INTRODUCTION TO DATA SCIENCE [C]

MIDTERM PROJECT REPORT

Group: 06

SUBMITTED BY

ID	NAME
20-43257-1	MD. SAHILUR RAHMAN
20-43507-1	S.M. ADNAN MAHMUD

SUBMITTED TO

TOHEDUL ISLAM
Assistant Professor, Computer Science

Dataset Description: 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, hyperlipidaemia or already established disease) need early detection and management wherein a machine learning model can be of great help.

Attribute Information

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**: output class [1: heart disease, 0: Normal]

1.Input CSV File & install and import libraries

```
install.packages("cowplot")
install.packages("tidyverse")
install.packages("GGally")
install.packages("psych")
install.packages("matrixStats")
library(cowplot)
library(GGally)
library(tidyverse)
library(psych)
```

input the dataset CSV file

```
data <-
read_csv('C:/Users/ungab/OneDrive/Desktop/Dataset.csv',show_col_types = FALSE)
data
```

```
     Age Sex   ChestPainType RestingBP Cholesterol FastingBS RestingECG
   <dbl> <chr> <chr>             <dbl>       <dbl>     <dbl> <chr>
 1    40 M     ATA                 140         289         0 Normal
 2    49 F     NAP                 160         180         0 Normal
 3    37 M     ATA                 130         283         0 ST
 4    NA F     ASY                 138         214         0 Normal
 5    54 M     NAP                -150         195         0 Normal
 6    39 M     NAP                 120         339         0 Normal
 7    45 F     ATA                 130         237         0 Normal
 8    54 M     ATA                 110         208         0 Normal
 9    37 NA    ASY                 140         207         0 Normal
10    48 F     ATA                 120         284         0 Normal
# i 140 more rows
# i 5 more variables: MaxHR <dbl>, ExerciseAngina <chr>, Oldpeak <dbl>,
#   ST_Slope <chr>, HeartDisease <dbl>
# i Use `print(n = ...)` to see more rows
```

2.Data Exploration

Dimension and variable names of the dataset

Code: `dim(data)`

```
variable_names <- names(data)
print(variable_names)
```

```
> dim(data)
[1] 150  12
> variable_names <- names(data)
> print(variable_names)
 [1] "Age"            "Sex"            "ChestPainType"  "RestingBP"      "Cholesterol"
 [6] "FastingBS"      "RestingECG"     "MaxHR"          "ExerciseAngina" "Oldpeak"
[11] "ST_Slope"       "HeartDisease"
>
```

Structure of the dataset

Code: `str(data)`

```
> str(data)
spc_tbl_ [150 × 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ Age           : num [1:150] 40 49 37 NA 54 39 45 54 37 48 ...
 $ Sex           : chr [1:150] "M" "F" "M" "F" ...
 $ ChestPainType : chr [1:150] "ATA" "NAP" "ATA" "ASY" ...
 $ RestingBP     : num [1:150] 140 160 130 138 -150 120 130 110 140 120 ...
 $ Cholesterol   : num [1:150] 289 180 283 214 195 339 237 208 207 284 ...
 $ FastingBS     : num [1:150] 0 0 0 0 0 0 0 0 0 0 ...
 $ RestingECG    : chr [1:150] "Normal" "Normal" "ST" "Normal" ...
 $ MaxHR         : num [1:150] 172 156 98 108 122 170 170 142 130 120 ...
 $ ExerciseAngina: chr [1:150] "N" "N" "N" "Y" ...
 $ Oldpeak       : num [1:150] 0 1 0 1.5 0 0 0 0 1.5 0 ...
 $ ST_Slope      : chr [1:150] "Up" "Flat" "Up" "Flat" ...
 $ HeartDisease  : num [1:150] 0 1 0 1 0 0 0 0 1 0 ...
 - attr(*, "spec")=
  .. cols(
  ..   Age = col_double(),
  ..   Sex = col_character(),
  ..   ChestPainType = col_character(),
  ..   RestingBP = col_double(),
  ..   Cholesterol = col_double(),
  ..   FastingBS = col_double(),
  ..   RestingECG = col_character(),
  ..   MaxHR = col_double(),
  ..   ExerciseAngina = col_character(),
  ..   Oldpeak = col_double(),
  ..   ST_Slope = col_character(),
  ..   HeartDisease = col_double()
  .. )
 - attr(*, "problems")=<externalptr>
```

Full Summary of the dataset

Code: `summary(data)`

```
> summary(data)
      Age            Sex            ChestPainType        RestingBP
 Min.   : 32.00   Length:150         Length:150         Min.   :-150.0
 1st Qu.: 42.00   Class :character   Class :character   1st Qu.: 120.0
 Median : 49.00   Mode  :character   Mode  :character   Median : 130.0
 Mean   : 49.81                                        Mean   : 129.3
 3rd Qu.: 54.00                                        3rd Qu.: 140.0
 Max.   :172.00                                        Max.   : 190.0
 NA's   :3
  Cholesterol       FastingBS        RestingECG           MaxHR
 Min.   :  85.0   Min.   :0.00000   Length:150         Min.   : 82.0
 1st Qu.: 205.2   1st Qu.:0.00000   Class :character   1st Qu.:124.0
 Median : 239.0   Median :0.00000   Mode  :character   Median :140.0
 Mean   : 258.3   Mean   :0.08667                      Mean   :140.1
 3rd Qu.: 277.0   3rd Qu.:0.00000                      3rd Qu.:155.8
 Max.   :1005.0   Max.   :1.00000                      Max.   :190.0

 ExerciseAngina       Oldpeak          ST_Slope          HeartDisease
 Length:150         Min.   :0.0000   Length:150         Min.   :0.00
 Class :character   1st Qu.:0.0000   Class :character   1st Qu.:0.00
 Mode  :character   Median :0.0000   Mode  :character   Median :0.00
                    Mean   :0.5933                      Mean   :0.38
                    3rd Qu.:1.0000                      3rd Qu.:1.00
                    Max.   :4.0000                      Max.   :1.00
```

