DATA621: Final Project

Farhana Akther, Bridget Boakye and Hazal Gunduz

2024-05-11

Contents

Abstract:

Loading and transforming data set:

Loading the data set:

The dataset utilized in this research has been sourced from Kaggle and is accessible via the following link: Kaggle Health Dataset. Upon downloading, the dataset was subsequently uploaded to a GitHub repository to facilitate its access within the RStudio environment to ensure the reproducibility of the study. To import the dataset into RStudio, the read.csv() function from base R will be used and we will utilize the URL of the stored repository.

```
set.seed(123)

hypert <- read.csv("https://raw.githubusercontent.com/FarhanaAkther23/DATA621/main/DATA621%20Final%20Pr
```

To confirm that the dataset has been loaded correctly and contains all the columns, we can utilize the head() function from Base R. This function conveniently displays the first six rows of the dataset, allowing us to inspect its structure and ensure proper loading.

```
head(hypert)
```

```
##   age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
## 1  57   1  3      145  233   1       0     150     0     2.3     0  0    1
## 2  64   0  2      130  250   0       1     187     0     3.5     0  0    2
## 3  52   1  1      130  204   0       0     172     0     1.4     2  0    2
## 4  56   0  1      120  236   0       1     178     0     0.8     2  0    2
## 5  66   0  0      120  354   0       1     163     1     0.6     2  0    2
## 6  51   1  0      140  192   0       1     148     0     0.4     1  0    1
##   target
## 1      1
## 2      1
## 3      1
## 4      1
## 5      1
## 6      1
```

```
dim(hypert)
```

```
## [1] 26083    14
```

Upon inspection, we observe that the dataset has been successfully loaded into our RStudio environment. However, before proceeding with the analysis, it's important to acknowledge that the dataset may require cleaning and transformation to ensure it is suitable for analysis. Let us understand the variables, perform data wrangling and transformation to prepare the dataset for analysis

Data Prepratation:

Understaing the Columns:

age: Age of the patient in years (numeric). *sex:* patient's gender (1: male; 0: female) (numeric). *cp:* Chest pain type: 0: asymptomatic 1: typical angina 2: atypical angina 3: non-anginal pain (integer). *trestbps:*

Resting blood pressure (in mm Hg) (integer). *chol:* Serum cholestoral in mg/dl (integer). *fbs:* if the patient's fasting blood sugar > 120 mg/dl (1: yes; 0: no) (integer). *restecg:* Resting ECG (electrocardiograph) results: 0: normal 1: ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV) 2: probable or definite left ventricular hypertrophy by Estes' criteria (integer). *thalach:* Maximum heart rate achieved by the patient (integer). *exang:* Exercise induced angina, experienced by the patient (1: yes; 0: no) (integer). *oldpeak:* ST depression induced by exercise relative to rest (numeric). *slope:* The slope of the peak exercise ST segment (integer). *ca:* Number of major vessels colored by fluoroscopy (integer). *thal:* Type of thalassemia (integer). *target:* Presence of hypertension, where 0 indicates absence and 1 indicates presence (integer).

Wrangling and Transforming:

Before proceeding with logistic regression analysis, it's essential to examine the structure of the dataset. We will use the str() function to inspect the types of data in each variable or column. This step will help us understand the composition of the dataset and ensure that it is suitable for logistic regression analysis.

```
str(hypert)
```

```
## 'data.frame':    26083 obs. of  14 variables:
##  $ age     : num  57 64 52 56 66 51 42 38 72 47 ...
##  $ sex     : num  1 0 1 0 0 1 0 0 0 0 ...
##  $ cp      : int  3 2 1 1 0 0 1 1 2 2 ...
##  $ trestbps: int  145 130 130 120 120 140 140 120 172 150 ...
##  $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
##  $ fbs     : int  1 0 0 0 0 0 0 0 1 0 ...
##  $ restecg : int  0 1 0 1 1 1 0 1 1 1 ...
##  $ thalach : int  150 187 172 178 163 148 153 173 162 174 ...
##  $ exang   : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
##  $ slope   : int  0 0 2 2 2 1 1 2 2 2 ...
##  $ ca      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ thal    : int  1 2 2 2 2 1 2 3 3 2 ...
##  $ target  : int  1 1 1 1 1 1 1 1 1 1 ...
```

The code above successfully displays the structure of the dataset hypert. It consists of 26083 observations and 14 variables. Each variable has its corresponding data type, such as numerical (num or int). This information will be helpful for further data analysis and modeling. However, if you look carefully, it seems that some of the columns in the dataset have been imported as numeric or integer data types, while they should be treated as categorical variables or factors. Specifically, columns like sex, cp, fbs, exang, slope, ca, thal, and target are categorical variables based on the dataset description.To address this issue, we'll need to convert these columns to factors in order to properly represent their categorical nature. Before changing data type of any columns we want to look out for any missing values in those rows. we will is.na() to see any missing values.

```
sum(is.na(hypert))
```

```
## [1] 25
```

we can see that there in total 25 missing values in the data set and if we check column wise all of those missing values are in sex column

```
(hypert[is.na(hypert$sex),])
```

```
##       age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
## 287    43  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 875    78  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 1450   65  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 1996   60  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 3940   58  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 4383   74  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 4813   66  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 5831   68  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 6609   54  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 6894   59  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 12903  33  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 13527  55  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 13797  50  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 14868  48  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 15109  64  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 15332  56  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 16468  44  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 16659  49  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 20186  53  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 20478  88  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 20756  75  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 21026  70  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 22182  84  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 22406  76  NA  3      134  204   0       1     162     0     0.8     2  2    2
## 23306  69  NA  3      134  204   0       1     162     0     0.8     2  2    2
##       target
## 287        0
## 875        0
## 1450       0
## 1996       0
## 3940       0
## 4383       0
## 4813       0
## 5831       0
## 6609       0
## 6894       0
## 12903      0
## 13527      0
## 13797      0
## 14868      0
## 15109      0
## 15332      0
## 16468      0
## 16659      0
## 20186      0
## 20478      0
## 20756      0
## 21026      0
## 22182      0
## 22406      0
```

```
## 23306      0
```

We can observe that apart from 'age' column in the data set every other variable/column has the same value for each of those missing values so we can ignore those rows and since they are only 25 out 26k (way below 5%) observations so we can easily impute it from our data set.

```
hypert <- hypert[!(is.na(hypert$sex)),]
sum(is.na(hypert))
```

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

Finding Correlation using pairs.panels():

In order for us to look at the relationship between all the variables, first we will need to convert the relevant columns to numeric format to ensure they are treated as numeric for our correlation analysis.

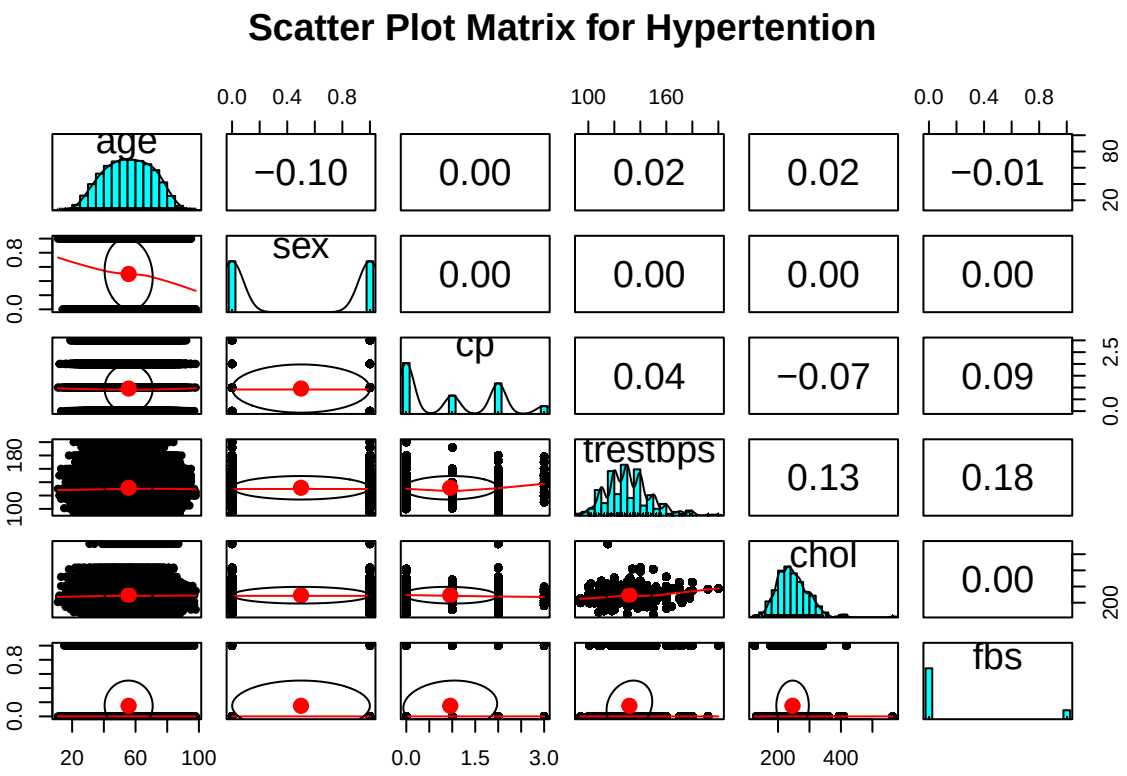
```
# Convert columns to numeric
hypert$sex <- as.numeric(as.character(hypert$sex))
hypert$fbs <- as.numeric(as.character(hypert$fbs))
hypert$restecg <- as.numeric(as.character(hypert$restecg))
hypert$exang <- as.numeric(as.character(hypert$exang))
hypert$slope <- as.numeric(as.character(hypert$slope))
hypert$thal <- as.numeric(as.character(hypert$thal))
hypert$target <- as.numeric(as.character(hypert$target))

cor(hypert)
```

