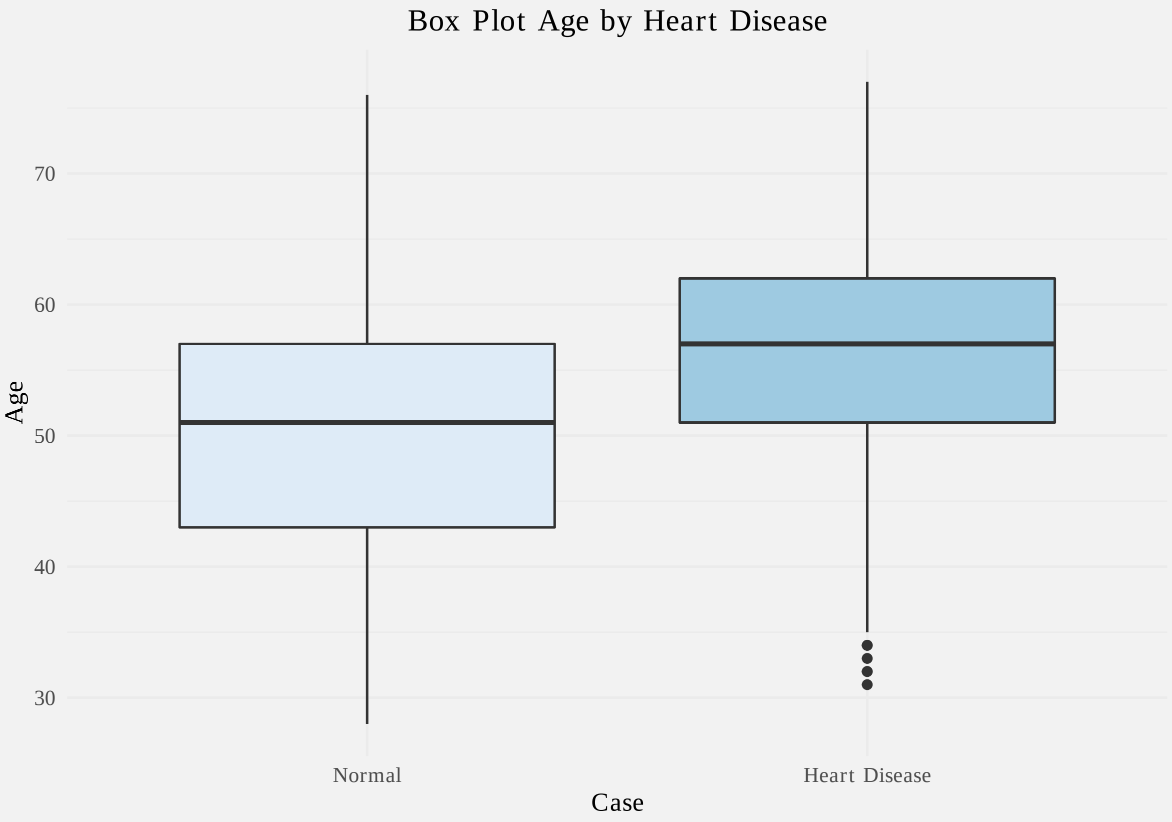


Comments: The rate of people have a heart disease and normal people are not much difference, the percentage of having disease cases larger than the percentage of normal cases about 11%.

2.2.2 “Age” attribute

Visualize a box plot for age of people suffer a heart failure and normal people.

```r
ggplot(heart, aes(x = Age, y = HeartDisease, fill = HeartDisease)) +
    geom_boxplot() +
    coord_flip() +
    ggtitle('Box Plot Age by Heart Disease') +
    xlab('Age') +
    ylab('Case') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5), legend.position = 'none')
```



Get the average age of normal cases and heart disease cases.

```r
avg_age_disease <- heart %>%
    group_by(HeartDisease) %>%
    summarise(avg_age = mean(Age))
avg_age_disease
# A tibble: 2 x 2
  HeartDisease  avg_age
  <fct>           <dbl>
1 Normal           50.6
2 Heart Disease    55.9
```

To find the number of people suffer a heart disease in age groups, age will be divided into 3 groups:

- Young adults (0-39)
- Middle-aged adults (40-59)
- Old-aged adults (above 60)

We then encode **categorized_age** to a vector of enumerate value by using ***factor*** function.

```
categorized_age <- seq_along(Age)
for (i in seq_along(Age)) {
    if (Age[i] < 40)
        categorized_age[i] <- 'Young'
    else if (Age[i] < 60)
        categorized_age[i] <- 'Middle aged'
    else
        categorized_age[i] <- 'Old aged'
}
categorized_age <- factor(categorized_age,
                          levels = c('Young', 'Middle aged', 'Old aged'))
```

Use ***table*** function to get the frequency table of these groups.

```
table(categorized_age)
categorized_age
      Young Middle aged    Old aged
         80         585         253
```

Next, we add the **categorized_age** as a new column named 'CategorizedAge' to our **heart** dataframe.

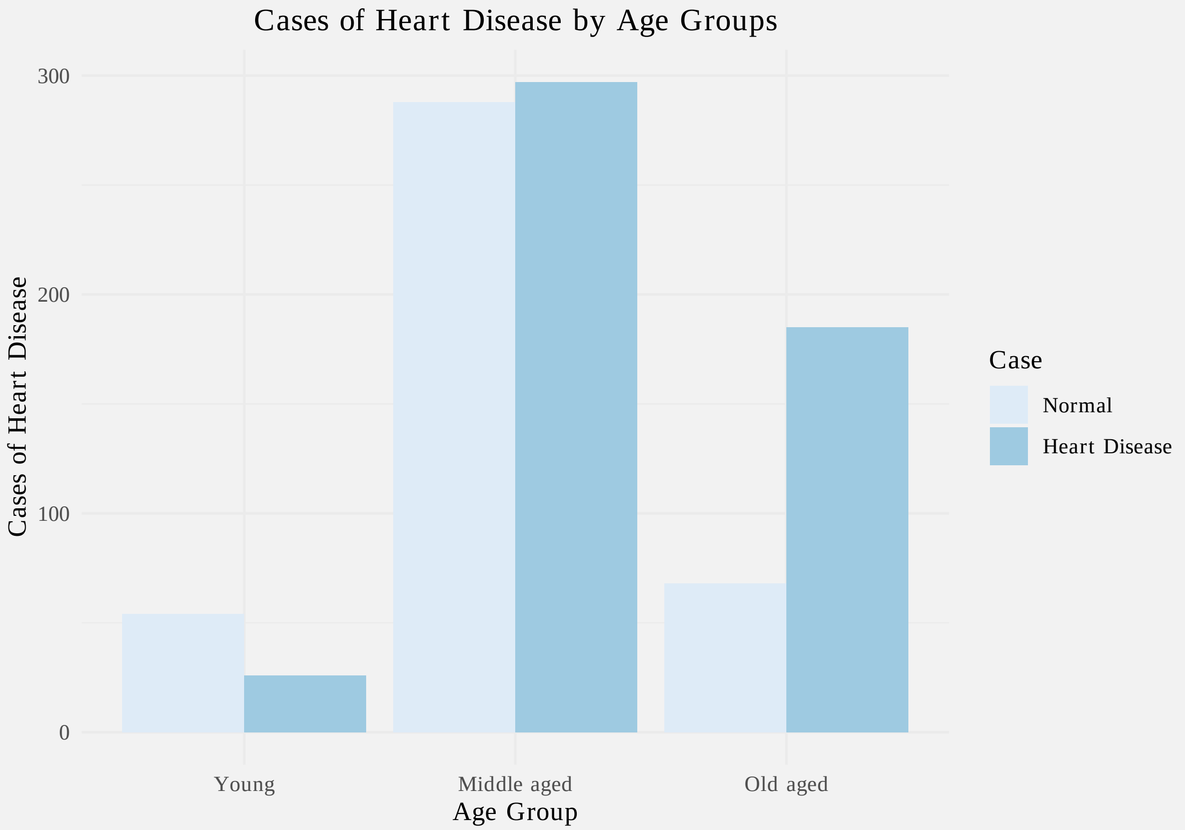
```
heart$CategorizedAge <- categorized_age
```

Next, The number of heart disease cases and normal cases in each age group is calculated.

```
age_disease <- heart %>%
    group_by(CategorizedAge, HeartDisease) %>%
    summarise(count = n()) %>%
    mutate(percentage = count / sum(count) * 100)
age_disease
# A tibble: 6 x 4
# Groups:   CategorizedAge [3]
  CategorizedAge HeartDisease  count percentage
  <fct>          <fct>         <int>      <dbl>
1 Young          Normal           54       67.5
2 Young          Heart Disease    26       32.5
3 Middle aged    Normal          288       49.2
4 Middle aged    Heart Disease   297       50.8
5 Old aged       Normal           68       26.9
6 Old aged       Heart Disease   185       73.1
```

We then plot the bar chart to show the frequency of number of normal cases and heart disease cases in each group.

```
ggplot(age_disease, aes(x = CategorizedAge, y = count, fill = HeartDisease)) +
    geom_col(position = 'dodge') +
    ggtitle('Cases of Heart Disease by Age Groups') +
    xlab('Age Group') +
    ylab('Cases of Heart Disease') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



Comments:

- The mean age of heart disease cases (≈ 56) larger than the mean age of normal cases (≈ 51).
- The range of age suffer a heart disease is between 51 and 62.
- Middle-aged group is the most surveyed group with 585 instances and over 50% of them have a heart disease.
- Old-aged group ranks second with 253 instances but has the highest ratio of suffering a heart disease which is about 73%
- Young group ranks last with 80 instances and 26 of them have a heart disease, this implies young people in the age below 39 also have a chance of suffering a heart disease.

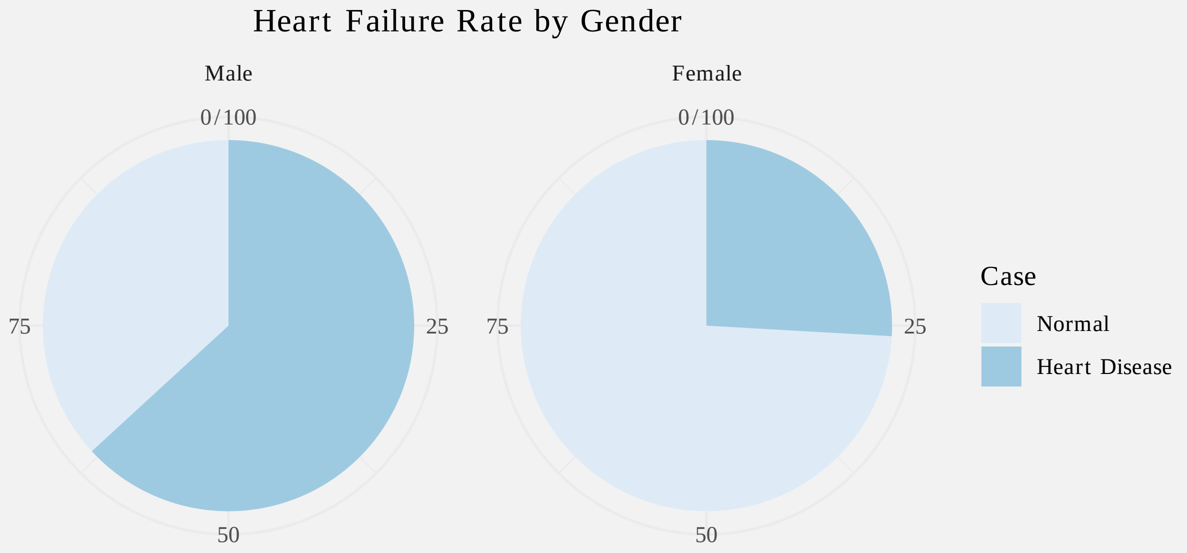
2.2.3 “Sex” attribute

Count the number of male and female instances by using ***table*** function. Then, calculate the number of normal cases and heart disease cases in each gender.

```
table(Sex)
Sex
  F   M
193 725
gender_disease <- heart %>%
    group_by(Sex, HeartDisease) %>%
    summarise(count = n()) %>%
    mutate(percentage = count / sum(count) * 100)
gender_disease
# A tibble: 4 x 4
# Groups:   Sex [2]
  Sex    HeartDisease  count percentage
  <fct>  <fct>         <int>      <dbl>
1 Male   Normal          267       36.8
2 Male   Heart Disease   458       63.2
3 Female Normal          143       74.1
4 Female Heart Disease    50       25.9
```

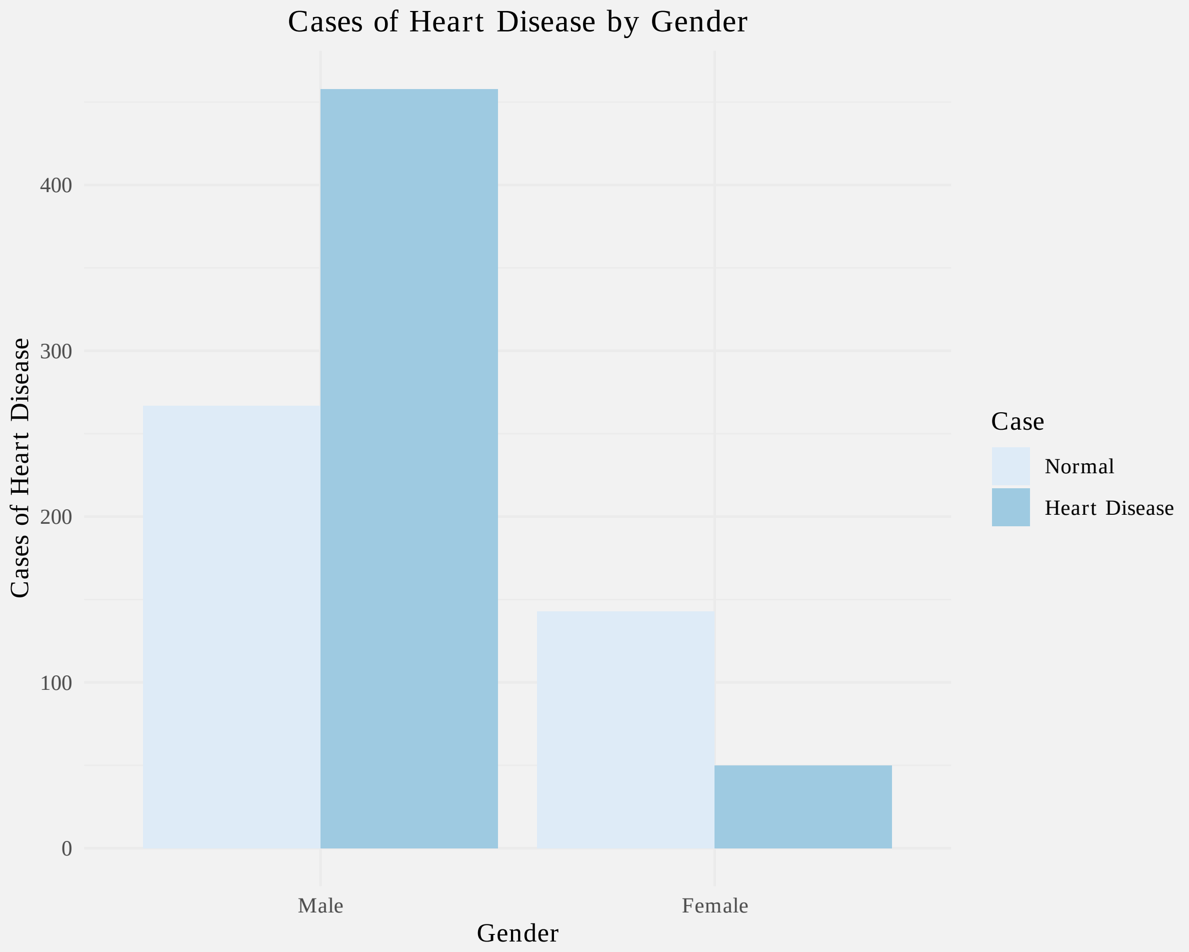
Use pie chart to show the rate of heart disease in each gender.

```
ggplot(gender_disease, aes(x = '', y = percentage, fill = HeartDisease)) +
    geom_bar(width = 2, stat = 'identity') +
    coord_polar('y', start = 0) +
    facet_wrap(~Sex, ncol = 2, scale = "fixed") +
    ggtitle('Heart Failure Rate by Gender') + xlab('') + ylab('') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



Plot a bar chart to show the number of heart disease cases and normal cases for each gender.

```
ggplot(gender_disease, aes(x = Sex, y = count, fill = HeartDisease)) +
    geom_col(position = 'dodge') +
    ggtitle('Cases of Heart Disease by Gender') +
    xlab('Gender') +
    ylab('Cases of Heart Disease') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



Comments:

- The number of male instances is over 3.5 times the number of female instances.
- For 725 male instances, the ratio of having a heart disease is over 60%.
- But for 193 the female instances, this ratio is only 26%.

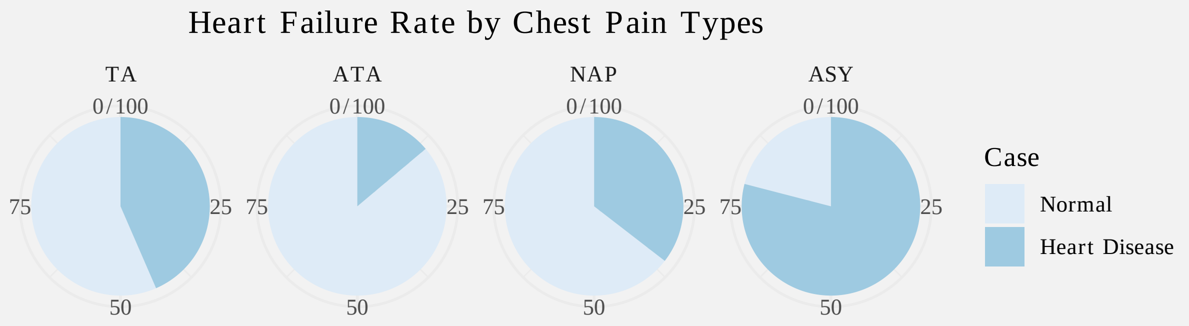
2.2.4 "ChestPainType" attribute

Count the number of instances in each type of chest pain. Then, calculate the number of normal cases and heart disease cases in each type.

```
table(heart$ChestPainType)
 TA ATA NAP ASY
 46 173 203 496
chestpain_disease <- heart %>%
    group_by(ChestPainType, HeartDisease) %>%
    summarise(count = n()) %>%
    mutate(percentage = count / sum(count) * 100)
chestpain_disease
# A tibble: 8 x 4
# Groups:   ChestPainType [4]
  ChestPainType HeartDisease  count percentage
  <fct>         <fct>         <int>      <dbl>
1 TA            Normal           26       56.5
2 TA            Heart Disease    20       43.5
3 ATA           Normal          149       86.1
4 ATA           Heart Disease    24       13.9
5 NAP           Normal          131       64.5
6 NAP           Heart Disease    72       35.5
7 ASY           Normal          104       21.0
8 ASY           Heart Disease   392       79.0
```

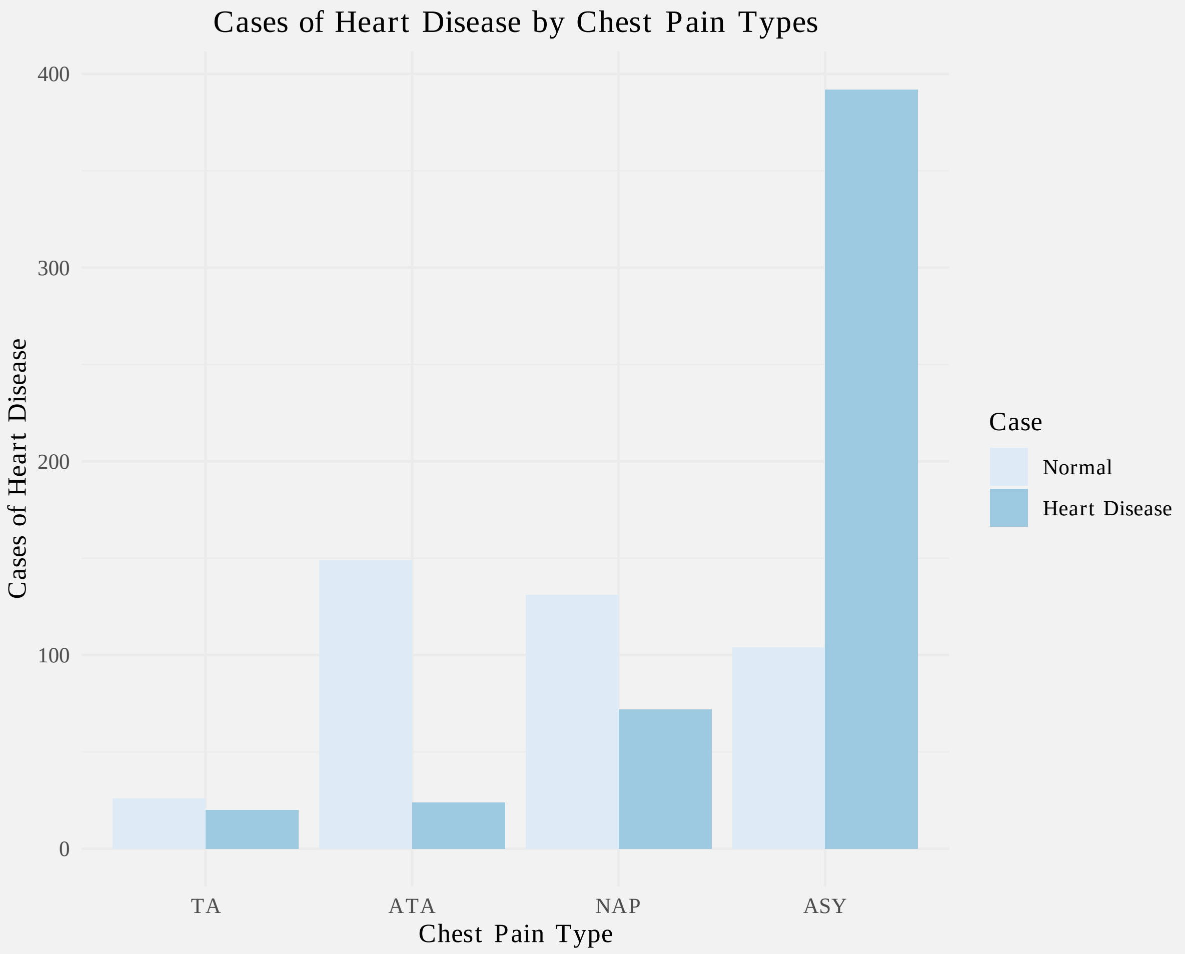
Plot a bar chart to show the number of heart disease cases and normal cases for each type of chest pain.