identifying missing values of the dataset

Code: `colSums(is.na(data))`

```
> colSums(is.na(data))
          Age           Sex  ChestPainType      RestingBP    Cholesterol
            3             3              0              0              0
    FastingBS     RestingECG          MaxHR ExerciseAngina        Oldpeak
            0             0              0              2              0
     ST_Slope   HeartDisease
            0             0
```

Standard Deviation, Mean, Variance of Numeric Attributes

Code: `data_sd <- data %>% summarise_if(is.numeric,sd)`

`View(data_sd)`

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
1	16.0804	28.19018	114.8757	0.2822885	23.84533	0.8962762	0.4870125

Code: `data_var <- data %>% summarise_if(is.numeric, var)`

`View(data_var)`

Filter

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
1	59.47204	268.4099	5662.042	0.0796868	568.5996	0.803311	0.2371812

Code: `data_mean <- data %>% summarise_if(is.numeric, mean)`

`View(data_mean)`

Filter

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
1	48.33333	131.1133	248.0733	0.08666667	140.0667	0.5933333	0.38

3.Missing Value

##Recover the missing values of the full dataset by omitting any missing values.

Code: `omitdata<-na.omit(data)`

`colSums(is.na(omitdata))`

```
> omitdata<-na.omit(data)
> colSums(is.na(omitdata))
           Age            Sex  ChestPainType      RestingBP    Cholesterol
             0              0              0              0              0
     FastingBS     RestingECG          MaxHR ExerciseAngina        Oldpeak
             0              0              0              0              0
      ST_Slope   HeartDisease
             0              0
>
```

##Recovering Missing Values in the Age Attribute

```
> data$Age
  [1]  40  49  37  NA  54  39  45  54  37  48  37  58  39  49  42  54  38  43  60  36  43
 [22]  44  49  NA  40  36  53  52  53  51  53  56  NA  41  43  54  65  41  48  48  54  54
 [43]  35  52  43  59  54  50  36  41 170  47  45  41  52  51 172  58  54  52  49  43  45
 [64]  46  50  37  45  32  52  44  54  44  52  44  55  46  32  35  52  49  55  54  63  52
 [85]  56  66  65  53  43  55  49  39  52  48  39  58  43  39  56  41  65  51  40  40  46
[106]  57  48  34  54  39  59  57  54  38  49  33  38  54  54  34  47  52  46  54  58  54
[127]  34  48  54  42  54  46  56  56  61  54  43  39  54  54  52  50  54  53  54  39  42
[148]  43  50  54
```

Code: `na.omit(data$Age)`

```
> na.omit(data$Age)
  [1]  40  49  37  54  39  45  54  37  48  37  58  39  49  42  54  38  43  60  36  43  44
 [22]  49  40  36  53  52  53  51  53  56  41  43  54  65  41  48  48  54  54  35  52  43
 [43]  59  54  50  36  41 170  47  45  41  52  51 172  58  54  52  49  43  45  46  50  37
 [64]  45  32  52  44  54  44  52  44  55  46  32  35  52  49  55  54  63  52  56  66  65
 [85]  53  43  55  49  39  52  48  39  58  43  39  56  41  65  51  40  40  46  57  48  34
[106]  54  39  59  57  54  38  49  33  38  54  54  34  47  52  46  54  58  54  34  48  54
[127]  42  54  46  56  56  61  54  43  39  54  54  52  50  54  53  54  39  42  43  50  54
attr(,"na.action")
[1]  4 24 33
attr(,"class")
[1] "omit"
```

Code: age_mode <- names(sort(table(data$Age), decreasing = TRUE))[1]

data$Age[is.na(data$Age)] <- as.numeric(age_mode)

```
> age_mode <- names(sort(table(data$Age), decreasing = TRUE))[1]
> data$Age[is.na(data$Age)] <- as.numeric(age_mode)
> data$Age
  [1]  40  49  37  54  54  39  45  54  37  48  37  58  39  49  42  54  38  43  60  36  43
 [22]  44  49  54  40  36  53  52  53  51  53  56  54  41  43  54  65  41  48  48  54  54
 [43]  35  52  43  59  54  50  36  41 170  47  45  41  52  51 172  58  54  52  49  43  45
 [64]  46  50  37  45  32  52  44  54  44  52  44  55  46  32  35  52  49  55  54  63  52
 [85]  56  66  65  53  43  55  49  39  52  48  39  58  43  39  56  41  65  51  40  40  46
[106]  57  48  34  54  39  59  57  54  38  49  33  38  54  54  34  47  52  46  54  58  54
[127]  34  48  54  42  54  46  56  56  61  54  43  39  54  54  52  50  54  53  54  39  42
[148]  43  50  54
>
```