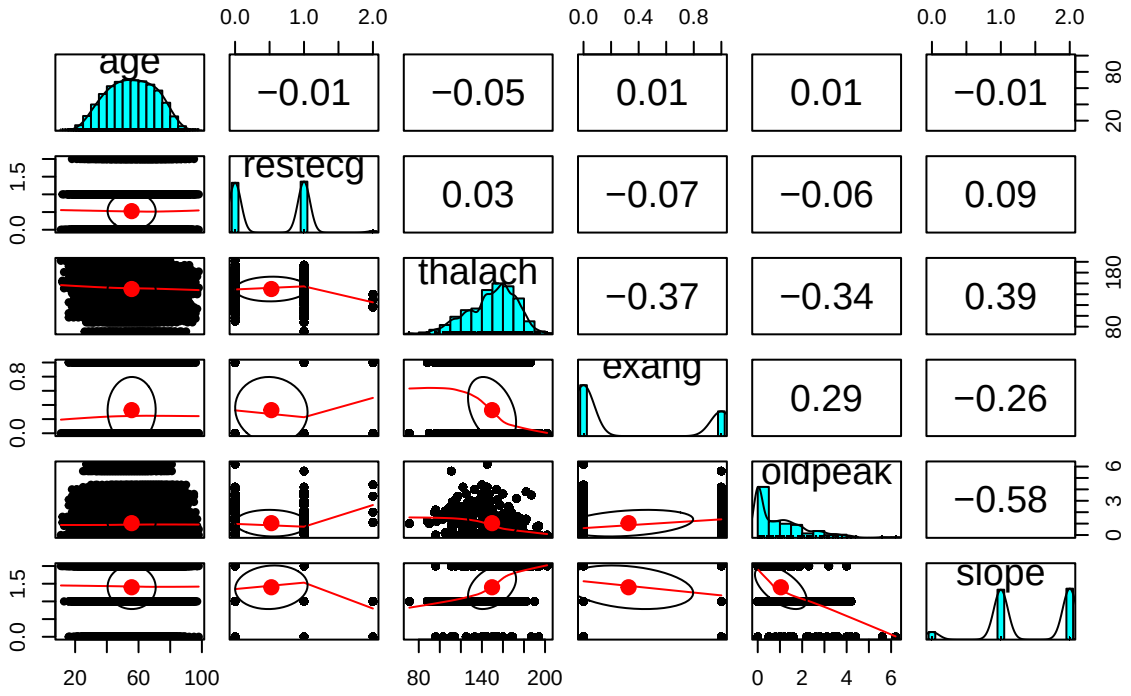
```
##                   age         sex            cp    trestbps         chol
## age       1.0000000000 -0.09874843  0.0006423732  0.02183637  0.018001387
## sex      -0.0987484277  1.00000000  0.0000000000  0.00000000  0.000000000
## cp        0.0006423732  0.00000000  1.0000000000  0.03849478 -0.070803197
## trestbps  0.0218363670  0.00000000  0.0384947828  1.00000000  0.133460097
## chol      0.0180013871  0.00000000 -0.0708031971  0.13346010  1.000000000
## fbs      -0.0064847662  0.00000000  0.0895182069  0.17732262  0.004372090
## restecg  -0.0072220709  0.00000000  0.0470129184 -0.11368567 -0.151471217
## thalach  -0.0504352873  0.00000000  0.2889693961 -0.05088878 -0.007399770
## exang     0.0120508468  0.00000000 -0.3924466848  0.07578560  0.064613573
## oldpeak   0.0081040033  0.00000000 -0.1522552081  0.20389176  0.047298379
## slope    -0.0092941724  0.00000000  0.1213751680 -0.13238470  0.007446393
## ca        0.0490279873  0.00000000 -0.2011470550  0.10137692  0.089288487
## thal      0.0079913573  0.00000000 -0.1680074133  0.06159422  0.085417715
## target   -0.0231610943  0.00000000  0.4370110824 -0.14845132 -0.083055173
##                  fbs      restecg      thalach       exang      oldpeak
## age      -0.006484766 -0.007222071 -0.050435287  0.01205085  0.008104003
## sex       0.000000000  0.000000000  0.000000000  0.00000000  0.000000000
## cp        0.089518207  0.047012918  0.288969396 -0.39244668 -0.152255208
## trestbps  0.177322618 -0.113685666 -0.050888782  0.07578560  0.203891755
## chol      0.004372090 -0.151471217 -0.007399770  0.06461357  0.047298379
## fbs       1.000000000 -0.088097287 -0.008116243  0.02777315  0.006545115
## restecg  -0.088097287  1.000000000  0.034217166 -0.06862642 -0.057028364
## thalach  -0.008116243  0.034217166  1.000000000 -0.37023644 -0.344757805
## exang     0.027773149 -0.068626415 -0.370236444  1.00000000  0.289717480
```

```
## oldpeak   0.006545115 -0.057028364 -0.344757805  0.28971748  1.000000000
## slope    -0.058471277  0.085325807  0.392343464 -0.25834664 -0.581924062
## ca        0.149595369 -0.087657043 -0.225839480  0.12714975  0.232065652
## thal     -0.025280889 -0.009612092 -0.110800496  0.22185995  0.225850880
## target   -0.034045785  0.137151203  0.419912450 -0.43920813 -0.438615003
##                 slope          ca         thal      target
## age      -0.009294172  0.04902799  0.007991357 -0.02316109
## sex       0.000000000  0.00000000  0.000000000  0.00000000
## cp        0.121375168 -0.20114705 -0.168007413  0.43701108
## trestbps -0.132384696  0.10137692  0.061594221 -0.14845132
## chol      0.007446393  0.08928849  0.085417715 -0.08305517
## fbs      -0.058471277  0.14959537 -0.025280889 -0.03404579
## restecg   0.085325807 -0.08765704 -0.009612092  0.13715120
## thalach   0.392343464 -0.22583948 -0.110800496  0.41991245
## exang    -0.258346644  0.12714975  0.221859954 -0.43920813
## oldpeak  -0.581924062  0.23206565  0.225850880 -0.43861500
## slope     1.000000000 -0.09909767 -0.118069941  0.35122533
## ca       -0.099097674  1.00000000  0.164544118 -0.40517023
## thal     -0.118069941  0.16454412  1.000000000 -0.35874634
## target    0.351225335 -0.40517023 -0.358746340  1.00000000
```

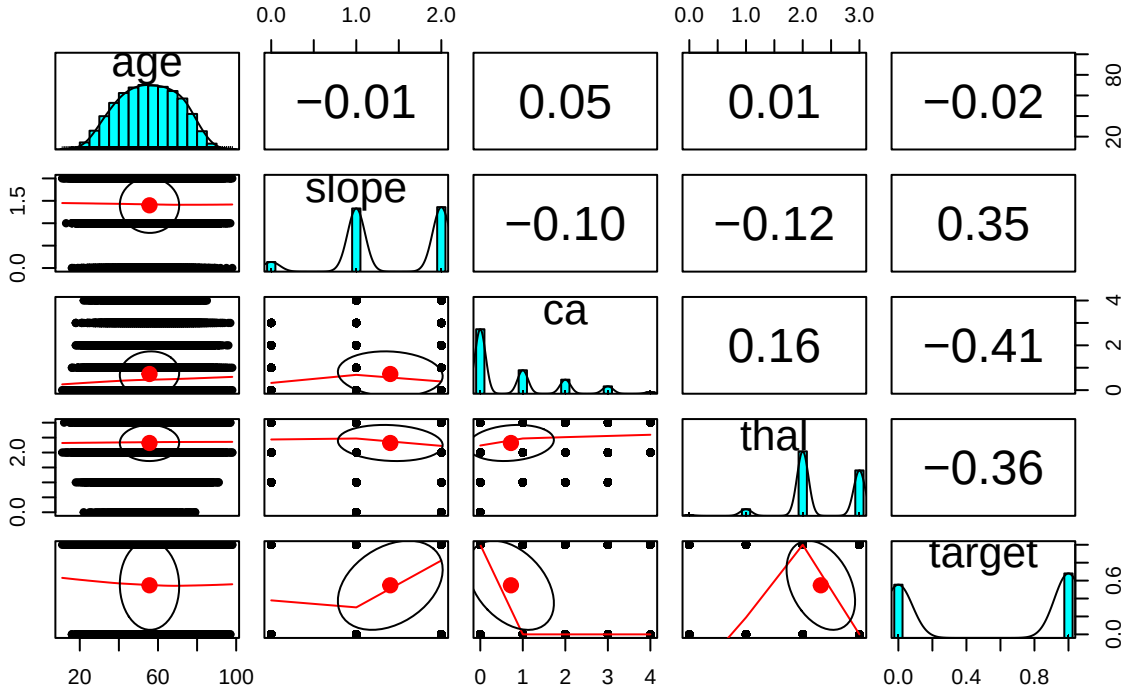
```
pairs.panels(hypert[, c(1, 2:6)], main = "Scatter Plot Matrix for Hypertention")
```




```
pairs.panels(hypert[, c(1, 7:11)], main = "")
```




```
pairs.panels(hypert[, c(1, 11:14)], main = "")
```



The diagonal panels display histograms or density plots of each variable. These plots show the distribution of values for each variable individually. The correlation analysis shows several associations within the dataset. Age shows weak positive correlations with resting blood pressure and weak negative correlations with maximum heart rate during exercise. Chest pain type has moderate positive correlations with maximum heart rate and hypertension presence, while resting blood pressure exhibits weak positive correlations with cholesterol levels. Maximum heart rate shows moderate positive correlations with chest pain type and hypertension presence. Exercise-induced angina and ST depression during exercise have moderate negative

correlations with maximum heart rate and hypertension presence. Slope of peak exercise ST segment has moderate positive correlations with maximum heart rate and hypertension presence. There are no significant correlations observed for sex, cholesterol, fasting blood sugar, and restecg. Moreover, there is no apparent multicollinearity issue observed among the independent variables.

Managing the data types of the columns for EDA: In the next step we will manage the data types of the columns. According to the source of the dataset, certain columns such as sex, cp, fbs, exang, slope, ca, thal, and target should be treated as factors. Therefore, we will adjust the data types accordingly. Additionally, to improve clarity and interpretation, we will convert the numeric codes in the sex column to descriptive labels, replacing 0 with "F" (female) and 1 with "M" (male). This ensures consistency and enhances the readability of the dataset.

```
hypert[hypert$sex == 0,]$sex <- "F" # Replacing 0 by F
hypert[hypert$sex == 1,]$sex <- "M" # Replacing 1 by M


hypert$sex <- as.factor(hypert$sex)
hypert$cp <- as.factor(hypert$cp)
hypert$fbs <- as.factor(hypert$fbs)
hypert$restecg <- as.factor(hypert$restecg)
hypert$exang <- as.factor(hypert$exang)
hypert$slope <- as.factor(hypert$slope)
hypert$thal <- as.factor(hypert$thal)


hypert$target <- ifelse(test = hypert$target == 0, yes = "Non-Hypertension", no = "Hypertension")
hypert$target <- as.factor(hypert$target)
```