```
ggplot(chestpain_disease, aes(x = '', y = percentage, fill = HeartDisease)) +
    geom_bar(width = 2, stat = 'identity') +
    coord_polar('y', start = 0) +
    facet_wrap(~ChestPainType, ncol = 4, scale = "fixed") +
    ggtitle('Heart Failure Rate by Chest Pain Types') +
    xlab('') +
    ylab('') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



Generate a bar plot to show number of people suffer a heart disease and normal people for each type of chest pain.

```
ggplot(chestpain_disease, aes(x = ChestPainType, y = count, fill = HeartDisease)) +
    geom_col(position = 'dodge') +
    ggtitle('Cases of Heart Disease by Chest Pain Types') +
    xlab('Chest Pain Type') +
    ylab('Cases of Heart Disease') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



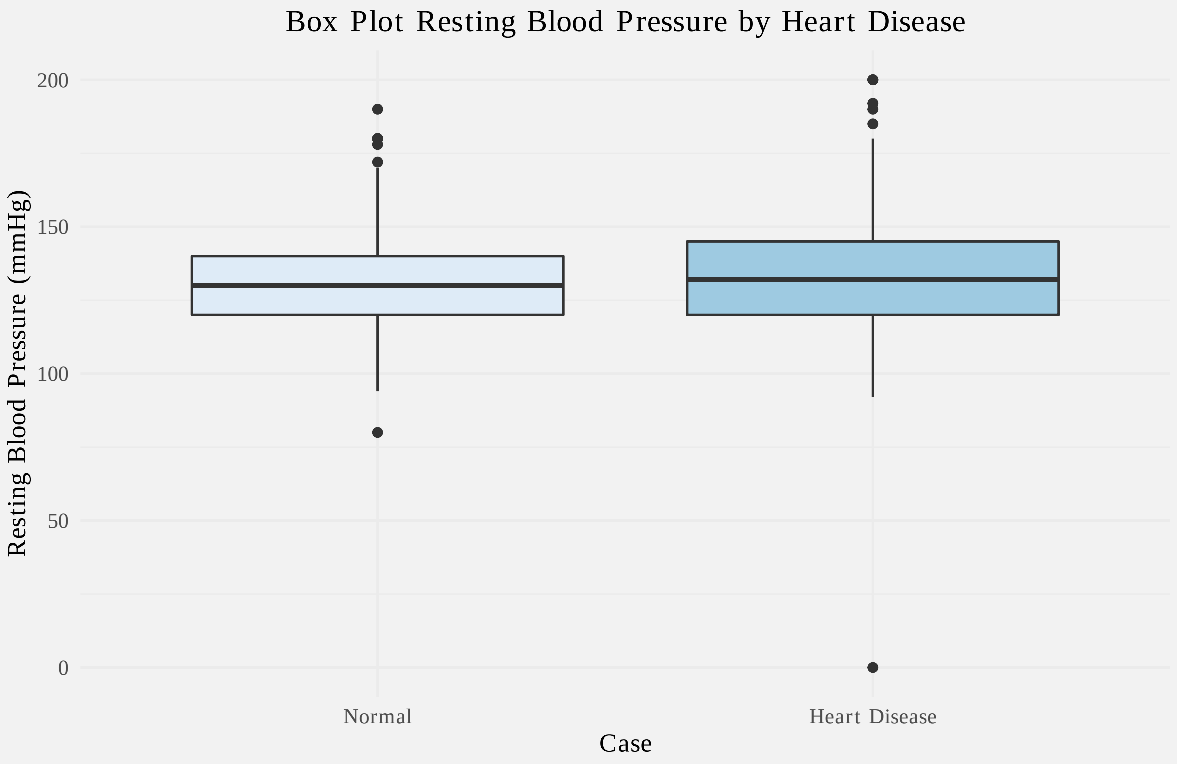
Comments:

- The most occurred chest pain is ASY (Asymptomatic) which is over 54% patients.
- Almost 80% patients, who suffer an ASY chest pain, have a heart disease.
- Over 86% patients, who suffer an ATA (Atypical Angina) chest pain, don’t have a heart disease.

2.2.5 “RestingBP” attribute

Visualize a box plot for resting blood pressure of patients who have a heart disease and who not.

```
ggplot(heart, aes(x = RestingBP, y = HeartDisease, fill = HeartDisease)) +
    geom_boxplot() +
    coord_flip() +
    ggtitle('Box Plot Resting Blood Pressure by Heart Disease') +
    xlab('Resting Blood Pressure (mmHg)') +
    ylab('Case') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5), legend.position = 'none')
```

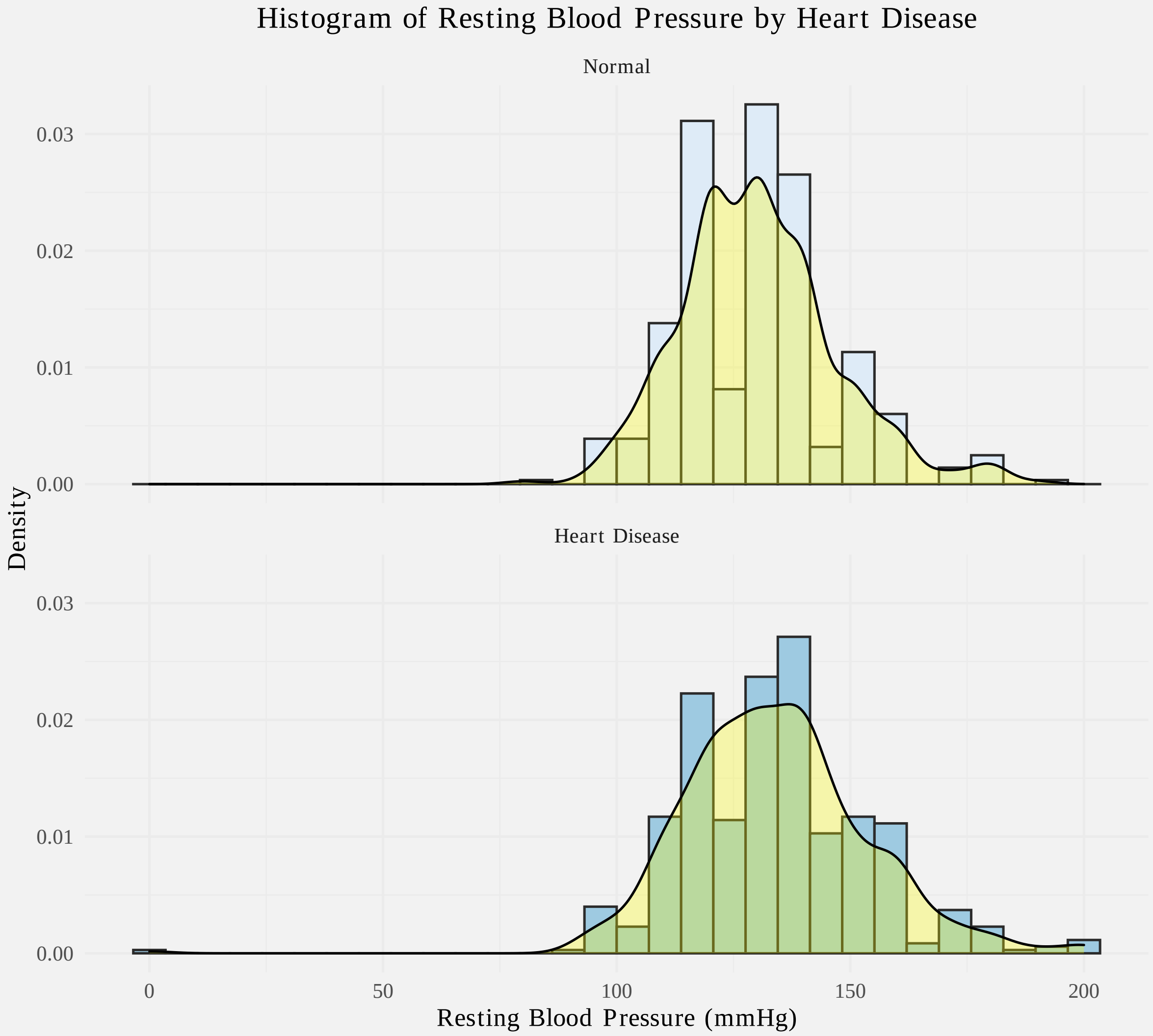


Calculate skewness and kurtosis of resting blood pressure.

```
bp_normal <- subset(heart, HeartDisease == 'Normal')$RestingBP
skewness(bp_normal)
[1] 0.5638636
kurtosis(bp_normal)
[1] 3.835007
bp_disease <- subset(heart, HeartDisease == 'Heart Disease')$RestingBP
skewness(bp_disease)
[1] -0.08056757
kurtosis(bp_disease)
[1] 7.167892
```

Visualize a histogram for resting blood pressure of patients who have a heart disease and who not.

```r
ggplot(heart, aes(x = RestingBP, fill = HeartDisease)) +
    geom_histogram(aes(y = ..density..), color = "grey17") +
    geom_density(alpha = .3, fill = "yellow") +
    facet_wrap(~HeartDisease, ncol = 1, scale = "fixed") +
    ggtitle("Histogram of Resting Blood Pressure by Heart Disease") +
    xlab("Resting Blood Pressure (mmHg)") +
    ylab("Density") +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5), legend.position = 'none')
```



RestingBP will be divided into 4 stage of Hypertension:

- Normal: resting blood pressure < 120 mm Hg.
- Prehypertension: resting blood pressure between 120 and 139 mm Hg.
- Stage 1: resting blood pressure between 140 and 159 mm Hg.
- Stage 2: resting blood pressure > 160 mm Hg.

We then encode categorized_restingBP to a vector of enumerate value by using ***factor*** function and add it as a new columns of our **heart** dataframe.

```
categorized_restingBP <- seq_along(RestingBP)
for (i in seq_along(RestingBP)) {
    if (RestingBP[i] < 120)
        categorized_restingBP[i] <- 'Normal'
    else if (RestingBP[i] < 140)
        categorized_restingBP[i] <- 'Prehypertension'
    else if (RestingBP[i] < 160)
        categorized_restingBP[i] <- 'Stage 1'
    else
        categorized_restingBP[i] <- 'Stage 2'
}
categorized_restingBP <- factor(categorized_restingBP,
                                levels = c('Normal', 'Prehypertension',
                                           'Stage 1', 'Stage 2'))
heart$CategorizedRestingBP <- categorized_restingBP
```

Use ***table*** function to get frequency table of these groups.

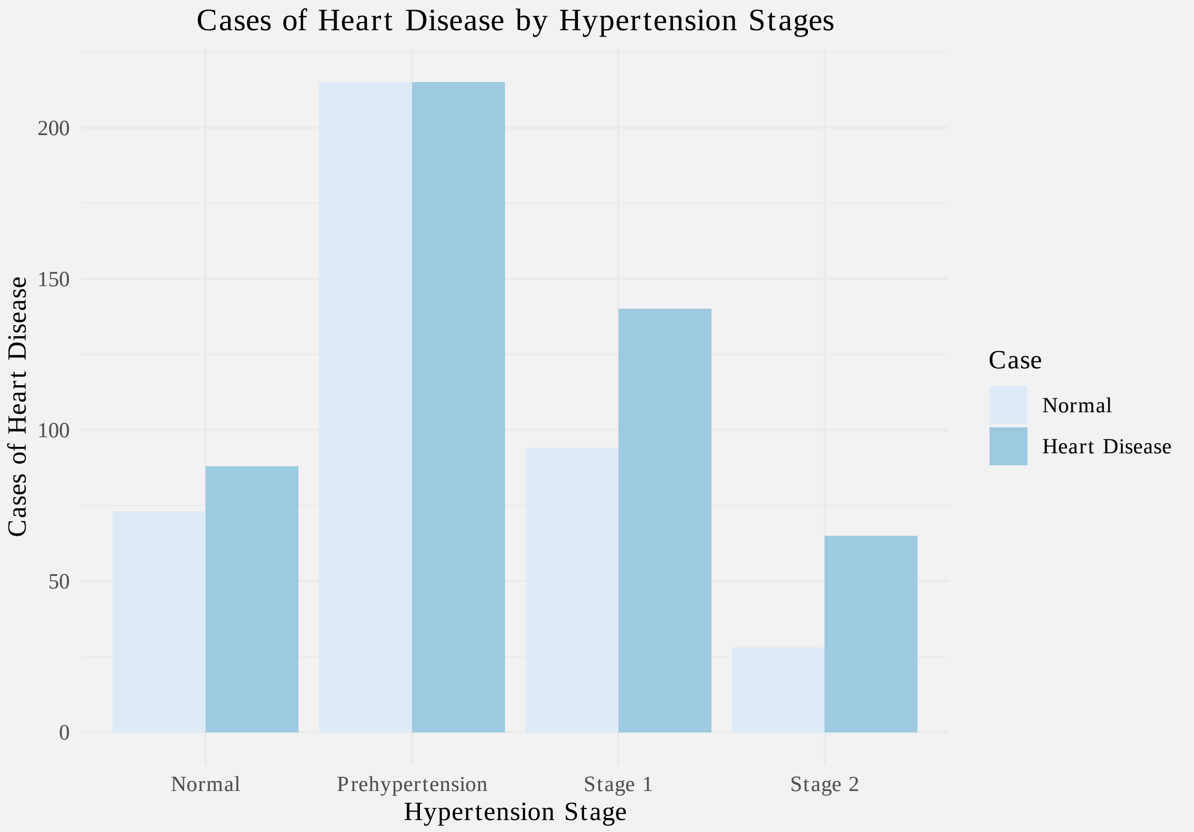
```
table(categorized_restingBP)
categorized_restingBP
         Normal Prehypertension         Stage 1         Stage 2
            161             430             234              93
```

Then, we count the normal patients and patients with heart disease in each group.

```
restingBP_disease <- heart %>%
    group_by(CategorizedRestingBP, HeartDisease) %>%
    summarise(count = n()) %>%
    mutate(percentage = count / sum(count) * 100)
restingBP_disease
# A tibble: 8 x 4
# Groups:   CategorizedRestingBP [4]
  CategorizedRestingBP HeartDisease  count percentage
  <fct>                <fct>         <int>      <dbl>
1 Normal               Normal           73       45.3
2 Normal               Heart Disease    88       54.7
3 Prehypertension      Normal          215       50
4 Prehypertension      Heart Disease   215       50
5 Stage 1              Normal           94       40.2
6 Stage 1              Heart Disease   140       59.8
7 Stage 2              Normal           28       30.1
8 Stage 2              Heart Disease    65       69.9
```

Graph a bar chart to show the frequency of heart disease by each stage of hypertension.

```r
ggplot(restingBP_disease,
       aes(x = CategorizedRestingBP, y = count, fill = HeartDisease)) +
    geom_col(position = 'dodge') +
    ggtitle('Cases of Heart Disease by Hypertension Stages') +
    xlab('Hypertension Stage') +
    ylab('Cases of Heart Disease') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



Comments:

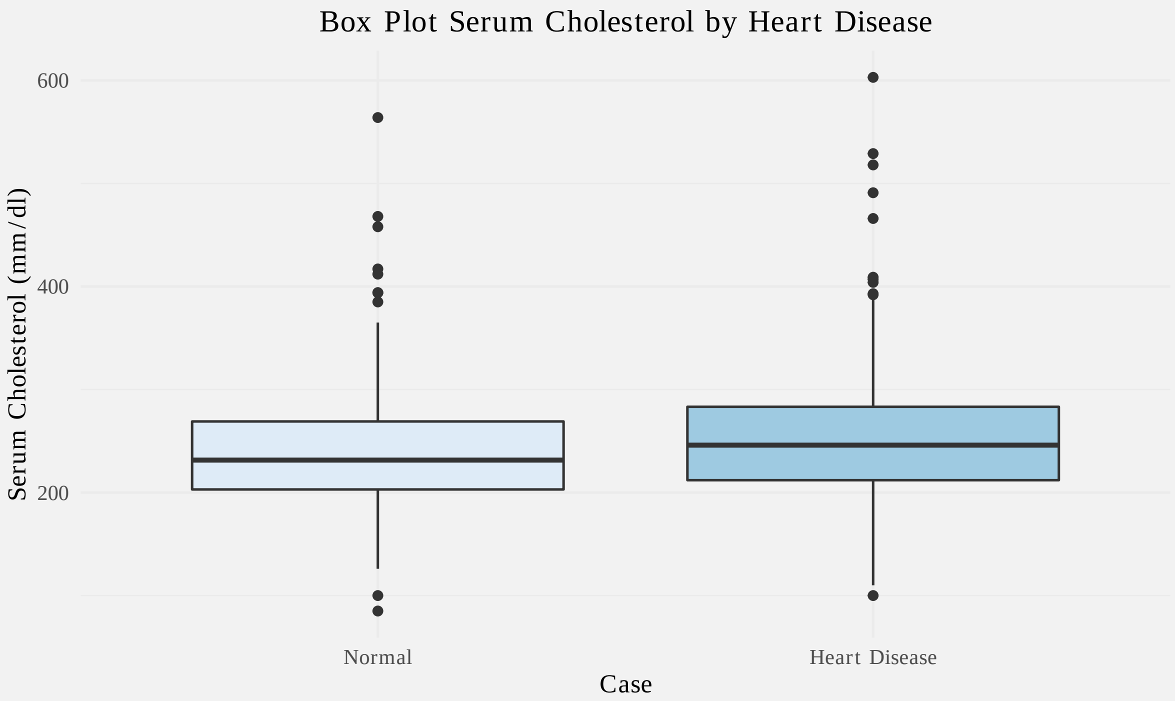
- Box plot shows that resting blood pressure of people have a heart disease nearly the same as the one of normal people and have corresponds median are 130 and 132.
- Patients, who have a heart disease, also have resting blood pressure in the range between 120 - 145 (mmHg) which is a little higher than normal resting blood pressure (< 120).
- Skewness and kurtosis of normal patients shows that the distribution of resting blood pressure is nearly normal distribution with a right skewness ($0.5638636 > 0$).
- For patients who have a heart disease the distribution of resting blood pressure is leptokurtic distribution ($7.167892 > 3$) with a little left skewness ($-0.08056757 < 0$).
- Prehypertension is the most occurrence stage, but only 50% patients have a heart disease.
- The chance of suffering a heart disease of normal, stage 1 and stage 2 is 54.7%, 59.8% and 70.0%.

2.2.6 “Cholesterol” attribute

There is many zero value of serum cholesterol which is the cause for the imbalanced data. Hence, we will plot patients records that have no zero serum cholesterol.

Graph a box plot for serum cholesterol of patients who have a heart disease and who not.

```
cholesterol_disease <- heart[heart$Cholesterol != 0, c('Cholesterol', 'HeartDisease')]
ggplot(cholesterol_disease,
       aes(x = Cholesterol, y = HeartDisease, fill = HeartDisease)) +
    geom_boxplot() + coord_flip() +
    ggtitle('Box Plot Serum Cholesterol by Heart Disease') +
    xlab('Serum Cholesterol (mm/dl)') + ylab('Case') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5), legend.position = 'none')
```

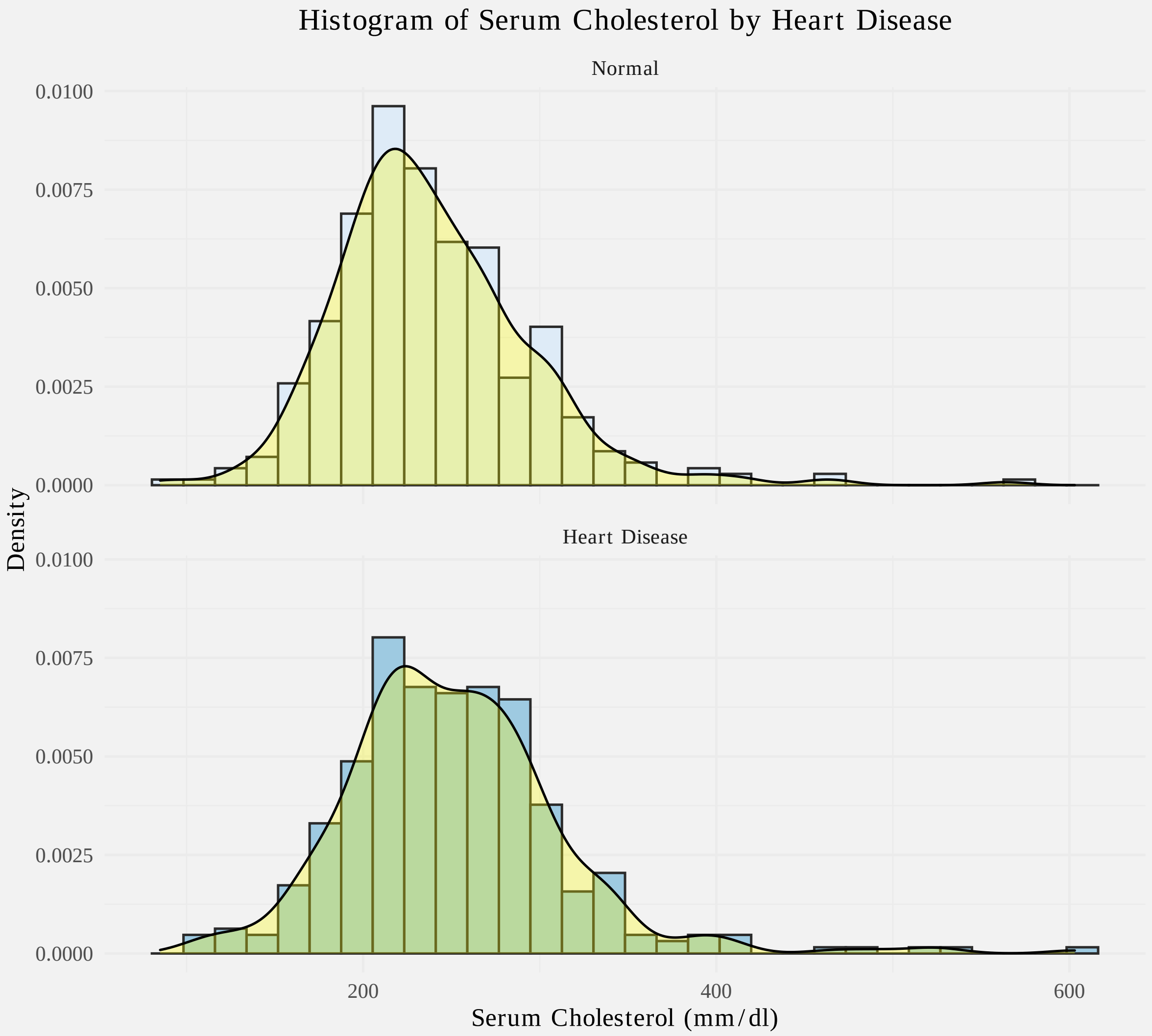


Calculate skewness and kurtosis of serum cholesterol.

```
cl_normal <- subset(cholesterol_disease, HeartDisease == 'Normal')$Cholesterol
skewness(cl_normal)
[1] 1.160466
kurtosis(cl_normal)
[1] 6.945497
cl_disease <- subset(cholesterol_disease, HeartDisease == 'Heart Disease')$Cholesterol
skewness(cl_disease)
[1] 1.251591
kurtosis(cl_disease)
[1] 7.661417
```

Visualize a histogram for serum cholesterol of patients who have a heart disease and who not.

```r
ggplot(cholesterol_disease, aes(x = Cholesterol, fill = HeartDisease)) +
    geom_histogram(aes(y = ..density..), color = "grey17") +
    geom_density(alpha = .3, fill = "yellow") +
    facet_wrap(~HeartDisease, ncol = 1, scale = "fixed") +
    ggtitle("Histogram of Serum Cholesterol by Heart Disease") +
    xlab("Serum Cholesterol (mm/dl)") +
    ylab("Density") +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5), legend.position = 'none')
```



Serum cholesterol will be divided into 3 levels:

- Normal: serum cholesterol < 200 mg/dl.
- High: serum cholesterol between 200 and 239 mg/dl.
- Very High: serum cholesterol > 240 mg/dl.