##Recovering Missing Values in the Sex Attribute

Code: Sex_mode <- names(sort(table(data$Sex), decreasing = FALSE))[1]

data$Sex[is.na(data$Sex)] <- as.character(Sex_mode)

```
> Sex_mode <- names(sort(table(data$Sex), decreasing = FALSE))[1]
> data$Sex[is.na(data$Sex)] <- as.character(Sex_mode)
> data$Sex
  [1] "M" "F" "M" "F" "M" "M" "F" "M" "F" "F" "F" "M" "M" "M" "M" "M" "M" "F" "M" "M" "M"
 [22] "M" "F" "M" "M" "F" "M" "M" "F" "M" "M" "M" "M" "M" "F" "M" "M" "F" "M" "F" "F" "F"
 [43] "M" "M" "M" "M" "M" "M" "M" "M" "M" "F" "M" "F" "F" "F" "M" "M" "M" "M" "M" "F" "M"
 [64] "M" "F" "F" "F" "M" "M" "M" "M" "M" "M" "F" "M" "M" "M" "F" "M" "M" "M" "M" "M" "M"
 [85] "M" "M" "M" "F" "M" "M" "F" "M" "F" "M" "F" "M" "M" "M" "M" "M" "M" "M" "F" "M" "M"
[106] "M" "F" "M" "M" "M" "F" "M" "M" "M" "F" "F" "M" "F" "F" "M" "F" "F" "M" "F" "M" "M"
[127] "F" "F" "F" "M" "M" "M" "M" "M" "F" "M" "F" "M" "M" "M" "M" "M" "M" "M" "F" "M" "M"
[148] "F" "M" "M"
```

##Recovering Missing Values in the ExerciseAngina Attribute

Code: ExerciseAngina_mode <- names(sort(table(data$ExerciseAngina), decreasing = FALSE))[1]
data$ExerciseAngina[is.na(data$ExerciseAngina)] <- as.character(ExerciseAngina_mode)

```
> ExerciseAngina_mode <- names(sort(table(data$ExerciseAngina), decreasing = FALSE))[1]
> data$ExerciseAngina[is.na(data$ExerciseAngina)] <- as.character(ExerciseAngina_mode)
> data$ExerciseAngina
  [1] "N" "N" "N" "Y" "N" "N" "Y" "N" "Y" "N" "N" "Y" "N" "Y" "N" "N" "N" "N" "N" "N" "N"
 [22] "N" "N" "Y" "N" "N" "Y" "N" "N" "N" "N" "N" "N" "N" "N" "N" "Y" "N" "N" "Y" "N" "Y"
 [43] "N" "N" "Y" "Y" "N" "N" "N" "N" "Y" "Y" "N" "N" "Y" "N" "Y" "N" "N" "Y" "N" "N" "N"
 [64] "Y" "N" "N" "N" "N" "Y" "N" "Y" "N" "N" "N" "Y" "N" "N" "N" "Y" "N" "N" "N" "N" "N"
 [85] "Y" "Y" "Y" "Y" "N" "Y" "N" "N" "N" "Y" "N" "Y" "N" "N" "N" "N" "Y" "N" "N" "Y" "N"
[106] "N" "N" "N" "N" "N" "N" "Y" "Y" "N" "N" "Y" "N" "Y" "N" "N" "N" "N" "N" "Y" "N" "N"
[127] "N" "N" "N" "Y" "N" "Y" "Y" "Y" "Y" "N" "N" "N" "Y" "Y" "Y" "Y" "Y" "N" "N" "N" "N"
[148] "N" "N" "Y"
> |
```

4.Data types and Conversion

Code: `data$Sex<-factor(data$Sex,levels=c('M','F'),labels=c(0,1))`

`View(data$Sex)`

	Age	Sex	ChestPainType
1	40	0	ATA
2	49	1	NAP
3	37	0	ATA
4	54	1	ASY
5	54	0	NAP
6	39	0	NAP

Code: `data$HeartDiseaseType <- factor(data$HeartDisease, levels = c(0, 1), labels = c("Normal", "Heart Disease"))`

HeartDisease	HeartDiseaseType
0	Normal
1	Heart Disease
0	Normal
1	Heart Disease
0	Normal
0	Normal
0	Normal

5.Outliers

##Detecting outliers in the Age attribute

Code:
```
age_counts <- table(data$Age)
mode_age <- as.numeric(names(age_counts)[which.max(age_counts)])
data$Age[data$Age %in% c(170, 172)] <- mode_age
```

```
> View(data)
> mode_age <- as.numeric(names(age_counts)[which.max(age_counts)])
> data$Age[data$Age %in% c(170, 172)] <- mode_age
> data$Age
  [1] 40 49 37 54 54 39 45 54 37 48 37 58 39 49 42 54 38 43 60 36 43 44 49 54 40 36 53 52
 [29] 53 51 53 56 54 41 43 54 65 41 48 48 54 54 35 52 43 59 54 50 36 41 54 47 45 41 52 51
 [57] 54 58 54 52 49 43 45 46 50 37 45 32 52 44 54 44 52 44 55 46 32 35 52 49 55 54 63 52
 [85] 56 66 65 53 43 55 49 39 52 48 39 58 43 39 56 41 65 51 40 40 46 57 48 34 54 39 59 57
[113] 54 38 49 33 38 54 54 34 47 52 46 54 58 54 34 48 54 42 54 46 56 56 61 54 43 39 54 54
[141] 52 50 54 53 54 39 42 43 50 54
>
```

