Cardio Disease

Mustapha Shehu Muhammad(CST/19/COM/00332)

2025-03-20

Understanding dataset

age: The person's age in days sex: The person's sex (1 = women, 2 = male) height: person's height in cm weigth: person's weigth in kg ap_hi: Systolic blood pressure ap_lo: Diastolic blood pressure cholestrol: person cholestrol level(1: normal, 2: above normal, 3: well above normal) glucose: person's glucose level (1: normal, 2: above normal, 3: well above normal) smoke: whether patient smokes or not(0: NO and 1: YES) alcohol: Alcohol intake(0: NO and 1: YES) active: Physical activity(0: NO and 1: YES) cardio: Heart Diseases—Target Variable(0: NO and 1: YES)

Load Data and Libraries

Load necessary libraries:

```
library(RWeka) # For J48 (C4.5 Implementation)
library(C50) # For C5.0
library(caret) # For cross-validation and performance evaluation
```

Load the dataset:

```
data <- read.csv("data.csv")
str(data)  # Check the structure
```

```
## 'data.frame':    70000 obs. of  14 variables:
##  $ id         : int  0 1 2 3 4 8 9 12 13 14 ...
##  $ age_days   : int  18393 20228 18857 17623 17474 21914 22113 22584 17668 19834 ...
##  $ age_year   : num  50.4 55.4 51.7 48.3 47.9 ...
##  $ gender     : int  2 1 1 2 1 1 1 2 1 1 ...
##  $ height     : int  168 156 165 169 156 151 157 178 158 164 ...
##  $ weight     : num  62 85 64 82 56 67 93 95 71 68 ...
##  $ ap_hi      : int  110 140 130 150 100 120 130 130 110 110 ...
##  $ ap_lo      : int  80 90 70 100 60 80 80 90 70 60 ...
##  $ cholesterol: int  1 3 3 1 1 2 3 3 1 1 ...
##  $ gluc       : int  1 1 1 1 1 2 1 3 1 1 ...
##  $ smoke      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ alco       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ active     : int  1 1 0 1 0 0 1 1 1 0 ...
##  $ cardio     : int  0 1 1 1 0 0 0 1 0 0 ...
```

Dataset Summary

```
summary(data)  # Basic summary
```

```
##        id           age_days        age_year         gender          height
##  Min.   :    0   Min.   :10798   Min.   :29.58   Min.   :1.00   Min.   : 55.0
##  1st Qu.:25007   1st Qu.:17664   1st Qu.:48.39   1st Qu.:1.00   1st Qu.:159.0
##  Median :50002   Median :19703   Median :53.98   Median :1.00   Median :165.0
##  Mean   :49972   Mean   :19469   Mean   :53.34   Mean   :1.35   Mean   :164.4
##  3rd Qu.:74889   3rd Qu.:21327   3rd Qu.:58.43   3rd Qu.:2.00   3rd Qu.:170.0
##  Max.   :99999   Max.   :23713   Max.   :64.97   Max.   :2.00   Max.   :250.0
##      weight           ap_hi             ap_lo          cholesterol
##  Min.   : 10.00   Min.   : -150.0   Min.   :  -70.00   Min.   :1.000
##  1st Qu.: 65.00   1st Qu.:  120.0   1st Qu.:   80.00   1st Qu.:1.000
##  Median : 72.00   Median :  120.0   Median :   80.00   Median :1.000
##  Mean   : 74.21   Mean   :  128.8   Mean   :   96.63   Mean   :1.367
##  3rd Qu.: 82.00   3rd Qu.:  140.0   3rd Qu.:   90.00   3rd Qu.:2.000
##  Max.   :200.00   Max.   :16020.0   Max.   :11000.00   Max.   :3.000
##       gluc           smoke              alco             active
##  Min.   :1.000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:1.000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:1.0000
##  Median :1.000   Median :0.00000   Median :0.00000   Median :1.0000
##  Mean   :1.226   Mean   :0.08813   Mean   :0.05377   Mean   :0.8037
##  3rd Qu.:1.000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:1.0000
##  Max.   :3.000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000
##      cardio
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.4997
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

Display First 6 Rows

```
#displays first 6 rows of the dataset

head(data)
```

```
##   id age_days age_year gender height weight ap_hi ap_lo cholesterol gluc smoke
## 1  0    18393 50.39178      2    168     62   110    80           1    1     0
## 2  1    20228 55.41918      1    156     85   140    90           3    1     0
## 3  2    18857 51.66301      1    165     64   130    70           3    1     0
## 4  3    17623 48.28219      2    169     82   150   100           1    1     0
## 5  4    17474 47.87397      1    156     56   100    60           1    1     0
## 6  8    21914 60.03836      1    151     67   120    80           2    2     0
##   alco active cardio
## 1    0      1      0
## 2    0      1      1
## 3    0      0      1
## 4    0      1      1
## 5    0      0      0
## 6    0      0      0
```

Data Preparation

Change gender to 0 and 1

```
# Load dplyr
library(dplyr)

# Map values: 1 → 0, 2 → 1
```

```r
data$gender <- recode(data$gender, `1` = 0, `2` = 1)

head(data)
```

```
##   id age_days age_year gender height weight ap_hi ap_lo cholesterol gluc smoke
## 1  0    18393 50.39178      1    168     62   110    80           1    1     0
## 2  1    20228 55.41918      0    156     85   140    90           3    1     0
## 3  2    18857 51.66301      0    165     64   130    70           3    1     0
## 4  3    17623 48.28219      1    169     82   150   100           1    1     0
## 5  4    17474 47.87397      0    156     56   100    60           1    1     0
## 6  8    21914 60.03836      0    151     67   120    80           2    2     0
##   alco active cardio
## 1    0      1      0
## 2    0      1      1
## 3    0      0      1
## 4    0      1      1
## 5    0      0      0
## 6    0      0      0
```

Check duplicate rows

```r
# Check for duplicates based on all columns
duplicate_row <- data[duplicated(data), ]

# Print the results
print(duplicate_row)
```

```
##  [1] id          age_days    age_year    gender      height      weight
##  [7] ap_hi       ap_lo       cholesterol gluc        smoke       alco
## [13] active      cardio
## <0 rows> (or 0-length row.names)
```

There's no duplicate

Handle missing values:

```r
# Check total missing values in the dataset
sum(is.na(data))
```

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

```r
# Check missing values per column
colSums(is.na(data))
```

```
##          id    age_days    age_year      gender      height      weight
##           0           0           0           0           0           0
##       ap_hi       ap_lo cholesterol        gluc       smoke        alco
##           0           0           0           0           0           0
##      active      cardio
##           0           0
```

There are no missing values

Drop age_days and rename age_year to age (integer)

```r
# Drop the 'age_days' column
data <- data[, !(names(data) == "age_days")]

# Rename 'age_year' to 'age'
```

```r
names(data)[names(data) == "age_year"] <- "age"

# Ensure 'age' is an integer
data$age <- as.integer(data$age)

# Verify the changes
head(data)
```

```
##   id age gender height weight ap_hi ap_lo cholesterol gluc smoke alco active
## 1  0  50      1    168     62   110    80           1    1     0    0      1
## 2  1  55      0    156     85   140    90           3    1     0    0      1
## 3  2  51      0    165     64   130    70           3    1     0    0      0
## 4  3  48      1    169     82   150   100           1    1     0    0      1
## 5  4  47      0    156     56   100    60           1    1     0    0      0
## 6  8  60      0    151     67   120    80           2    2     0    0      0
##   cardio
## 1      0
## 2      1
## 3      1
## 4      1
## 5      0
## 6      0
```

Drop Id column

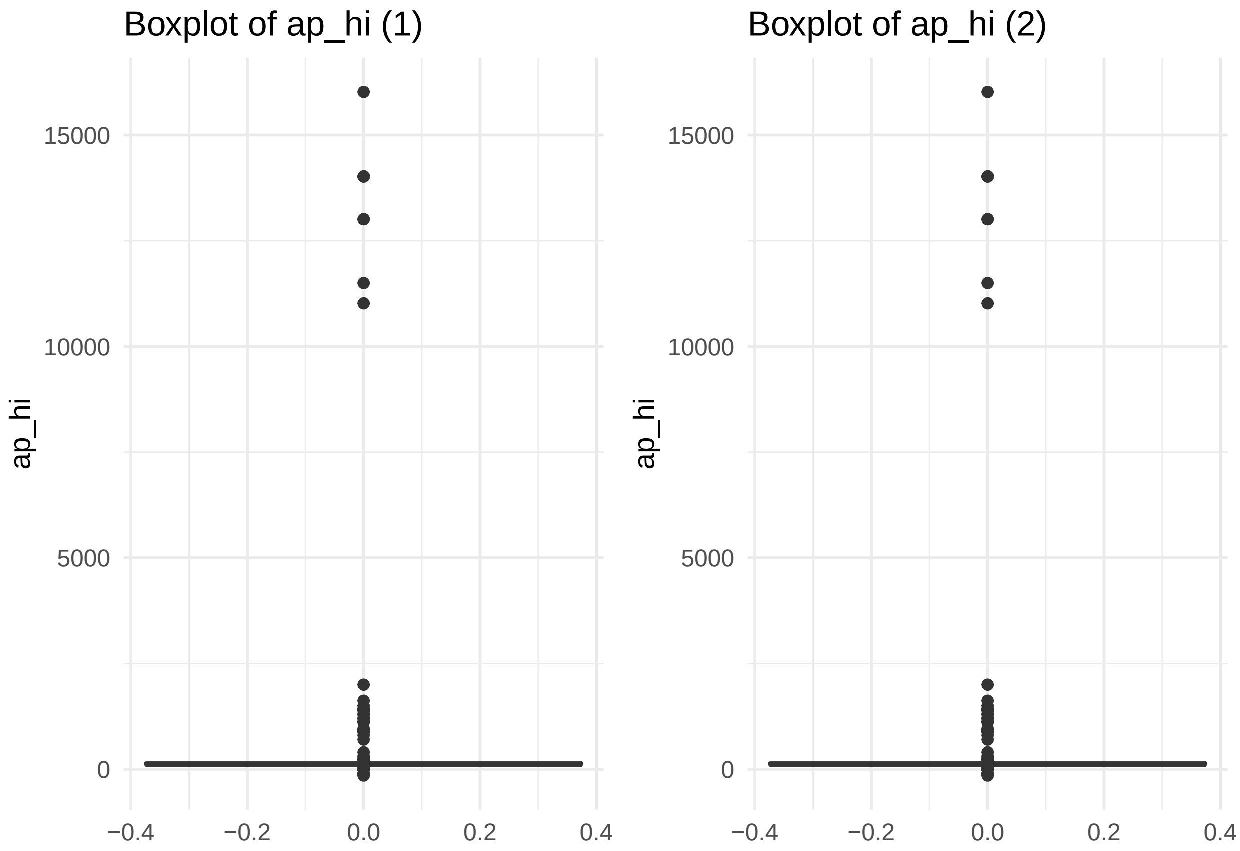
```r
# Drop 'id' column
data <- data[, !(names(data) == "id")]

# Verify the changes
head(data)
```

```
##   age gender height weight ap_hi ap_lo cholesterol gluc smoke alco active
## 1  50      1    168     62   110    80           1    1     0    0      1
## 2  55      0    156     85   140    90           3    1     0    0      1
## 3  51      0    165     64   130    70           3    1     0    0      0
## 4  48      1    169     82   150   100           1    1     0    0      1
## 5  47      0    156     56   100    60           1    1     0    0      0
## 6  60      0    151     67   120    80           2    2     0    0      0
##   cardio
## 1      0
## 2      1
## 3      1
## 4      1
## 5      0
## 6      0
```

Outliers Detection

```r
library(ggplot2)
library(gridExtra)  # For side-by-side plots

# Create the two boxplots
plot1 <- ggplot(data, aes(y = ap_hi)) +
  geom_boxplot() +
  ggtitle("Boxplot of ap_hi (1)") +
  theme_minimal()
```

```r
plot2 <- ggplot(data, aes(y = ap_hi)) +
  geom_boxplot() +
  ggtitle("Boxplot of ap_hi (2)") +
  theme_minimal()