We then encode categorized_cholesterol to a vector of enumerate value by using ***factor*** function and add it as a new columns of our **cholesterol_disease** dataframe.

```
categorized_cholesterol <- seq_along(cholesterol_disease$Cholesterol)
for (i in seq_along(cholesterol_disease$Cholesterol)) {
    if (cholesterol_disease$Cholesterol[i] < 200)
        categorized_cholesterol[i] <- 'Normal'
    else if (cholesterol_disease$Cholesterol[i] < 240)
        categorized_cholesterol[i] <- 'High'
    else
        categorized_cholesterol[i] <- 'Very High'
}
categorized_cholesterol <- factor(categorized_cholesterol,
                                  levels = c('Normal', 'High', 'Very High'))
cholesterol_disease$CategorizedCholesterol <- categorized_cholesterol
```

Use **table** function to get frequency table of these groups.

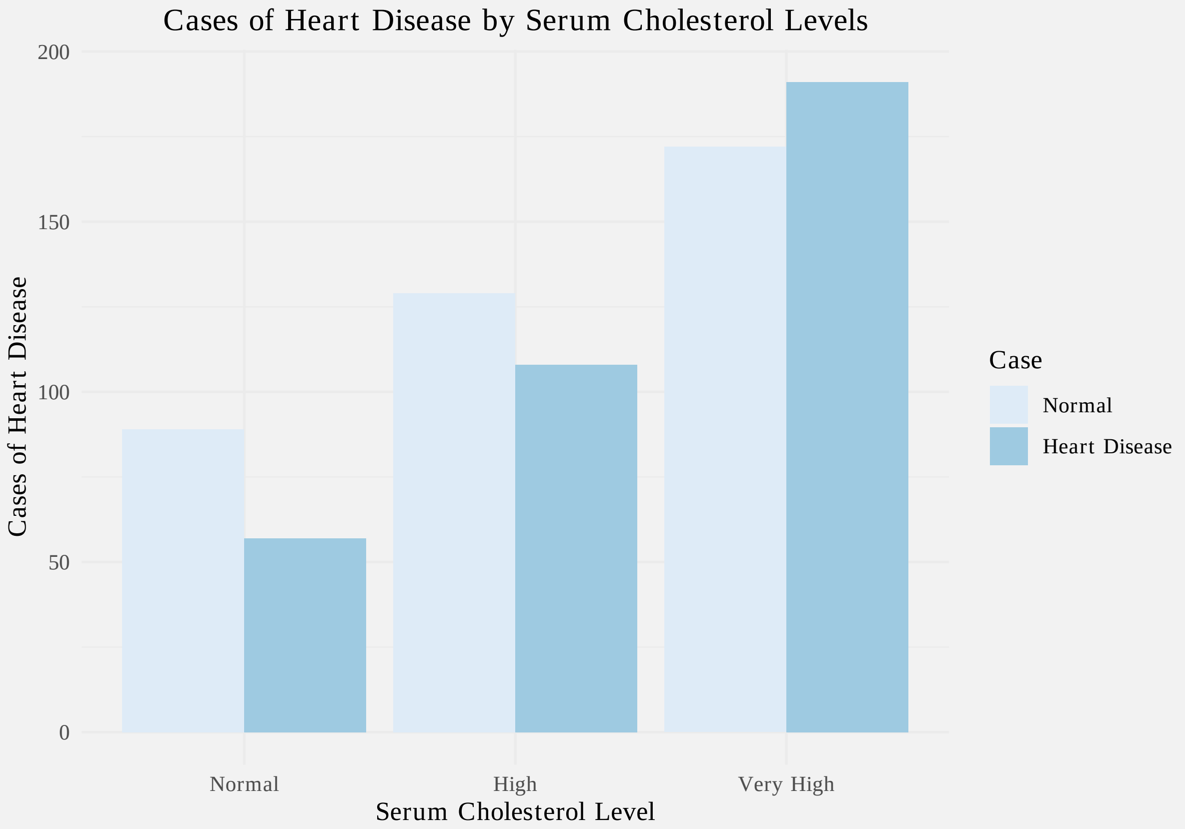
```
table(categorized_cholesterol)
categorized_cholesterol
   Normal      High Very High
      146       237       363
```

Then, we count the normal patients and patients with heart disease in each group.

```
categorizedCholesterol_disease <- cholesterol_disease %>%
    group_by(CategorizedCholesterol, HeartDisease) %>%
    summarise(count = n()) %>%
    mutate(percentage = count / sum(count) * 100)
categorizedCholesterol_disease
# A tibble: 6 x 4
# Groups:   CategorizedCholesterol [3]
  CategorizedCholesterol HeartDisease  count percentage
  <fct>                  <fct>         <int>      <dbl>
1 Normal                 Normal           89       61.0
2 Normal                 Heart Disease    57       39.0
3 High                   Normal          129       54.4
4 High                   Heart Disease   108       45.6
5 Very High              Normal          172       47.4
6 Very High              Heart Disease   191       52.6
```

Graph a bar chart to show the frequency of heart disease by each level of serum cholesterol.

```r
ggplot(categorizedCholesterol_disease,
       aes(x = CategorizedCholesterol, y = count, fill = HeartDisease)) +
    geom_col(position = 'dodge') +
    ggtitle('Cases of Heart Disease by Serum Cholesterol Levels') +
    xlab('Serum Cholesterol Level') +
    ylab('Cases of Heart Disease') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



Comments

- Box plot shows that serum cholesterol of two groups is nearly the same but the median of the patients who have a heart disease (246.0) is higher than the one of normal patients (231.5).
- Patients, who have a heart disease, also have serum cholesterol in the range between 212 - 283.25 (mm/dl) which is a higher than normal serum cholesterol ($<$ 200).
- Skewness and kurtosis of two groups show that the two distributions are leptokurtic distribution (kurtosis $>$ 3) with a positive skewness (skewness $>$ 0).
- Normal and High serum cholesterol patients have a chance of suffering a heart disease are 39.0% and 45.57%.
- But Very High serum cholesterol patients have the highest chance of suffering a heart disease which is about 52.6%.

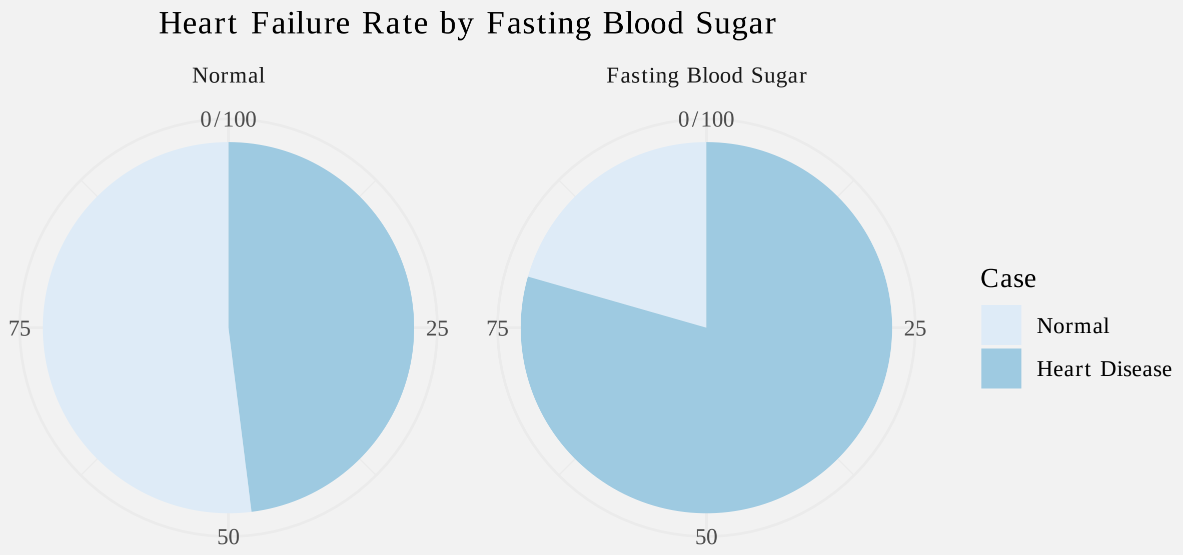
2.2.7 “FastingBS” attribute

Count the number of records of normal patients and patients who suffer a disease by each type of fasting blood sugar.

```
fastingBS_disease <- heart %>%
    group_by(FastingBS, HeartDisease) %>%
    summarise(count = n()) %>%
    mutate(percentage = count / sum(count) * 100)
fastingBS_disease
# A tibble: 4 x 4
# Groups:   FastingBS [2]
  FastingBS           HeartDisease  count percentage
  <fct>               <fct>         <int>      <dbl>
1 Normal              Normal          366       52.0
2 Normal              Heart Disease   338       48.0
3 Fasting Blood Sugar Normal           44       20.6
4 Fasting Blood Sugar Heart Disease   170       79.4
```

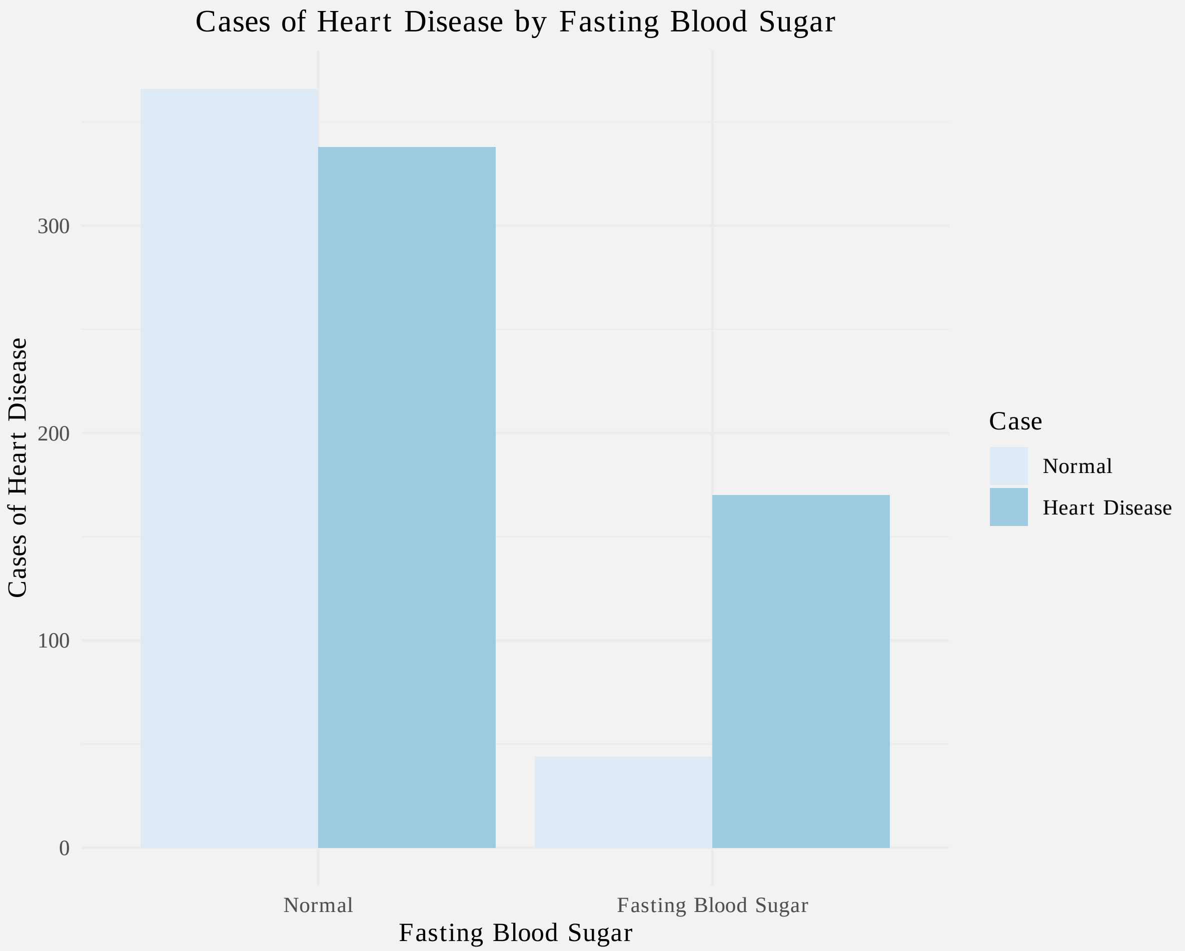
Graph a pie chart for fasting blood sugar of normal patients and patients suffer a heart disease.

```
ggplot(fastingBS_disease, aes(x = '', y = percentage, fill = HeartDisease)) +
    geom_bar(width = 2, stat = 'identity') +
    coord_polar('y', start = 0) +
    facet_wrap(~FastingBS, ncol = 2, scale = 'fixed') +
    ggtitle('Heart Failure Rate by Fasting Blood Sugar') +
    xlab('') +
    ylab('') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



Graph a bar chart for fasting blood sugar of normal patients and patients suffer a heart disease.

```r
ggplot(fastingBS_disease, aes(x = FastingBS, y = count, fill = HeartDisease)) +
    geom_col(position = 'dodge') +
    ggtitle('Cases of Heart Disease by Fasting Blood Sugar') +
    xlab('Fasting Blood Sugar') +
    ylab('Cases of Heart Disease') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



Comments:

- The records of normal blood sugar patients is over 3 times of patients with fasting blood sugar.
- The pie chart of normal blood sugar show that patients, who have normal blood sugar, have almost 52% chance for not suffer a heart disease.
- For the pie chart of fasting blood sugar, it is almost 80% patients suffer a heart disease.

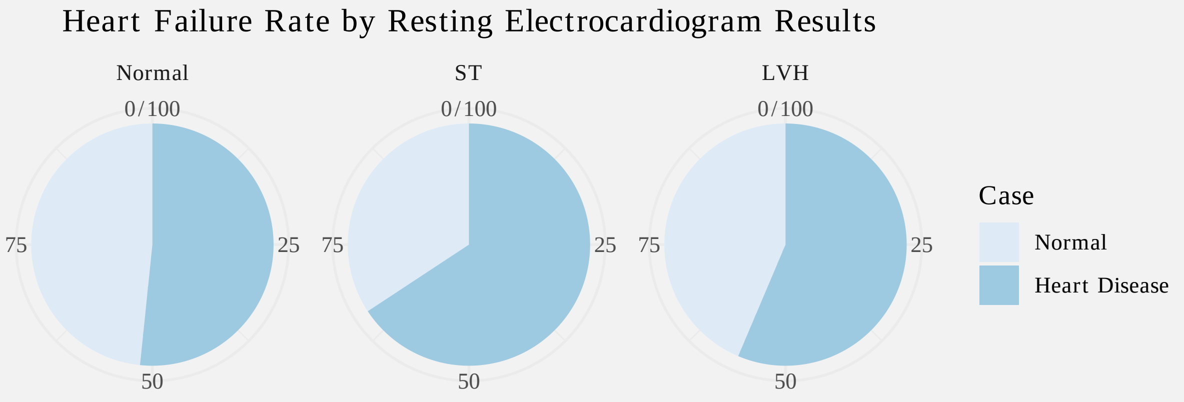
2.2.8 “RestingECG” attribute

Count the number of normal patients and patients suffer a heart disease by each type of resting electrocardiogram results.

```
restingECG_disease <- heart %>%
    group_by(RestingECG, HeartDisease) %>%
    summarise(count = n()) %>%
    mutate(percentage = count / sum(count) * 100)
restingECG_disease
# A tibble: 6 x 4
# Groups:   RestingECG [3]
  RestingECG HeartDisease  count percentage
  <fct>      <fct>         <int>      <dbl>
1 Normal     Normal          267       48.4
2 Normal     Heart Disease   285       51.6
3 ST         Normal           61       34.3
4 ST         Heart Disease   117       65.7
5 LVH        Normal           82       43.6
6 LVH        Heart Disease   106       56.4
```

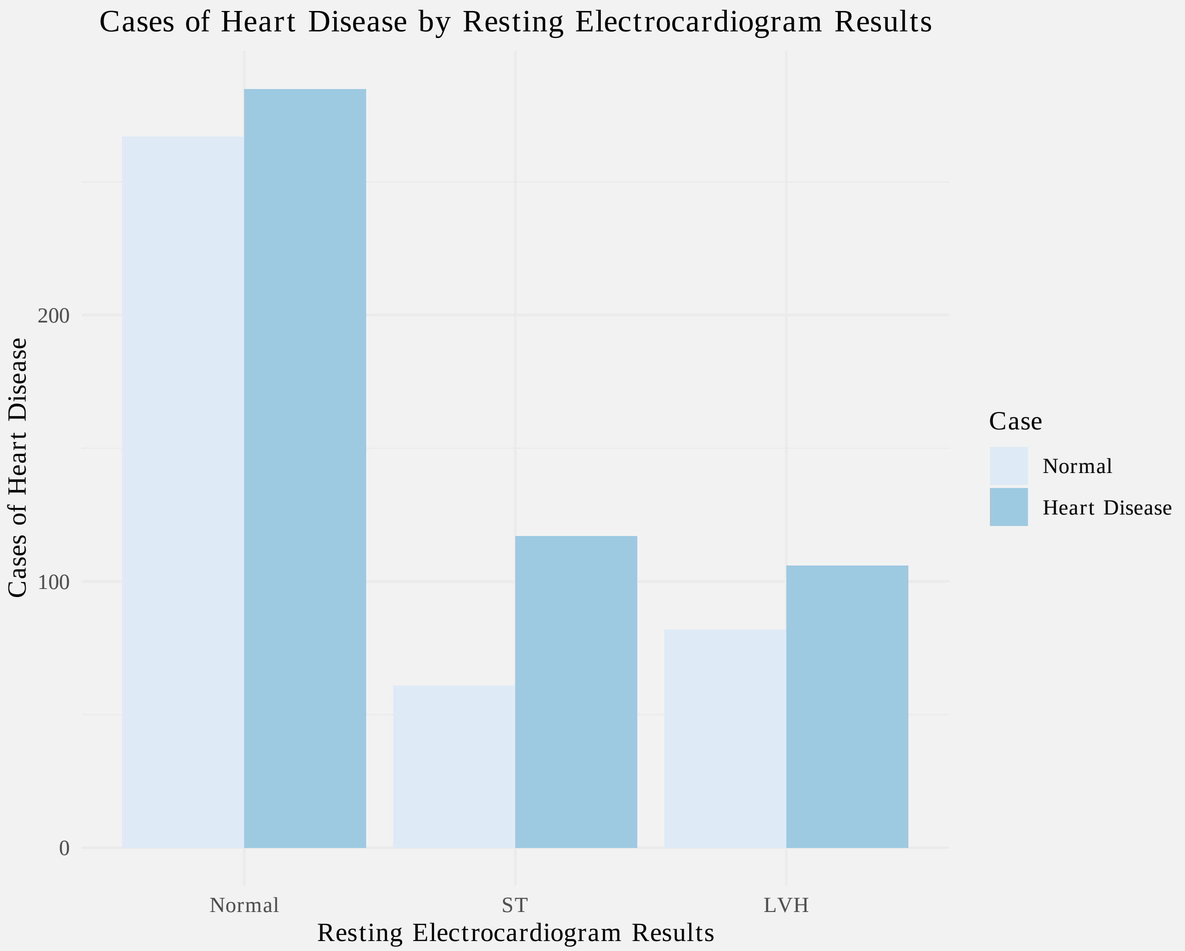
Graph a pie chart for resting electrocardiogram results of normal patients and patients suffer a heart disease.