##Detecting outliers in the RestingBP attribute

Code:

```
RestingBP_counts <- table(data$RestingBP)
mode_RestingBP <-
as.numeric(names(RestingBP_counts)[which.max(RestingBP_counts)])
data$RestingBP[data$RestingBP %in% c(-150,120)] <- mode_RestingBP
```

```
> mode_RestingBP <- as.numeric(names(RestingBP_counts)[which.max(RestingBP_counts)])
> data$RestingBP[data$RestingBP %in% c(-150,120)] <- mode_RestingBP
> data$RestingBP
  [1] 140 160 130 138 120 120 130 110 140 120 130 136 120 140 115 120 110 120 100 120 100
 [22] 120 124 150 130 130 124 120 113 125 145 130 125 130 150 125 140 110 120 150 150 130
 [43] 150 140 120 130 120 140 112 110 130 120 140 130 130 160 120 130 150 112 100 150 140
 [64] 120 110 120 132 110 160 150 140 130 120 120 140 150 118 140 140 130 110 120 150 160
 [85] 150 140 170 140 120 140 110 130 120 160 110 130 142 160 120 125 130 130 150 120 118
[106] 140 120 150 140 190 130 150 140 140 130 100 120 130 120 140 135 125 110 180 130 120
[127] 130 108 120 120 145 110 170 150 130 115 120 120 140 150 160 140 160 140 120 110 120
[148] 120 120 130
>
```

##Detecting outliers in the Cholesterol attribute

Code:

```
data$Cholesterol <- ifelse(data$Cholesterol %in% c(1000, 1005),
median(data$Cholesterol, na.rm = TRUE), data$Cholesterol)
```

```
> data$Cholesterol <- ifelse(data$Cholesterol %in% c(1000, 1005), median(data$Cholesterol, n
a.rm = TRUE), data$Cholesterol)
> data$Cholesterol
  [1] 289 180 283 214 195 339 237 208 207 284 239 239 204 234 211 273 196 201 248 267 223
 [22] 184 201 288 215 209 260 284 468 188 518 167 224 172 186 254 306 250 177 227 230 294
 [43] 264 259 175 318 223 216 340 289 233 205 224 245 180 194 270 213 365 342 253 254 224
 [64] 277 202 260 297 225 246 412 265 215 182 218 268 163 529 167 100 206 277 238 223 196
 [85] 213 139 263 216 291 229 208 307 210 329 182 263 207 147  85 269 275 179 392 466 186
[106] 260 254 214 129 241 188 255 276 297 207 246 282 338 160 156 248 272 240 393 230 246
[127] 161 163 230 228 292 202 388 230 294 265 215 241 166 247 331 341 291 243 279 273 198
[148] 249 168 603
>
```

6.Transformation

Code:

```
min_value <- min(data$RestingBP)
max_value <- max(data$RestingBP)
# Normalization function
normalize <- function(x) {
  return((x - min_value) / (max_value - min_value))
}
```

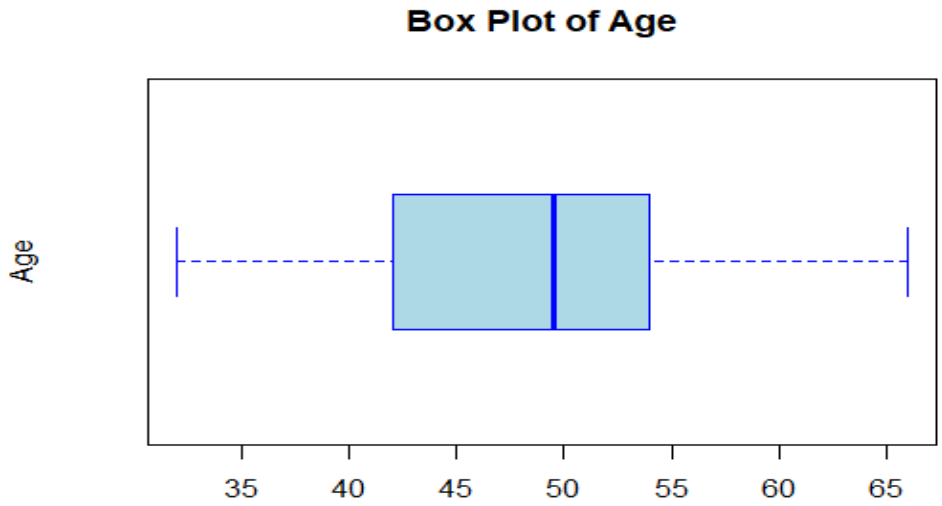
```
data$Normalized_RestingBP <- normalize(data$RestingBP)
```

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease	Normalized_RestingBP
1	40	0	ATA	140	289	0	Normal	172	N	0.0	Up	0	0.4444444
2	49	1	NAP	160	180	0	Normal	156	N	1.0	Flat	1	0.6666667
3	37	0	ATA	130	283	0	ST	98	N	0.0	Up	0	0.3333333
4	54	1	ASY	138	214	0	Normal	108	Y	1.5	Flat	1	0.4222222
5	54	0	NAP	120	195	0	Normal	122	N	0.0	Up	0	0.2222222
6	39	0	NAP	120	339	0	Normal	170	N	0.0	Up	0	0.2222222
7	45	1	ATA	130	237	0	Normal	170	Y	0.0	Up	0	0.3333333