# Arrange side by side
grid.arrange(plot1, plot2, ncol = 2)
```



clearly we can see that ouliers are present in dataset as such high value of blood-pressure is not possible

** After looking for systolic and diastolic pressure we found that: **

```r
# Blood Pressure Categories Data
bp_data <- data.frame(
  Category = c("Normal", "Elevated", "High Blood Pressure (Hypertension) Stage 1",
               "High Blood Pressure (Hypertension) Stage 2", "Hypertensive Crisis (Consult Doctor Immedi
  Systolic = c("Less than 120", "120 - 129", "130 - 139", "140 or higher", "Higher than 180"),
  Diastolic = c("Less than 80", "Less than 80", "80 - 89", "90 or higher", "Higher than 120")
)

print(bp_data)
```

```
##                                           Category        Systolic
## 1                                           Normal   Less than 120
## 2                                         Elevated       120 - 129
## 3       High Blood Pressure (Hypertension) Stage 1       130 - 139
## 4       High Blood Pressure (Hypertension) Stage 2   140 or higher
## 5 Hypertensive Crisis (Consult Doctor Immediately) Higher than 180
##         Diastolic
## 1    Less than 80
## 2    Less than 80
## 3         80 - 89
## 4    90 or higher
## 5 Higher than 120
```

Treating Outliers

```r
# Drop rows where systolic pressure (ap_hi) > 230 or diastolic pressure (ap_lo) > 150
data <- data[!(data$ap_hi > 230 | data$ap_lo > 150), ]

# Check the shape (rows and columns)
dim(data)
```

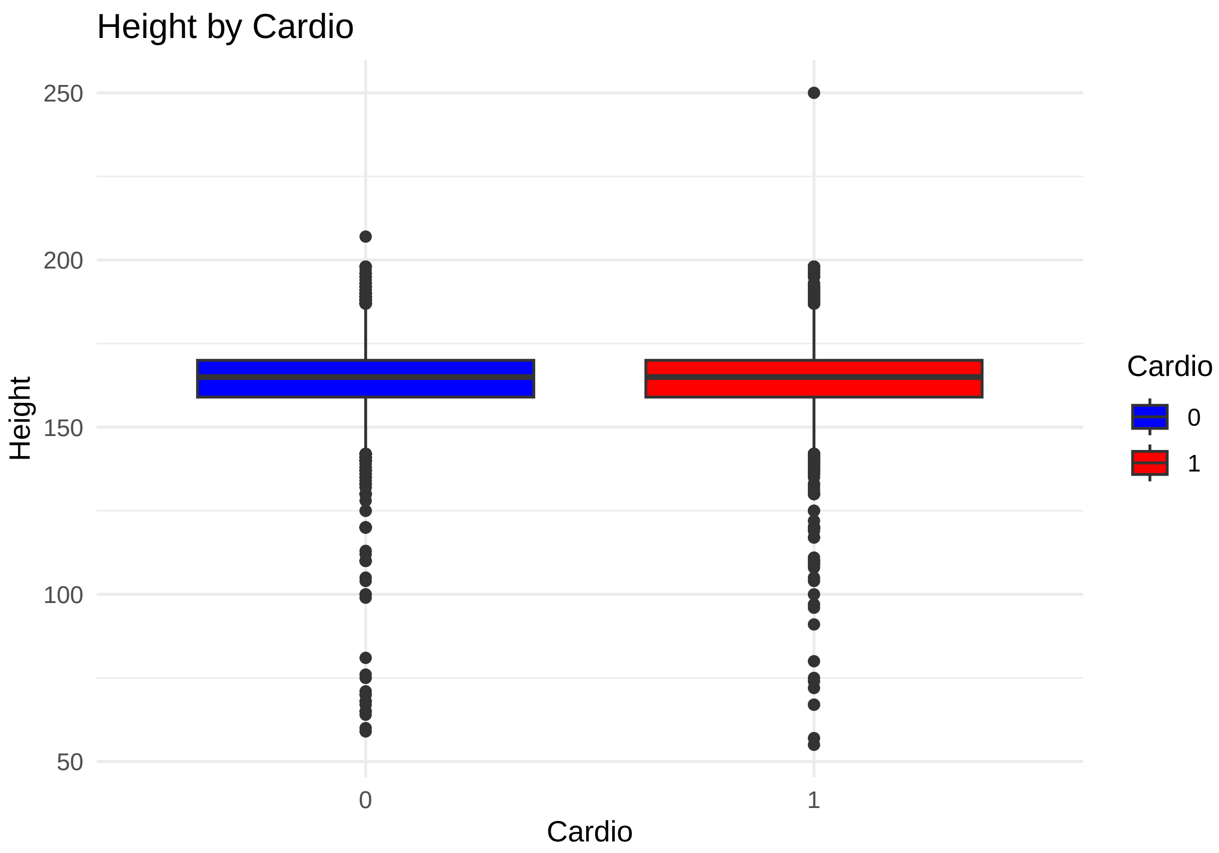
```
## [1] 68978    12
```

Also we found that a blood pressure reading lower than 90 millimeters of mercury (mm Hg) for the top number (systolic) or 60 mm Hg for the bottom number (diastolic) is generally considered low blood pressure.

```r
# Drop rows where systolic pressure (ap_hi) < 70 or diastolic pressure (ap_lo) < 55
data <- data[!(data$ap_hi < 70 | data$ap_lo < 55), ]

# Check the shape (rows and columns)
dim(data)
```

```
## [1] 68666    12
```

```r
# Load necessary libraries
library(ggplot2)

# Create the boxplot
ggplot(data, aes(x = factor(cardio), y = height, fill = factor(cardio))) +
  geom_boxplot() +
  scale_fill_manual(values = c("0" = "blue", "1" = "red")) +
  theme_minimal() +
  labs(x = "Cardio", y = "Height", fill = "Cardio") +
  ggtitle("Height by Cardio")
```



```
# Calculate IQR, upper limit, and lower limit
Q1 <- quantile(data$height, 0.25, na.rm = TRUE)
Q3 <- quantile(data$height, 0.75, na.rm = TRUE)
IQR <- Q3 - Q1
ul <- Q3 + 1.5 * IQR
ll <- Q1 - 1.5 * IQR

# Print the results
cat("IQR:", IQR, "\n")
```

```
## IQR: 11
```

```
cat("Upper limit:", ul, "\n")
```

```
## Upper limit: 186.5
```

```
cat("Lower limit:", ll, "\n")
```

```
## Lower limit: 142.5
```

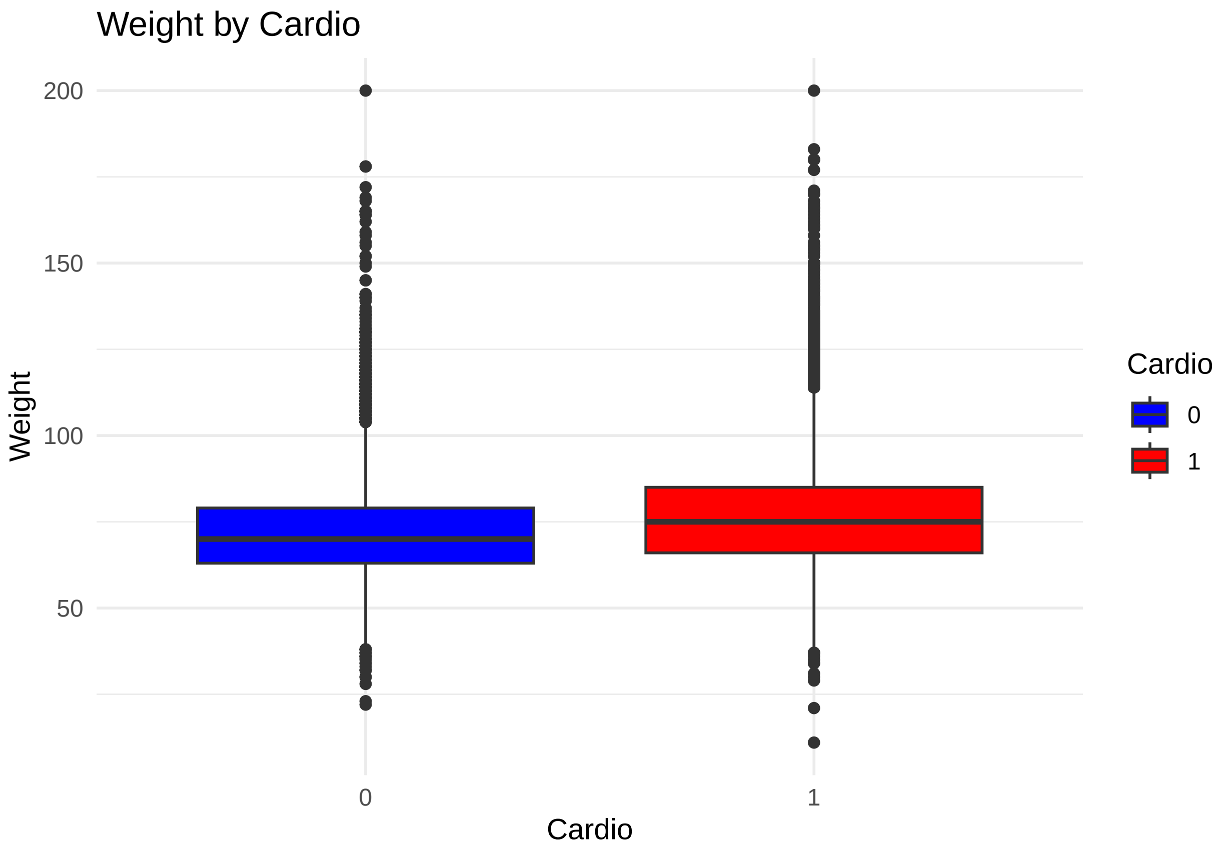
outliers present

```
# Drop rows where height is less than 120 cm
data <- data[data$height >= 120, ]

# Reset row indices (optional, if needed)
rownames(data) <- NULL

# Check the updated data
```

```r
dim(data)
```

```
## [1] 68617    12
```

```r
# Load necessary libraries
library(ggplot2)

# Create the boxplot with blue and red colors
ggplot(data, aes(x = factor(cardio), y = weight, fill = factor(cardio))) +
  geom_boxplot() +
  scale_fill_manual(values = c("0" = "blue", "1" = "red")) +
  theme_minimal() +
  labs(x = "Cardio", y = "Weight", fill = "Cardio") +
  ggtitle("Weight by Cardio")
```



```r
# Calculate IQR, upper limit, and lower limit
Q1 <- quantile(data$weight, 0.25, na.rm = TRUE)
Q3 <- quantile(data$weight, 0.75, na.rm = TRUE)
IQR <- Q3 - Q1
ul <- Q3 + 1.5 * IQR
ll <- Q1 - 1.5 * IQR

# Print the results
cat("IQR:", IQR, "\n")
```

```
## IQR: 17
```

```
cat("Upper limit:", ul, "\n")
```

```
## Upper limit: 107.5
```

```
cat("Lower limit:", ll, "\n")
```

```
## Lower limit: 39.5
```

Outlier detected

```
# Drop rows where weight is greater than 180
data <- data[data$weight <= 180, ]