```
ggplot(restingECG_disease, aes(x = '', y = percentage, fill = HeartDisease)) +
    geom_bar(width = 2, stat = 'identity') +
    coord_polar('y', start = 0) +
    facet_wrap(~RestingECG, ncol = 3, scale = 'fixed') +
    ggtitle('Heart Failure Rate by Resting Electrocardiogram Results') +
    xlab('') +
    ylab('') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



Graph a bar chart for resting electrocardiogram results of normal patients and patients suffer a heart disease.

```
ggplot(restingECG_disease, aes(x = RestingECG, y = count, fill = HeartDisease)) +
    geom_col(position = 'dodge') +
    ggtitle('Cases of Heart Disease by Resting Electrocardiogram Results') +
    xlab('Resting Electrocardiogram Results') +
    ylab('Cases of Heart Disease') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



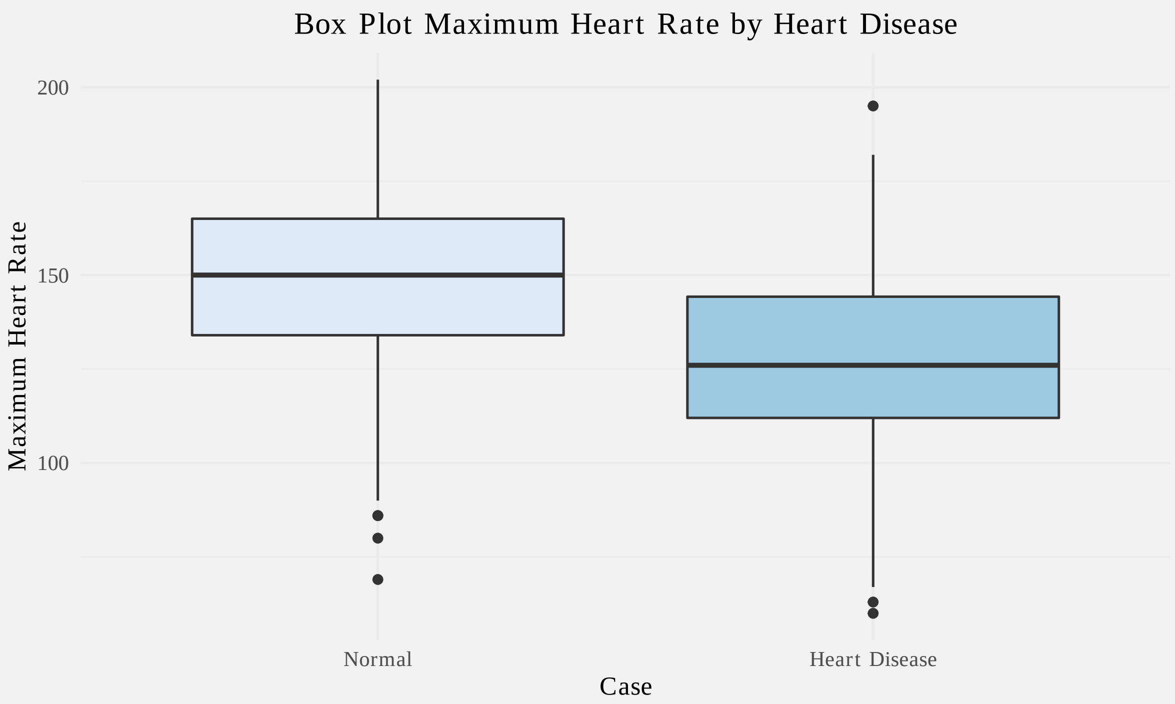
Comments:

- The most RestingECG results is normal which is about 552 records and over 3 times ST and LVH type.
- Patients with normal RestingECG have over 51% that suffer a heart disease.
- This ratio of ST and LVH is over 65% and 56%, respectively.

2.2.9 “MaxHR” attribute

Graph a box plot for maximum heart rate of normal patients and patients with heart disease.

```r
ggplot(heart, aes(x = MaxHR, y = HeartDisease, fill = HeartDisease)) +
    geom_boxplot() +
    coord_flip() +
    ggtitle('Box Plot Maximum Heart Rate by Heart Disease') +
    xlab('Maximum Heart Rate') +
    ylab('Case') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5), legend.position = 'none')
```

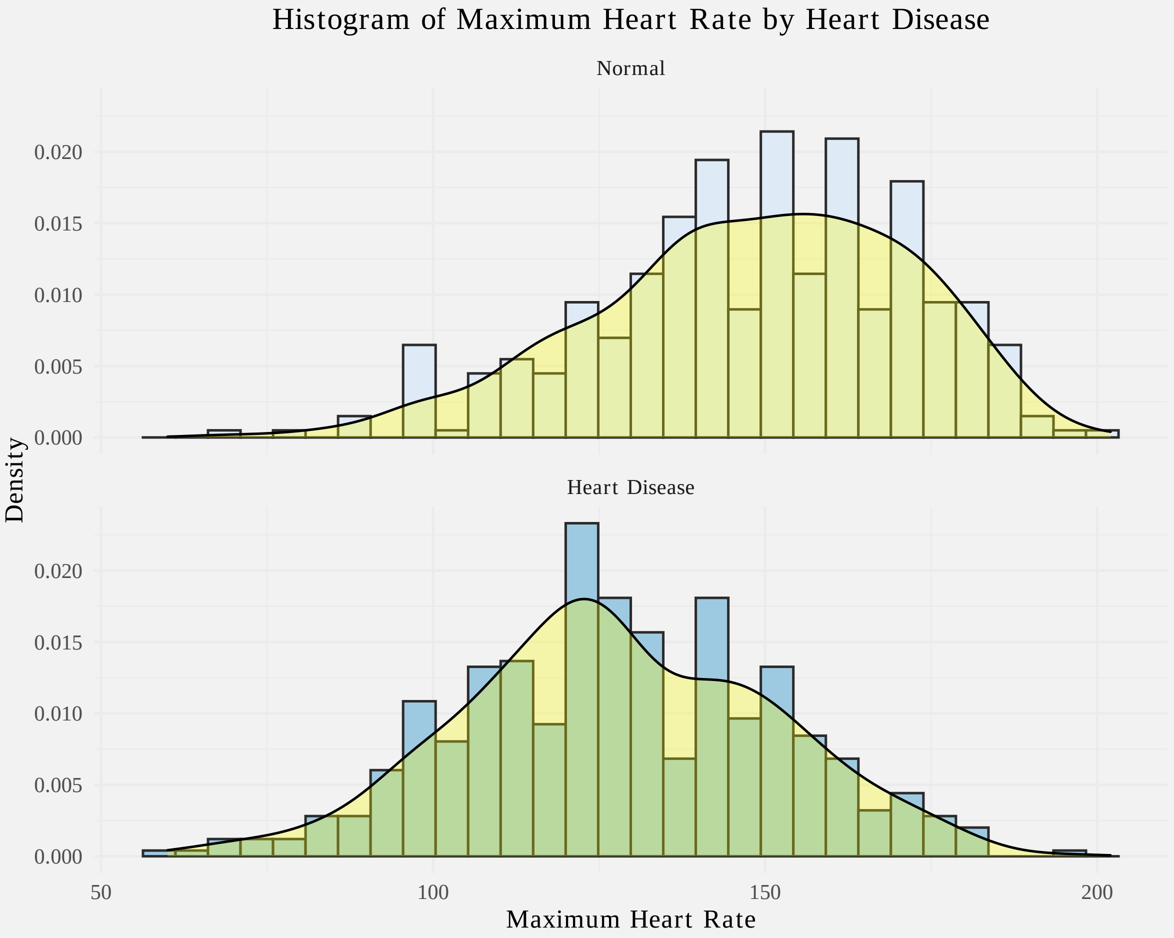


Calculate skewness and kurtosis of maximum heart rate

```r
maxHR_normal <- subset(heart, HeartDisease == 'Normal')$MaxHR
skewness(maxHR_normal)
[1] -0.4452592
kurtosis(maxHR_normal)
[1] 2.833873
maxHR_disease <- subset(heart, HeartDisease == 'Heart Disease')$MaxHR
skewness(maxHR_disease)
[1] -0.003628364
kurtosis(maxHR_disease)
[1] 2.793901
```

Graph a histogram for maximum heart rate of normal patients and patients with heart disease.

```
ggplot(heart, aes(x = MaxHR, fill = HeartDisease)) +
    geom_histogram(aes(y = ..density..), color = "grey17") +
    geom_density(alpha = .3, fill = "yellow") +
    facet_wrap(~HeartDisease, ncol = 1, scale = "fixed") +
    ggtitle("Histogram of Maximum Heart Rate by Heart Disease") +
    xlab("Maximum Heart Rate") + ylab("Density") +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5), legend.position = 'none')
```



- The median of maximum heart of patients with heart disease (126) is much smaller than the one of normal patients (150).
- Patients, who have a heart disease, have a maximum heart rate between 112 and 144.
- Box plot also indicates that both groups have a maximum rate heart which is larger than normal maximum heart rate (> 60 and < 100).
- The distribution of patients with heart disease is nearly normal with a little platykurtotic (2.793901) and a little left skewness (-0.0036).
- The distribution of normal patients is also nearly normal with a little platykurtotic and left skewness (0.4452592).

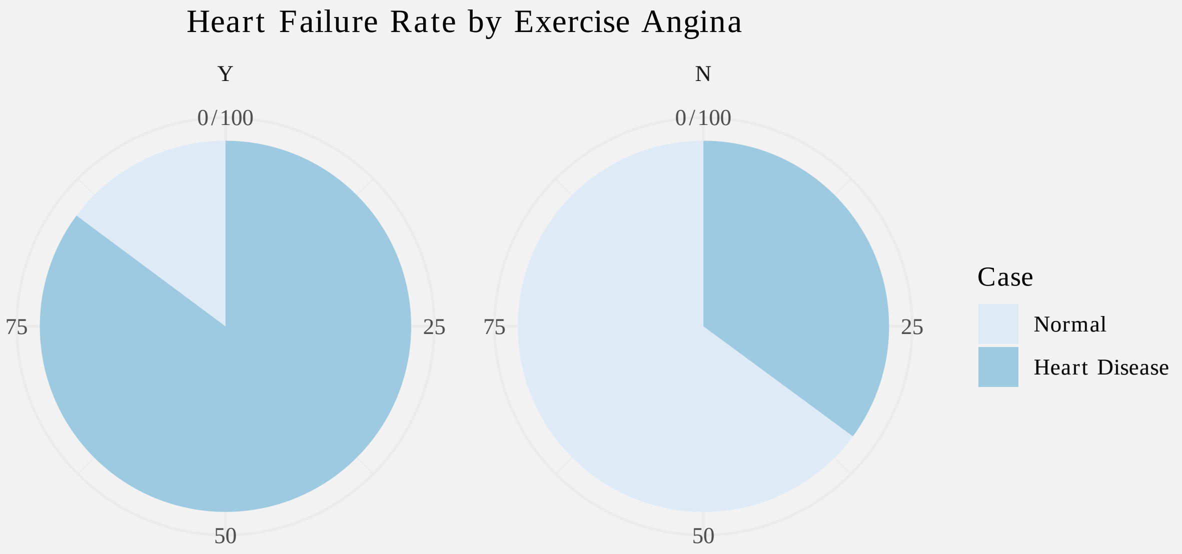
2.2.10 “ExerciseAngina” attribute

Count the number of angina when exercising of normal patients and patients with a heart disease.

```
angina_disease <- heart %>%
    group_by(ExerciseAngina, HeartDisease) %>%
    summarise(count = n()) %>%
    mutate(percentage = count / sum(count) * 100)
angina_disease
# A tibble: 4 x 4
# Groups:   ExerciseAngina [2]
  ExerciseAngina HeartDisease  count percentage
  <fct>          <fct>         <int>      <dbl>
1 Y              Normal           55       14.8
2 Y              Heart Disease   316       85.2
3 N              Normal          355       64.9
4 N              Heart Disease   192       35.1
```

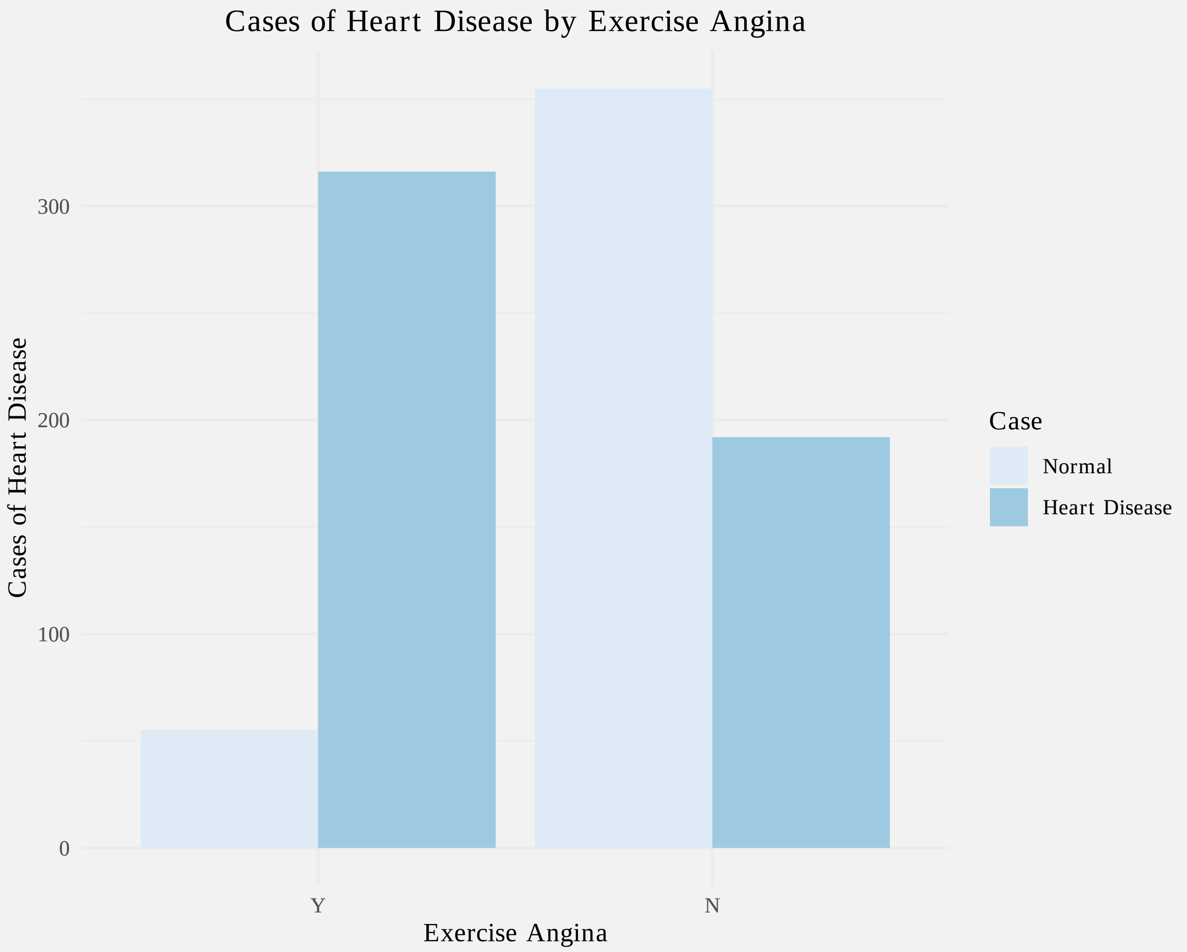
Graph a pie chart for whether there is an angina when exercising of normal patients and patients with a heart disease.

```
ggplot(angina_disease, aes(x = '', y = percentage, fill = HeartDisease)) +
    geom_bar(width = 2, stat = 'identity') + coord_polar('y', start = 0) +
    facet_wrap(~ExerciseAngina, ncol = 2, scale = 'fixed') +
    ggtitle('Heart Failure Rate by Exercise Angina') +
    xlab('') +
    ylab('') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



Graph a bar chart for whether there is an angina when exercising of normal patients and patients with a heart disease.

```r
ggplot(angina_disease, aes(x = ExerciseAngina, y = count, fill = HeartDisease)) +
    geom_col(position = 'dodge') +
    ggtitle('Cases of Heart Disease by Exercise Angina') +
    xlab('Exercise Angina') +
    ylab('Cases of Heart Disease') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



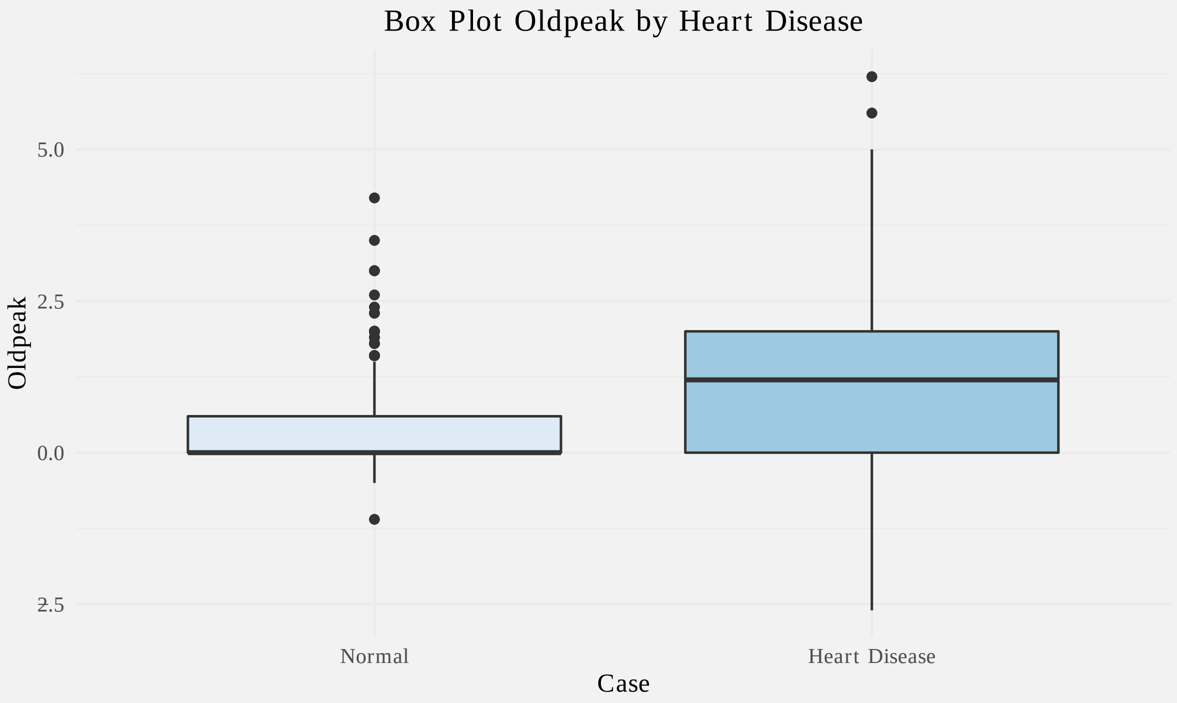
Comments:

- The number of records of patients without angina when exercising is almost 1.5 times the number of records of patients with angina when exercising.
- Pie chart shows that patients with angina when exercising have a high rate about 85% to suffer a heart disease.
- For patients with no angina when exercising this ratio is only about 35%.

2.2.11 “Oldpeak” attribute

Graph a box plot for old-peak of normal patients and patients with heart disease.

```
ggplot(heart, aes(x = Oldpeak, y = HeartDisease, fill = HeartDisease)) +
    geom_boxplot() +
    coord_flip() +
    ggtitle('Box Plot Oldpeak by Heart Disease') +
    xlab('Oldpeak') +
    ylab('Case') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5), legend.position = 'none')
```

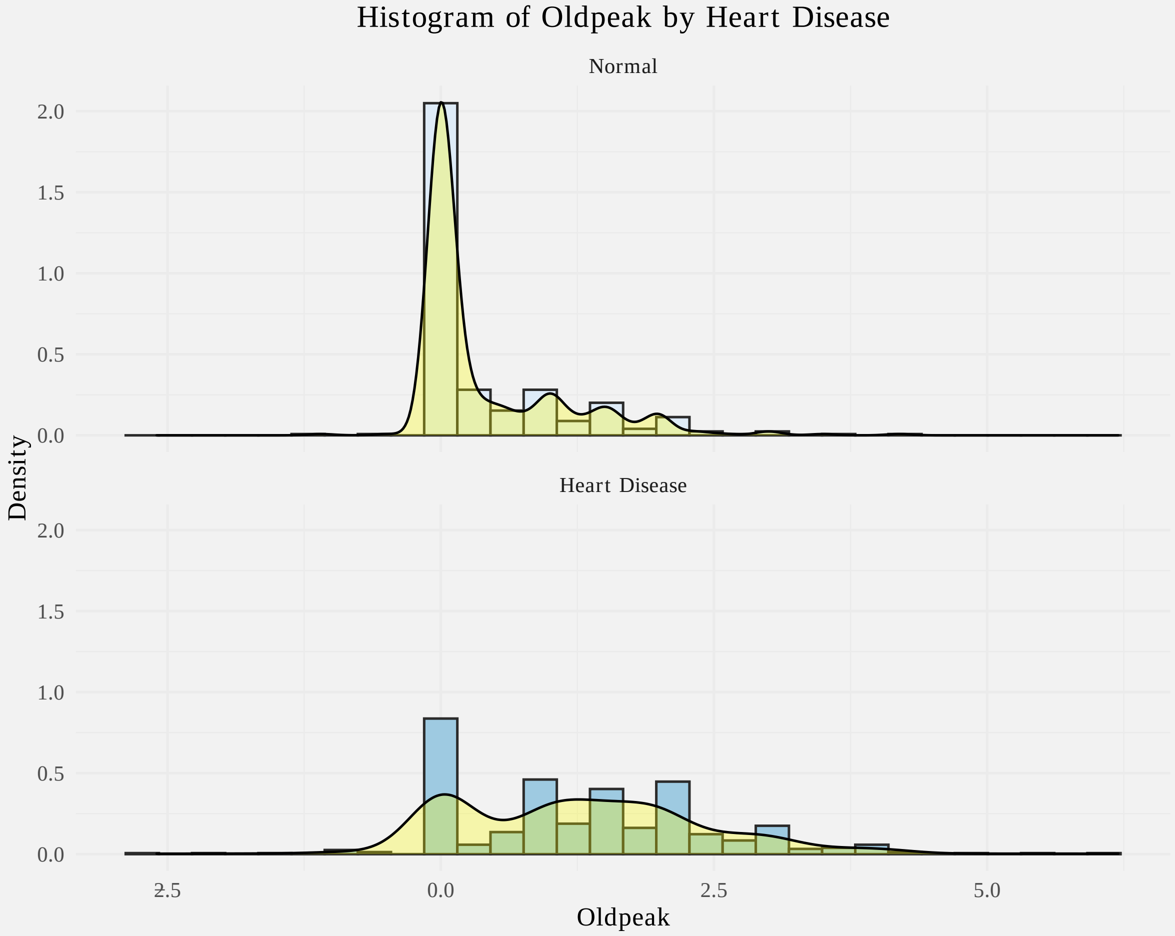


Calculate skewness and kurtosis of maximum heart rate

```
oldpeak_normal <- subset(heart, HeartDisease == 'Normal')$Oldpeak
skewness(oldpeak_normal)
[1] 1.882717
kurtosis(oldpeak_normal)
[1] 6.825291
oldpeak_disease <- subset(heart, HeartDisease == 'Heart Disease')$Oldpeak
skewness(oldpeak_disease)
[1] 0.5103819
kurtosis(oldpeak_disease)
[1] 3.68208
```

Graph a histogram for old-peak of normal patients and patients with heart disease.