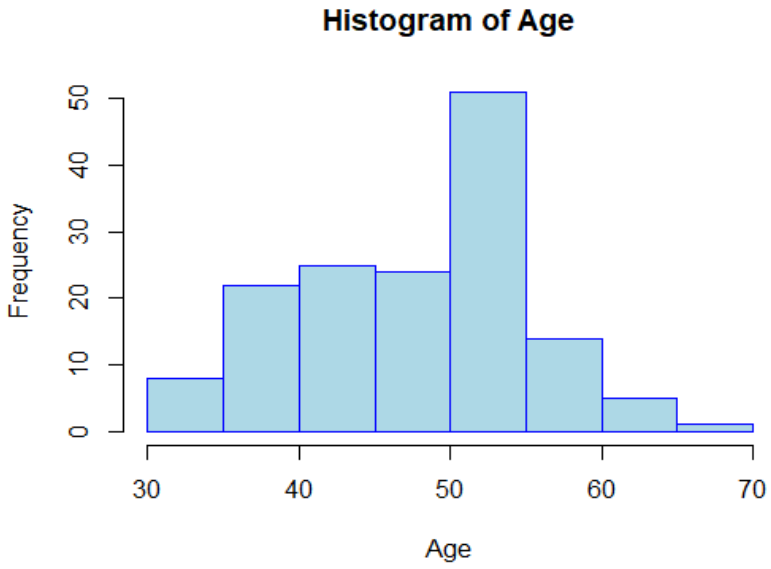
7.Data Visualization

##boxplot & Histrogram of Age Attribute

Code: `boxplot(data$Age, main = "Box Plot of Age", ylab = "Age", col = "lightblue", border = "blue", horizontal = TRUE)`

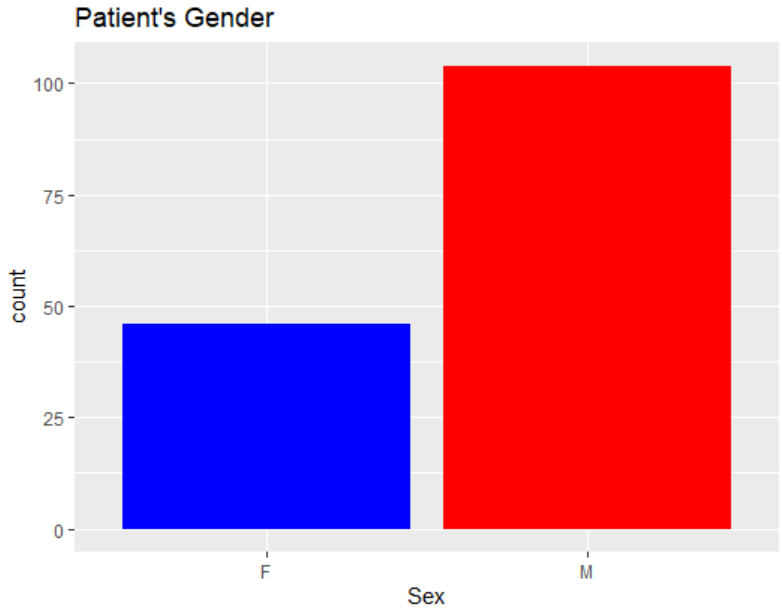


Code: `hist(data$Age, main = "Histogram of Age", xlab = "Age", ylab = "Frequency", col = "lightblue", border = "blue")`



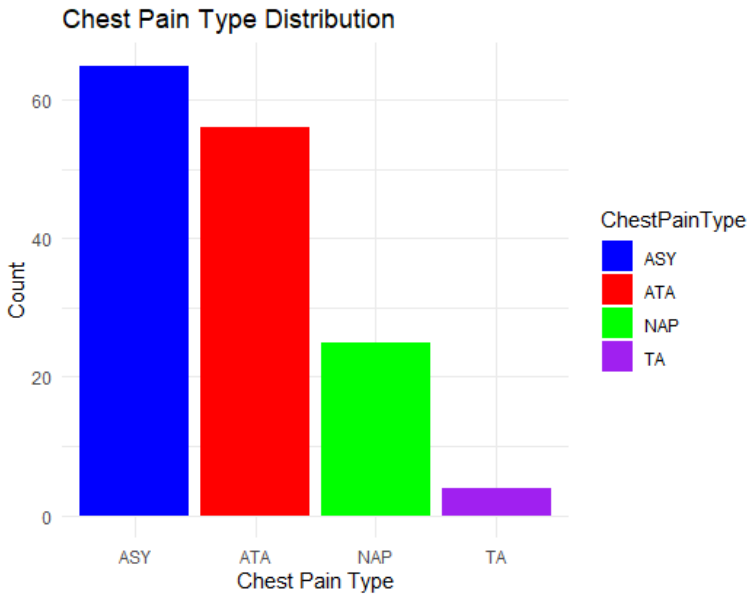
##Barplot of Sex Attribute

Code:
```
barplot(table(data$Sex), col = c("blue", "red"), main = "Sex Distribution")
ggplot(data, aes(x = Sex)) +
  geom_bar(fill = c("blue", "red")) +
  labs(title = " Patient's Gender ")
```