# Reset row indices (optional, if needed)
rownames(data) <- NULL

# Check the updated data
dim(data)
```

```
## [1] 68614    12
```

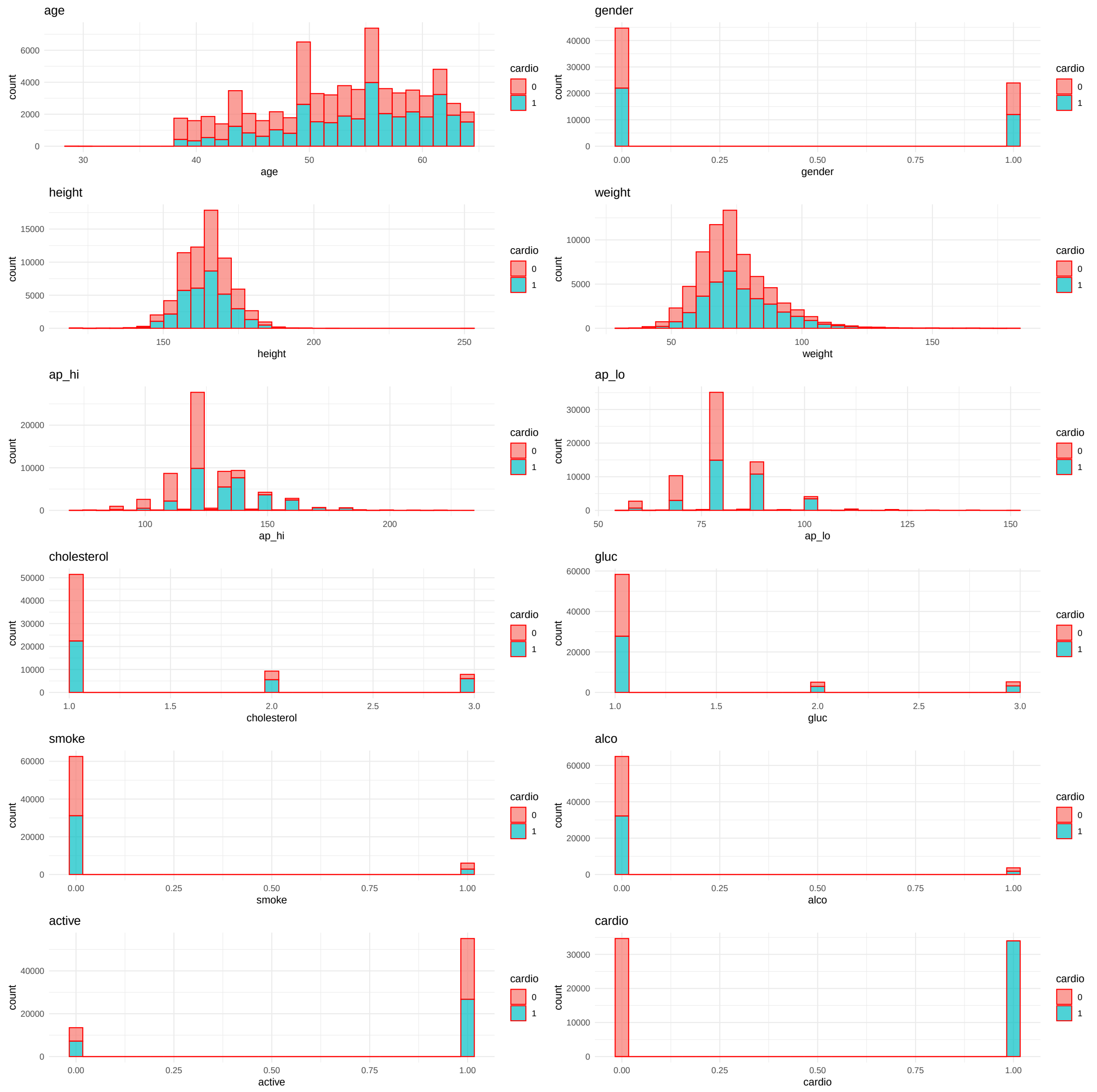
```
# Drop rows where weight is less than 30
data <- data[data$weight >= 30, ]

# Reset row indices (optional, if needed)
rownames(data) <- NULL

# Check the updated data
head(data)
```

```
##   age gender height weight ap_hi ap_lo cholesterol gluc smoke alco active
## 1  50      1    168     62   110    80           1    1     0    0      1
## 2  55      0    156     85   140    90           3    1     0    0      1
## 3  51      0    165     64   130    70           3    1     0    0      0
## 4  48      1    169     82   150   100           1    1     0    0      1
## 5  47      0    156     56   100    60           1    1     0    0      0
## 6  60      0    151     67   120    80           2    2     0    0      0
##   cardio
## 1      0
## 2      1
## 3      1
## 4      1
## 5      0
## 6      0
```

Exploratory Data Analysis (EDA)

Visualize feature distributions:

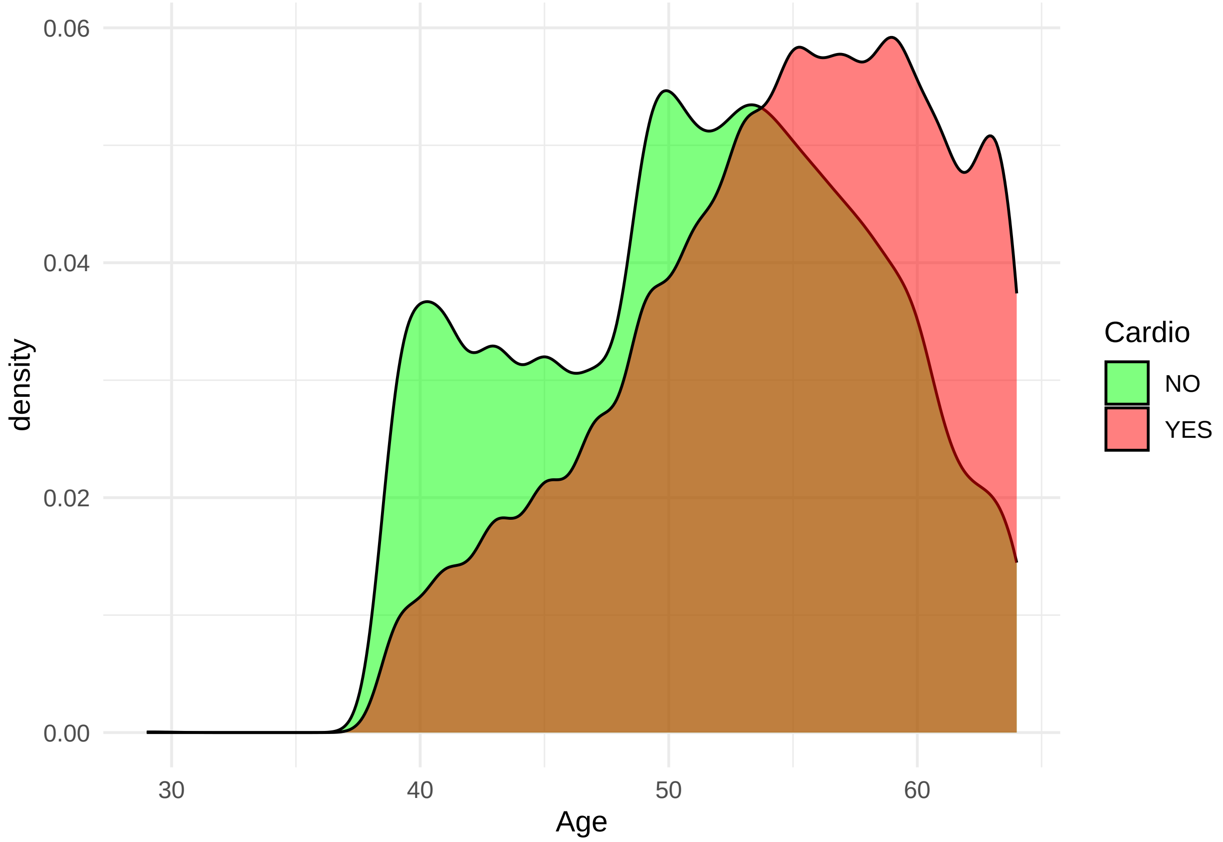
```
# Load ggplot2 library
library(ggplot2)

# Set plot layout to 2 plots per row
gridExtra::grid.arrange(
  grobs = lapply(names(data), function(col) {
    ggplot(data, aes(x = .data[[col]], fill = as.factor(cardio))) +
      geom_histogram(color = "red", bins = 30, alpha = 0.7) +
      labs(title = col, fill = "cardio") +
      theme_minimal()
  }),
  ncol = 2
)
```



Key Insights: Blood pressure, cholesterol, and glucose levels seem to be the most telling indicators of cardiovascular disease in this dataset. Age and weight also show noticeable trends. Lifestyle factors don't show a strong correlation visually, but could still play a role when combined with other features.

UNIVARIATE ANALYSIS

```
library(ggplot2)

ggplot(data, aes(x = age, fill = as.factor(cardio))) +
  geom_density(alpha = 0.5) +
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("NO", "YES")) +
  labs(x = "Age", fill = "Cardio") +
  theme_minimal()
```



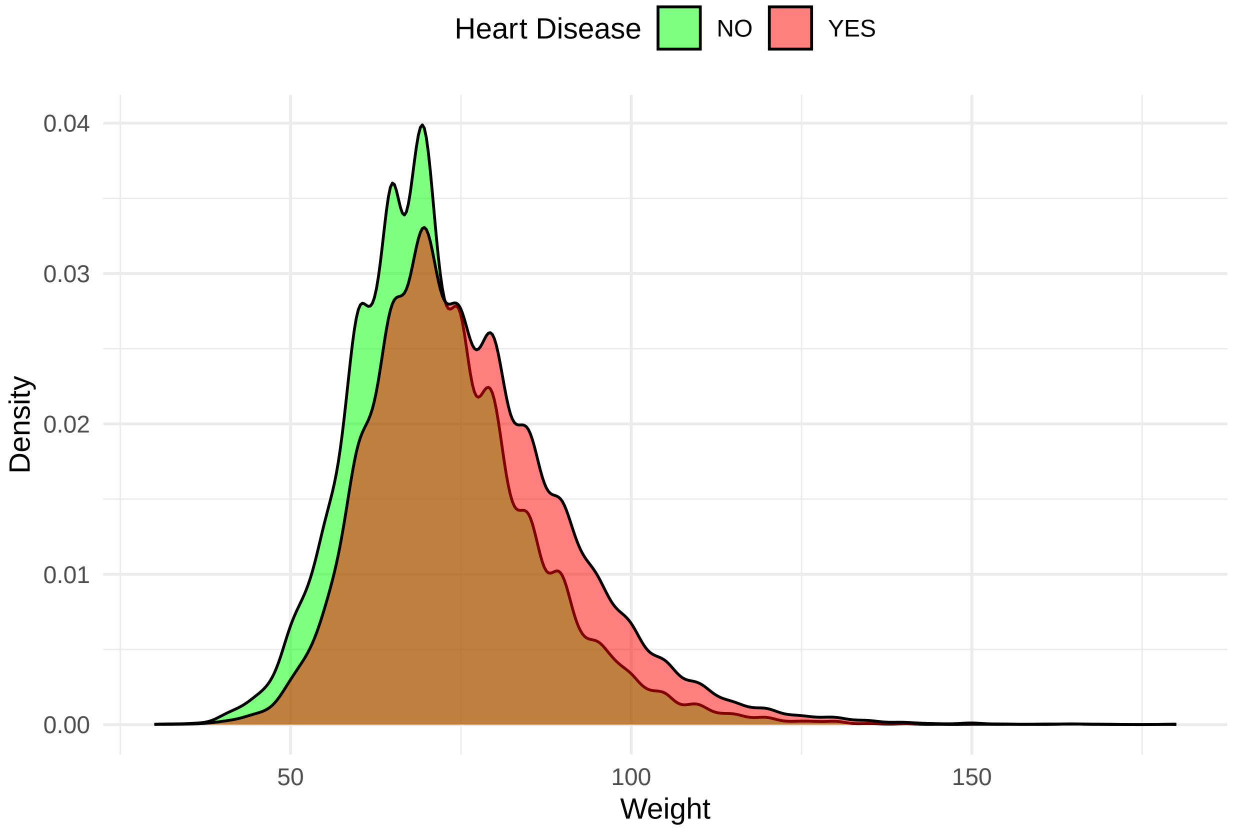
```
library(dplyr)
library(psych)

# Group by 'cardio' and summarize 'age'
data %>%
  group_by(cardio) %>%
  summarise(
    count = n(),
    mean = mean(age, na.rm = TRUE),
    sd = sd(age, na.rm = TRUE),
    min = min(age, na.rm = TRUE),
    Q1 = quantile(age, 0.25, na.rm = TRUE),
    median = median(age, na.rm = TRUE),
    Q3 = quantile(age, 0.75, na.rm = TRUE),
    max = max(age, na.rm = TRUE)
  )
```

```
## # A tibble: 2 x 9
##   cardio count  mean    sd   min    Q1 median    Q3   max
##    <int> <int> <dbl> <dbl> <int> <dbl>  <int> <dbl> <int>
## 1      0 34653  51.2  6.78    29    46     52    57    64
## 2       1 33955   54.5   6.35    39     50     55     60     64
```

Observation:

Person's suffering from heart related issue tend to have higher age than other ones. As we can see here that avg age of person having heart disease is 54.43 and others have **51.21** As per the plot we can see that **pdf curve of person having heart disease is left skewed means person with lower age have less chance of having heart problems.