Now that we've adjusted the data types of the columns, let's revisit the str() function to ensure that our data frame aligns with the descriptions provided in the data source. This step allows us to verify that the data types and factor levels are consistent with our expectations and the information provided in the source.

```
str(hypert)
```

```
## 'data.frame':    26058 obs. of  14 variables:
##  $ age     : num  57 64 52 56 66 51 42 38 72 47 ...
##  $ sex     : Factor w/ 2 levels "F","M": 2 1 2 1 1 2 1 1 1 1 ...
##  $ cp      : Factor w/ 4 levels "0","1","2","3": 4 3 2 2 1 1 2 2 3 3 ...
##  $ trestbps: int  145 130 130 120 120 140 140 120 172 150 ...
##  $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
##  $ fbs     : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 2 1 ...
##  $ restecg : Factor w/ 3 levels "0","1","2": 1 2 1 2 2 2 1 2 2 2 ...
##  $ thalach : int  150 187 172 178 163 148 153 173 162 174 ...
##  $ exang   : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 1 1 ...
##  $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
##  $ slope   : Factor w/ 3 levels "0","1","2": 1 1 3 3 3 2 2 3 3 3 ...
##  $ ca      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ thal    : Factor w/ 4 levels "0","1","2","3": 2 3 3 3 3 2 3 4 4 3 ...
##  $ target  : Factor w/ 2 levels "Hypertension",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Exploratory Data Analysis

In this section, we will explore various facets of the dataset. We aim to visualize and assess the dispersion of data. We will begin by examining the summary statistics of columns hypothesized to influence hypertension.

Age (age):

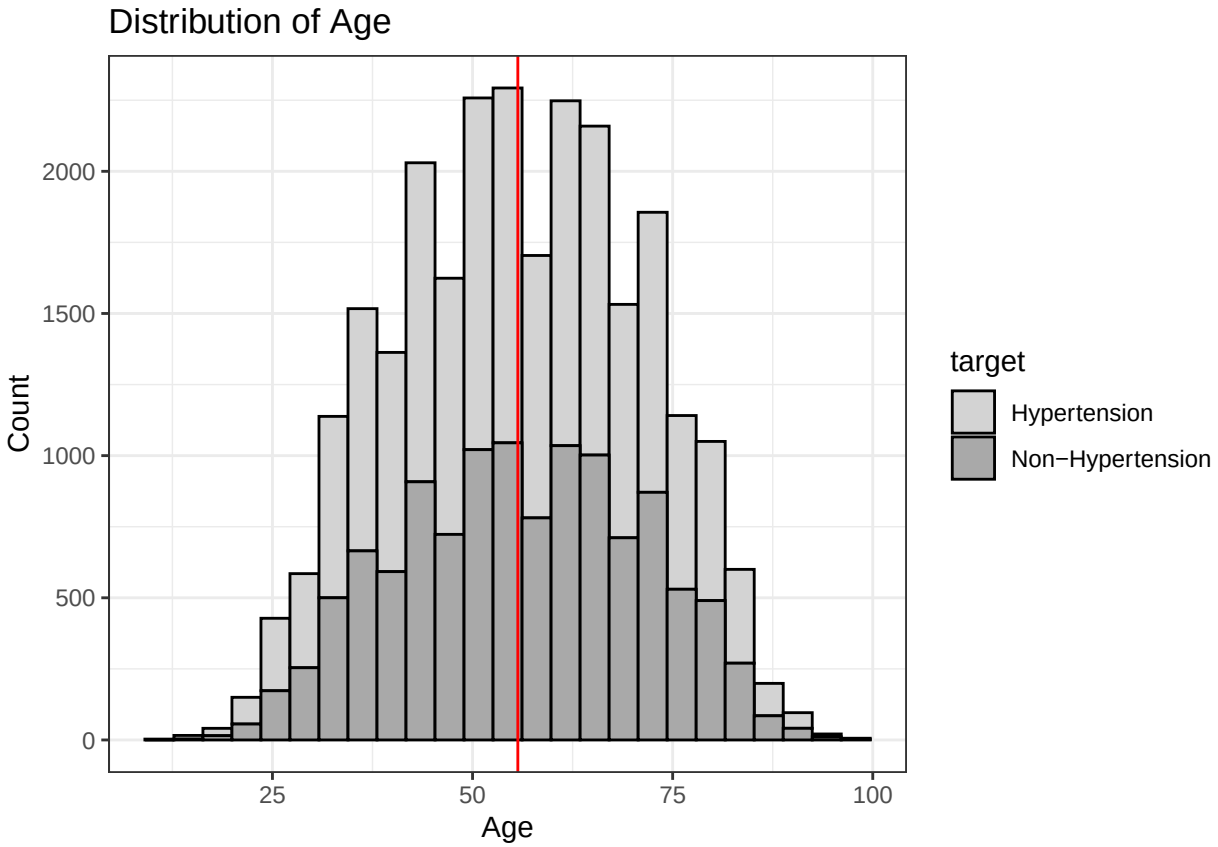
Age is a crucial demographic factor that often correlates with the risk of hypertension. Older individuals are generally more prone to hypertension. Let's take a look at a detailed statistics using the describe() function from the psych package.

```
describe(hypert$age)
```

```
##    vars     n  mean    sd median trimmed   mad min max range  skew kurtosis
## X1    1 26058 55.66 15.19     56   55.74 17.79  11  98    87 -0.04    -0.74
##      se
## X1 0.09
```

Although we thought that the older individuals are generally more prone to hypertension, from the above output we can see that our mean age is around 55-56 years with minimum of 11 and maximum of 98 years old. We can also see the distribution more clearly by plotting a graph.

```
# histogram plot of age distribution, vertical line for mean age
ggplot(data = hypert, aes(x = age, fill = target)) +
  geom_histogram(color = "black", bins = 25) +
  geom_vline(xintercept = mean(hypert$age), color = 'red') +
  labs(x = "Age", y = "Count", title = "Distribution of Age") +
  theme_bw() +
  scale_fill_manual(values = c("lightgray", "darkgray"))
```



From the graph above we observe a diverse distribution of ages that encompasses a range from young to elderly individuals. Despite the higher mean age, we will explore age as a potential factor contributing to hypertension.

contingency table:

We create a contingency table is by using the xtabs() function. This table provides the count of cases in each subgroup. For example, when applying the xtabs() function to the "target" and "age" columns, it yields the number of individuals with and without hypertension in each age subgroup, as illustrated below:

```
# Create and view  contingency table using xtabs()

xtabs(~target+age, data = hypert)
```

```
##                   age
## target              11  12  13  14  15  16  17  18  19  20  21  22  23  24  25
##   Hypertension       1   2   2   2   3   6   5   5  16  10  20  31  33  51  57
##   Non-Hypertension   0   0   0   0   0   3   0   6   9   7  16  17  16  31  41
##                   age
## target              26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   Hypertension      66  81  90 121 120 129 163 169 177 189 213 223 227 252 259
##   Non-Hypertension  45  56  80  76  98 115 109 130 146 150 156 176 183 176 202
##                   age
## target              41  42  43  44  45  46  47  48  49  50  51  52  53  54  55
##   Hypertension     260 271 279 289 283 297 305 299 308 315 308 306 316 312 310
##   Non-Hypertension 214 213 218 232 245 231 243 249 248 254 258 261 260 262 267
##                   age
## target              56  57  58  59  60  61  62  63  64  65  66  67  68  69  70
```

```
##   Hypertension     310 314 307 302 310 304 297 302 299 288 283 287 277 271 273
##   Non-Hypertension 256 261 262 258 258 255 262 260 255 256 246 245 246 237 228
##                   age
## target              71  72  73  74  75  76  77  78  79  80  81  82  83  84  85
##   Hypertension     254 250 245 236 208 198 205 157 149 145 109 104  87  83  56
##   Non-Hypertension 231 230 208 202 196 180 154 145 138 106 101  91  72  58  49
##                   age
## target              86  87  88  89  90  91  92  93  94  95  96  97  98
##   Hypertension      40  49  25  21  15   8  11   3   3   3   1   2   2
##   Non-Hypertension  41  22  22  17   7  10   7   3   3   3   2   2   0
```

The table above displays the distribution of individuals across age groups and their respective hypertension status. Each row represents an age group, while the columns represent the counts of individuals with and without hypertension within each age group. For instance, there are 1 individual aged 11 with hypertension and 0 individuals aged 11 without hypertension. This table provides a comprehensive overview of the relationship between age and hypertension status.

Resting Blood Pressure (trestbps):

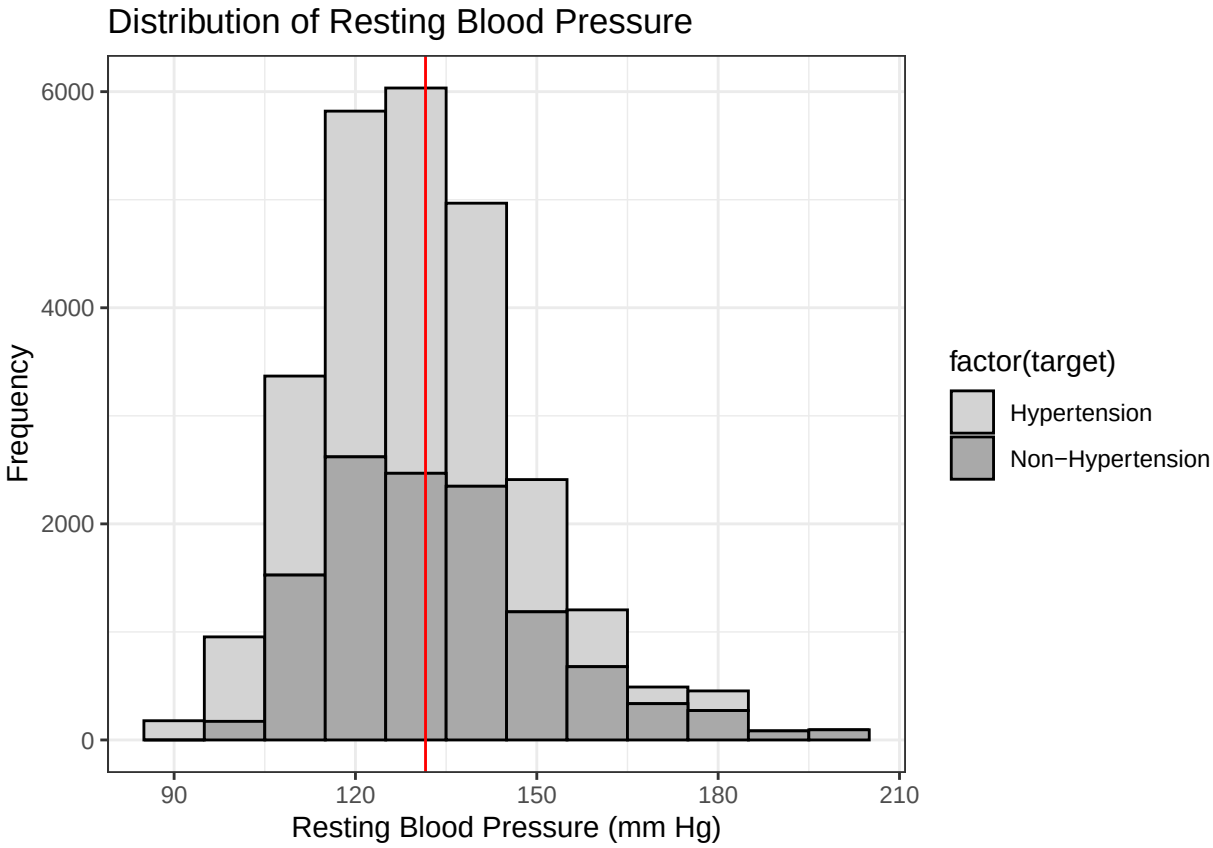
Resting blood pressure or systolic blood pressure is another significant factor that can contribute to hypertension. It is a fundamental clinical measure directly associated with hypertension. Higher resting blood pressure levels are indicative of hypertension risk. Hypertension is often synonymous with high blood pressure, particularly elevated systolic blood pressure. However, in this study, we aim to explore whether hypertension is solely attributable to high resting systolic blood pressure or if there are additional factors involved. Let's begin by examining the summary statistics using the describe() function:

```
describe(hypert$trestbps)
```

```
##    vars     n   mean   sd median trimmed   mad min max range skew kurtosis   se
## X1    1 26058 131.59 17.6    130  130.38 14.83  94 200   106 0.72     0.92 0.11
```

The summary statistics indicates that the average resting blood pressure is approximately 131.59 mm Hg, with a standard deviation of about 17.6 mm Hg. The values range from 94 mm Hg to 200 mm Hg, with a median value of 130 mm Hg. The distribution appears slightly skewed to the right (skewness = 0.72), and it has a positive kurtosis value (0.92), suggesting slightly heavier tails than a normal distribution. We can visualize this with a bar graph below:

```
# Plotting histogram of resting blood pressure (trestbps) with differentiation based on hypertension st
ggplot(data = hypert, aes(x = trestbps, fill = factor(target))) +
  geom_histogram(binwidth = 10, color = "black") +
  labs(x = "Resting Blood Pressure (mm Hg)", y = "Frequency", title = "Distribution of Resting Blood Pr
  scale_fill_manual(values = c("lightgray", "darkgray")) +
  theme_bw()+
  geom_vline(xintercept=mean(hypert$trestbps), color='red')
```



The graph shows hypertension at the peak with a frequency of approximately 6000 and non-hypertension at approximately 2500 suggests that the majority of individuals in the dataset are classified as hypertensive rather than non-hypertensive.

This observation is important because it indicates an imbalance in the distribution of hypertension status within the dataset. Such an imbalance can influence the performance and accuracy of predictive models trained on this data, especially as our goal is to predict hypertension. Therefore, it's crucial to address this class imbalance during the modeling process to ensure fair and accurate predictions.

Similarly we can also look at the contingency table:

```
xtabs(~target+trestbps, data = hypert)
```

```
##                   trestbps
## target               94  100  101  102  104  105  106  108  110  112  114  115
##   Hypertension      178  170   96  168   82  266   82  342  702  454    0  262
##   Non-Hypertension    0  172    0    0    0    0    0  174  962  310   80    0
##                   trestbps
## target              117  118  120  122  123  124  125  126  128  129  130  132
##   Hypertension        0  460 1968  260    0  170  342   88  532   82 2018  256
##   Non-Hypertension   94  156 1244   86   82  344  614  168  510    0 1080  422
##                   trestbps
## target              134  135  136  138  140  142  144  145  146  148  150  152
##   Hypertension      166  424   84  800 1484  168    0   84   96   88  774  168
##   Non-Hypertension  208   80  158  232 1318   88  184  368   92   78  676  250
##                   trestbps
## target              154  155  156  160  164  165  170  172  174  178  180  192
##   Hypertension        0   98   88  438    0    0   74   80    0   82  100    0
```

```
##   Non-Hypertension   90    0    0  510   86   82  254    0   82   92  180   84
##                   trestbps
## target              200
##   Hypertension        0
##   Non-Hypertension   94
```

The table presents the distribution of individuals based on their resting blood pressure (trestbps) and hypertension status. Each row represents a specific blood pressure value, with columns indicating the counts of individuals with and without hypertension. Notably, there's a diverse distribution of hypertension across different blood pressure levels. While individuals with low systolic blood pressure show instances of hypertension, the number of hypertensive patients decreases as blood pressure approaches the normal range. However, beyond approximately 126 mm Hg, there's a resurgence in the number of hypertensive patients, although there are exceptions. Surprisingly, individuals with extremely high blood pressure readings (at 192 and 200 mm Hg) are not classified as hypertensive.

```
hyper_115 <- hypert|>
  filter(trestbps == 115)
hyper_t <- hypert|>
  filter(trestbps != 115)
```

```
describe(hyper_115)
```

```
##          vars   n   mean     sd median trimmed   mad min   max range  skew
## age         1 262  57.04  16.46   57.0   57.37 17.79  13  92.0  79.0 -0.16
## sex*        2 262   1.50   0.50    1.5    1.50  0.74   1   2.0   1.0  0.00
## cp*         3 262   1.61   0.92    1.0    1.51  0.00   1   3.0   2.0  0.84
## trestbps    4 262 115.00   0.00  115.0  115.00  0.00 115 115.0   0.0   NaN
## chol        5 262 368.58 131.04  303.0  357.83 63.75 260 564.0 304.0  0.78
## fbs*        6 262   1.00   0.00    1.0    1.00  0.00   1   1.0   0.0   NaN
## restecg*    7 262   1.37   0.48    1.0    1.33  0.00   1   2.0   1.0  0.55
## thalach     8 262 175.90  10.69  181.0  176.74  5.93 160 185.0  25.0 -0.76
## exang*      9 262   1.00   0.00    1.0    1.00  0.00   1   1.0   0.0   NaN
## oldpeak    10 262   0.93   0.67    1.2    0.96  0.59   0   1.6   1.6 -0.54
## slope*     11 262   2.33   0.47    2.0    2.29  0.00   2   3.0   1.0  0.73
## ca         12 262   0.00   0.00    0.0    0.00  0.00   0   0.0   0.0   NaN
## thal*      13 262   3.31   0.46    3.0    3.26  0.00   3   4.0   1.0  0.84
## target*    14 262   1.00   0.00    1.0    1.00  0.00   1   1.0   0.0   NaN
##          kurtosis   se
## age         -0.60 1.02
## sex*        -2.01 0.03
## cp*         -1.30 0.06
## trestbps      NaN 0.00
## chol        -1.31 8.10
## fbs*          NaN 0.00
## restecg*    -1.70 0.03
## thalach     -1.32 0.66
## exang*        NaN 0.00
## oldpeak     -1.48 0.04
## slope*      -1.48 0.03
## ca            NaN 0.00
## thal*       -1.30 0.03
## target*       NaN 0.00
```

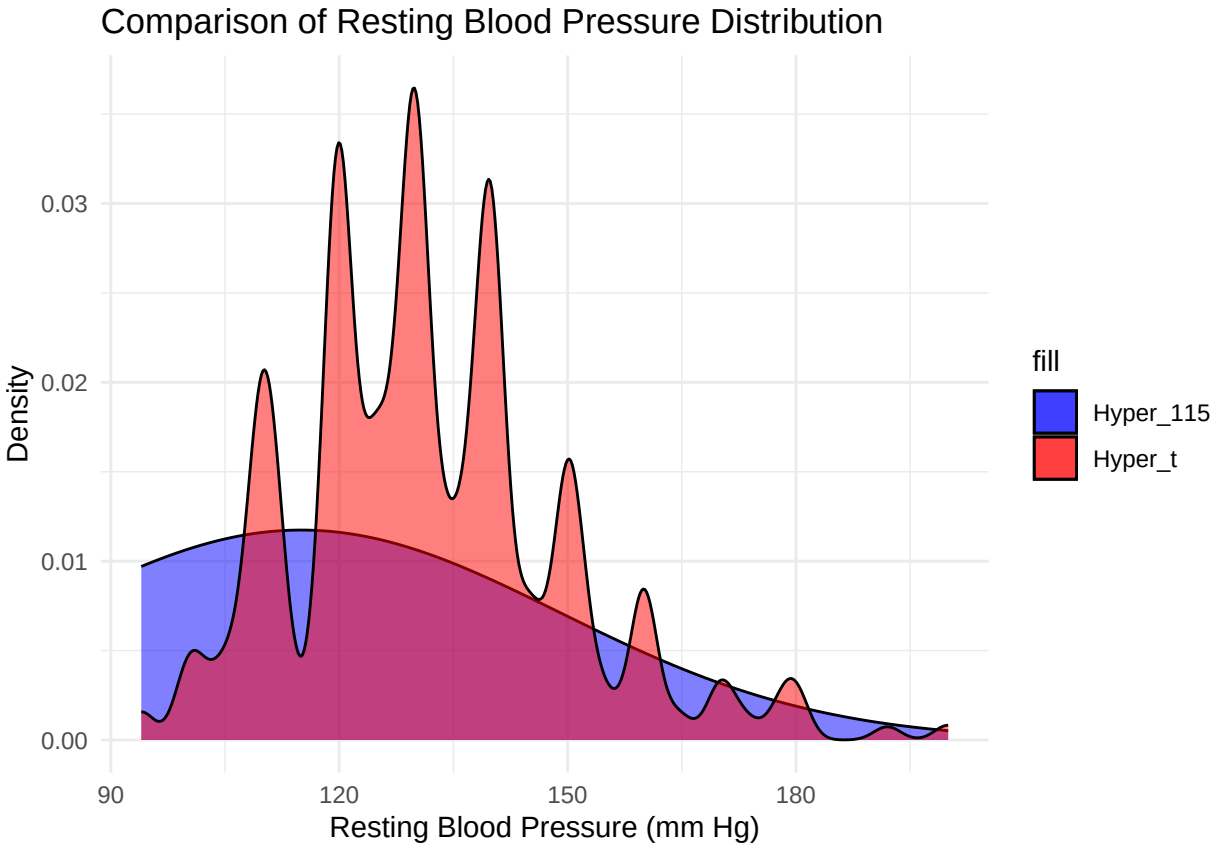
```r
describe(hyper_t)
```

```
##          vars     n   mean    sd median trimmed   mad min   max range  skew
## age         1 25796  55.64 15.18  56.00   55.72 17.79  11  98.0  87.0 -0.04
## sex*        2 25796   1.50  0.50   1.50    1.50  0.74   1   2.0   1.0  0.00
## cp*         3 25796   1.96  1.02   2.00    1.86  1.48   1   4.0   3.0  0.49
## trestbps    4 25796 131.76 17.61 130.00  130.57 14.83  94 200.0 106.0  0.71
## chol        5 25796 245.04 48.66 240.00  242.91 47.44 126 417.0 291.0  0.54
## fbs*        6 25796   1.15  0.36   1.00    1.06  0.00   1   2.0   1.0  1.94
## restecg*    7 25796   1.53  0.53   2.00    1.52  0.00   1   3.0   2.0  0.17
## thalach     8 25796 149.38 22.80 152.00  150.67 23.72  71 202.0 131.0 -0.52
## exang*      9 25796   1.33  0.47   1.00    1.29  0.00   1   2.0   1.0  0.72
## oldpeak    10 25796   1.04  1.17   0.75    0.85  1.11   0   6.2   6.2  1.26
## slope*     11 25796   2.40  0.62   2.00    2.46  1.48   1   3.0   2.0 -0.52
## ca         12 25796   0.73  1.01   0.00    0.54  0.00   0   4.0   4.0  1.28
## thal*      13 25796   3.32  0.61   3.00    3.36  0.00   1   4.0   3.0 -0.45
## target*    14 25796   1.46  0.50   1.00    1.45  0.00   1   2.0   1.0  0.17
##          kurtosis   se
## age         -0.74 0.09
## sex*        -2.00 0.00
## cp*         -1.17 0.01
## trestbps     0.91 0.11
## chol         0.58 0.30
## fbs*         1.78 0.00
## restecg*    -1.34 0.00
## thalach     -0.07 0.14
## exang*      -1.48 0.00
## oldpeak      1.50 0.01
## slope*      -0.63 0.00
## ca           0.71 0.01
## thal*        0.22 0.00
## target*     -1.97 0.00
```

Comparing the filtered data frame hyper_115 to the rest of the data hyper_t, we observed distinct differences in their descriptive statistics. In hyper_115, all individuals shared a fixed resting blood pressure of 115 mm Hg, along with a mean age of approximately 57.04 years. Conversely, hyper_t exhibited a wider range of blood pressure values, with a mean resting blood pressure of 131.76 mm Hg and a slightly lower mean age of about 55.64 years. Furthermore, hyper_t displayed higher mean values for cholesterol, heart rate, and other variables compared to hyper_115. These findings suggest that hypertension may involve multiple factors beyond elevated blood pressure alone, such as age and cholesterol levels. Understanding these additional parameters could provide valuable insights into the complexity of hypertension and inform more holistic approaches to its diagnosis and management.

We can also visualize using a density plot to compare the distributions of resting blood pressure between the two groups hyper_115 and hyper_t:

```r
ggplot() +
  geom_density(data = hyper_115, aes(x = trestbps, fill = "Hyper_115"), alpha = 0.5) +
  geom_density(data = hyper_t, aes(x = trestbps, fill = "Hyper_t"), alpha = 0.5) +
  labs(x = "Resting Blood Pressure (mm Hg)", y = "Density",
       title = "Comparison of Resting Blood Pressure Distribution") +
  scale_fill_manual(values = c("Hyper_115" = "blue", "Hyper_t" = "red")) +
  theme_minimal()
```



from above it seems like there is a concentration of individuals with hypertension (Hyper_t group) around a resting blood pressure of 125 mm Hg. This could imply that 125 mm Hg is a common or significant blood pressure level among individuals with hypertension in the Hyper_t group.

Fasting Blood Sugar(fbs > 120 mg/dl (1: yes; 0: no):

Elevated fasting blood sugar levels, indicated by values greater than 120 mg/dl, are often associated with conditions such as diabetes mellitus, which is a known risk factor for hypertension. Individuals with diabetes are more likely to develop hypertension due to various physiological mechanisms, including insulin resistance, endothelial dysfunction, and dysregulation of the renin-angiotensin-aldosterone system.

Including fasting blood sugar as a predictor variable in predictive models for hypertension can provide valuable insights into the relationship between glucose metabolism and blood pressure regulation. Individuals with elevated fasting blood sugar levels may have an increased risk of hypertension, and incorporating this variable into predictive models can help identify high-risk populations and guide preventive interventions. Lets check out the how many people with hypertension have a high blood pressure

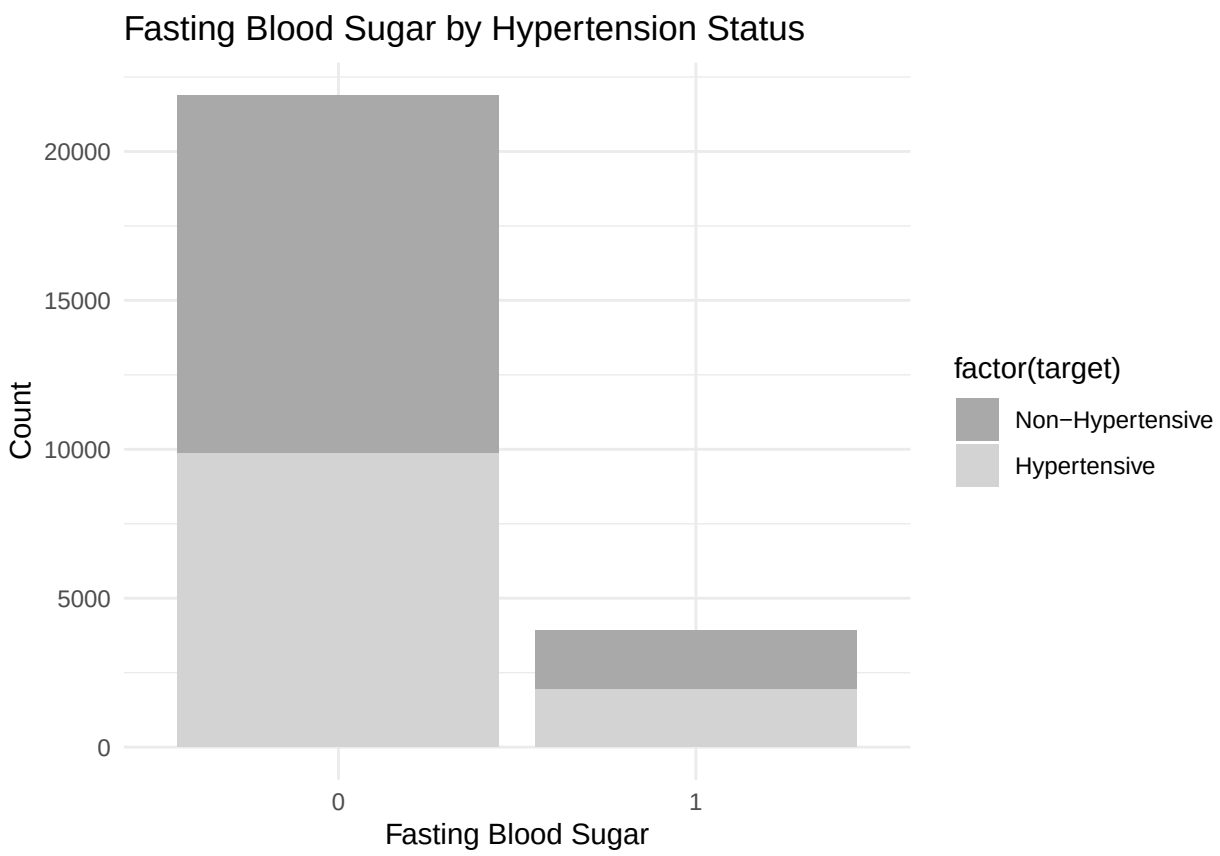
```
xtabs(~target+fbs, data = hypert)
```

```
##                   fbs
## target                0     1
##   Hypertension    12292  1982
##   Non-Hypertension  9860  1924
```

From the contingency table reveals that there is a notable difference in the number of individuals classified as hypertensive between the two fasting blood sugar groups. Specifically, there are more individuals with normal

fasting blood sugar levels (less than 120 mg/dl) who are hypertensive compared to those with elevated fasting blood sugar levels (120 mg/dl or higher). However, this difference is not overwhelmingly large, suggesting that while fasting blood sugar levels may play a role in hypertension, they are not the sole determinant. Additionally, within each fasting blood sugar group, there is a similar distribution between hypertensive and non-hypertensive individuals, indicating that other variables and factors likely contribute to the development of hypertension.We can also visualize this in the bar graph below.

```
ggplot(hyper_t, aes(x = fbs, fill = factor(target))) +
  geom_bar() +
  labs(x = "Fasting Blood Sugar", y = "Count", title = "Fasting Blood Sugar by Hypertension Status") +
  scale_fill_manual(values = c("darkgray", "lightgray"), labels = c("Non-Hypertensive", "Hypertensive"))
  theme_minimal()
```



Cholesterol (chol: Serum cholestoral in mg/dl):

When cholesterol accumulates in the bloodstream due to insufficient removal by the body, it can adhere to the walls of arteries. This buildup can lead to the condition known as atherosclerosis, characterized by the stiffening and narrowing of arteries over time. Consequently, the heart must exert greater effort to pump blood through these constricted arteries, resulting in an increase in blood pressure. To gain further insight, let's examine the summary statistics for the "cholesterol" column in our dataset using the describe() function:

```
describe(hypert$chol)
```

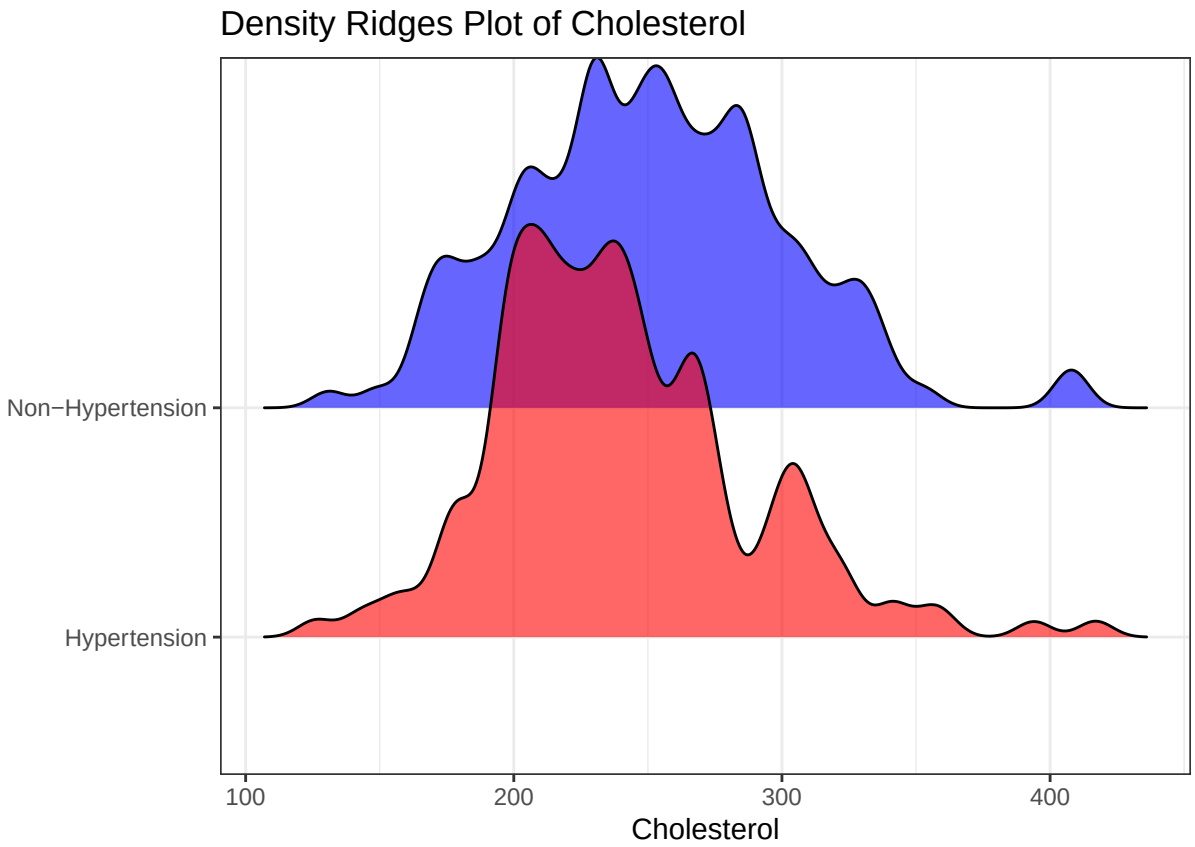
```
##    vars     n   mean    sd median trimmed   mad min max range skew kurtosis
## X1    1 26058 246.29 51.65    240   243.5 47.44 126 564   438  1.1     4.15
```