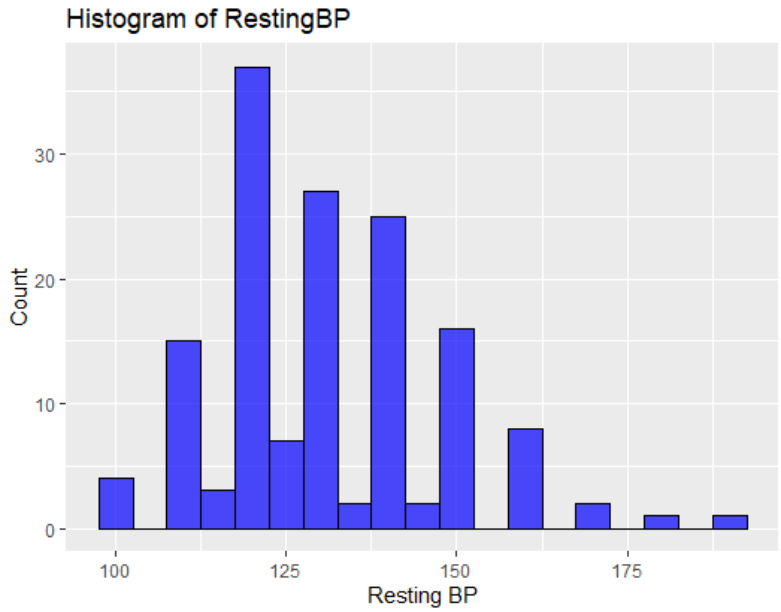
##Barplot of ChestPainType Attribute

Code:
```
chest_pain_counts <- table(data$ChestPainType)
chest_pain_df <- as.data.frame(chest_pain_counts)
names(chest_pain_df) <- c("ChestPainType", "Count")
bar_plot <- ggplot(chest_pain_df, aes(x = ChestPainType, y = Count,
fill = ChestPainType)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("blue", "red", "green", "purple")) +
  labs(title = "Chest Pain Type Distribution") +
  xlab("Chest Pain Type") +
  ylab("Count") +
  theme_minimal()
print(bar_plot)
```



##Histrogram of RestingBP Attribute

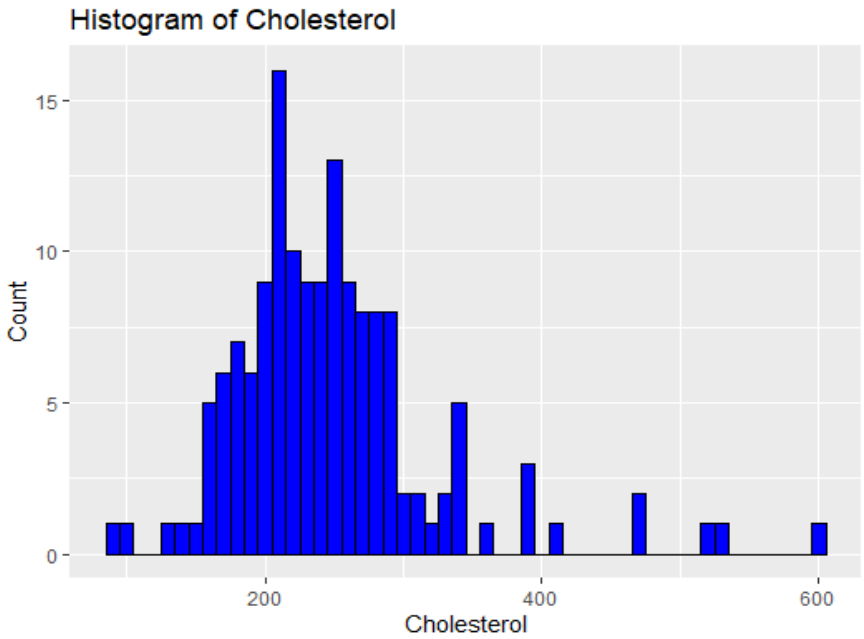
Code:
```
ggplot(data, aes(x = RestingBP)) +
  geom_histogram(binwidth = 5, fill = "blue", color = "black", alpha
= 0.7) +
  labs(title = "Histogram of RestingBP",
       x = "Resting BP",
       y = "Count")
```



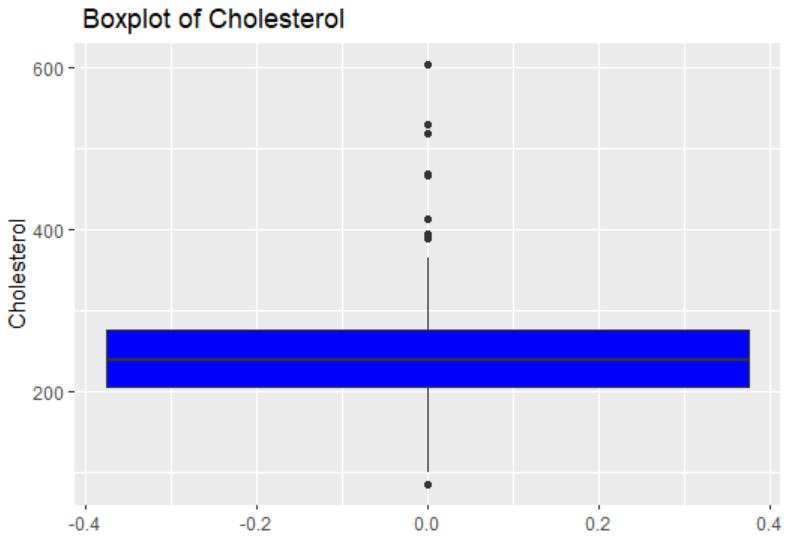
##Histrogtam & Boxplot of Cholesterol Attribute

Code:

```
ggplot(data, aes(x = Cholesterol)) +
  geom_histogram(binwidth = 10, fill = "blue", color = "black") +
  labs(
    title = "Histogram of Cholesterol",
    x = "Cholesterol",
    y = "Count"
  )
```