```r
library(ggplot2)

# Create the plot
ggplot(data, aes(x = weight, fill = factor(cardio))) +
  geom_density(alpha = 0.5) +
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("NO", "YES")) +
  labs(x = "Weight", y = "Density", fill = "Heart Disease") +
  theme_minimal() +
  theme(legend.position = "top")
```



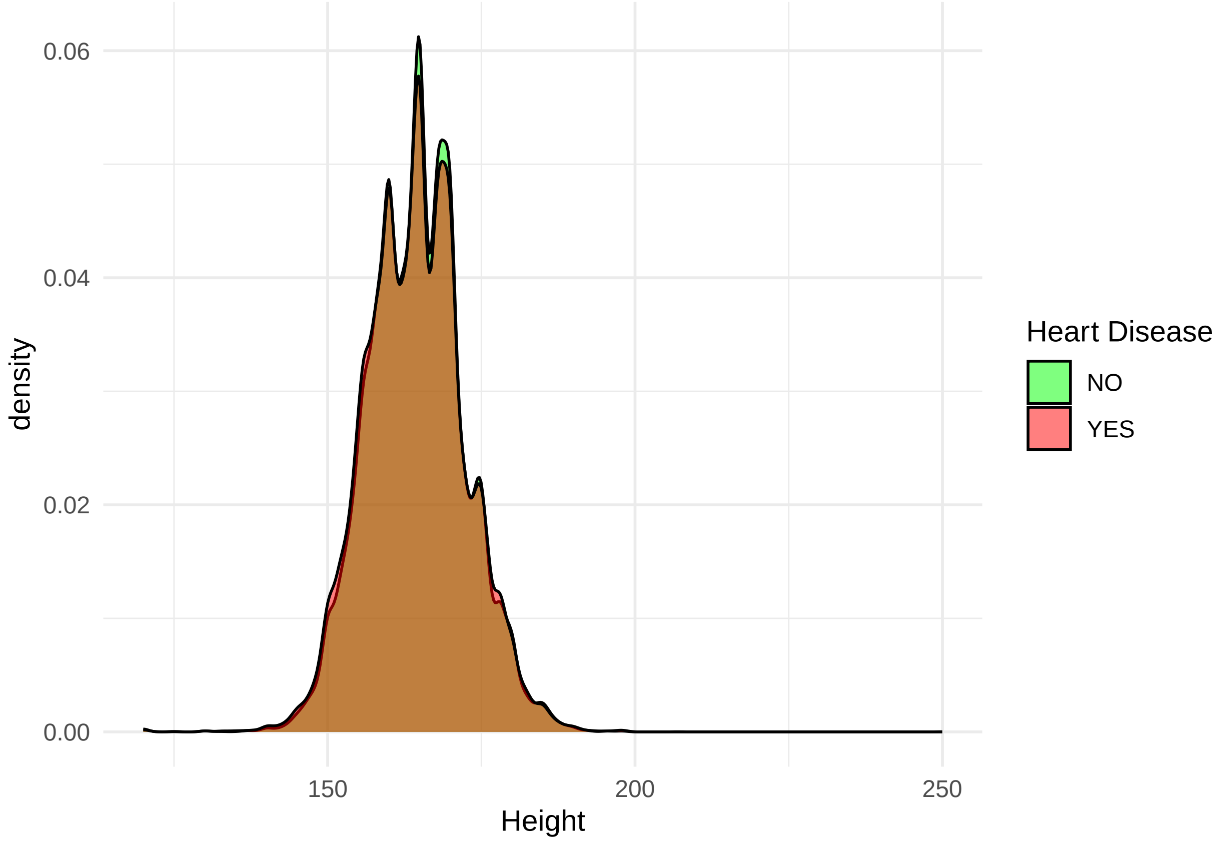
```r
library(dplyr)

# Group by cardio and summarize weight
data %>%
  group_by(cardio) %>%
  summarize(
    count = n(),
    mean = mean(weight, na.rm = TRUE),
    std = sd(weight, na.rm = TRUE),
    min = min(weight, na.rm = TRUE),
    q1 = quantile(weight, 0.25, na.rm = TRUE),
    median = median(weight, na.rm = TRUE),
    q3 = quantile(weight, 0.75, na.rm = TRUE),
    max = max(weight, na.rm = TRUE)
  )
```

```
## # A tibble: 2 x 9
##   cardio count  mean   std   min    q1 median    q3   max
##    <int> <int> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1      0 34653  71.6  13.2    30    63     70    79   178
## 2      1 33955  76.7  14.8    30    66     75    85   180
```

Observation:

person having more weight have higher chance of having heart diseases. As we can see from the graphs and also from the statistical data that Avg. weight of person having heart problems(76.91) is more than those who don't have.

```
library(ggplot2)

ggplot(data, aes(x = height, fill = as.factor(cardio))) +
  geom_density(alpha = 0.5) +
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("NO", "YES")) +
  labs(x = "Height", fill = "Heart Disease") +
  theme_minimal()
```



```r
library(dplyr)

# Group by 'cardio' and summarize 'height'
data %>%
  group_by(cardio) %>%
  summarise(
    count = n(),
    mean = mean(height, na.rm = TRUE),
    std = sd(height, na.rm = TRUE),
    min = min(height, na.rm = TRUE),
    q1 = quantile(height, 0.25, na.rm = TRUE),
    median = median(height, na.rm = TRUE),
    q3 = quantile(height, 0.75, na.rm = TRUE),
    max = max(height, na.rm = TRUE)
  )
```

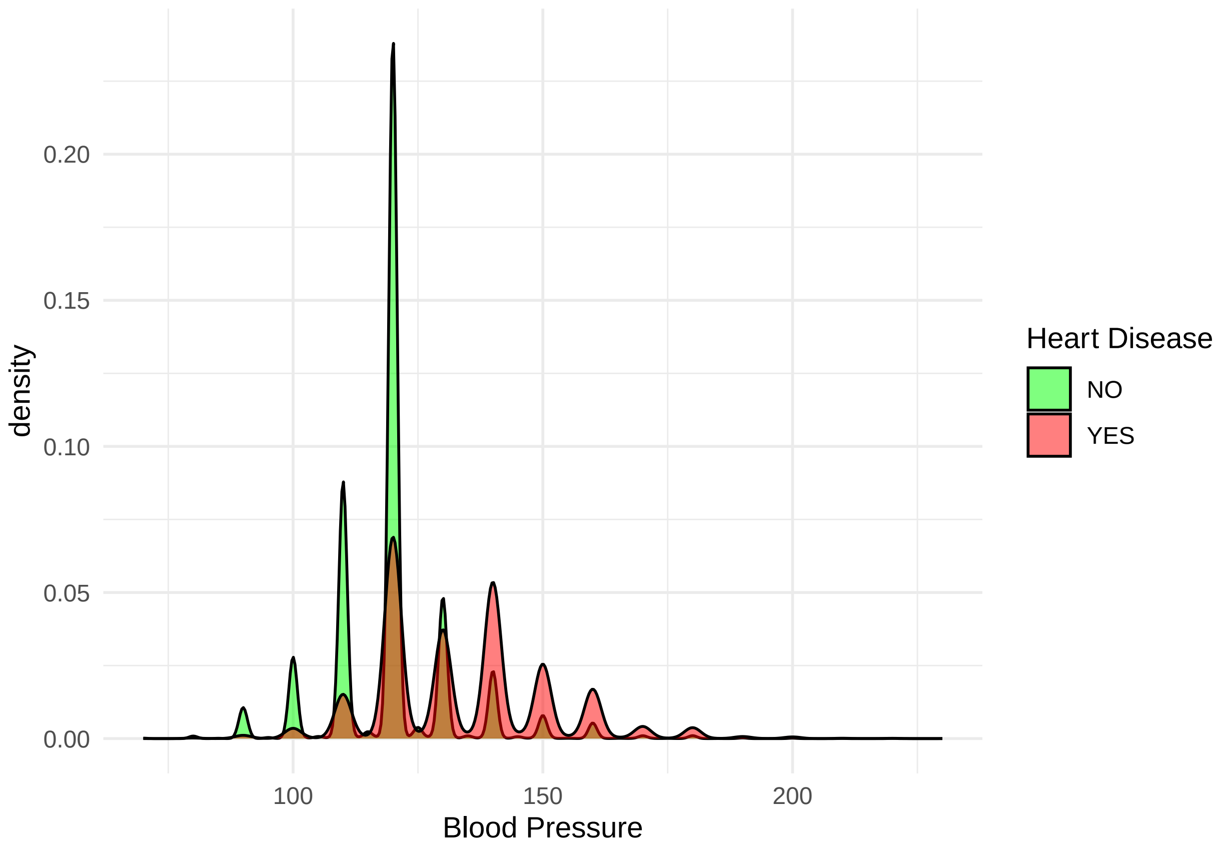
```
## # A tibble: 2 x 9
##   cardio count  mean   std   min    q1 median    q3   max
##    <int> <int> <dbl> <dbl> <int> <dbl>  <int> <dbl> <int>
## 1      0 34653  165.  7.82   120   159    165   170   207
## 2      1 33955  164.  8.01   120   159    165   170   250
```

Observation:

Height is doesn't play any role in determining heart diseases. As we can see both the plot overlaps and avg height of person with heart problem and without heart problems is also same.

Blood pressure analysis

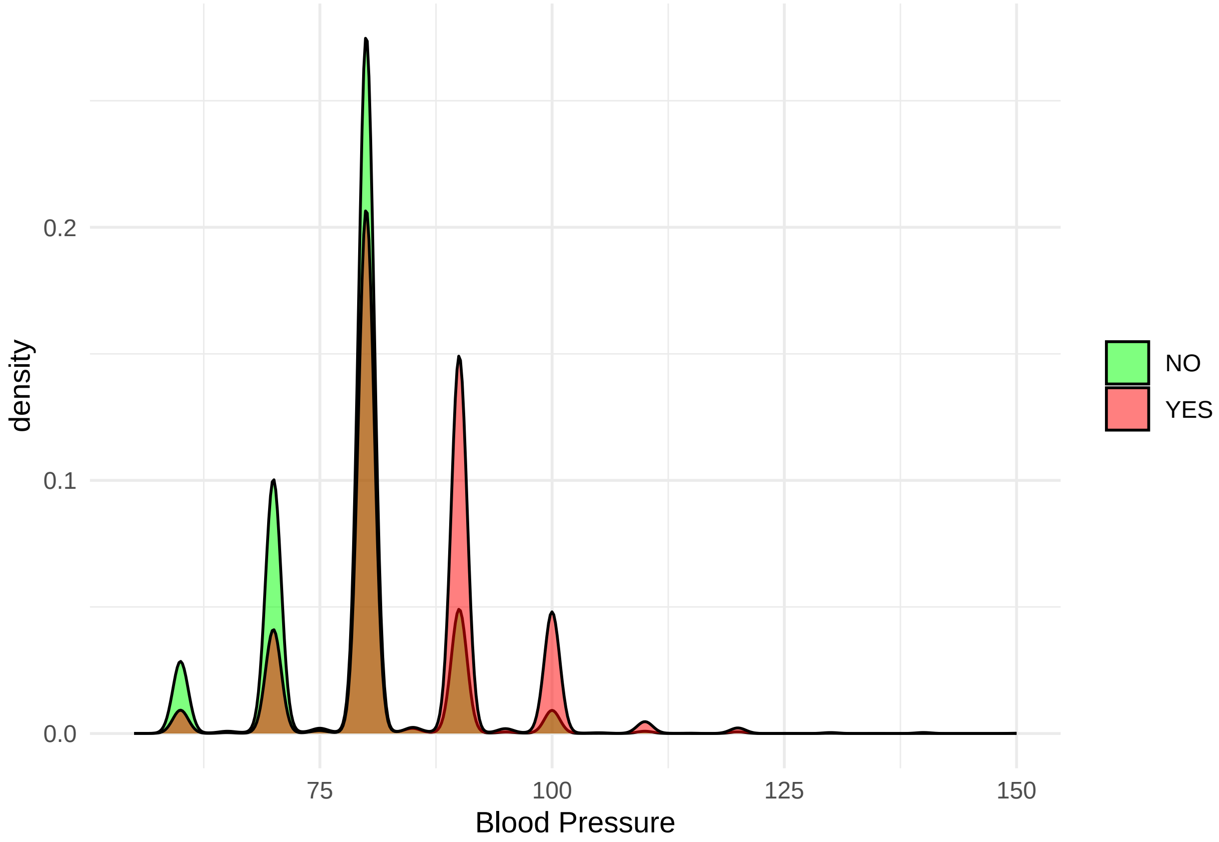
```r
library(ggplot2)

ggplot(data, aes(x = ap_hi, fill = as.factor(cardio))) +
  geom_density(alpha = 0.5) +
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("NO", "YES")) +
  labs(x = "Blood Pressure", fill = "Heart Disease") +
  theme_minimal()
```