```
##      se
## X1 0.32
```

The summary statistics above reveal that the dataset consists of one variable with 26,058 observations. The mean cholesterol level of approximately 246.29 mg/dl, along with a standard deviation of 51.65 mg/dl, reflects the central tendency and variability of cholesterol values. The median value of 240 mg/dl further signifies the midpoint of the dataset's distribution. The range spanning from 126 mg/dl to 564 mg/dl illustrates the diversity of cholesterol levels among individuals. The positively skewed distribution, indicated by a skewness value of 1.1, and leptokurtic shape with a kurtosis value of 4.15 suggest potential outliers and a heavier tail in the distribution. Considering optimal cardiovascular health, the mean cholesterol level exceeds the recommended average of 200 mg/dl, requires attention to individual health factors and reference ranges utilized by healthcare organizations for accurate interpretation. We will plot a ridge density plot and a bar plto distrebution to see how cholesterol is distributed in hypertensive and non hypertensive patients:

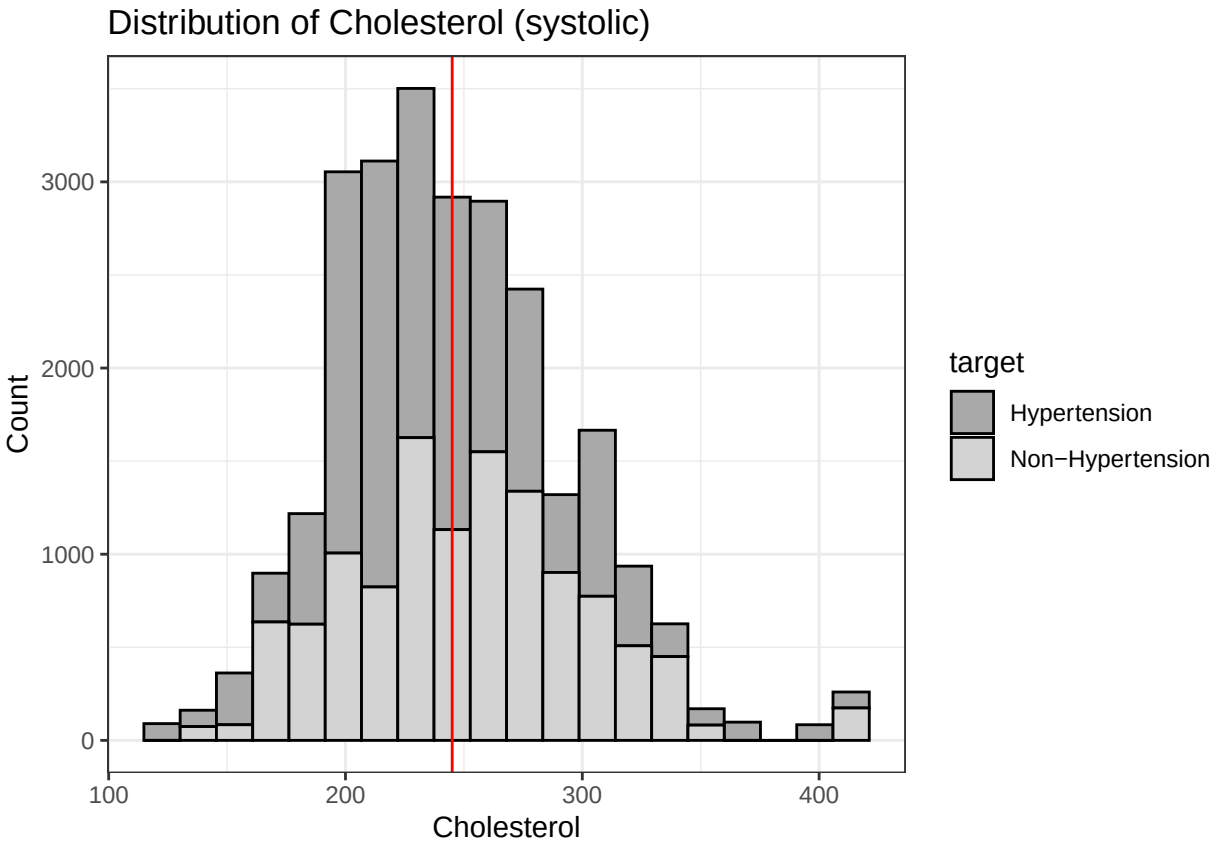
```
ggplot(hyper_t, aes(x = chol, y = target, fill = target)) +
  geom_density_ridges(alpha = 0.6, color = 'black') +
  labs(x = "Cholesterol", y = "", title = "Density Ridges Plot of Cholesterol") +
  theme_bw() +
  scale_fill_manual(values = c("red", "blue")) +
  theme(legend.position = "none")
```

```
## Picking joint bandwidth of 6.33
```




```
ggplot(hyper_t, aes(x = chol, fill = target)) +
  geom_histogram(color = "black", bins = 20) +
```

```
  geom_vline(xintercept = mean(hyper_t$chol), color = 'red') +
  labs(x = "Cholesterol", y = "Count", title = "Distribution of Cholesterol (systolic)") +
  theme_bw() +
  scale_fill_manual(values = c("darkgray", "lightgray"))
```



Maximum Heart Rate Achieved (thalach):

The maximum heart rate achieved during exercise can provide insights into cardiovascular fitness and potential risk factors for hypertension. let take a look at the summary statistics.

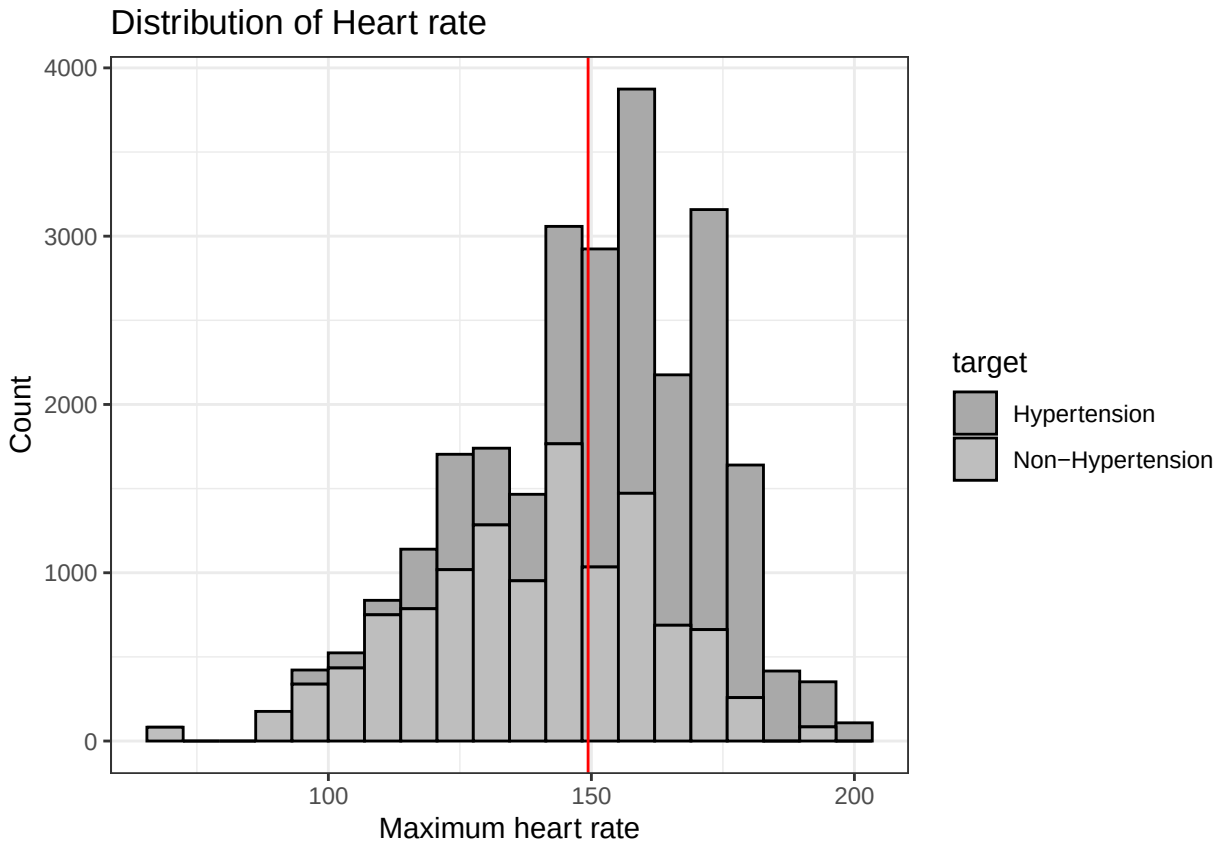
```
describe(hypert$thalach)
```

```
##    vars     n   mean    sd median trimmed   mad min max range  skew kurtosis
## X1    1 26058 149.64 22.87    153  150.94 22.24  71 202   131 -0.52    -0.08
##      se
## X1 0.14
```

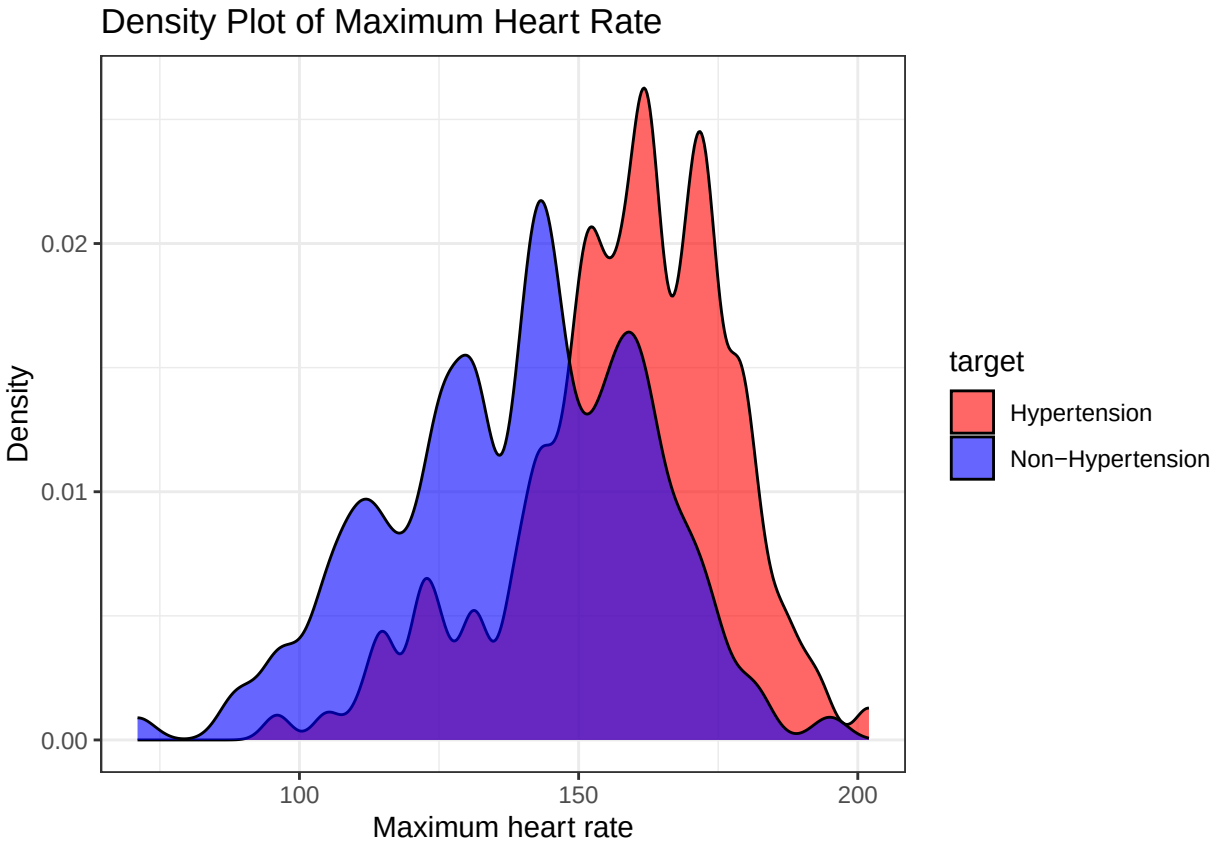
The summary statistics for the variable thalach (maximum heart rate achieved during exercise) shows the distribution within the dataset with a mean heart rate of approximately 149.64 beats per minute and a standard deviation of 22.87 beats per minute, we get a sense of the average heart rate and the variability around that mean. The median heart rate of 153 beats per minute further emphasizes the central tendency of the data. Notably, the minimum heart rate observed during exercise was 71 beats per minute, while the maximum reached 202 beats per minute, highlighting the range of heart rates among individuals. The slightly negative skewness and near-normal kurtosis suggest a distribution that leans slightly to the left but

is generally within a typical range. Together, these statistics offer valuable insights into the cardiovascular fitness levels of the individuals and their potential association with hypertension. Let's also vusialize in a barlot and a density plot.