```r
ggplot(heart, aes(x = Oldpeak , fill = HeartDisease)) +
    geom_histogram(aes(y = ..density..), color = "grey17") +
    geom_density(alpha = .3, fill = "yellow") +
    facet_wrap(~HeartDisease, ncol = 1, scale = "fixed") +
    ggtitle("Histogram of Oldpeak by Heart Disease") +
    xlab("Oldpeak") +
    ylab("Density") +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5), legend.position = 'none')
```



- The median of old-peak of normal patients (0.0) is much smaller than the one of patients with heart disease (1.2).
- Patients, who have a heart disease, have an old-peak between 0.0 and 2.0.
- Box plot also indicates that both groups have old-peak values which are in low range of old-peak value.
- The distribution of normal patients is leptokurtic (6.825291) and right skewness (1.882717).
- The distribution of patients with heart disease is nearly normal with a little leptokurtic (3.68208) right skewness (0.5103819).

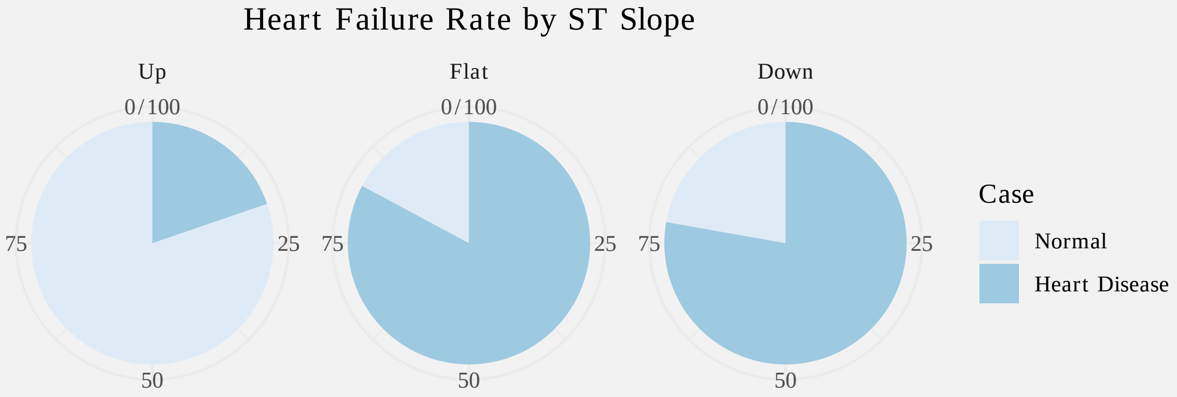
2.2.12 “ST_Slope” attribute

Count number of ST Slope types when exercising of normal patients and patients with a heart disease.

```
slope_disease <- heart %>%
        group_by(ST_Slope, HeartDisease) %>%
        summarise(count = n()) %>%
        mutate(percentage = count / sum(count) * 100)
slope_disease
# A tibble: 6 x 4
# Groups:   ST_Slope [3]
  ST_Slope HeartDisease  count percentage
  <fct>    <fct>         <int>      <dbl>
1 Up       Normal          317       80.3
2 Up       Heart Disease    78       19.7
3 Flat     Normal           79       17.2
4 Flat     Heart Disease   381       82.8
5 Down     Normal           14       22.2
6 Down     Heart Disease    49       77.8
```

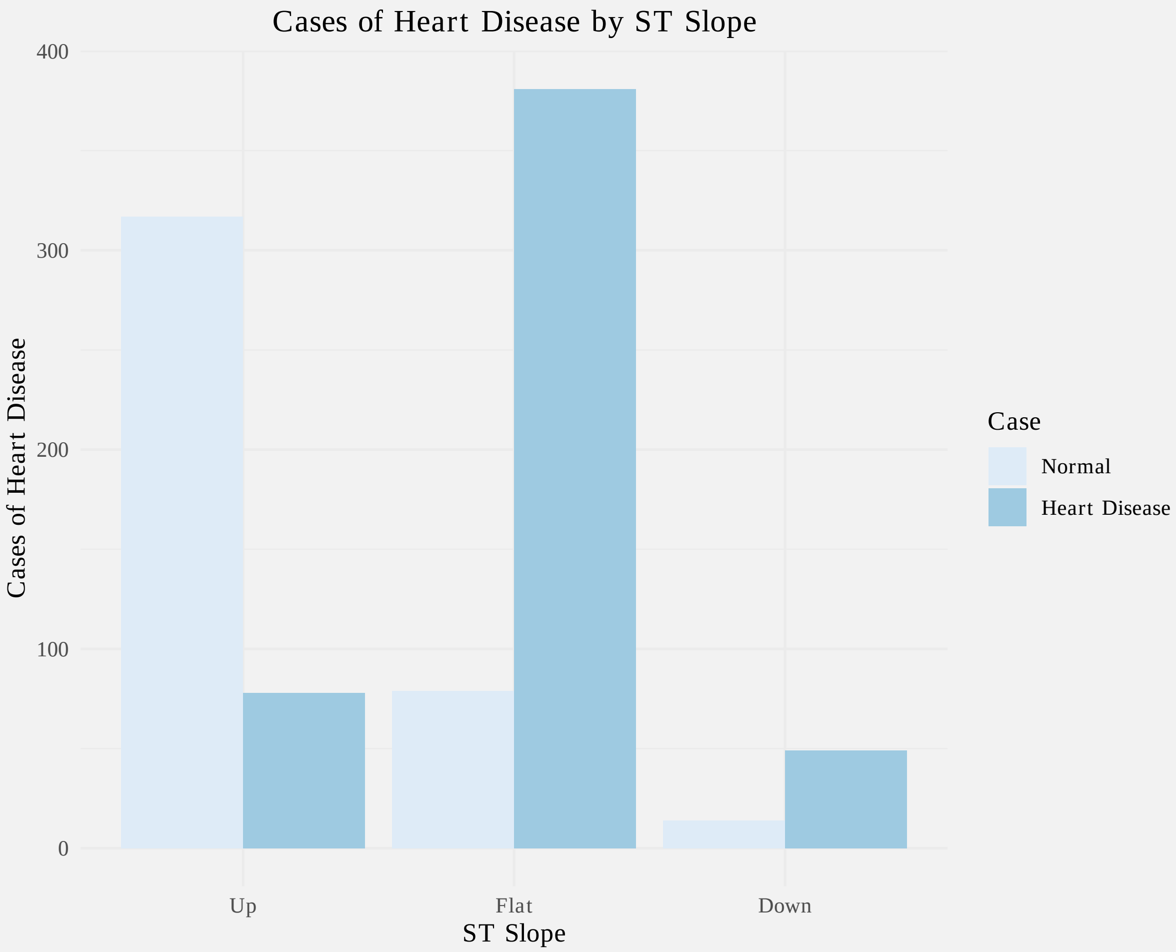
Graph a pie chart for ST Slope when exercising of normal patients and patients with a heart disease.

```
ggplot(slope_disease, aes(x = '', y = percentage, fill = HeartDisease)) +
    geom_bar(width = 2, stat = 'identity') +
    coord_polar('y', start = 0) +
    facet_wrap(~ST_Slope, ncol = 3, scale = 'fixed') +
    ggtitle('Heart Failure Rate by ST Slope') +
    xlab('') + ylab('') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



Graph a bar chart for ST Slope when exercising of normal patients and patients with a heart disease.

```
ggplot(slope_disease, aes(x = ST_Slope, y = count, fill = HeartDisease)) +
    geom_col(position = 'dodge') +
    ggtitle('Cases of Heart Disease by ST Slope') +
    xlab('ST Slope') +
    ylab('Cases of Heart Disease') +
    scale_fill_brewer(palette = 'Blues', name = 'Case',
                      labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



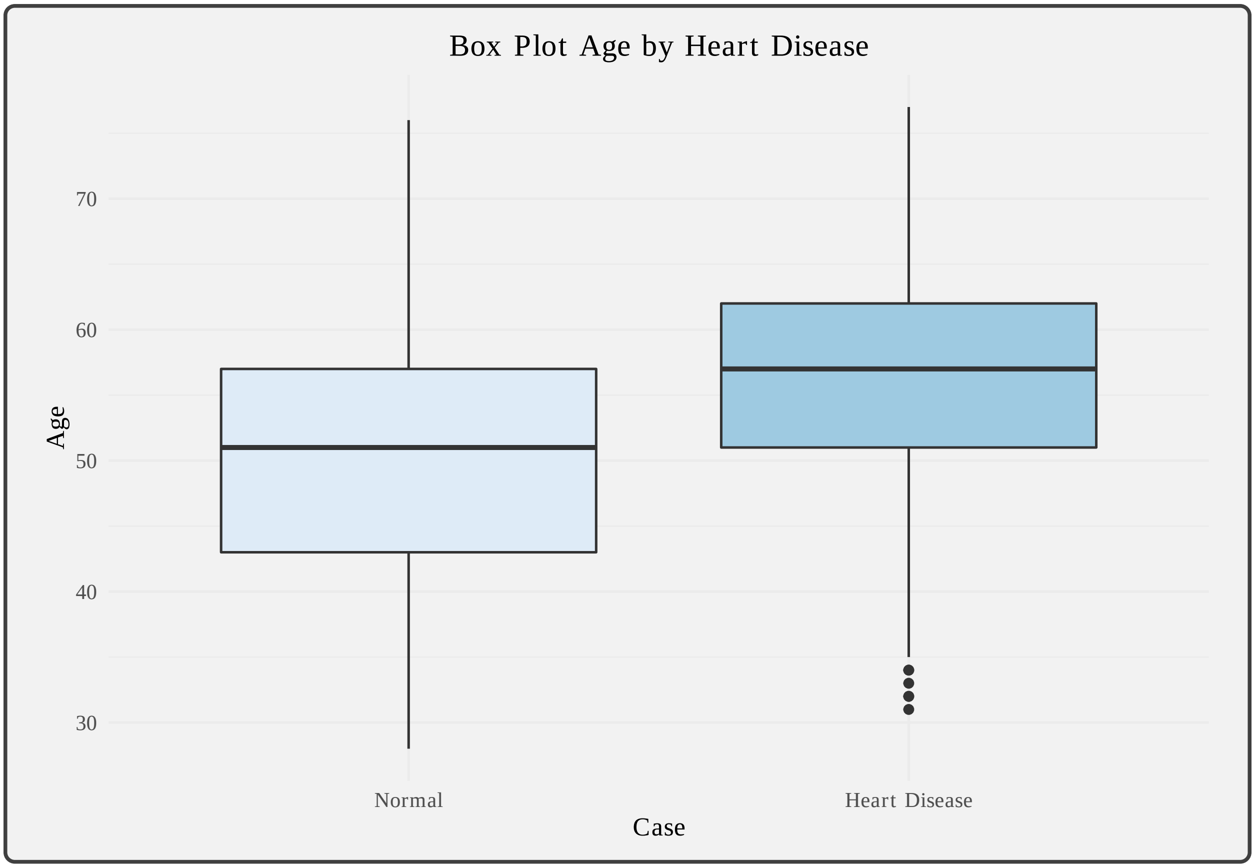
Comments:

- Down type have the least number of records.
- The number of records of Up and Flat type is almost equal.
- Patients, who have ST Slope Up when exercising, have the least chance of suffering a heart disease which is almost 20%.
- Patients, who have ST Slop Flat or Down when exercising, have very high chance of suffering a heart disease which are about 83% and 78%, respectively.

3 Inferential Statistics

3.1 Hypothesis Testing for Means and Proportions

3.1.1 “Age” attribute



H_0: The mean age of patients with heart disease less than or equal to the mean age of normal patients.

H_α: The mean age of patients with heart disease greater than the mean age of normal patients.

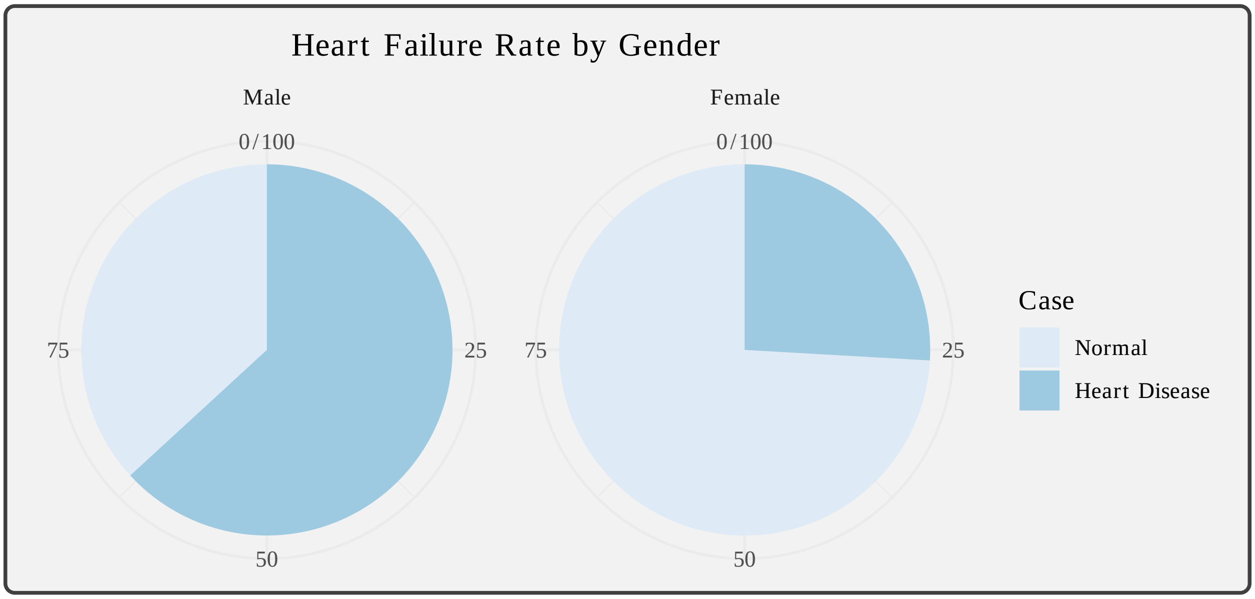
Hypothesis Testing: Use ***t.test*** function to test whether the mean age of patients with heart disease greater than the mean age of normal patients.

```
t.test(Age ~ HeartDisease, data = heart, alternative = 'less')
    Welch Two Sample t-test

data:  Age by HeartDisease
t = -8.8225, df = 843.69, p-value < 2.2e-16
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
     -Inf -4.35015
sample estimates:
       mean in group Normal mean in group Heart Disease
                   50.55122                    55.89961
```

Comment: Because p-value < 2.2e-16, which is very close to 0, is less than significant level $\alpha = 5\%$. We reject the null hypothesis H_0 and accept the alternative hypothesis H_α that the mean age of patients with heart disease greater than the mean age of normal patients.

3.1.2 “Sex” attribute



H_0: The rate of men have a heart disease less than or equal to the rate of women have a heart disease.

H_α: The rate of men have a heart disease greater than the rate of women have a heart disease.

Use ***table*** function to group and count the number of normal patients and patients with heart disease by gender. Then, we need to put the heart disease group in the first column.

```
gender_disease_table <- table(heart$Sex, heart$HeartDisease)
gender_disease_table <- gender_disease_table[, c('Heart Disease', 'Normal')]
gender_disease_table
         Heart Disease Normal
  Male             458    267
  Female            50    143
```

Hypothesis Testing: Use ***prop.test*** function to test whether the rate of men have a heart disease greater than the rate of women have a heart disease.

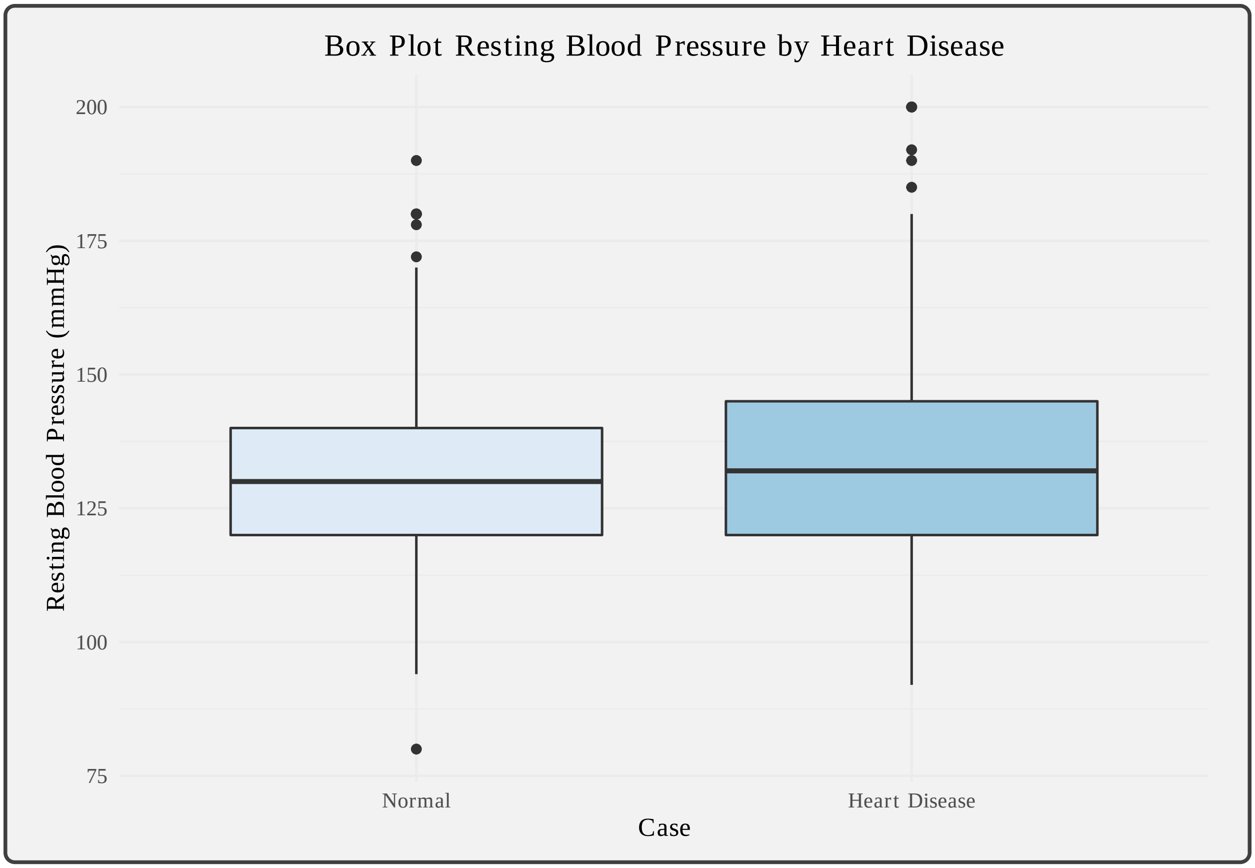
```
prop.test(gender_disease_table, correct = FALSE, alternative = 'greater')
    2-sample test for equality of proportions without continuity
    correction

data:  gender_disease_table
X-squared = 85.646, df = 1, p-value < 2.2e-16
alternative hypothesis: greater
95 percent confidence interval:
 0.3129991 1.0000000
sample estimates:
   prop 1    prop 2
0.6317241 0.2590674
```

Comment: Because p-value < 2.2e-16, which is very close to 0, is less than significant level $\alpha = 5\%$. We reject the null hypothesis H_0 and accept the alternative hypothesis H_α that the rate of men have a heart disease greater than the rate of women have a heart disease.

3.1.3 "RestingBP" attribute

From boxplot of resting blood pressure of descriptive statistics section. We can see that there is one instance of the data that has 0 resting blood pressure, so we need to replace its resting blood pressure with the mean of resting blood pressure. Then, we graph a new boxplot for resting blood pressure.



H_0: The mean of resting blood pressure of normal patients and patients with heart disease is equal.

H_α: The mean of resting blood pressure of normal patients and patients with heart disease is not equal.

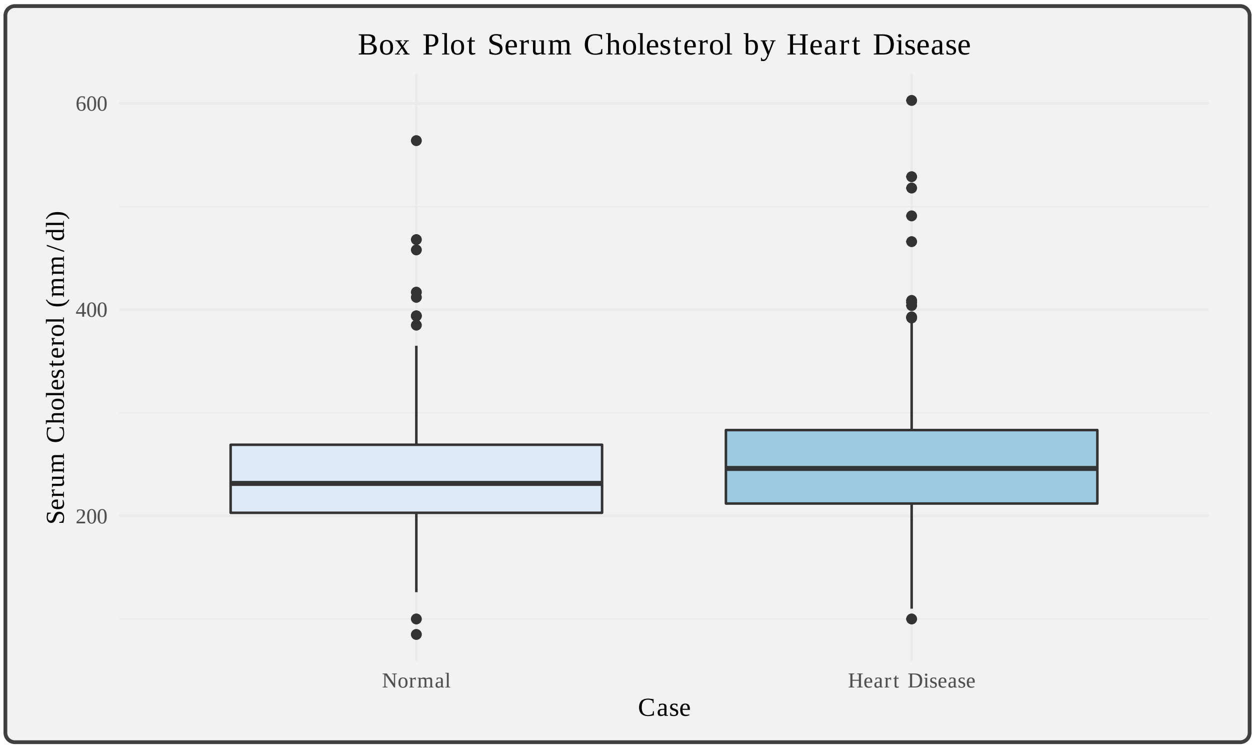
Hypothesis Testing: Use ***t.test*** function to test whether the mean of resting blood pressure of normal patients and patients with heart disease is equal.