```r
library(dplyr)

data %>%
  group_by(cardio) %>%
  summarise(
    count = n(),
    mean = mean(ap_hi, na.rm = TRUE),
    std = sd(ap_hi, na.rm = TRUE),
    min = min(ap_hi, na.rm = TRUE),
    q1 = quantile(ap_hi, 0.25, na.rm = TRUE),
    median = median(ap_hi, na.rm = TRUE),
    q3 = quantile(ap_hi, 0.75, na.rm = TRUE),
    max = max(ap_hi, na.rm = TRUE)
  )
```

```
## # A tibble: 2 x 9
##   cardio count  mean   std   min    q1 median    q3   max
##    <int> <int> <dbl> <dbl> <int> <dbl>  <int> <dbl> <int>
## 1      0 34653  120.  12.6    70   110    120   120   220
## 2      1 33955  134.  17.3    70   120    130   140   230
```

```r
library(ggplot2)

ggplot(data, aes(x = ap_lo, fill = factor(cardio))) +
  geom_density(alpha = 0.5) +
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("NO", "YES")) +
  labs(x = "Blood Pressure", fill = "Heart Disease") +
  theme_minimal() +
  theme(legend.title = element_blank())
```



```
data %>%
  group_by(cardio) %>%
  summarise(
    count = n(),
    mean = mean(ap_lo, na.rm = TRUE),
    std = sd(ap_lo, na.rm = TRUE),
    min = min(ap_lo, na.rm = TRUE),
    q1 = quantile(ap_lo, 0.25, na.rm = TRUE),
    median = median(ap_lo, na.rm = TRUE),
    q3 = quantile(ap_lo, 0.75, na.rm = TRUE),
    max = max(ap_lo, na.rm = TRUE)
  )
```

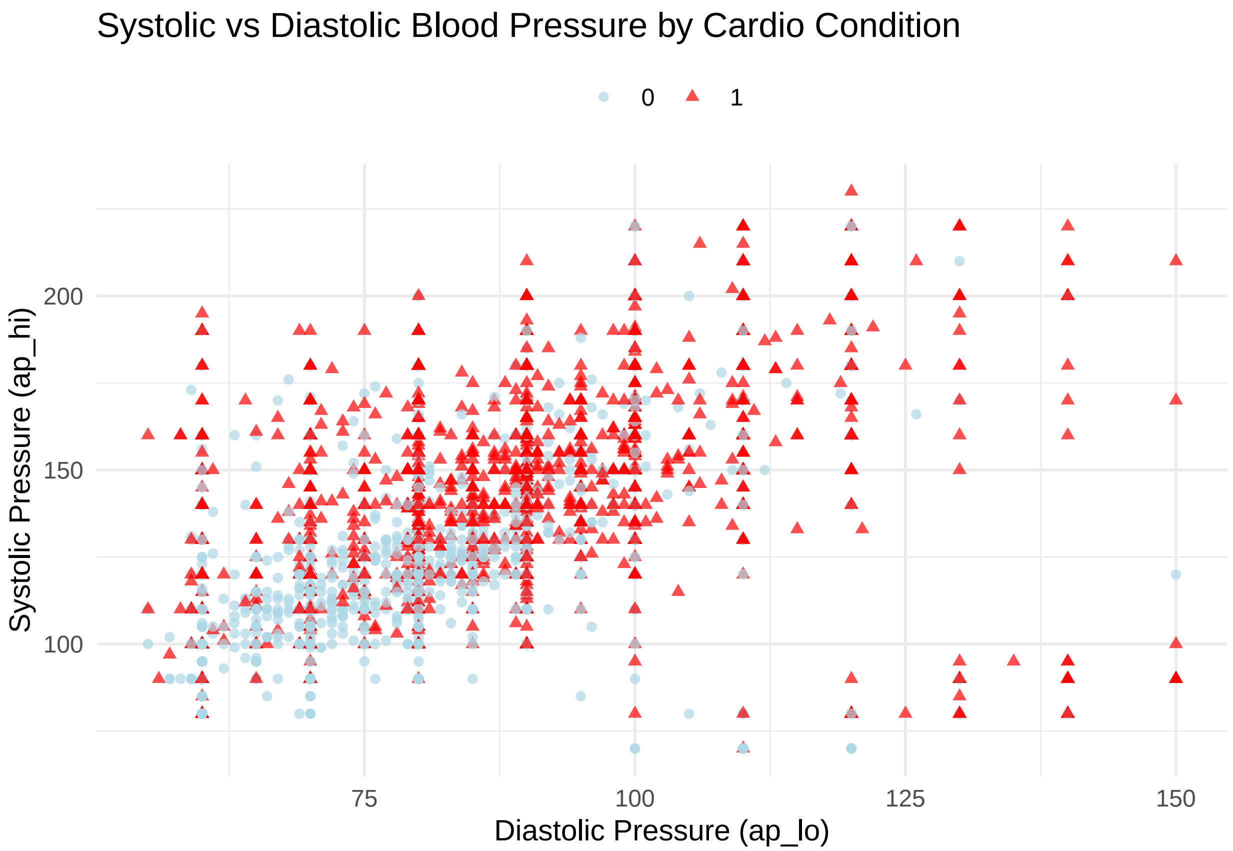
```
## # A tibble: 2 x 9
##   cardio count  mean   std   min    q1 median    q3   max
##    <int> <int> <dbl> <dbl> <int> <dbl>  <int> <dbl> <int>
## 1      0 34653  78.2  8.17    55    70     80    80   150
## 2      1 33955  84.6  9.63    55    80     80    90   150
```

Observation:

As we can see that people who have high blood pressure have high chance that they have heart disease. Avg. Blood pressure of person having heart problems is 134.07 is more than the normal level.

BIVARIATE ANALYSIS

```r
library(ggplot2)

ggplot(data, aes(x = ap_lo, y = ap_hi, color = factor(cardio), shape = factor(cardio))) +
  geom_point(alpha = 0.7) +
  scale_color_manual(values = c("0" = "lightblue", "1" = "red")) +
  labs(x = "Diastolic Pressure (ap_lo)", y = "Systolic Pressure (ap_hi)", color = "Cardio", shape = "Ca
  theme_minimal() +
  theme(legend.title = element_blank()) +
  theme(legend.position = "top") +
  ggtitle("Systolic vs Diastolic Blood Pressure by Cardio Condition")
```



Observation: People with heart problems have higher systolic pressure and diastolic pressure values as per the plot.

Smoke and alcohol analysis

```r
library(dplyr)

data %>%
  group_by(smoke) %>%
  summarise(
    ap_hi_count = n(),
    ap_hi_mean = mean(ap_hi, na.rm = TRUE),
    ap_hi_sd = sd(ap_hi, na.rm = TRUE),
    ap_hi_min = min(ap_hi, na.rm = TRUE),
    ap_hi_q1 = quantile(ap_hi, 0.25, na.rm = TRUE),
    ap_hi_median = median(ap_hi, na.rm = TRUE),
    ap_hi_q3 = quantile(ap_hi, 0.75, na.rm = TRUE),
    ap_hi_max = max(ap_hi, na.rm = TRUE),
    ap_lo_mean = mean(ap_lo, na.rm = TRUE),
    ap_lo_sd = sd(ap_lo, na.rm = TRUE),
    ap_lo_min = min(ap_lo, na.rm = TRUE),
    ap_lo_q1 = quantile(ap_lo, 0.25, na.rm = TRUE),
    ap_lo_median = median(ap_lo, na.rm = TRUE),
    ap_lo_q3 = quantile(ap_lo, 0.75, na.rm = TRUE),
    ap_lo_max = max(ap_lo, na.rm = TRUE)
  )
```

```
## # A tibble: 2 x 16
##   smoke ap_hi_count ap_hi_mean ap_hi_sd ap_hi_min ap_hi_q1 ap_hi_median ap_hi_q3
##   <int>       <int>      <dbl>    <dbl>     <int>    <dbl>        <dbl>    <dbl>
## 1     0       62570       126.     16.6        70      120          120      140
## 2     1        6038       128.     17.4        70      120          120      140
## # i 8 more variables: ap_hi_max <int>, ap_lo_mean <dbl>, ap_lo_sd <dbl>,
## #   ap_lo_min <int>, ap_lo_q1 <dbl>, ap_lo_median <dbl>, ap_lo_q3 <dbl>,
## #   ap_lo_max <int>
```

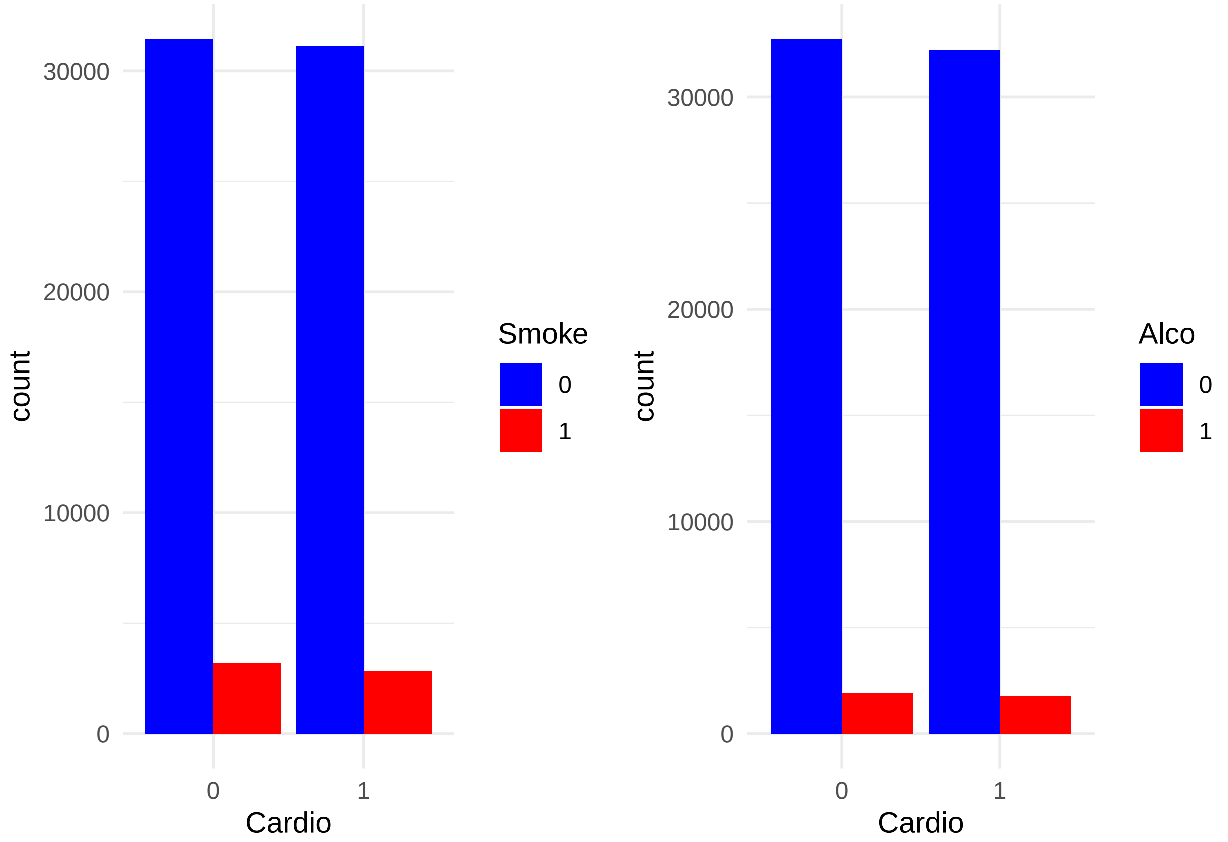
```
library(dplyr)