```
ggplot() +
  geom_histogram(data = hyper_t, mapping = aes(x = thalach, fill = target), color = "black", bins = 20)
  theme_bw() +
  geom_vline(xintercept = mean(hyper_t$thalach), color = 'red') +
  labs(x = "Maximum heart rate", y = "Count", title = "Distribution of Heart rate") +
  scale_fill_manual(values = c("darkgray", "gray"))
```




```
ggplot(hyper_t, aes(x = thalach, fill = target)) +
  geom_density(alpha = 0.6) +
    labs(x = "Maximum heart rate", y = "Density", title = "Density Plot of Maximum Heart Rate") +
  scale_fill_manual(values = c("red", "blue")) +
  theme_bw()
```



The observations from the graph suggest interesting patterns in the relationship between maximum heart rate during exercise and hypertension status. When the maximum heart rate falls within the range of approximately 155-160 beats per minute, the count of individuals classified as hypertensive peaks at around 3800. Conversely, when the maximum heart rate is in the range of 125-130 beats per minute, the count of individuals classified as non-hypertensive reaches a peak of approximately 1700. This implies that there may be an association between maximum heart rate during exercise and hypertension status. Specifically, higher maximum heart rates during exercise appear to be more prevalent among individuals classified as hypertensive, while lower maximum heart rates are associated with a higher count of individuals classified as non-hypertensive. This observation suggests that maximum heart rate during exercise could potentially serve as a predictive factor or indicator for hypertension status,

BUILDING MODELS:

We'll start by splitting the dataset into training and testing sets. With the training set, we'll construct our models, subsequently employing it to predict outcomes using the testing set. To validate our predictions, we can utilize a confusion matrix.

Splitting the Dataset:

We'll divide the dataset into an 80/20 split, reserving 80% for training our model and using the remaining 20% for testing. Essentially, we'll engage in supervised machine learning, employing logistic regression to predict outcomes.

Before we begin, we confirm that our data is not imbalanced by checking the class distribution of the target variable.

```r
class_distribution <- table(hypert$target) / nrow(hypert)
print(class_distribution)
```

```
##
##     Hypertension Non-Hypertension
##         0.547778         0.452222
```

There is a slight imbalance in the data but it is not severe enough to warrant specialized techniques for oversampling.

```r
set.seed(456)
split <- createDataPartition(hypert$target, p = 0.8, list = FALSE)
train_data <- hypert[split, ]
test_data <- hypert[-split, ]
```

MODEL 1

In Model 1, we run a baseline model on the entire dataset.

```r
m1 <- glm(formula = target ~ ., family = binomial(link = "logit"), data = train_data)
summary(m1)
```

```
##
## Call:
## glm(formula = target ~ ., family = binomial(link = "logit"),
##     data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4376  -0.5362  -0.2117   0.3717   2.9596
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.3141232  0.3370962   0.932   0.3514
## age         -0.0007802  0.0014126  -0.552   0.5807
## sexM         0.0186803  0.0426111   0.438   0.6611
## cp1         -1.0575371  0.0661944 -15.976  < 2e-16 ***
## cp2         -1.7991025  0.0559616 -32.149  < 2e-16 ***
## cp3         -1.8180456  0.0781878 -23.252  < 2e-16 ***
## trestbps     0.0138923  0.0012257  11.335  < 2e-16 ***
## chol         0.0007031  0.0004128   1.703   0.0885 .
## fbs1        -0.0333470  0.0673157  -0.495   0.6203
## restecg1    -0.6929247  0.0443530 -15.623  < 2e-16 ***
## restecg2    -0.4015722  0.2619309  -1.533   0.1252
## thalach     -0.0131158  0.0011362 -11.544  < 2e-16 ***
## exang1       0.7384446  0.0495717  14.897  < 2e-16 ***
## oldpeak      0.5410450  0.0275987  19.604  < 2e-16 ***
## slope1       0.6590031  0.1032719   6.381 1.76e-10 ***
## slope2      -0.0151200  0.1128153  -0.134   0.8934
## ca           0.8327554  0.0245819  33.877  < 2e-16 ***
## thal1       -1.1139509  0.2374552  -4.691 2.72e-06 ***
```

```
## thal2       -1.7869560  0.2264164  -7.892 2.97e-15 ***
## thal3        0.1442161  0.2266564   0.636   0.5246
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28711  on 20847  degrees of freedom
## Residual deviance: 14340  on 20828  degrees of freedom
## AIC: 14380
##
## Number of Fisher Scoring iterations: 6
```

The baseline logistic regression model shows several key predictors that significantly influence the likelihood of the outcome, hypertention, with various factors showing strong statistical significance. Types of chest pain (cp1, cp2, cp3) and the maximum heart rate achieved (thalach) have negative coefficients, indicating that increases in these variables are associated with a reduced probability of the outcome. Conversely, trestbps (resting blood pressure) and chol (cholesterol levels) are positively associated with the outcome, suggesting that higher values increase the likelihood of the outcome. The model also identifies oldpeak (ST depression induced by exercise relative to rest), slope1 (the slope of the peak exercise ST segment), and ca (number of major vessels colored by fluoroscopy) as significant predictors with positive associations. Exercise-induced angina (exang1) and certain types of thalassemia (thal2) are also significant.

However, age, sexM, fbs1, restecg2, thal3, and slope2 do not show statistically significant effects on the outcome in this model, suggesting that they may not be useful predictors in this context. The model's fit, as indicated by the Akaike Information Criterion (AIC) of 14364 and the substantial reduction in deviance from the null model to the residual, suggests that it adequately captures the relationship between predictors and the outcome. Let's see if we can get a lower AIC from the baseline.

The Variance Inflation Factor (VIF):

Lets check Variance Inflation Factor (VIF) to detect 'multicollinearity' in our models as quantifies the correlation and its strength between independent variables in a regression model. The interpretation of VIF values is as follows:

- VIF < 1: No correlation
- $1 <$ VIF < 5: Moderate correlation
- VIF > 5: Severe correlation

```
knitr::kable(vif(m1))
```

	GVIF	Df	GVIF^(1/(2*Df))
age	1.017731	1	1.008827
sex	1.010049	1	1.005012
cp	1.500764	3	1.070004
trestbps	1.097534	1	1.047633
chol	1.088483	1	1.043304
fbs	1.129918	1	1.062976
restecg	1.132382	2	1.031569
thalach	1.292891	1	1.137053
exang	1.161064	1	1.077527

	GVIF	Df	GVIF^(1/(2*Df))
oldpeak	1.549640	1	1.244845
slope	1.700923	2	1.142013
ca	1.125263	1	1.060784
thal	1.220964	3	1.033833

We focus on the GVIF, as it measures how much the variance of the estimated regression coefficients is increased due to multicollinearity. As we can see, there is little/moderate correlation among the variables so we do not need to address multicollinearity.

MODEL 2

In this second model, we maintain all of our variables but take the log transformation of the variables that we identified as skewed during EDA (chol, trestbps, and thalach) to see if this will improve our model performance.

```
m2 <- glm(formula = target ~ age + sex + fbs + cp + restecg + exang + oldpeak + slope + thal + log(chol
summary(m2)
```

```
##
## Call:
## glm(formula = target ~ age + sex + fbs + cp + restecg + exang +
##     oldpeak + slope + thal + log(chol + 1) + log(trestbps + 1) +
##     log(thalach + 1), family = binomial(link = "logit"), data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3316  -0.5689  -0.2286   0.4397   2.7799
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        2.016479   1.136587   1.774   0.0760 .
## age                0.001035   0.001343   0.771   0.4409
## sexM               0.018622   0.040633   0.458   0.6467
## fbs1               0.313647   0.060048   5.223 1.76e-07 ***
## cp1               -1.050589   0.060376 -17.401  < 2e-16 ***
## cp2               -1.763159   0.052950 -33.298  < 2e-16 ***
## cp3               -2.010223   0.077822 -25.831  < 2e-16 ***
## restecg1          -0.684746   0.042629 -16.063  < 2e-16 ***
## restecg2          -0.388787   0.217839  -1.785   0.0743 .
## exang1             0.717166   0.047260  15.175  < 2e-16 ***
## oldpeak            0.630484   0.025846  24.394  < 2e-16 ***
## slope1             0.911858   0.099018   9.209  < 2e-16 ***
## slope2             0.475193   0.108262   4.389 1.14e-05 ***
## thal1             -0.242970   0.231175  -1.051   0.2932
## thal2             -1.143919   0.220736  -5.182 2.19e-07 ***
## thal3              0.888886   0.221185   4.019 5.85e-05 ***
## log(chol + 1)      0.538086   0.107446   5.008 5.50e-07 ***
## log(trestbps + 1)  1.607182   0.157877  10.180  < 2e-16 ***
## log(thalach + 1)  -2.638030   0.150624 -17.514  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28711  on 20847  degrees of freedom
## Residual deviance: 15573  on 20829  degrees of freedom
## AIC: 15611
##
## Number of Fisher Scoring iterations: 5
```

Model 2, m2, shows significant changes in comparison to m1. The logarithmic transformations of chol, trestbps, and thalach significantly improved the precision of the model's estimates, evident from the strong statistical significance ($p < 0.001$) and high z-values of these predictors. For instance, log(thalach + 1) has a notably large negative coefficient, indicating a strong inverse relationship between the logarithm of maximum heart rate achieved and the probability of the outcome, which contrasts sharply with m1 where the relationship was only slighly negative. This, log(chol + 1) and log(trestbps + 1), suggests that cholesterol and resting blood pressure have nonlinear effects on the outcome, which were better captured with the logarithmic transformation, thereby potentially resolving some nonlinearities or scaling issues present in m1.

Several predictors like cp1, cp2, cp3, restecg1, exang1, and oldpeak remain highly significant with large z-values, suggesting robust associations with the outcome. The continued significance and strength of these predictors affirm their importance in the model. However, the model's AIC, 15611, is higher, compared with and m1's AIC, 14380, suggesting that m2 may be overfitting or that it includes additional complexity that does not necessarily improve the model's prediction accuracy relative to the number of predictors used. This could indicate that while m2 incorporates more variables or more complex transformations, these adjustments may not provide a proportionate improvement in the model's explanatory power over m1.

MODEL 3

In this final model, we return to a simplier model as our more complex model, m2, did not perform as expected. We select predictors that were highly significant ($p < 0.05$) in the baseline model, cp, restecg1, thalach, exang1, oldpeak, slope1, ca, thal1 without log transformations to assess if they perform better than the baseline model.

