```r
options(repos = c(CRAN = "https://cloud.r-project.org/"))
```

Install packages

```r
install.packages("visdat")
```

```
##
## The downloaded binary packages are in
##  /var/folders/dz/tsf1k8z17z1gzfc1zzgksk6r0000gn/T//RtmpmyT5vn/downloaded_packages
```

```r
install.packages("corrplot")
```

```
##
## The downloaded binary packages are in
##  /var/folders/dz/tsf1k8z17z1gzfc1zzgksk6r0000gn/T//RtmpmyT5vn/downloaded_packages
```

```r
install.packages("plotly")
```

```
##
## The downloaded binary packages are in
##  /var/folders/dz/tsf1k8z17z1gzfc1zzgksk6r0000gn/T//RtmpmyT5vn/downloaded_packages
```

```r
install.packages("randomForest")
```

```
##
## The downloaded binary packages are in
##  /var/folders/dz/tsf1k8z17z1gzfc1zzgksk6r0000gn/T//RtmpmyT5vn/downloaded_packages
```

Import library

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(visdat)
library(corrplot)
```

```
## corrplot 0.94 loaded
```

```r
library(plotly)
```

```
##
## Attaching package: 'plotly'
##
## The following object is masked from 'package:ggplot2':
##
```

```
##     last_plot
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(randomForest)
```

```
## randomForest 4.7-1.2
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
## The following object is masked from 'package:ggplot2':
##
##     margin
```

Import dataset

```
heart_data <- read.csv("heart_disease_uci.csv")
```

1. Exploratory Data Analysis (EDA)

1.1 Explore basic information about the dataset

```
str(heart_data)
```

```
## 'data.frame':    920 obs. of  16 variables:
##  $ id      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ age     : int  63 67 67 37 41 56 62 57 63 53 ...
##  $ sex     : chr  "Male" "Male" "Male" "Male" ...
##  $ dataset : chr  "Cleveland" "Cleveland" "Cleveland" "Cleveland" ...
##  $ cp      : chr  "typical angina" "asymptomatic" "asymptomatic" "non-anginal" ...
##  $ trestbps: int  145 160 120 130 130 120 140 120 130 140 ...
##  $ chol    : int  233 286 229 250 204 236 268 354 254 203 ...
##  $ fbs     : logi  TRUE FALSE FALSE FALSE FALSE FALSE ...
##  $ restecg : chr  "lv hypertrophy" "lv hypertrophy" "lv hypertrophy" "normal" ...
##  $ thalch  : int  150 108 129 187 172 178 160 163 147 155 ...
##  $ exang   : logi  FALSE TRUE TRUE FALSE FALSE FALSE ...
##  $ oldpeak : num  2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...
##  $ slope   : chr  "downsloping" "flat" "flat" "downsloping" ...
##  $ ca      : int  0 3 2 0 0 0 2 0 1 0 ...
##  $ thal    : chr  "fixed defect" "normal" "reversable defect" "normal" ...
##  $ num     : int  0 2 1 0 0 0 3 0 2 1 ...
```

This UCI Heart Disease dataset has a comprehensive collection of medical data used for predicting the presence of heart disease in patients. Key features include: - Source: Cleveland Clinic Foundation, part of the UCI Machine Learning Repository - Sample Size: 303 patients - Features: 14 attributes (including the target variable) The target variable in this dataset is particularly valuable for our project. It's integer-valued from 0 (no presence) to 4, indicating increasing severity of heart disease. While our risk prediction model aims to output risk levels rather than diagnose disease severity directly, this classification in the dataset provides a crucial source for training our heart disease risk prediction model.

1.2 Clean the dataset

Step 1: Check the data type of the dataset

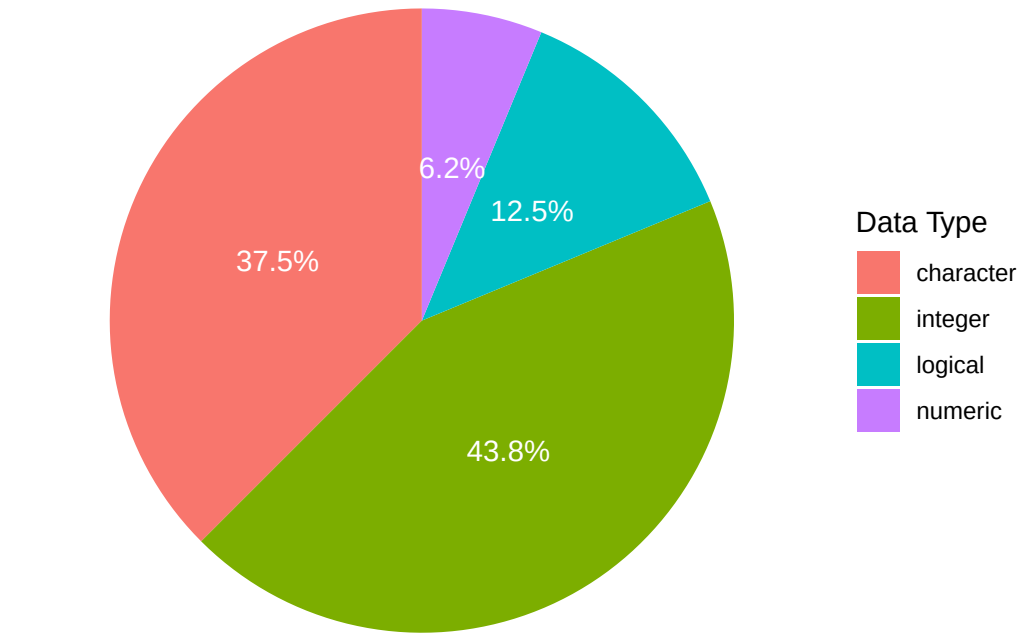
```
# Check types of each column
column_types <- sapply(heart_data, class)
column_types
```