```
t.test(RestingBP ~ HeartDisease, data = heart)
    Welch Two Sample t-test

data:  RestingBP by HeartDisease
t = -3.647, df = 910.43, p-value = 0.0002804
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -6.560854 -1.970064
sample estimates:
       mean in group Normal mean in group Heart Disease
                   130.1805                    134.4459
```

Comment: Because p-value = 0.0002804 is less than significant level $\alpha = 5\%$, we reject the null hypothesis H_0 accept the alternative hypothesis H_α that the mean of resting blood pressure of normal patients and patients with heart disease is not equal.

3.1.4 “Cholesterol” attribute



H_0: The mean of serum cholesterol of patients with heart disease is less than or equal to than the mean of serum cholesterol of normal patients.

H_α: The mean of serum cholesterol of patients with heart disease is greater than the mean of serum cholesterol of normal patients.

Hypothesis Testing: Use ***t.test*** function to test whether the mean of serum cholesterol of patients with heart disease is greater than the mean of serum cholesterol of normal patients.

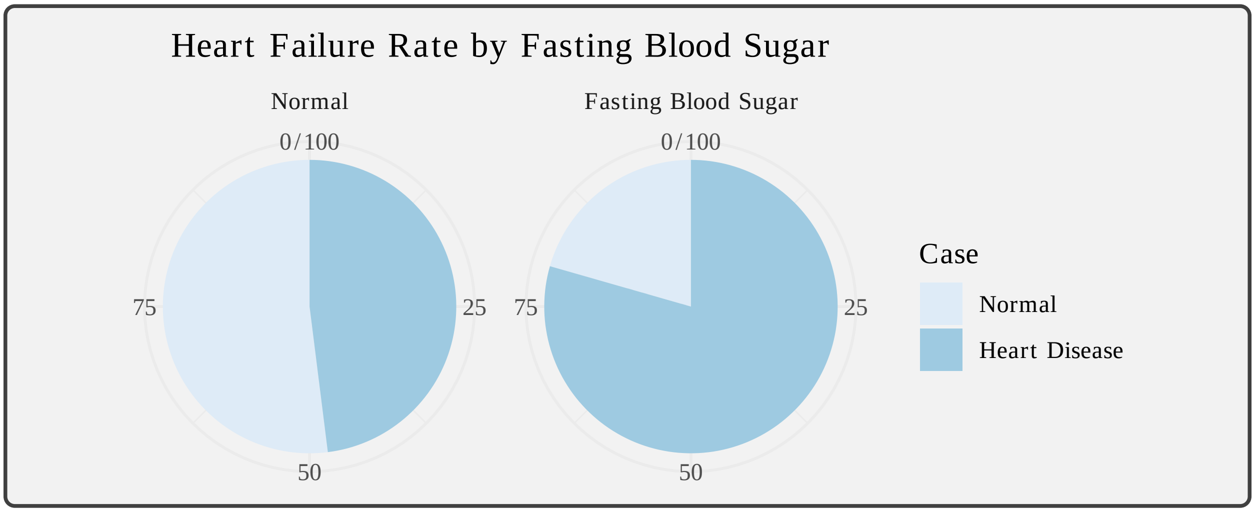
Noticed that, we use the **cholesterol_disease** dataframe created before because we need to remove records that have zero value of cholesterol.

```
t.test(cholesterol_disease$Cholesterol ~ cholesterol_disease$HeartDisease,
       alternative = 'less')
    Welch Two Sample t-test

data:  cholesterol_disease$Cholesterol by cholesterol_disease$HeartDisease
t = -2.833, df = 712.54, p-value = 0.002371
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
     -Inf -5.14612
sample estimates:
       mean in group Normal mean in group Heart Disease
                   238.7692                    251.0618
```

Comment: Because p-value = 0.002371 is less than significant level $\alpha = 5\%$. We reject the null hypothesis H_0 and accept the alternative hypothesis H_α that the mean of serum cholesterol of patients with heart disease is greater than the mean of serum cholesterol of normal patients.

3.1.5 “FastingBS” attribute



H_0: Heart failure rate of fasting blood sugar patients is less than or equal to heart failure rate of normal blood sugar patients.

H_α: Heart failure rate of fasting blood sugar patients is greater than heart failure rate of normal blood sugar patients.

Use ***table*** function to group and count the number of normal patients and patients with heart disease by each type of blood sugar. Then, we need to put the heart disease group in the first column.

```
fastingBS_disease_table <- table(heart$FastingBS, heart$HeartDisease)
fastingBS_disease_table <- fastingBS_disease_table[, c('Heart Disease', 'Normal')]
fastingBS_disease_table
                      Heart Disease Normal
  Normal                        338    366
  Fasting Blood Sugar           170     44
```

Hypothesis Testing: Use ***prop.test*** function to test whether heart failure rate of fasting blood sugar patients is greater than heart failure rate of normal blood sugar patients.

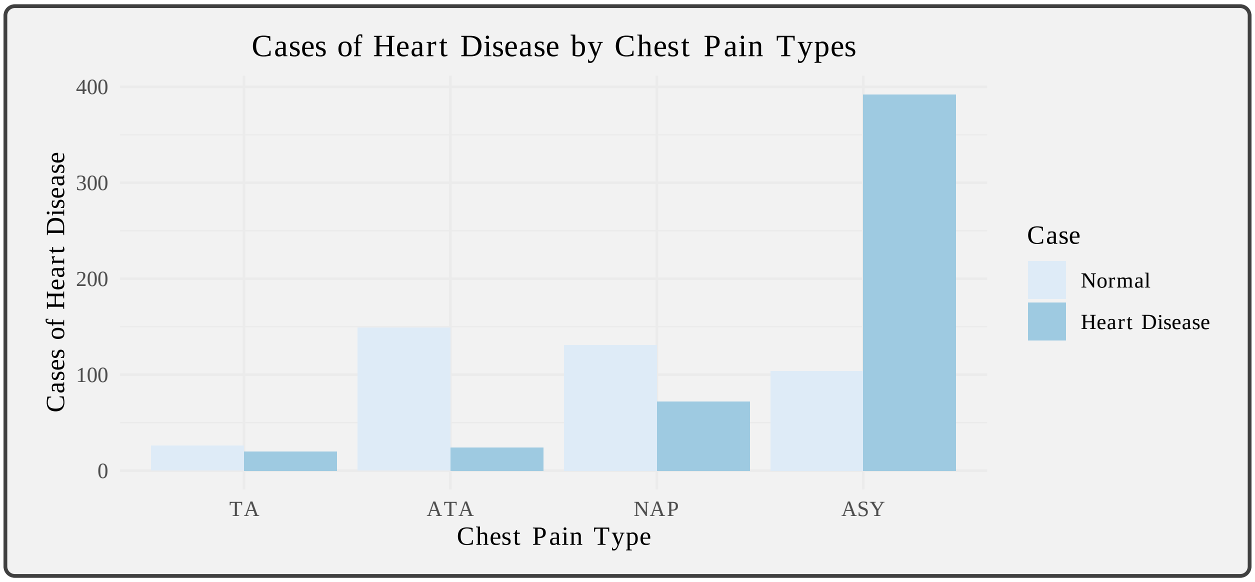
```
prop.test(fastingBS_disease_table, correct = FALSE, alternative = 'less')
    2-sample test for equality of proportions without continuity
    correction

data:  fastingBS_disease_table
X-squared = 65.586, df = 1, p-value = 2.781e-16
alternative hypothesis: less
95 percent confidence interval:
 -1.0000000 -0.2592859
sample estimates:
   prop 1    prop 2
0.4801136 0.7943925
```

Comment: Because p-value = 2.781e-16, which is very close to 0, is less than significant level $\alpha = 5\%$. We reject the null hypothesis H_0 and accept the alternative hypothesis H_α that heart failure rate of fasting blood sugar patients is greater than heart failure rate of normal blood sugar patients.

3.2 Test of Independence

3.2.1 “HeartDisease” and “ChestPainType”



We will test whether there is a relationship between type of chest pain of patients and whether they suffer a heart disease.

H_0: “HeartDisease” and “ChestPainType” are independent.

H_α: “HeartDisease” and “ChestPainType” are not independent.

Use ***table*** function to group and count the number of normal patients and patients with heart disease by each type of chest pain.

```
chestpain_disease <- table(heart$ChestPainType, heart$HeartDisease)
chestpain_disease
      Normal Heart Disease
  TA      26            20
  ATA    149            24
  NAP    131            72
  ASY    104           392
```

Hypothesis Testing: Use ***chisq.test*** function to test whether there is a relationship between type of chest pain of patients and whether they suffer a heart disease.

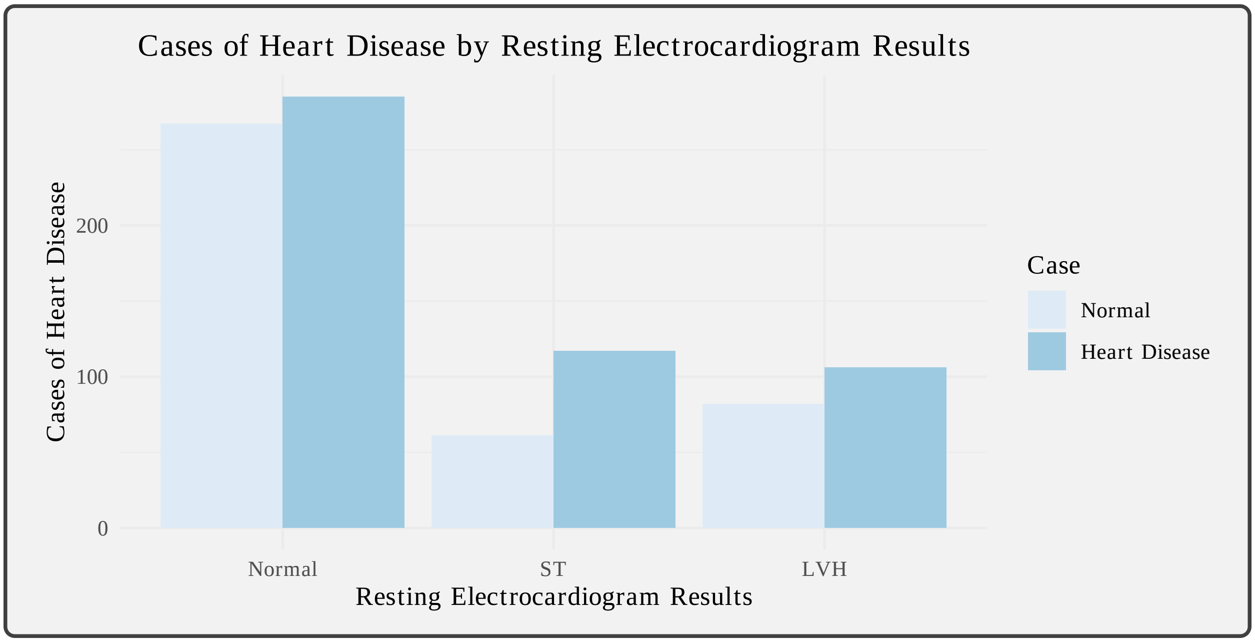
```
chisq.test(chestpain_disease)
    Pearson's Chi-squared test

data:  chestpain_disease
X-squared = 268.07, df = 3, p-value < 2.2e-16
```

Comment: Because p-value $<$ 2.2e-16, which is very close to 0, is less than the significant level $\alpha =$ 5%. We reject the null hypothesis H_0 and accept the alternative hypothesis H_α that “HeartDisease” and “ChestPainType” are not independent.

Hence, the test shows that there is a relationship between type of chest pain of patients and whether they suffer a heart disease.

3.2.2 "HeartDisease" and "RestingECG"



We will test whether there is a relationship between resting electrocardiogram result of patients and whether they suffer a heart disease.

H_0: "HeartDisease" and "RestingECG" are independent.

H_α: "HeartDisease" and "RestingECG" are not independent.

Use ***table*** function to group and count the number of normal patients and patients with heart disease by each resting electrocardiogram result.

```
restingECG_disease <- table(heart$RestingECG, heart$HeartDisease)
restingECG_disease
         Normal Heart Disease
  Normal    267           285
  ST         61           117
  LVH        82           106
```

Hypothesis Testing: Use ***chisq.test*** function to test whether there is a relationship between resting electrocardiogram result of patients and whether they suffer a heart disease.

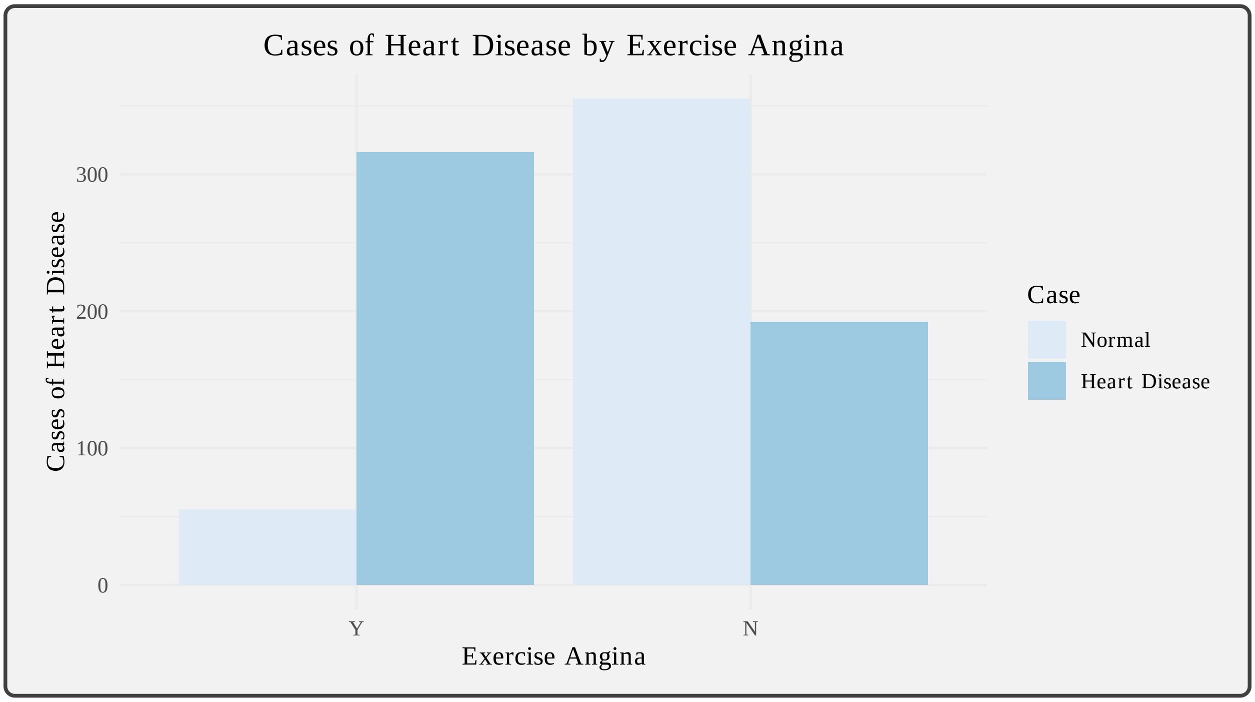
```
chisq.test(restingECG_disease)
    Pearson's Chi-squared test

data:  restingECG_disease
X-squared = 10.931, df = 2, p-value = 0.004229
```

Comment: Because p-value = 0.004229 is less than the significant level $\alpha = 5\%$. We reject the null hypothesis H_0 and accept the alternative hypothesis H_α that "HeartDisease" and "RestingECG" are not independent.

Hence, the test shows that there is a relationship between resting electrocardiogram result of patients and whether they suffer a heart disease.

3.2.3 “HeartDisease” and “ExerciseAngina”



We will test whether there is a relationship between whether patients have angina when exercising and whether they suffer a heart disease.

H_0: “HeartDisease” and “ExerciseAngina” are independent.

H_α: “HeartDisease” and “ExerciseAngina” are not independent.

Use ***table*** function to group and count the number of normal patients and patients with heart disease by whether they have angina when exercising.

```
angina_disease <- table(heart$ExerciseAngina, heart$HeartDisease)
angina_disease
    Normal Heart Disease
  Y     55            316
  N    355            192
```

Hypothesis Testing: Use ***chisq.test*** function to test whether there is a relationship between whether patients have angina when exercising and whether they suffer a heart disease.

```
chisq.test(angina_disease)
    Pearson's Chi-squared test with Yates' continuity correction

data:  angina_disease
X-squared = 222.26, df = 1, p-value < 2.2e-16
```

Comment: Because p-value < 2.2e-16, which is very close to 0, is less than the significant level $\alpha = 5\%$. We reject the null hypothesis H_0 and accept the alternative hypothesis H_α that “HeartDisease” and “ExerciseAngina” are not independent.

Hence, the test shows that there is a relationship between whether patients have angina when exercising and whether they suffer a heart disease.

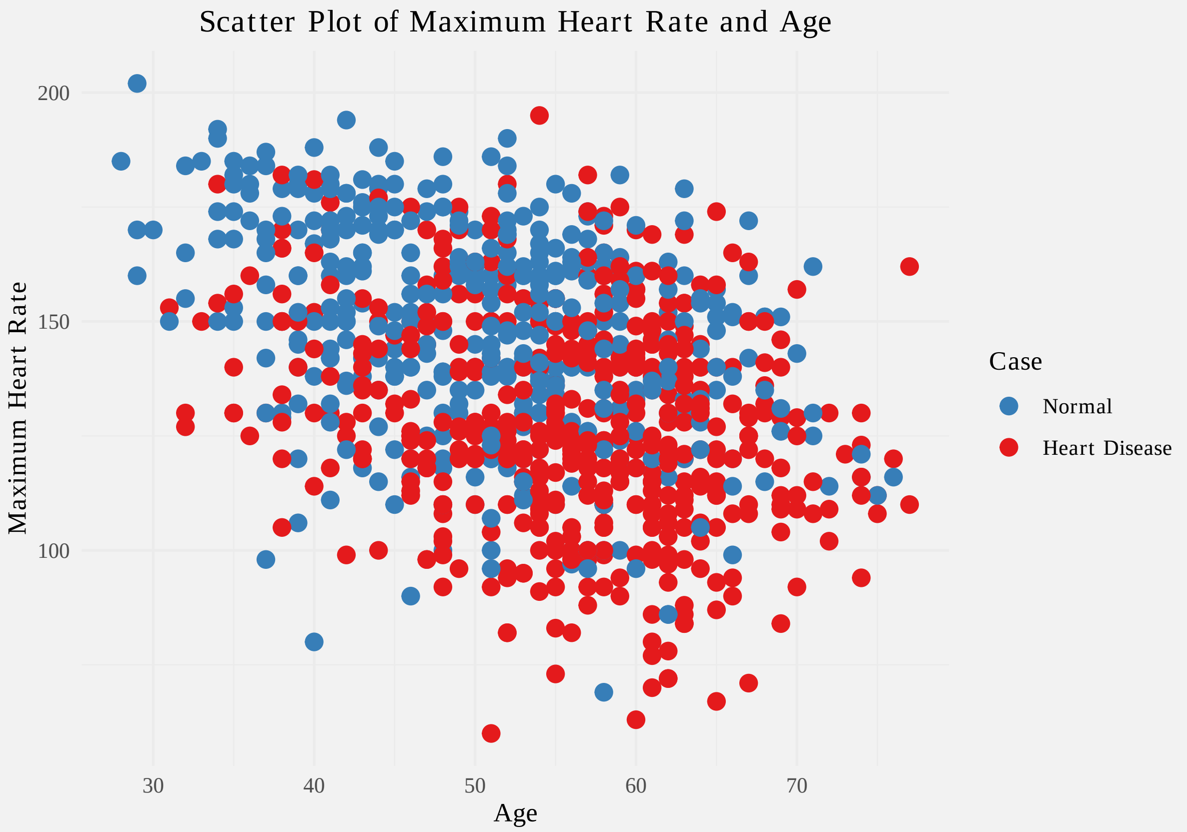
4 Regression

4.1 Simple Linear Regression Model

4.1.1 "MaxHR" from "Age"

Graph a scatter plot between "MaxHR" and "Age".