data %>%
  group_by(alco) %>%
  summarise(
    ap_hi_count = n(),
    ap_hi_mean = mean(ap_hi, na.rm = TRUE),
    ap_hi_sd = sd(ap_hi, na.rm = TRUE),
    ap_hi_min = min(ap_hi, na.rm = TRUE),
    ap_hi_q1 = quantile(ap_hi, 0.25, na.rm = TRUE),
    ap_hi_median = median(ap_hi, na.rm = TRUE),
    ap_hi_q3 = quantile(ap_hi, 0.75, na.rm = TRUE),
    ap_hi_max = max(ap_hi, na.rm = TRUE),
    ap_lo_mean = mean(ap_lo, na.rm = TRUE),
    ap_lo_sd = sd(ap_lo, na.rm = TRUE),
    ap_lo_min = min(ap_lo, na.rm = TRUE),
    ap_lo_q1 = quantile(ap_lo, 0.25, na.rm = TRUE),
    ap_lo_median = median(ap_lo, na.rm = TRUE),
    ap_lo_q3 = quantile(ap_lo, 0.75, na.rm = TRUE),
    ap_lo_max = max(ap_lo, na.rm = TRUE)
  )
```

```
## # A tibble: 2 x 16
##    alco ap_hi_count ap_hi_mean ap_hi_sd ap_hi_min ap_hi_q1 ap_hi_median ap_hi_q3
##   <int>       <int>      <dbl>    <dbl>     <int>    <dbl>        <int>    <dbl>
## 1     0       64935       126.     16.6        70      120          120      140
## 2     1        3673       129.     18.1        70      120          120      140
## # i 8 more variables: ap_hi_max <int>, ap_lo_mean <dbl>, ap_lo_sd <dbl>,
## #   ap_lo_min <int>, ap_lo_q1 <dbl>, ap_lo_median <int>, ap_lo_q3 <dbl>,
## #   ap_lo_max <int>
```

Observation: Although there is not much difference but still we can say that person who smoke or take alcohol have higher blood pressure than other person. And hence chances that they can have heart problems also increases.

Cardio Vs Smoke and Cardio Vs Alco

```r
library(ggplot2)
library(gridExtra)

# Plot 1: Cardio vs Smoke
plot1 <- ggplot(data, aes(x = as.factor(cardio), fill = as.factor(smoke))) +
  geom_bar(position = "dodge") +
  scale_fill_manual(values = c("blue", "red")) +
  labs(x = "Cardio", fill = "Smoke") +
  theme_minimal()

# Plot 2: Cardio vs Alco
plot2 <- ggplot(data, aes(x = as.factor(cardio), fill = as.factor(alco))) +
  geom_bar(position = "dodge") +
  scale_fill_manual(values = c("blue", "red")) +
  labs(x = "Cardio", fill = "Alco") +
  theme_minimal()

# Arrange side by side
grid.arrange(plot1, plot2, ncol = 2)
```



Observation: plot shows that those who smoke and have high blood pressure are mostly having heart problems

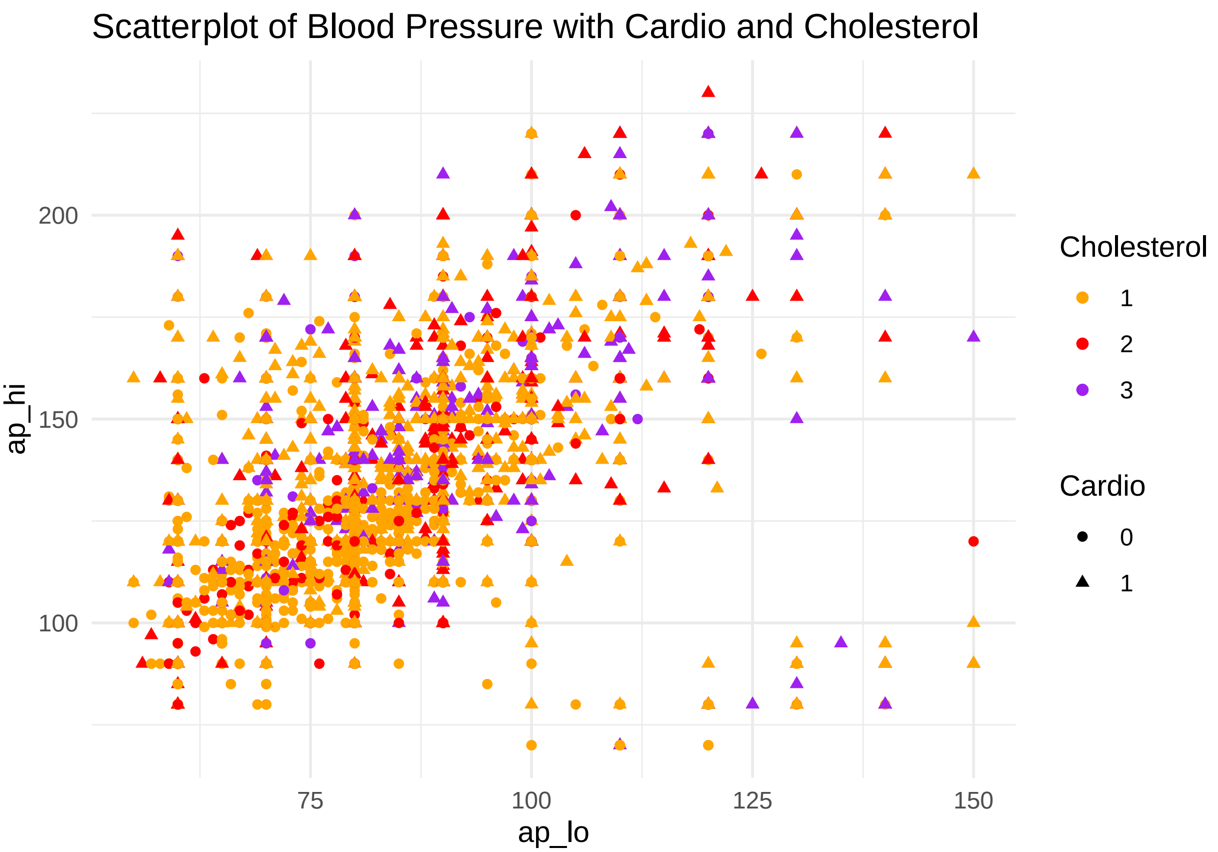
Cholesterol analysis

```r
library(dplyr)
```

```r
# Group by cholesterol and summarize ap_hi and ap_lo
data %>%
  group_by(cholesterol) %>%
  summarise(
    ap_hi_mean = mean(ap_hi, na.rm = TRUE),
    ap_hi_sd = sd(ap_hi, na.rm = TRUE),
    ap_hi_min = min(ap_hi, na.rm = TRUE),
    ap_hi_max = max(ap_hi, na.rm = TRUE),
    ap_lo_mean = mean(ap_lo, na.rm = TRUE),
    ap_lo_sd = sd(ap_lo, na.rm = TRUE),
    ap_lo_min = min(ap_lo, na.rm = TRUE),
    ap_lo_max = max(ap_lo, na.rm = TRUE)
  )
```

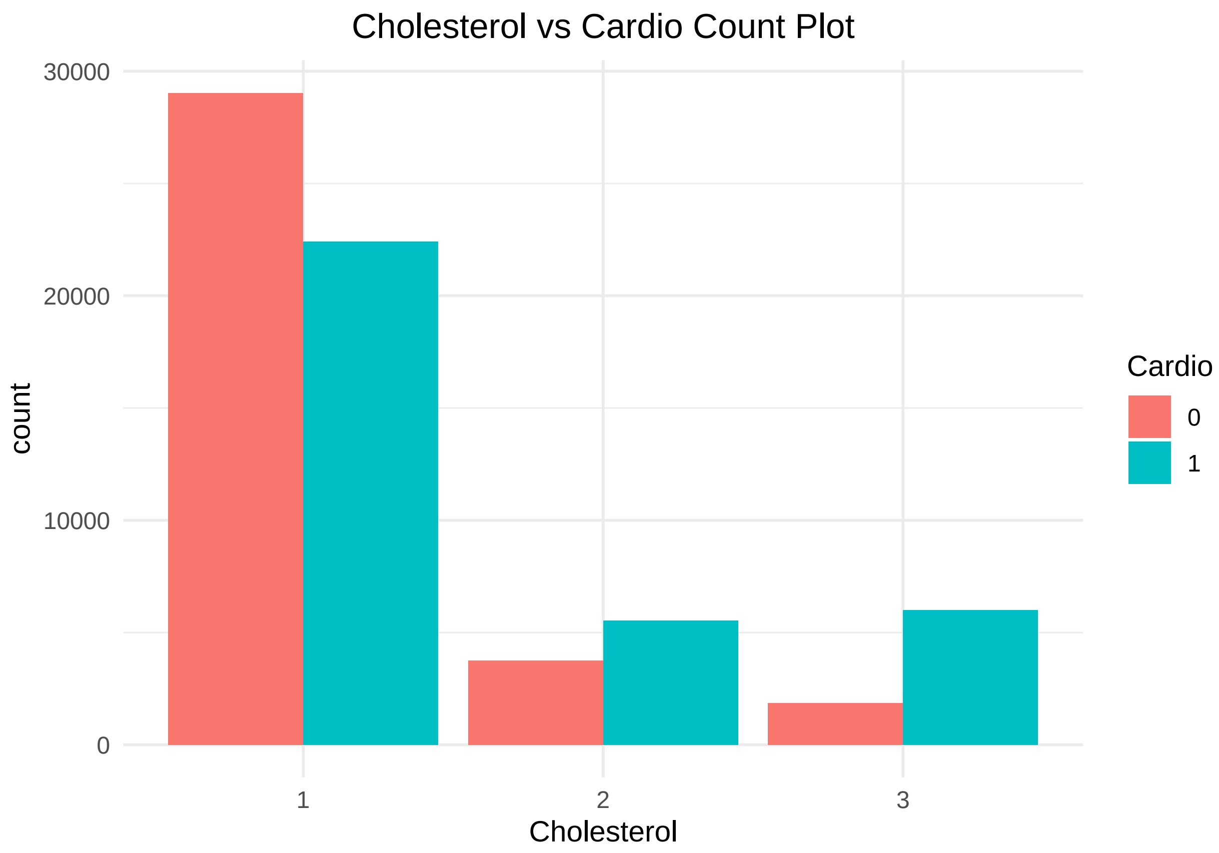
```
## # A tibble: 3 x 9
##   cholesterol ap_hi_mean ap_hi_sd ap_hi_min ap_hi_max ap_lo_mean ap_lo_sd
##         <int>      <dbl>    <dbl>     <int>     <int>      <dbl>    <dbl>
## 1           1       125.     15.7        70       220       80.5     9.10
## 2           2       131.     19.2        80       230       83.2    10.5 
## 3           3       134.     16.8        70       220       84.9     9.54
## # i 2 more variables: ap_lo_min <int>, ap_lo_max <int>
```

```r
library(ggplot2)