```
##          id         age         sex     dataset          cp    trestbps
##   "integer"   "integer" "character" "character" "character"   "integer"
##        chol         fbs     restecg      thalch       exang     oldpeak
##   "integer"   "logical" "character"   "integer"   "logical"   "numeric"
##       slope          ca        thal         num
## "character"   "integer" "character"   "integer"
```

```
# Create a data frame to store column types and their counts
type_distribution <- as.data.frame(table(column_types))
type_distribution
```

```
##   column_types Freq
## 1    character    6
## 2      integer    7
## 3      logical    2
## 4      numeric    1
```

```
# Calculate type distribution percentages
type_distribution$percentage <- type_distribution$Freq / sum(type_distribution$Freq) * 100

# Create a pie chart to visualize type distribution
ggplot(type_distribution, aes(x = "", y = Freq, fill = column_types)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar("y", start = 0) +
  geom_text(aes(label = paste0(round(percentage, 1), "%")),
            position = position_stack(vjust = 0.5),
            color = "white") +  # Set text color to white
  labs(title = "Distribution of Column Types in Heart Disease Dataset",
       fill = "Data Type") +
  theme_minimal() +
  theme(axis.text = element_blank(),
        axis.title = element_blank(),
        panel.grid = element_blank(),
        plot.title = element_text(hjust = 0.5))
```



Step 2: Convert character columns to factors

As we can see, sex, dataset, cp, restecg, slope, and thal columns are in character format. I convert them to factors, which is more appropriate for categorical data analysis.

```
# Convert character columns to factors
heart_data$sex <- as.factor(heart_data$sex)
heart_data$dataset <- as.factor(heart_data$dataset)
heart_data$cp <- as.factor(heart_data$cp)
heart_data$restecg <- as.factor(heart_data$restecg)
heart_data$slope <- as.factor(heart_data$slope)
heart_data$thal <- as.factor(heart_data$thal)
```

Step 3: Convert logical columns to 0/1

As we can see, fbs and exang coluns are in logical format. I convert them from TRUE/FALSE values to 0/1, i.e. 0 means TRUE, 1 means FALSE.

```
# Convert logical columns to factors
heart_data$fbs <- as.factor(heart_data$fbs)
heart_data$exang <- as.factor(heart_data$exang)

# Display the result after converting
str(heart_data)
```

```
## 'data.frame':    920 obs. of  16 variables:
##  $ id      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ age     : int  63 67 67 37 41 56 62 57 63 53 ...
##  $ sex     : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 2 1 1 2 2 ...
##  $ dataset : Factor w/ 4 levels "Cleveland","Hungary",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ cp      : Factor w/ 4 levels "asymptomatic",..: 4 1 1 3 2 2 1 1 1 1 ...
```