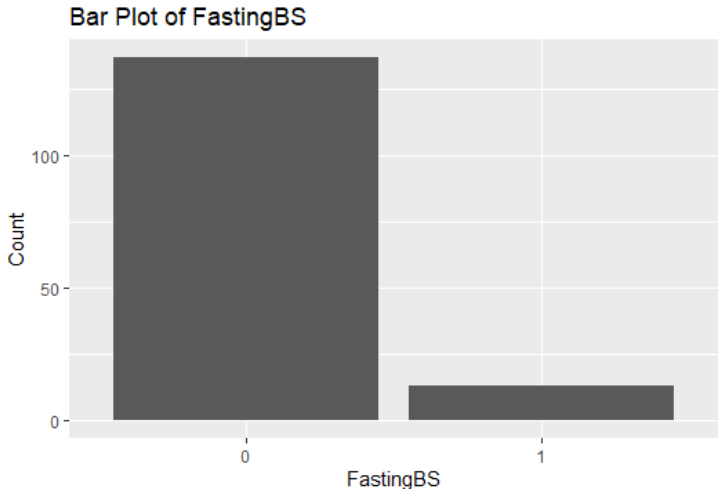


Code:
```
ggplot(data, aes(y = Cholesterol)) +
  geom_boxplot(fill = "blue") +
  labs(
    title = " Boxplot of Cholesterol",
    y = "Cholesterol"
  )
```



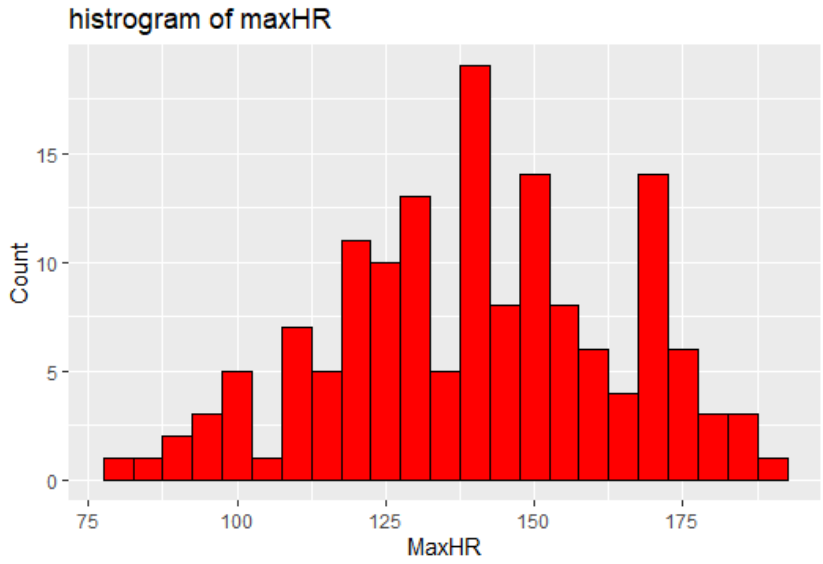
##Barplot of FastingBS

Code:
```
ggplot(data, aes(x = factor(FastingBS))) +
  geom_bar() +
  labs(title = "Bar Plot of FastingBS",
       x = "FastingBS",
       y = "Count")
```

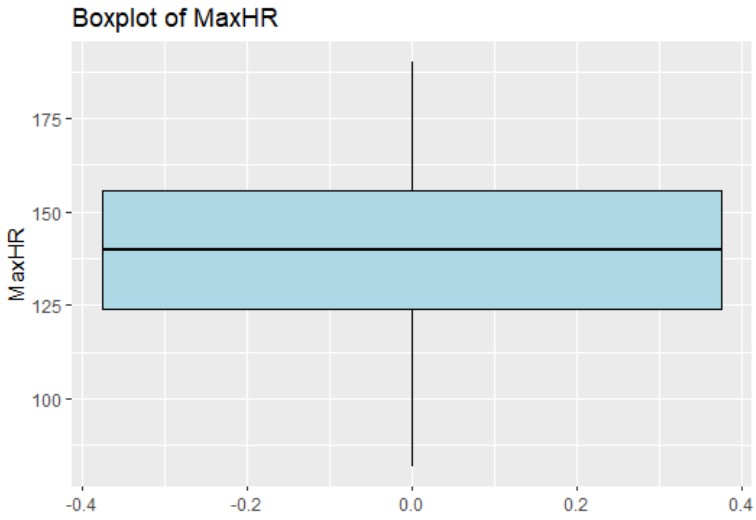


##Histrogram & Boxplot of MaxHR attribute

Code:
```
ggplot(data, aes(x = MaxHR)) +
  geom_histogram(binwidth = 5, fill = "red", color = "black") +
  labs(title = "histrogram of maxHR", x = "MaxHR", y = "Count")
```



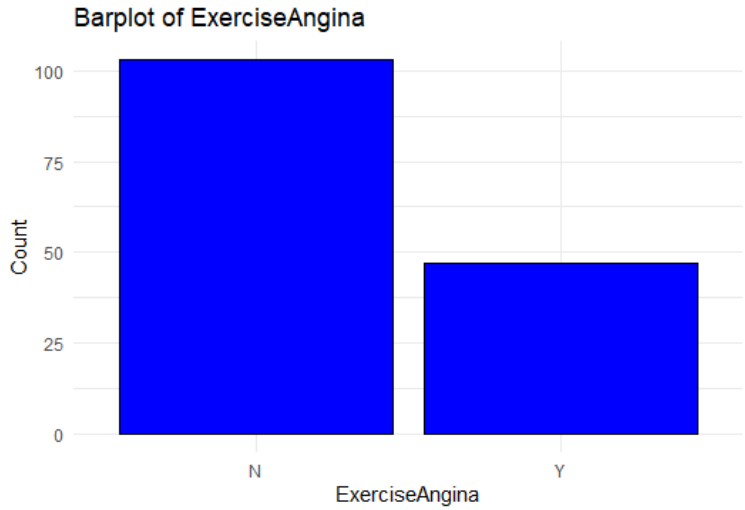
Code:
```
ggplot(data, aes(y = MaxHR)) +
  geom_boxplot(fill = "lightblue", color = "black") +
  labs(title = "Boxplot of MaxHR", y = "MaxHR")
```