# Scatter plot with ap_lo on x-axis, ap_hi on y-axis, cardio as shape, and cholesterol as color
ggplot(data, aes(x = ap_lo, y = ap_hi, shape = as.factor(cardio), color = as.factor(cholesterol))) +
  geom_point() +
  scale_color_manual(values = c("orange", "red", "purple")) +
  labs(x = "ap_lo", y = "ap_hi", shape = "Cardio", color = "Cholesterol") +
  theme_minimal() +
  theme(legend.position = "right") +
  ggtitle("Scatterplot of Blood Pressure with Cardio and Cholesterol")
```



Observation: person with high cholesterol level have higher blood pressure and have more chance to have heart problems

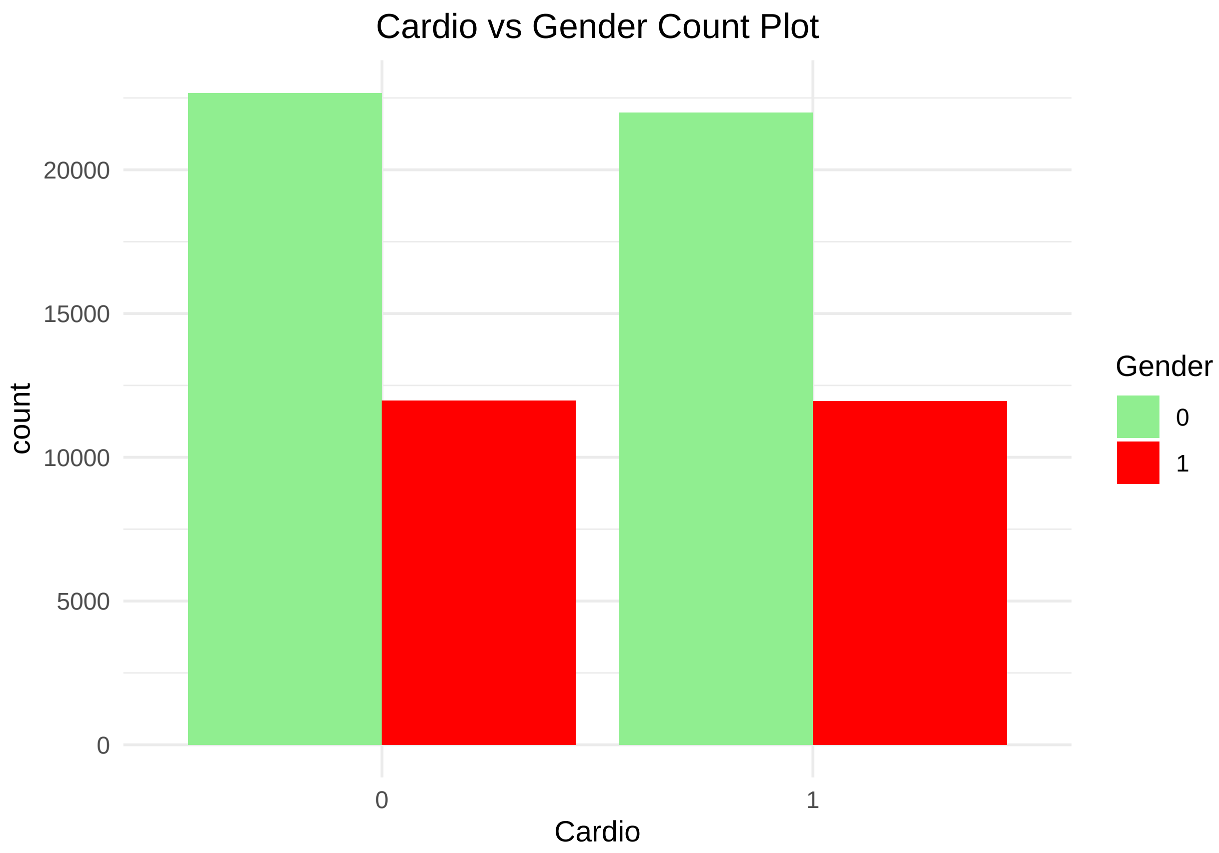
```
library(ggplot2)

# Count plot for cholesterol with cardio as hue
ggplot(data, aes(x = as.factor(cholesterol), fill = as.factor(cardio))) +
  geom_bar(position = "dodge") +
  labs(x = "Cholesterol", fill = "Cardio") +
  theme_minimal() +
  ggtitle("Cholesterol vs Cardio Count Plot") +
  theme(plot.title = element_text(hjust = 0.5))
```



Observation: Most of the Person with **cholesterol level 2 and 3** have heart disease.

Gender analysis

```
library(ggplot2)

# Count plot for cardio with gender as hue
ggplot(data, aes(x = as.factor(cardio), fill = as.factor(gender))) +
  geom_bar(position = "dodge") +
  scale_fill_manual(values = c("lightgreen", "red")) +
  labs(x = "Cardio", fill = "Gender") +
  theme_minimal() +
  ggtitle("Cardio vs Gender Count Plot") +
  theme(plot.title = element_text(hjust = 0.5))
```



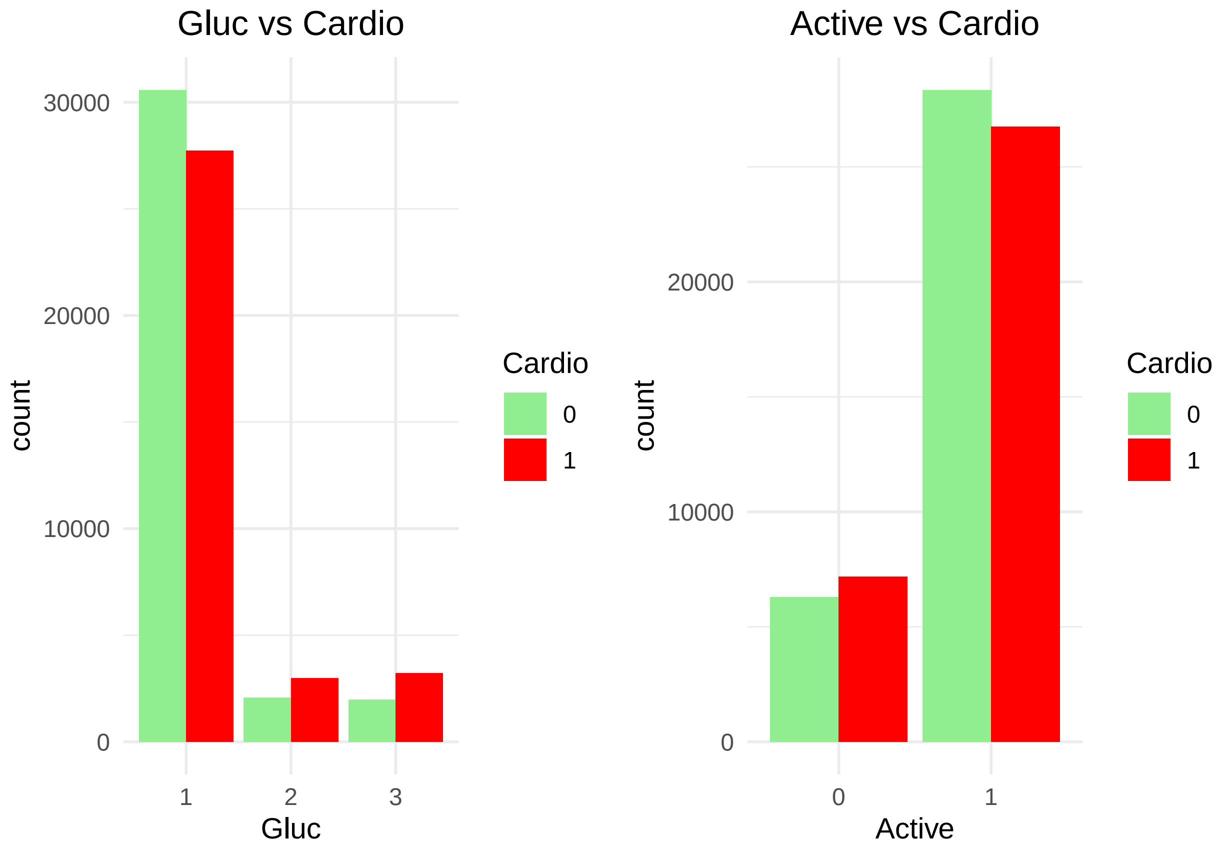
Observation:

Over 20000 females and 10000 males have heart diseases.

Glucose and Physical Activity

```r
library(ggplot2)
library(gridExtra)

# Gluc vs Cardio Count Plot
p1 <- ggplot(data, aes(x = as.factor(gluc), fill = as.factor(cardio))) +
  geom_bar(position = "dodge") +
  scale_fill_manual(values = c("lightgreen", "red")) +
  labs(x = "Gluc", fill = "Cardio") +
  theme_minimal() +
  ggtitle("Gluc vs Cardio") +
  theme(plot.title = element_text(hjust = 0.5))

# Active vs Cardio Count Plot
p2 <- ggplot(data, aes(x = as.factor(active), fill = as.factor(cardio))) +
  geom_bar(position = "dodge") +
  scale_fill_manual(values = c("lightgreen", "red")) +
  labs(x = "Active", fill = "Cardio") +
  theme_minimal() +
  ggtitle("Active vs Cardio") +
  theme(plot.title = element_text(hjust = 0.5))

# Arrange plots side by side
```

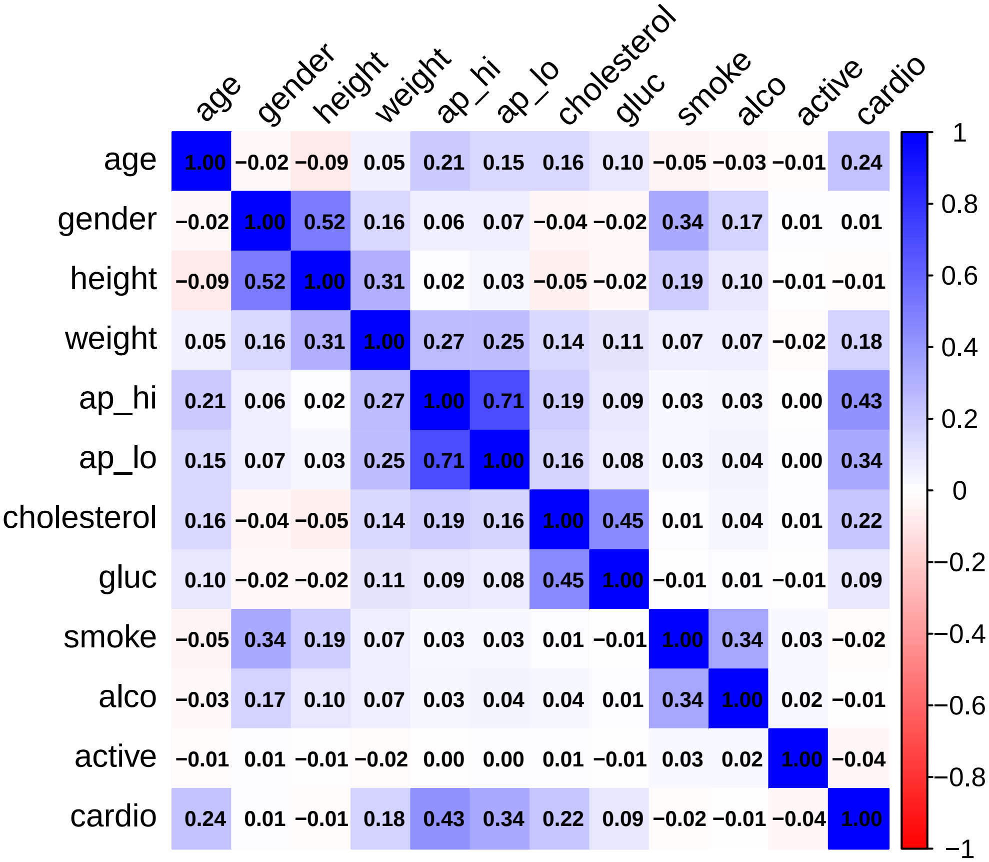
```r
grid.arrange(p1, p2, ncol = 2)
```



Observation: Both the feature doesn't seem to have much correlation with cardiac problems. Hence these features are not that important as others.

Correlation Matrix