```r
ggplot(heart, aes(x = Age, y = MaxHR, color = HeartDisease)) +
    geom_point(alpha = 1, size = 3) +
    ggtitle("Scatter Plot of Maximum Heart Rate and Age") +
    xlab("Age") +
    ylab("Maximum Heart Rate") +
    scale_color_brewer(palette = 'Set1', direction = -1, name = 'Case',
                       labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



Comment: From the previous descriptive statistics and this scatter plot we can see that most of the patients with heart disease have age between 51 and 62 and maximum heart rate between 112 and 144.

We consider the least squares line is:

$$MaxHR = \beta_0 + \beta_1 \times Age$$

Use ***lm*** function to get the coefficients of our simple linear regression model.

```
maxHR_age_model <- lm(MaxHR ~ Age)
maxHR_age_model
Call:
lm(formula = MaxHR ~ Age)

Coefficients:
(Intercept)          Age
    191.990       -1.031
```

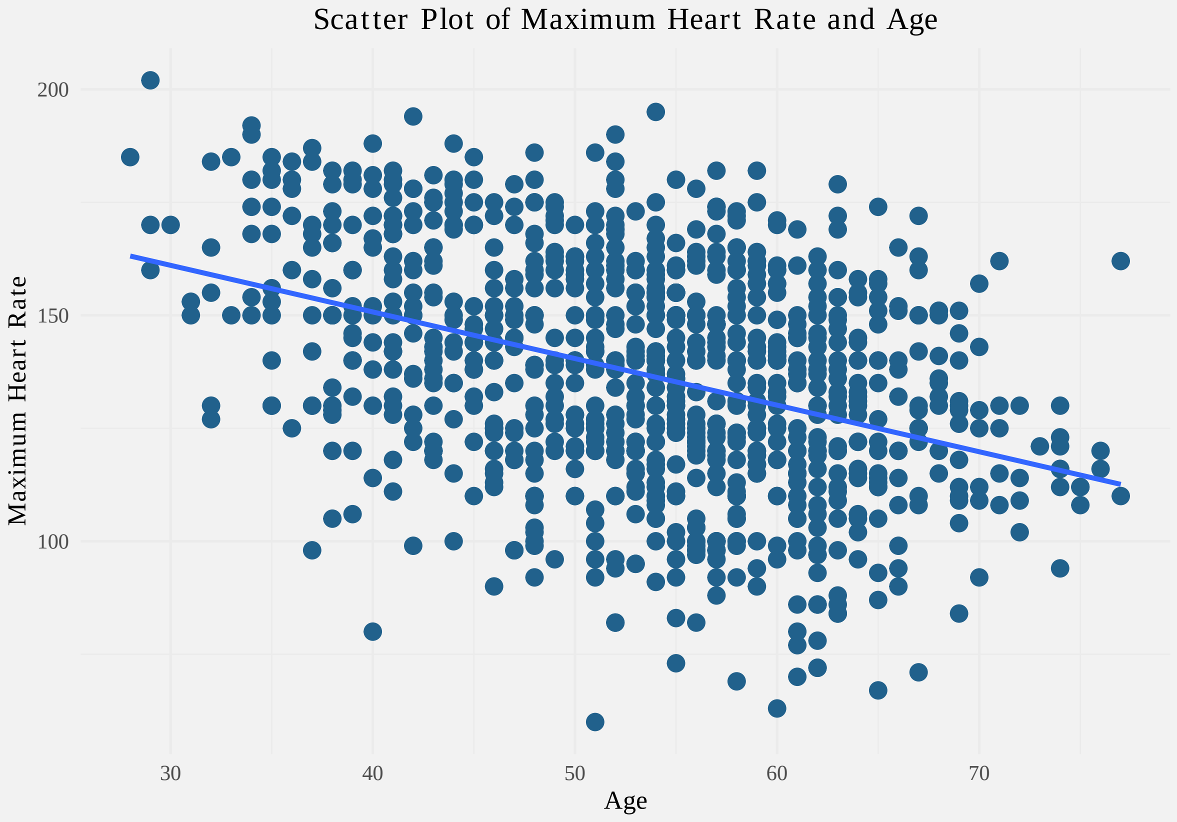
Hence, we have the equation:

$$MaxHR = 191.990 - 1.031 \times Age$$

Comment: The model presents that if the age is 0, the maximum heart rate is 191.990 (based on β_0) and if the age is increased by 1 unit, the maximum heart rate is decreased by 1.031 unit (based on β_1).

Fit the regression line to the scatter plot above by using ***geom_smooth*** function.

```
ggplot(heart, aes(x = Age, y = MaxHR)) +
    geom_point(alpha = 1, size = 3, color = '#21618c') +
    geom_smooth(method = 'lm', se = FALSE) + theme_minimal() +
    ggtitle("Scatter Plot of Maximum Heart Rate and Age") +
    xlab("Age") + ylab("Maximum Heart Rate") +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



Use ***summary*** function to extract more information about our model.

```
summary(maxHR_age_model)
Call:
lm(formula = MaxHR ~ Age)

Residuals:
    Min      1Q  Median      3Q     Max
-79.399 -15.922   0.726  18.196  58.695

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 191.99020    4.47820   42.87   <2e-16 ***
Age          -1.03121    0.08242  -12.51   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 23.54 on 916 degrees of freedom
Multiple R-squared:  0.146, Adjusted R-squared:  0.145
F-statistic: 156.5 on 1 and 916 DF,  p-value: < 2.2e-16
```

Comment: p-value $<$ 2.2e-16, which is very close to 0, is less than significant level $\alpha = 5\%$, so we can conclude that β_1 is different from 0. Hence, age has a statically significant effect on maximum heart rate.

Use ***confint*** function to estimate the 95% confidence interval for the coefficients.

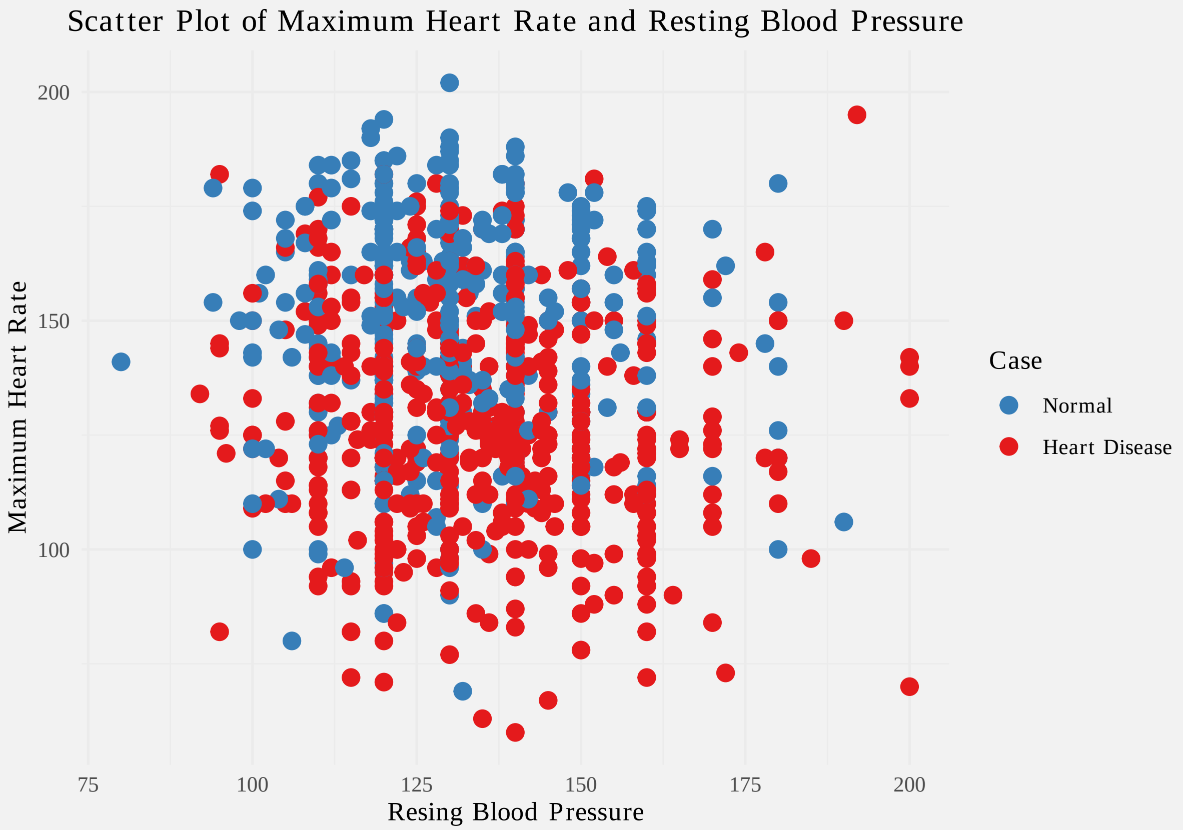
```
confint(maxHR_age_model)
                 2.5 %      97.5 %
(Intercept) 183.201480 200.7789274
Age          -1.192958  -0.8694569
```

Comment: The 95% confidence interval of β_0 is (183.201480, 200.7789274) and β_1 is (-1.192958, -0.8694569).

4.1.2 “MaxHR” from “RestingBP”

Graph a scatter plot between “MaxHR” and “RestingBP”.

```r
ggplot(heart, aes(x = RestingBP, y = MaxHR, color = HeartDisease)) +
    geom_point(alpha = 1, size = 3) +
    ggtitle("Scatter Plot of Maximum Heart Rate and Resting Blood Pressure") +
    xlab("Resing Blood Pressure") +
    ylab("Maximum Heart Rate") +
    scale_color_brewer(palette = 'Set1', direction = -1, name = 'Case',
                       labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



Comment: From the previous descriptive statistics and this scatter plot we can see that most of the patients with heart disease have resting blood pressure between 120 and 145 and maximum heart rate between 112 and 144.

We consider the least squares line is:

$$MaxHR = \beta_0 + \beta_1 \times ResingBP$$

Use ***lm*** function to get the coefficients of our simple linear regression model.

```
maxHR_restingBP_model <- lm(MaxHR ~ RestingBP)
maxHR_restingBP_model
Call:
lm(formula = MaxHR ~ RestingBP)

Coefficients:
(Intercept)    RestingBP
   157.2257      -0.1542
```

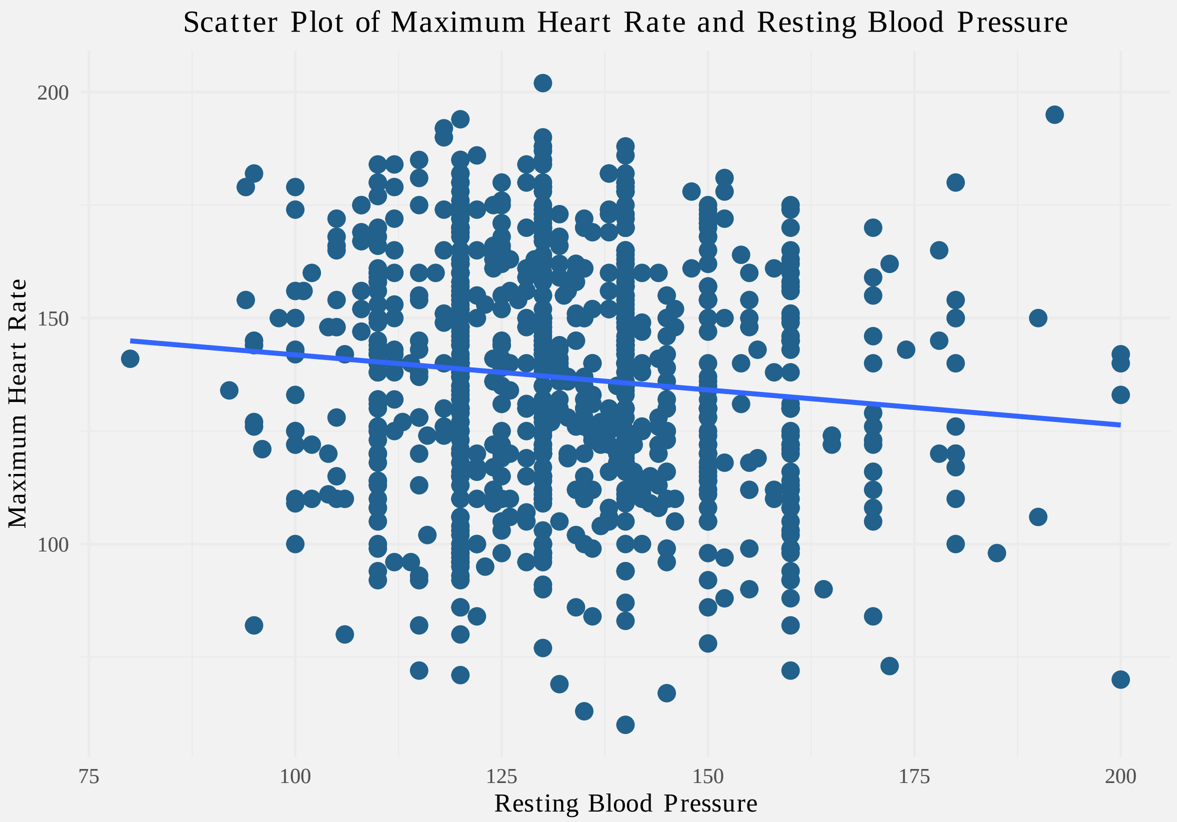
Hence, we have the equation:

$$MaxHR = 157.2257 - 0.1542 \times RestingBP$$

Comment: The model presents that if the resting blood pressure is 0, the maximum heart rate is 157.2257 (based on β_0) and if the resting blood pressure is increased by 1 unit, the maximum heart rate is decreased by 0.1542 unit (based on β_1).

Fit the regression line to the scatter plot above by using ***geom_smooth*** function.

```
ggplot(heart, aes(x = RestingBP, y = MaxHR)) +
    geom_point(alpha = 1, size = 3, color = '#21618c') +
    geom_smooth(method = 'lm', se = FALSE) + theme_minimal() +
    ggtitle("Scatter Plot of Maximum Heart Rate and Resting Blood Pressure") +
    xlab("Resting Blood Pressure") + ylab("Maximum Heart Rate") +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



Use ***summary*** function to extract more information about our model.

```
summary(maxHR_restingBP_model)
Call:
lm(formula = MaxHR ~ RestingBP)

Residuals:
    Min      1Q  Median      3Q     Max 
-75.637 -17.720   0.319  19.343  67.382 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 157.22571    6.03590  26.048  < 2e-16 ***
RestingBP    -0.15421    0.04515  -3.415 0.000665 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 25.31 on 916 degrees of freedom
Multiple R-squared:  0.01257,   Adjusted R-squared:  0.0115 
F-statistic: 11.66 on 1 and 916 DF,  p-value: 0.000665
```

Comment: p-value = 0.000665, which is less than significant level $\alpha = 5\%$, so we can conclude that β_1 is different from 0. Hence, resting blood pressure has a statically significant effect on maximum heart rate.

Use ***confint*** function to estimate the 95% confidence interval for the coefficients.

```
confint(maxHR_restingBP_model)
                  2.5 %       97.5 %
(Intercept) 145.3799126 169.07150955
RestingBP    -0.2428169  -0.06559517
```

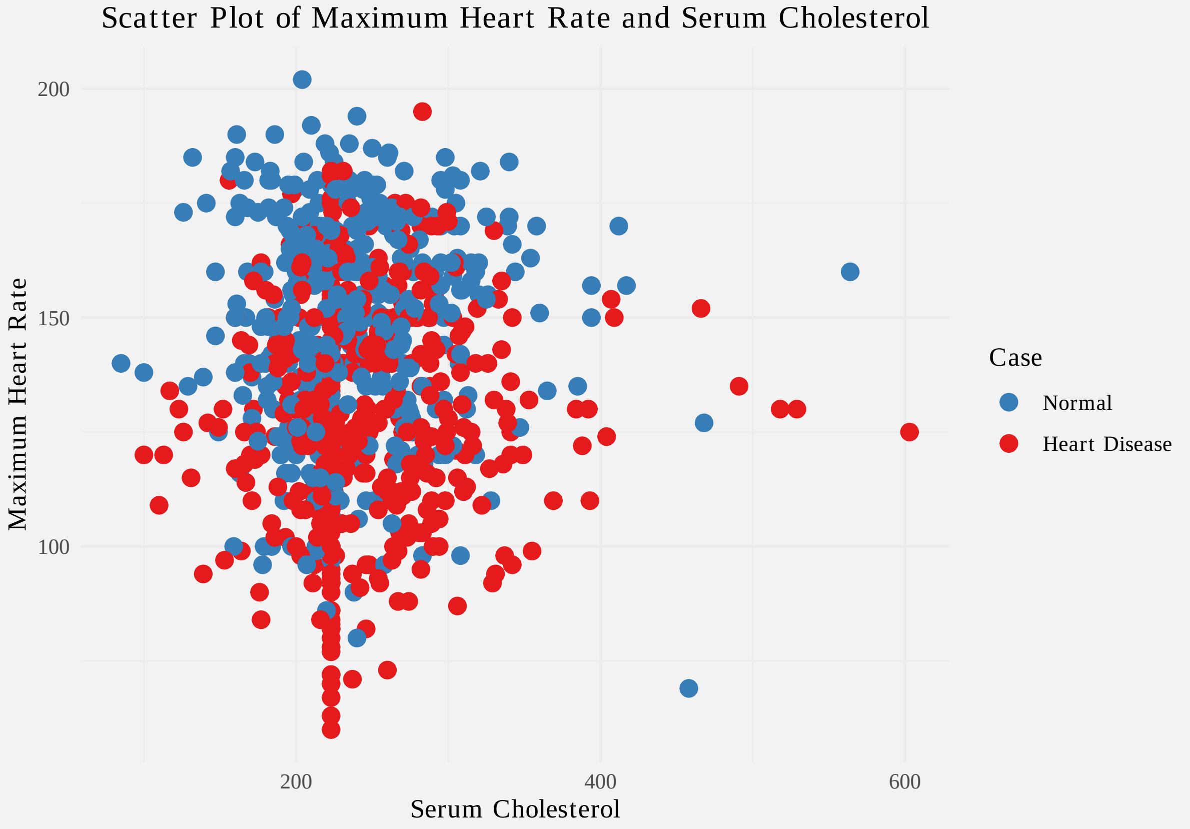
Comment: The 95% confidence interval of β_0 is (145.3799126, 169.07150955) and β_1 is (-0.2428169, -0.06559517).

4.1.3 “MaxHR” from “Cholesterol”

First, because there are many outliers (0 value of cholesterol) we need to replace all records that have 0 cholesterol with the median of cholesterol. Then, we graph a scatter plot between “MaxHR” and “Cholesterol”.

```r
cholesterol_disease <- heart[, c('Cholesterol', 'MaxHR', 'HeartDisease')]
cholesterol_disease$Cholesterol[Cholesterol == 0] <- median(Cholesterol)

ggplot(cholesterol_disease, aes(x = Cholesterol, y = MaxHR, color = HeartDisease)) +
    geom_point(alpha = 1, size = 3) +
    ggtitle("Scatter Plot of Maximum Heart Rate and Serum Cholesterol") +
    xlab("Serum Cholesterol") +
    ylab("Maximum Heart Rate") +
    scale_color_brewer(palette = 'Set1', direction = -1, name = 'Case',
                       labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



Comment: From the previous descriptive statistics and this scatter plot we can see that most of the patients with heart disease have serum cholesterol between 212 and 284 and maximum heart rate between 112 and 144.

We consider the least squares line is:

$$MaxHR = \beta_0 + \beta_1 \times Cholesterol$$

Use ***lm*** function to get the coefficients of our simple linear regression model.

```
maxHR_cholesterol_model <- lm(MaxHR ~ Cholesterol, data = cholesterol_disease)
maxHR_cholesterol_model
Call:
lm(formula = MaxHR ~ Cholesterol, data = cholesterol_disease)

Coefficients:
(Intercept)  Cholesterol
  133.77597      0.01261
```

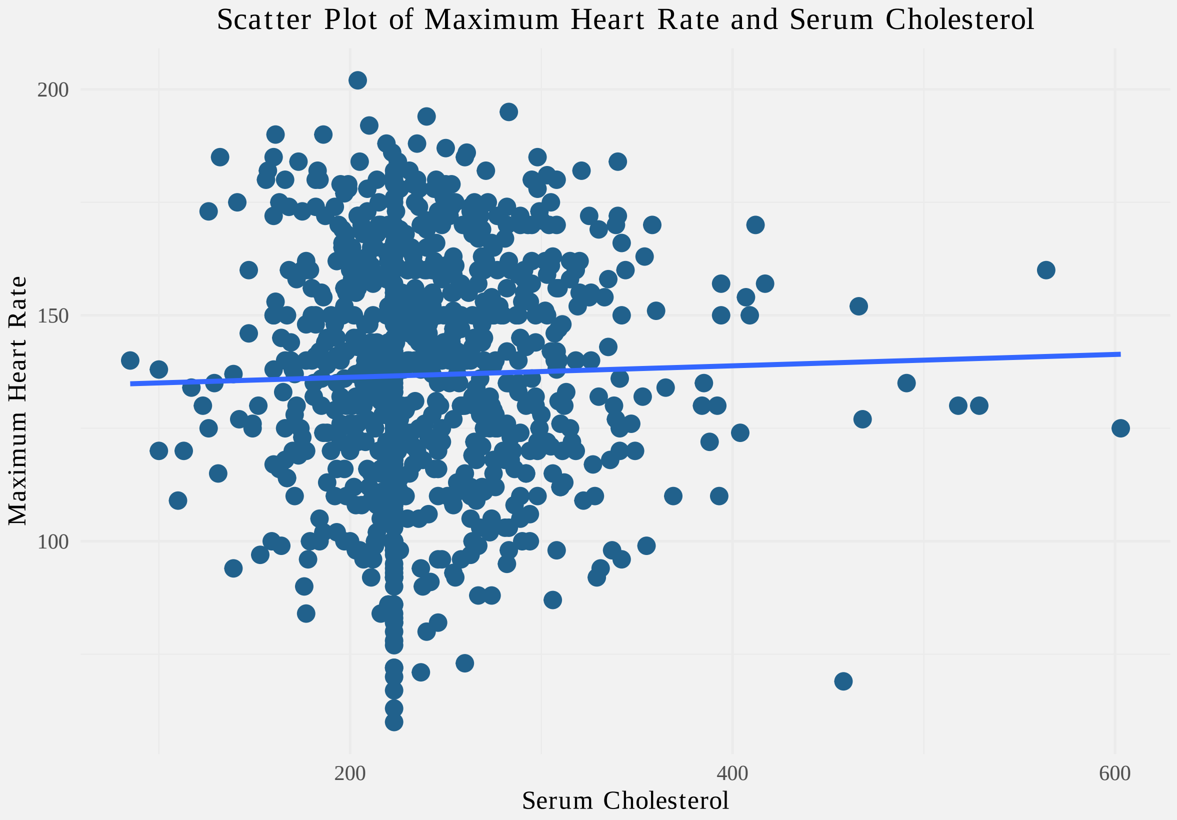
Hence, we have the equation:

$$MaxHR = 133.77597 + 0.01261 \times Cholesterol$$

Comment: The model presents that if the serum cholesterol is 0, the maximum heart rate is 133.77597 (based on β_0) and if the serum cholesterol is increased by 1 unit, the maximum heart rate is increased by 0.01261 unit (based on β_1).