```
m3 <- glm(formula = target ~ cp + restecg + trestbps + thalach + exang + thal + ca + oldpeak + slope, fa
summary(m3)
```

```
##
## Call:
## glm(formula = target ~ cp + restecg + trestbps + thalach + exang +
##     thal + ca + oldpeak + slope, family = binomial(link = "logit"),
##     data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4253  -0.5419  -0.2139   0.3703   2.9549
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.371249   0.322624   1.151    0.250
## cp1         -1.054064   0.065972 -15.977  < 2e-16 ***
## cp2         -1.805732   0.055100 -32.772  < 2e-16 ***
## cp3         -1.835025   0.077223 -23.763  < 2e-16 ***
```

```
## restecg1    -0.705935   0.043543 -16.212  < 2e-16 ***
## restecg2    -0.407319   0.258281  -1.577    0.115
## trestbps     0.014030   0.001215  11.549  < 2e-16 ***
## thalach     -0.013017   0.001131 -11.509  < 2e-16 ***
## exang1       0.738452   0.049466  14.928  < 2e-16 ***
## thal1       -1.098012   0.237944  -4.615 3.94e-06 ***
## thal2       -1.750113   0.225333  -7.767 8.05e-15 ***
## thal3        0.180958   0.225589   0.802    0.422
## ca           0.829727   0.024277  34.177  < 2e-16 ***
## oldpeak      0.544859   0.027458  19.843  < 2e-16 ***
## slope1       0.678418   0.102463   6.621 3.56e-11 ***
## slope2       0.001020   0.112094   0.009    0.993
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28711  on 20847  degrees of freedom
## Residual deviance: 14344  on 20832  degrees of freedom
## AIC: 14376
##
## Number of Fisher Scoring iterations: 6
```

This model, m3, effectively captures the essential predictors with a better balance of model complexity (lower AIC compared to m2 and m1) and accuracy, making it potentially more suitable for practical applications than m2. Moreover, all of the predictors in this model are highly significant. m3 offers a refinement over m1 in terms of predictive power and parsimony, making it a strong candidate model depending on the specific use case and the importance of various predictors in the practical context.

4. SELECTING MODELS:

Model Evaluation

For these three model evaluations, let's consider at metric of AUC to gain a comprehensive understanding of each model's performance. Here's how we can compute this metric for each model:

```
#Model Evaluation

pred_m1 <- predict(m1, newdata = test_data, type = "response")
pred_m2 <- predict(m2, newdata = test_data, type = "response")
pred_m3 <- predict(m3, newdata = test_data, type = "response")

#Extract true labels from test data
true_labels <- test_data$target

#AUC
auc_m1 <- roc(true_labels, pred_m1)$auc
```

```
## Setting levels: control = Hypertension, case = Non-Hypertension
```

```
## Setting direction: controls < cases
```

```r
auc_m2 <- roc(true_labels, pred_m2)$auc
```

```
## Setting levels: control = Hypertension, case = Non-Hypertension
## Setting direction: controls < cases
```

```r
auc_m3 <- roc(true_labels, pred_m3)$auc
```

```
## Setting levels: control = Hypertension, case = Non-Hypertension
## Setting direction: controls < cases
```

```r
print("Evaluation Metrics for Model 1:")
```

```
## [1] "Evaluation Metrics for Model 1:"
```

```r
print(paste("AUC:", auc_m1))
```

```
## [1] "AUC: 0.924201787501056"
```

```r
print("Evaluation Metrics for Model 2:")
```

```
## [1] "Evaluation Metrics for Model 2:"
```

```r
print(paste("AUC:", auc_m2))
```

```
## [1] "AUC: 0.908394288896054"
```

```r
print("Evaluation Metrics for Model 3:")
```

```
## [1] "Evaluation Metrics for Model 3:"
```

```r
print(paste("AUC:", auc_m3))
```

```
## [1] "AUC: 0.92428075806987"
```

Conducted an evaluation of three different models (Model 1, Model 2, and Model 3) by predicting their outcomes using test data and comparing them with the true labels. Predictions are made for each model using the predict function, specifying the type of response. The true labels are extracted from the test data. Subsequently, the AUC is calculated for each model using the 'roc' function. To ensure consistency in the levels of factor variables between the predicted and true labels, their levels are matched. This is done by converting the predicted and true labels into factors and setting their levels to be the same.

Finally, the evaluation results for each model are printed, providing insights into their performance across the metric.

Model Selection

```r
# Compare AUCs of different models and select the one with the highest AUC

best_model <- NULL

if (auc_m1 > auc_m2 & auc_m1 > auc_m3) {
  best_model <- "Model 1"
} else if (auc_m2 > auc_m1 & auc_m2 > auc_m3) {
  best_model <- "Model 2"
} else {
  best_model <- "Model 3"
}

print(paste("The best model is", best_model, "with AUC:", max(auc_m1, auc_m2, auc_m3)))
```

```
## [1] "The best model is Model 3 with AUC: 0.92428075806987"
```

```r
DF <- data.frame(Model = c("Model 1", "Model 2", "Model 3"),
                         AUC = c(auc_m1, auc_m2, auc_m3))
print(DF)
```

```
##     Model       AUC
## 1 Model 1 0.9242018
## 2 Model 2 0.9083943
## 3 Model 3 0.9242808
```

A higher AUC value generally indicates better performance in distinguishing between positive and negative instances. Therefore, based on these results, Model 3 appears to have slightly better performance than Model 1 and Model 2, as it has the highest AUC value. This model exhibits an AUC of 0.924280, indicating its superior discriminatory power compared to Model 1 (AUC: 0.924201) and Model 2 (AUC: 0.908394). While Model 1 also shows a high AUC, Model 3 outperforms it marginally. Model 2, on the other hand, demonstrates a slightly lower AUC compared to both Model 1 and Model 3.

Predictions on Evaluation Dataset:

We make our final prediction, create a dataframe with the predictions.

```r
write.csv(test_data, 'predictions.csv', row.names=FALSE)
head(test_data)
```

```
##    age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
## 11  55   F  0      140  239   0       1     160     0     1.2     2  0    2
## 28  49   M  2      110  175   0       1     123     0     0.6     2  0    2
## 30  55   M  2      130  197   1       0     152     0     1.2     0  0    2
## 32  46   F  0      120  177   0       1     140     0     0.4     2  0    3
## 34  55   F  2      125  273   0       0     152     0     0.5     0  1    2
## 39  46   F  2      155  269   0       1     148     0     0.8     2  0    2
##          target
## 11 Hypertension
## 28 Hypertension
## 30 Hypertension
```

```
## 32 Hypertension
## 34 Hypertension
## 39 Hypertension
```

```
levels(test_data$target) <- c("Non-Hypertension", "Hypertension")

# Calculate AUC for Model, Model 2, Model 3
AUC_m1 <- roc(test_data$target, as.numeric(pred_m1))
```

```
## Setting levels: control = Non-Hypertension, case = Hypertension
```

```
## Setting direction: controls < cases
```

```
AUC_m2 <- roc(test_data$target, as.numeric(pred_m2))
```

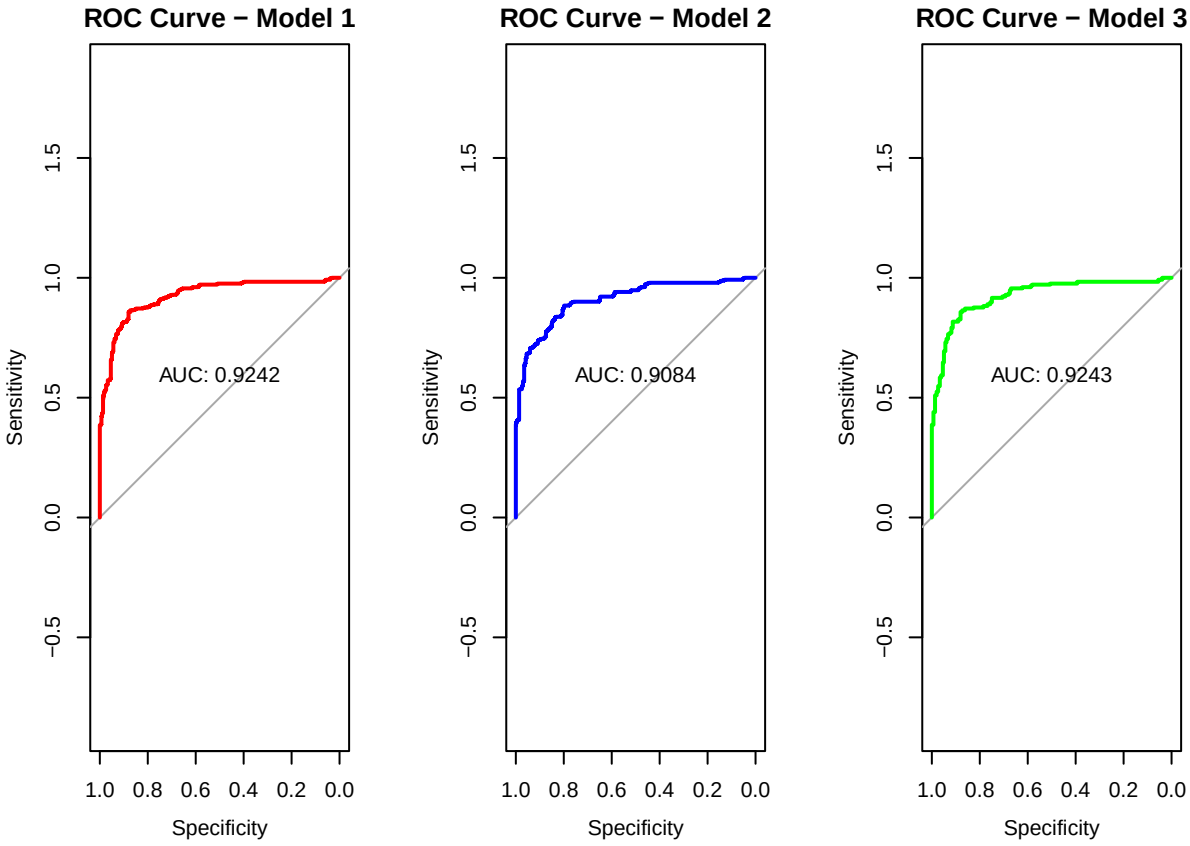
```
## Setting levels: control = Non-Hypertension, case = Hypertension
## Setting direction: controls < cases
```

```
AUC_m3 <- roc(test_data$target, as.numeric(pred_m3))
```

```
## Setting levels: control = Non-Hypertension, case = Hypertension
## Setting direction: controls < cases
```

Here's the plot the ROC curve for the AUC of three of our models:

```
par(mfrow = c(1,3))
# Plot ROC curve
plot(AUC_m1, col = "red", main = "ROC Curve - Model 1")
text(0.5, 0.5, paste("AUC:", round(AUC_m1$auc, digits = 4)), adj = c(0.5, -1))
plot(AUC_m2, col = "blue", main = "ROC Curve - Model 2")
text(0.5, 0.5, paste("AUC:", round(AUC_m2$auc, digits = 4)), adj = c(0.5, -1))
plot(AUC_m3, col = "green", main = "ROC Curve - Model 3")
text(0.5, 0.5, paste("AUC:", round(AUC_m3$auc, digits = 4)), adj = c(0.5, -1))
```



CONCLUSION:

In conclusion, this evaluation process systematically assessed the performance of three distinct models (Model 1, Model 2, and Model 3) in predicting outcomes using test data. Each model's predictions were compared against the true labels extracted from the test dataset. The evaluation metrics included the calculation of the Area Under the ROC Curve (AUC) to quantify the models' discriminative power. Despite encountering challenges such as mismatched levels between predicted and true labels, efforts were made to ensure consistency for accurate evaluation. Visualization techniques, including ROC curve plots, were employed to facilitate a deeper interpretation of the AUC values. The evaluation revealed variations in performance across the models, with Model 3 demonstrating a slightly higher AUC compared to the others. Overall, this evaluation process serves as a crucial step in assessing and selecting the most effective model for the task at hand, thereby enhancing decision-making in practical applications.