```r
library(corrplot)

# Calculate the correlation matrix
corr_matrix <- cor(data)

# Plot the heatmap
corrplot(corr_matrix, method = "color", addCoef.col = "black", tl.col = "black",
         tl.srt = 45, col = colorRampPalette(c("red", "white", "blue"))(200),
         number.cex = 0.7)
```



Modelling

Convert target variable to factor (if not already)

```
data$cardio <- as.factor(data$cardio)

# Split dataset into training (80%) and testing (20%)
set.seed(123)
trainIndex <- createDataPartition(data$cardio, p = 0.8, list = FALSE)
trainData <- data[trainIndex, ]
testData <- data[-trainIndex, ]

dim(trainData)
```

```
## [1] 54887    12
```

```
dim(testData)
```

```
## [1] 13721    12
```

Apply J48 (C4.5 Implementation in RWeka)

```
# Train J48 model
library(RWeka)
library(caret)

j48_model <- J48(cardio ~ ., data = trainData)
```

```r
# Predict on test set
j48_pred <- predict(j48_model, testData)

# Evaluate Model Performance
conf_matrix <- confusionMatrix(j48_pred, testData$cardio)

# Print confusion matrix
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    0    1
##          0 5323 2125
##          1 1607 4666
## 
##                Accuracy : 0.728           
##                  95% CI : (0.7205, 0.7354)
##     No Information Rate : 0.5051          
##     P-Value [Acc > NIR] : < 2.2e-16       
## 
##                   Kappa : 0.4555          
## 
##  Mcnemar's Test P-Value : < 2.2e-16       
## 
##             Sensitivity : 0.7681          
##             Specificity : 0.6871          
##          Pos Pred Value : 0.7147          
##          Neg Pred Value : 0.7438          
##              Prevalence : 0.5051          
##          Detection Rate : 0.3879          
##    Detection Prevalence : 0.5428          
##       Balanced Accuracy : 0.7276          
## 
##        'Positive' Class : 0               
## 
```

```r
# Extract metrics
precision <- conf_matrix$byClass["Pos Pred Value"]  # Precision
recall <- conf_matrix$byClass["Sensitivity"]        # Recall
f1_score <- 2 * (precision * recall) / (precision + recall)  # F1-Score

# Display metrics
cat("Precision:", precision, "\n")
```

```
## Precision: 0.7146885
```

```r
cat("Recall:", recall, "\n")
```

```
## Recall: 0.7681097
```

```r
cat("F1-Score:", f1_score, "\n")
```

```
## F1-Score: 0.7404368
```

Apply C5.0 Decision Tree

```r
# Load necessary libraries
library(C50)
library(caret)

# Train C5.0 model
c50_model <- C5.0(cardio ~ ., data = trainData)

# Predict on test set
c50_pred <- predict(c50_model, testData)

# Evaluate Model Performance
conf_matrix <- confusionMatrix(c50_pred, testData$cardio)

# Print confusion matrix
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    0    1
##          0 5385 2123
##          1 1545 4668
## 
##                Accuracy : 0.7327
##                  95% CI : (0.7252, 0.7401)
##     No Information Rate : 0.5051
##     P-Value [Acc > NIR] : < 2.2e-16
## 
##                   Kappa : 0.4648
## 
##  Mcnemar's Test P-Value : < 2.2e-16
## 
##             Sensitivity : 0.7771
##             Specificity : 0.6874
##          Pos Pred Value : 0.7172
##          Neg Pred Value : 0.7513
##              Prevalence : 0.5051
##          Detection Rate : 0.3925
##    Detection Prevalence : 0.5472
##       Balanced Accuracy : 0.7322
## 
##        'Positive' Class : 0
## 
```

```r
# Extract metrics
precision <- conf_matrix$byClass["Pos Pred Value"]  # Precision
recall <- conf_matrix$byClass["Sensitivity"]        # Recall
f1_score <- 2 * (precision * recall) / (precision + recall)  # F1-Score

# Display metrics
cat("Precision:", precision, "\n")
```

```
## Precision: 0.7172349
```

```r
cat("Recall:", recall, "\n")
```

```
## Recall: 0.7770563
```

```r
cat("F1-Score:", f1_score, "\n")
```

```
## F1-Score: 0.7459482
```

Apply C5.0 Rules-Based Model

```r
# Load necessary libraries
library(C50)
library(caret)

# Train C5.0 model with rule-based classifier
c50_rules_model <- C5.0(cardio ~ ., data = trainData, rules = TRUE)

# Predict on test set
c50_rules_pred <- predict(c50_rules_model, testData)

# Evaluate Model Performance
conf_matrix <- confusionMatrix(c50_rules_pred, testData$cardio)

# Print confusion matrix
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 5413 2143
##          1 1517 4648
##
##                Accuracy : 0.7333
##                  95% CI : (0.7258, 0.7406)
##     No Information Rate : 0.5051
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.466
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.7811
##             Specificity : 0.6844
##          Pos Pred Value : 0.7164
##          Neg Pred Value : 0.7539
##              Prevalence : 0.5051
##          Detection Rate : 0.3945
##    Detection Prevalence : 0.5507
##       Balanced Accuracy : 0.7328
##
##        'Positive' Class : 0
##
```

```r
# Extract metrics
precision <- conf_matrix$byClass["Pos Pred Value"]  # Precision
```

```
recall <- conf_matrix$byClass["Sensitivity"]        # Recall
f1_score <- 2 * (precision * recall) / (precision + recall)  # F1-Score

# Display metrics
cat("Precision:", precision, "\n")
```

```
## Precision: 0.7163843
```

```
cat("Recall:", recall, "\n")
```

```
## Recall: 0.7810967
```

```
cat("F1-Score:", f1_score, "\n")
```

```
## F1-Score: 0.7473423
```

Apply Cross-Validation

```
# Set seed for reproducibility
set.seed(123)

# Define cross-validation method (10-fold CV)
train_control <- trainControl(method = "cv", number = 10)

# Train the model using cross-validation
model <- train(
  cardio ~ .,                # cardio is the target variable
  data = data,               # your dataset
  method = "rpart",          # Decision Tree (you can use other methods)
  trControl = train_control
)

# Print model summary
print(model)
```

```
## CART
##
## 68608 samples
##    11 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 61747, 61747, 61748, 61747, 61747, 61747, ...
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.00553674  0.7227000  0.4448064
##   0.00970402  0.7167970  0.4326047
##   0.41940804  0.6273898  0.2500200
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.00553674.
```

```
# Check model performance
model$results
```

```
##            cp  Accuracy     Kappa AccuracySD    KappaSD
## 1 0.00553674 0.7227000 0.4448064 0.00544238 0.01101801
## 2 0.00970402 0.7167970 0.4326047 0.00966988 0.01952144
## 3 0.41940804 0.6273898 0.2500200 0.10529455 0.21524223
```

Accuracy comparison

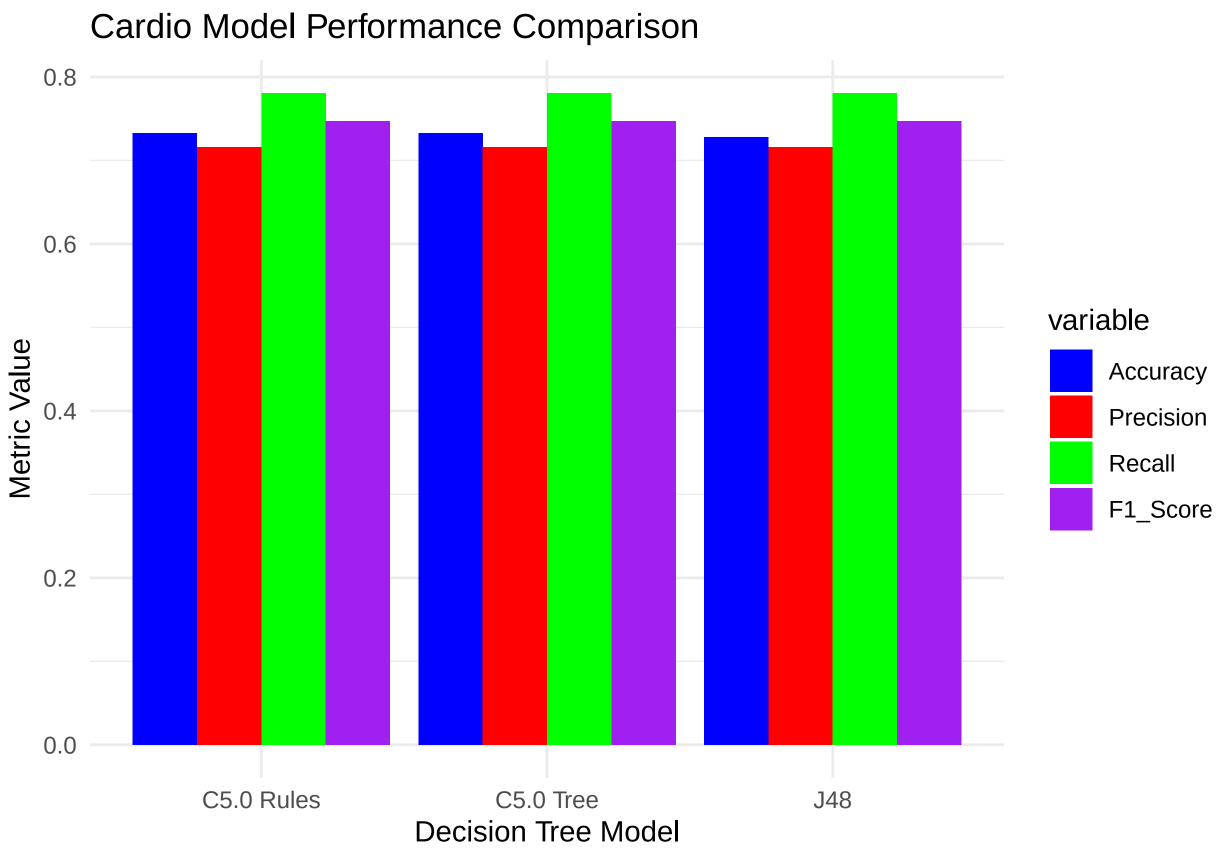
```
j48_acc <- mean(j48_pred == testData$cardio)
c50_acc <- mean(c50_pred == testData$cardio)
c50_rules_acc <- mean(c50_rules_pred == testData$cardio)

accuracy_results <- data.frame(
  Model = c("J48", "C5.0 Tree", "C5.0 Rules"),
  Accuracy = c(j48_acc, c50_acc, c50_rules_acc)
)

print(accuracy_results)
```

```
##        Model  Accuracy
## 1        J48 0.7280082
## 2  C5.0 Tree 0.7326725
## 3 C5.0 Rules 0.7332556
```

Model Comparison

Here i reshape the data for visualization for the model comparison

```
# Load libraries
library(ggplot2)
library(reshape2)

# Create a data frame for your model metrics
model_comparison <- data.frame(
  Model = c("J48", "C5.0 Tree", "C5.0 Rules"),
  Accuracy = c(0.728, 0.733, 0.733),
  Precision = c(0.716, 0.716, 0.716),
  Recall = c(0.781, 0.781, 0.781),
  F1_Score = c(0.747, 0.747, 0.747)
)

# Reshape data for visualization
model_comparison_long <- melt(model_comparison, id.vars = "Model")

# Plot performance metrics
ggplot(model_comparison_long, aes(x = Model, y = value, fill = variable)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Cardio Model Performance Comparison", x = "Decision Tree Model", y = "Metric Value") +
  theme_minimal() +
  scale_fill_manual(values = c("blue", "red", "green", "purple"))
```



Comments: The C5.0 Rules model performed marginally better than the other two. The difference is very small, suggesting that the models are learning similar patterns in the data. Accuracy around 72–73% is decent,

These metrics indicate a reasonably balanced model performance:

Precision (0.716): About 71.6% of the positive predictions were accurate. Recall (0.781): The model correctly identified 78.1% of actual positives. F1-Score (0.747): This value balances precision and recall, showing an overall performance of 74.7%.