##Barplot of ExcerciseAngina Attribute

Code :

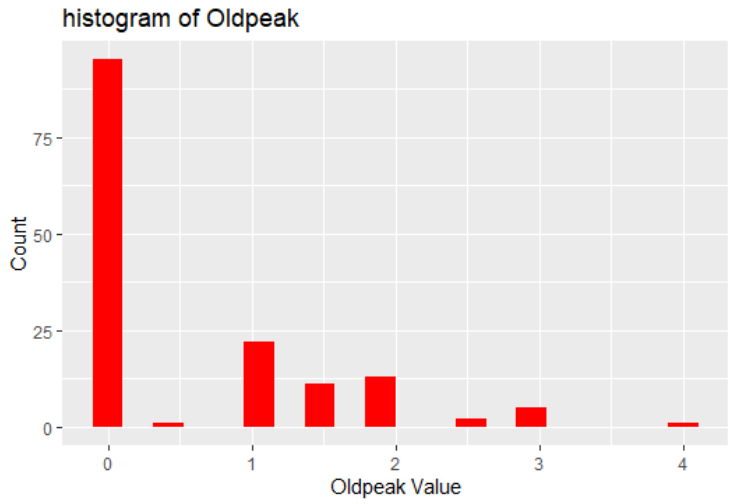
```
ggplot(data, aes(x = factor(ExerciseAngina))) +
  geom_bar(fill = "blue", color = "black") +
  labs(title = "Barplot of ExerciseAngina", x = "ExerciseAngina", y
= "Count") +
  theme_minimal()
```



##Histrogram of Oldpeak Attribute

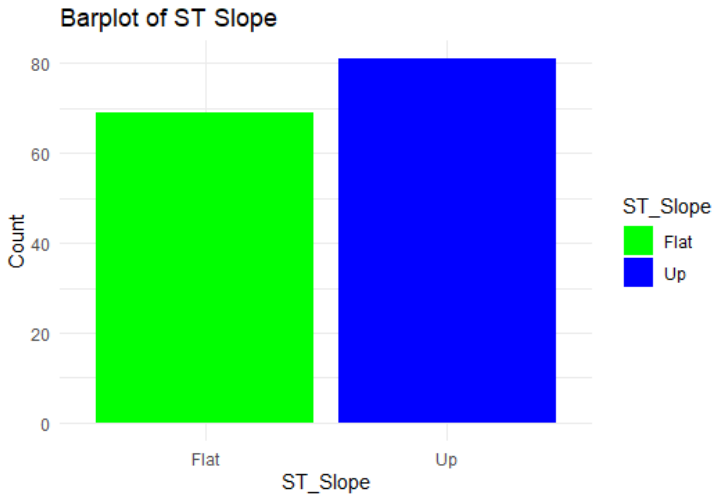
Code:

```
ggplot(data, aes(x = Oldpeak)) +
  geom_histogram(fill = "red", bins = 20) +
  labs(title = "histogram of Oldpeak", x = "Oldpeak Value", y =
"Count")
```



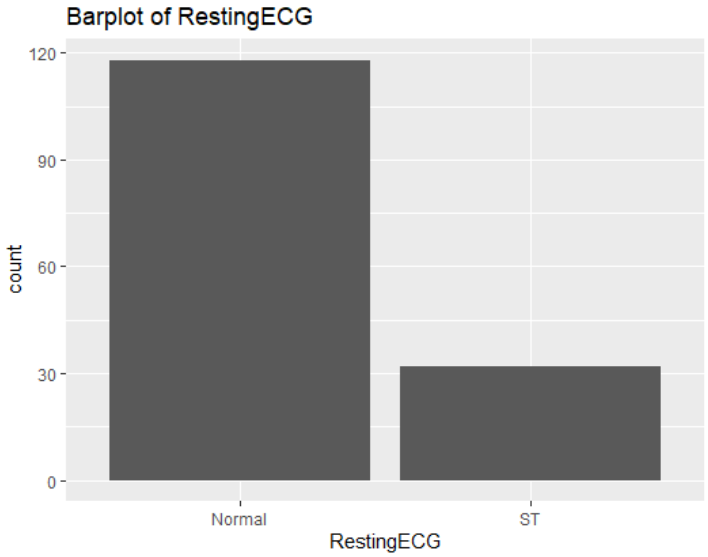
##Barplot of ST Slope Attribute

Code:
```
ggplot(data, aes(x = ST_Slope, fill = ST_Slope)) +
  geom_bar(position = "dodge") +
  labs(title = "Barplot of ST Slope",
       y = "Count") +
  scale_fill_manual(values = c("Up" = "blue", "Flat" = "green")) +
  theme_minimal()
```



##Barplot of RestingECG Attribute

Code:
```
ggplot(data, aes(x = RestingECG)) +
  geom_bar() +
  labs(title = "Barplot of RestingECG ")
```



##Barplot of HeartDisease Attribute

Code:

```
ggplot(data, aes(x = factor(HeartDisease))) +
  geom_bar(fill = c("blue", "green")) +
  labs(title = "Heart Disease of patient's", x = "HeartDisease", y =
"Count") +
  scale_x_discrete(labels = c("Normal", "Heart Disease")) +
  theme_minimal()
```