Fit the regression line to the scatter plot above by using ***geom_smooth*** function.

```
ggplot(cholesterol_disease, aes(x = Cholesterol, y = MaxHR)) +
    geom_point(alpha = 1, size = 3, color = '#21618c') +
    geom_smooth(method = 'lm', se = FALSE) + theme_minimal() +
    ggtitle("Scatter Plot of Maximum Heart Rate and Serum Cholesterol") +
    xlab("Serum Cholesterol") + ylab("Maximum Heart Rate") +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



Use ***summary*** function to extract more information about our model.

```
summary(maxHR_cholesterol_model)
Call:
lm(formula = MaxHR ~ Cholesterol, data = cholesterol_disease)

Residuals:
    Min      1Q  Median      3Q     Max
-76.588 -17.455   1.538  18.601  65.652

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 133.77597    3.84081  34.830   <2e-16 ***
Cholesterol   0.01261    0.01558   0.809    0.418
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 25.47 on 916 degrees of freedom
Multiple R-squared:  0.0007147, Adjusted R-squared:  -0.0003762
F-statistic: 0.6551 on 1 and 916 DF,  p-value: 0.4185
```

Comment: p-value = 0.4185, which is much greater than significant level $\alpha = 5\%$, so we can conclude that β_1 is equal to 0. Hence, cholesterol has no statically significant effect on maximum heart rate.

Use ***confint*** function to estimate the 95% confidence interval for the coefficients.

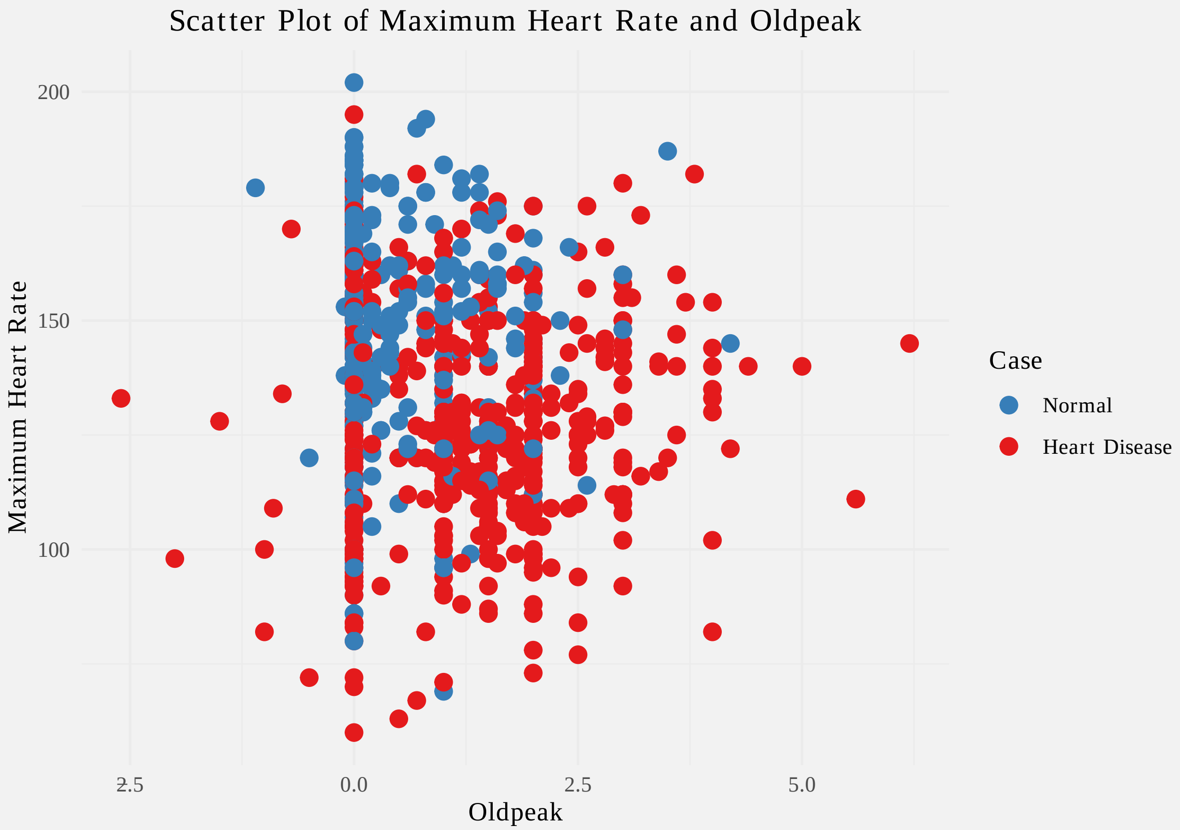
```
confint(maxHR_cholesterol_model)
                   2.5 %       97.5 %
(Intercept) 126.23816673 141.31376921
Cholesterol  -0.01796359   0.04318081
```

Comment: The 95% confidence interval of β_0 is (126.23816673, 141.31376921) and β_1 is (-0.01796359, 0.04318081).

4.1.4 “MaxHR” from “Oldpeak”

Graph a scatter plot between “MaxHR” and “Oldpeak”.

```
ggplot(heart, aes(x = Oldpeak, y = MaxHR, color = HeartDisease)) +
    geom_point(alpha = 1, size = 3) +
    ggtitle("Scatter Plot of Maximum Heart Rate and Oldpeak") +
    xlab("Oldpeak") +
    ylab("Maximum Heart Rate") +
    scale_color_brewer(palette = 'Set1', direction = -1, name = 'Case',
                       labels = c('Normal', 'Heart Disease')) +
    theme_minimal() +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



Comment: From the previous descriptive statistics and this scatter plot we can see that most of the patients with heart disease have old-peak between 0.0 and 2.0 and maximum heart rate between 112 and 144.

We consider the least squares line is:

$$MaxHR = \beta_0 + \beta_1 \times Oldpeak$$

Use ***lm*** function to get the coefficients of our simple linear regression model.

```
maxHR_oldpeak_model <- lm(MaxHR ~ Oldpeak)
maxHR_oldpeak_model
Call:
lm(formula = MaxHR ~ Oldpeak)

Coefficients:
(Intercept)      Oldpeak
    140.213       -3.836
```

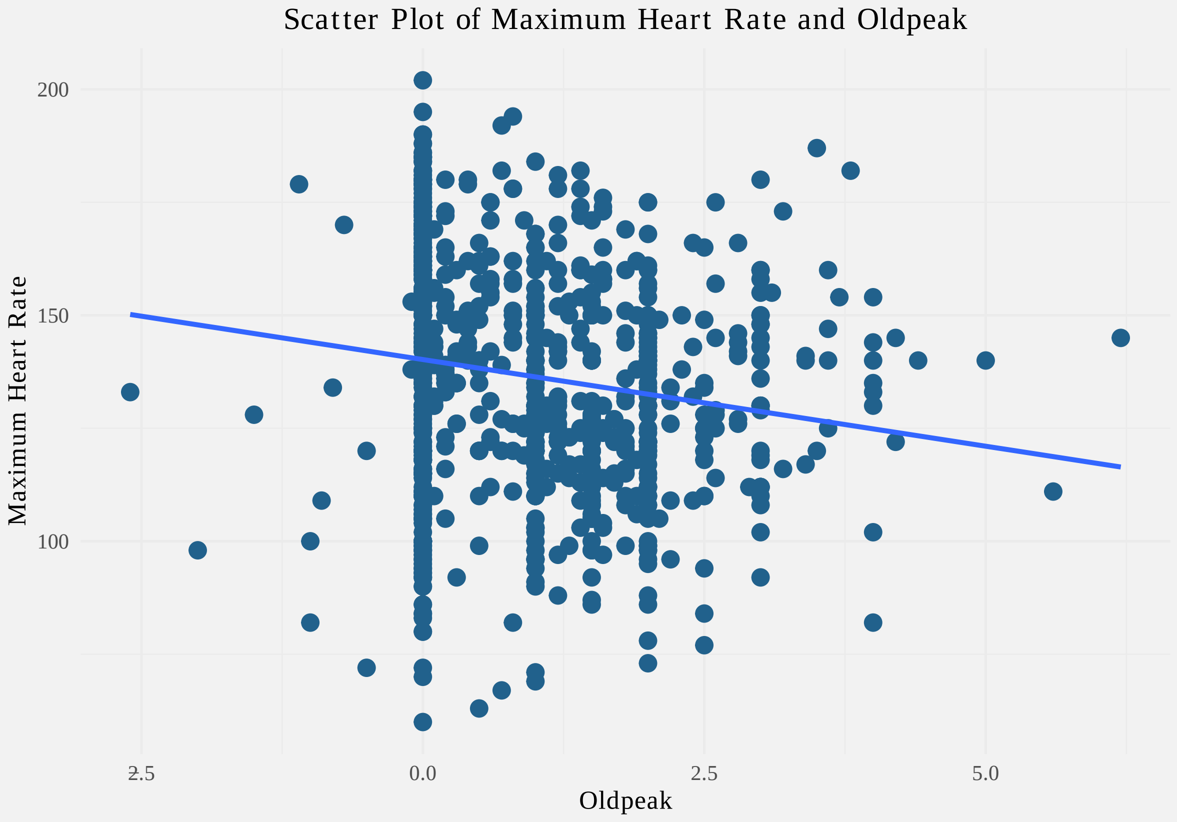
Hence, we have the equation:

$$MaxHR = 140.213 - 3.836 \times Oldpeak$$

Comment: The model presents that if the old-peak is 0, the maximum heart rate is 140.213 (based on β_0) and if the old-peak is increased by 1 unit, the maximum heart rate is decreased by 3.836 unit (based on β_1).

Fit the regression line to the scatter plot above by using ***geom_smooth*** function.

```
ggplot(heart, aes(x = Oldpeak, y = MaxHR)) +
    geom_point(alpha = 1, size = 3, color = '#21618c') +
    geom_smooth(method = 'lm', se = FALSE) + theme_minimal() +
    ggtitle("Scatter Plot of Maximum Heart Rate and Oldpeak") +
    xlab("Oldpeak") + ylab("Maximum Heart Rate") +
    theme(text = element_text(family = 'CMU Serif'),
          plot.title = element_text(hjust = 0.5))
```



Use ***summary*** function to extract more information about our model.

```
summary(maxHR_oldpeak_model)
Call:
lm(formula = MaxHR ~ Oldpeak)

Residuals:
    Min      1Q  Median      3Q     Max
-80.213 -17.528  -0.213  19.150  61.787

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 140.2132     1.0797 129.858  < 2e-16 ***
Oldpeak      -3.8359     0.7785  -4.927 9.89e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 25.14 on 916 degrees of freedom
Multiple R-squared:  0.02582,   Adjusted R-squared:  0.02476
F-statistic: 24.28 on 1 and 916 DF,  p-value: 9.886e-07
```

Comment: p-value = 9.886e-07, which is very close to 0, is less than significant level $\alpha = 5\%$, so we can conclude that β_1 is different from 0. Hence, old-peak has a statically significant effect on maximum heart rate.

Use ***confint*** function to estimate the 95% confidence interval for the coefficients.

```
confint(maxHR_oldpeak_model)
                 2.5 %     97.5 %
(Intercept) 138.094133 142.332246
Oldpeak      -5.363687  -2.308073
```

Comment: The 95% confidence interval of β_0 is (138.094133, 142.332246) and β_1 is (-5.363687, -2.308073).

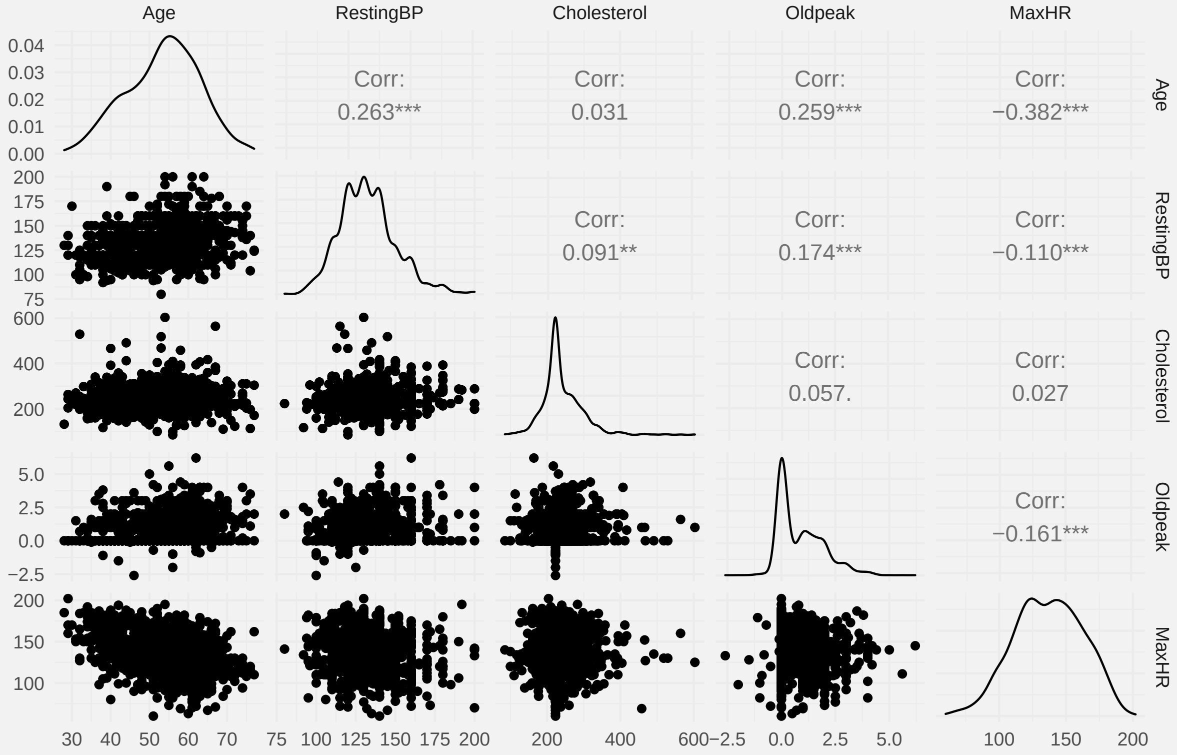
4.2 Multiple Linear Regression Model

We consider the multiple linear regression model of maximum heart rate from age, resting blood pressure, cholesterol and old-peak:

$$MaxHR = \beta_0 + \beta_1 \times Age + \beta_2 \times RestingBP + \beta_3 \times Cholesterol + \beta_4 \times Oldpeak$$

First, we need to create a new dataframe that only includes these attributes. Again, let take a look at the scatter plot of these attributes with maximum heart rate (the last row of the graph).

```
maxHR <- heart[, c('Age', 'RestingBP', 'Cholesterol', 'Oldpeak', 'MaxHR')]
maxHR$Cholesterol <- cholesterol_disease$Cholesterol
ggpairs(maxHR) +
    theme_minimal()
```



Use ***lm*** function to get the coefficients of our multiple linear regression model.

```
maxHR_model <- lm(MaxHR ~ Age + RestingBP + Cholesterol + Oldpeak, data = maxHR)
maxHR_model
Call:
lm(formula = MaxHR ~ Age + RestingBP + Cholesterol + Oldpeak,
    data = maxHR)

Coefficients:
(Intercept)          Age    RestingBP  Cholesterol      Oldpeak
 187.147263    -0.983217    -0.008287     0.019998    -1.620203
```

Hence, we have the equation:

$$MaxHR = 187.147263 - 0.983217 \times Age - 0.008287 \times RestingBP + 0.019998 \times Cholesterol - 1.620203 \times Oldpeak$$

Comments: Based on β_0, β_1, β_2, β_3 and β_4, we have:

- If the age, resting blood pressure, serum cholesterol and old-peak are all 0, the maximum heart rate is 187.147263.
- If the age is increased by 1 unit, the maximum heart rate is decreased by 0.983217 unit.
- If the resting blood pressure is increased by 1 unit, the maximum heart rate is decreased by 0.008287 unit.
- If the serum cholesterol is increased by 1 unit, the maximum heart rate is increased by 0.019998 unit.
- If the old-peak is increased by 1 unit, the maximum heart rate is decreased by 1.620203 unit.

Use ***summary*** function to extract more information about our model.

```
summary(maxHR_model)
Call:
lm(formula = MaxHR ~ Age + RestingBP + Cholesterol + Oldpeak,
    data = maxHR)

Residuals:
    Min      1Q  Median      3Q     Max
-80.302 -15.679   0.838  18.169  56.878

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 187.147263   7.230411  25.883   <2e-16 ***
Age          -0.983217   0.087495 -11.237   <2e-16 ***
RestingBP    -0.008287   0.045150  -0.184    0.854
Cholesterol   0.019998   0.014447   1.384    0.167
Oldpeak      -1.620203   0.758774  -2.135    0.033 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 23.5 on 913 degrees of freedom
Multiple R-squared:  0.1518,    Adjusted R-squared:  0.1481
F-statistic: 40.86 on 4 and 913 DF,  p-value: < 2.2e-16
```

Comments:

- P-value of β_1 is less than 2e-16, which is very close to 0, is less than significant level $\alpha = 5\%$, so we can conclude that β_1 is different from 0. Hence, age has a statically significant effect on maximum heart rate.
- On the other hands, p-value of β_2 and β_3 are 0.854 and 0.167 respectively, which are much greater than significant level $\alpha = 5\%$, so we can conclude that they are equal to 0. Hence, resting blood pressure and cholesterol have no statically significant effect on maximum heart rate.
- P-value of β_4 is 0.033, which is less than significant level $\alpha = 5\%$, so we can conclude that β_4 is different from 0. Hence, old-peak has a statically significant effect on maximum heart rate.

Use ***confint*** function to estimate the 95% confidence interval for the coefficients.

```
confint(maxHR_model)
                   2.5 %       97.5 %
(Intercept) 172.957105713 201.33741945
Age          -1.154932176  -0.81150189
RestingBP    -0.096898073   0.08032347
Cholesterol  -0.008356139   0.04835122
Oldpeak      -3.109345690  -0.13105958
```

Comment: The 95% confidence interval:

- β_0: (172.957105713, 201.33741945).
- β_1: (-1.154932176, -0.81150189).
- β_2: (-0.096898073, 0.08032347).
- β_3: (-0.008356139, 0.04835122).
- β_4: (-3.109345690, -0.13105958).

From the results showed above, we can just use Age and Oldpeak attribute to build a new multiple linear regression and compare it with the first one.

$$MaxHR = \beta_0 + \beta_1 \times Age + \beta_2 \times Oldpeak$$

```
maxHR_age_oldpeak_model <- lm(MaxHR ~ Age + Oldpeak)
summary(maxHR_age_oldpeak_model)
Call:
lm(formula = MaxHR ~ Age + Oldpeak)

Residuals:
    Min      1Q  Median      3Q     Max
-80.687 -15.778   0.629  18.252  57.267

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 190.91771    4.49889  42.437   <2e-16 ***
Age          -0.98491    0.08516 -11.565   <2e-16 ***
Oldpeak      -1.58326    0.75316  -2.102   0.0358 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 23.5 on 915 degrees of freedom
Multiple R-squared:  0.1501,    Adjusted R-squared:  0.1482
F-statistic: 80.78 on 2 and 915 DF,  p-value: < 2.2e-16
```

Hence, we have the equation:

$$MaxHR = 190.91771 - 0.98491 \times Age - 1.58326 \times Oldpeak$$

Comments:

- If the age and the old-peak are 0, the maximum heart rate is 190.91771 (based on β_0).
- If the age is increased by 1 unit, the maximum heart rate is decreased by 0.98491 unit (based on β_1).
- If the old-peak is increased by 1 unit, the maximum heart rate is decreased by 1.58326 unit (based on β_2).
- P-value of β_1 and β_2 still indicate that age and old-peak have a statically significant effect on maximum heart rate.

Use ***confint*** function to estimate the 95% confidence interval for the coefficients.

```
confint(maxHR_age_oldpeak_model)
                  2.5 %      97.5 %
(Intercept) 182.088372 199.7470465
Age          -1.152046  -0.8177743
Oldpeak      -3.061391  -0.1051372
```

Comment: The 95% confident interval:

- β_0: (182.088372, 199.7470465).
- β_1: (-1.152046, -0.8177743).
- β_2: (-3.061391, -0.1051372).

Compare the new model with the first one by using ***anova*** function.

```
anova(maxHR_age_oldpeak_model, maxHR_model)
Analysis of Variance Table

Model 1: MaxHR ~ Age + Oldpeak
Model 2: MaxHR ~ Age + RestingBP + Cholesterol + Oldpeak
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    915 505224
2    913 504163  2    1060.9 0.9606 0.3831
```

Comment: Because Pr(>F) = 0.3831, which is much greater than significant level $\alpha = 5\%$. Hence, there is strong evidence to accept that coefficients of resting blood pressure and serum cholesterol is equal to 0.

Conclusion About Multiple Linear Regression Model: After showing the ineffectiveness of resting blood pressure and serum cholesterol attributes to our model, we would choose **maxHR_age_oldpeak_model**, which was built from age and old-peak attributes, to estimate the maximum heart rate attribute:

$$MaxHR = 190.91771 - 0.98491 \times Age - 1.58326 \times Oldpeak$$