```
##  $ trestbps: int  145 160 120 130 130 120 140 120 130 140 ...
##  $ chol    : int  233 286 229 250 204 236 268 354 254 203 ...
##  $ fbs     : Factor w/ 2 levels "FALSE","TRUE": 2 1 1 1 1 1 1 1 1 2 ...
##  $ restecg : Factor w/ 4 levels "","lv hypertrophy",..: 2 2 2 3 2 3 2 3 2 2 ...
##  $ thalch  : int  150 108 129 187 172 178 160 163 147 155 ...
##  $ exang   : Factor w/ 2 levels "FALSE","TRUE": 1 2 2 1 1 1 1 2 1 2 ...
##  $ oldpeak : num  2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...
##  $ slope   : Factor w/ 4 levels "","downsloping",..: 2 3 3 2 4 4 2 4 3 2 ...
##  $ ca      : int  0 3 2 0 0 0 2 0 1 0 ...
##  $ thal    : Factor w/ 4 levels "","fixed defect",..: 2 3 4 3 3 3 3 3 4 4 ...
##  $ num     : int  0 2 1 0 0 0 3 0 2 1 ...
```

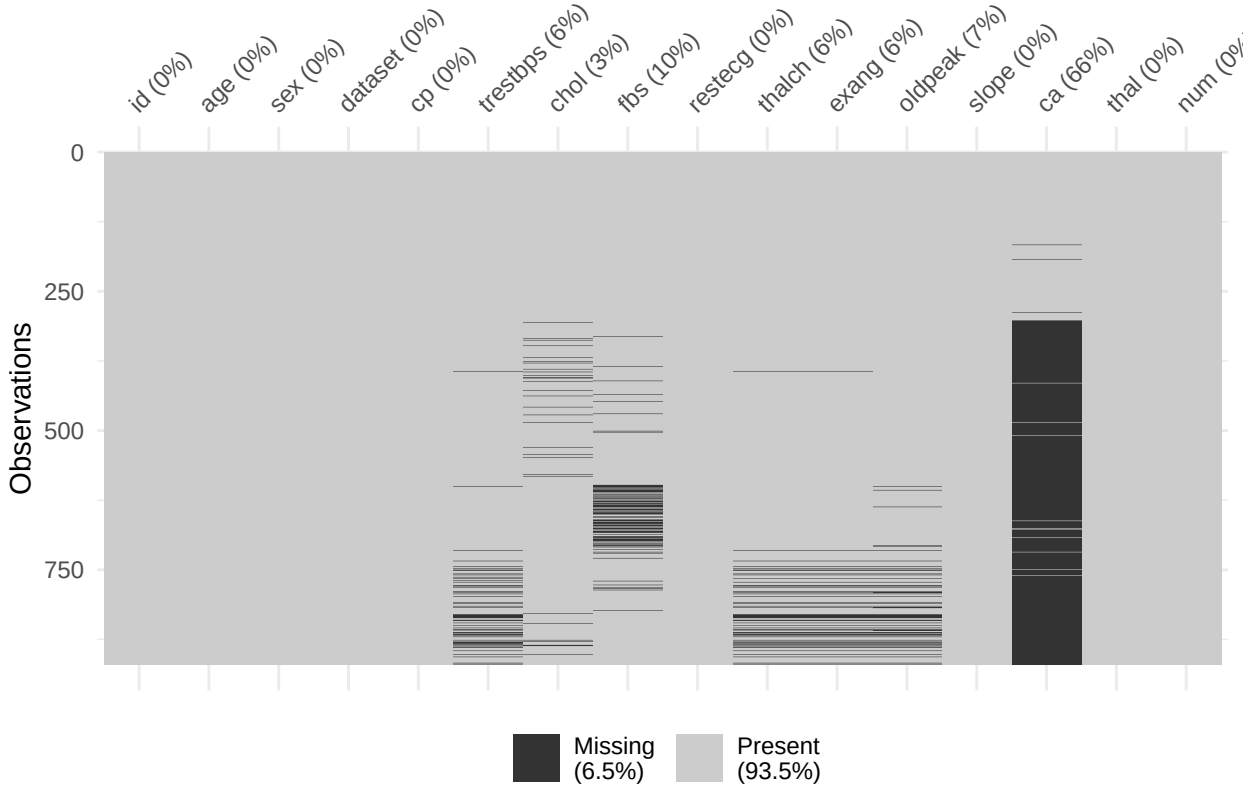
The other values are already integers or numeric, so no change is needed.

Step 4: Check for duplicate and missin values, and remove them

```r
# Remove duplicate rows
heart_data <- heart_data %>% distinct()

# Check and visualize missing values in each column
missing_values <- sapply(heart_data, function(x)sum(is.na(x)))
missing_values
```

```
##       id      age      sex  dataset       cp trestbps     chol      fbs
##        0        0        0        0        0       59       30       90
##  restecg   thalch    exang  oldpeak    slope       ca     thal      num
##        0       55       55       62        0      611        0        0
```

```r
vis_miss(heart_data)
```



```r
# Remove rows with missing values
cleaned_heart_data <- na.omit(heart_data)
```

Display summary of the cleaned dataset

```r
summary(cleaned_heart_data)
```

```
##        id              age            sex               dataset
##  Min.   :  1.0   Min.   :29.00   Female: 97   Cleveland    :299
##  1st Qu.: 76.5   1st Qu.:48.00   Male  :206   Hungary      :  2
##  Median :152.0   Median :56.00                Switzerland  :  0
##  Mean   :156.9   Mean   :54.51                VA Long Beach:  2
##  3rd Qu.:229.5   3rd Qu.:61.00
##  Max.   :760.0   Max.   :77.00
##                cp         trestbps          chol          fbs
##  asymptomatic   :146   Min.   : 94.0   Min.   :  0.0   FALSE:259
##  atypical angina: 50   1st Qu.:120.0   1st Qu.:211.0   TRUE : 44
##  non-anginal    : 84   Median :130.0   Median :240.0
##  typical angina : 23   Mean   :131.7   Mean   :245.5
##                        3rd Qu.:140.0   3rd Qu.:275.0
##                        Max.   :200.0   Max.   :564.0
##              restecg        thalch        exang        oldpeak
##                  :  0   Min.   : 71.0   FALSE:202   Min.   :0.000
##  lv hypertrophy  :147   1st Qu.:132.0   TRUE :101   1st Qu.:0.000
##  normal          :151   Median :152.0               Median :0.800
##  st-t abnormality:  5   Mean   :149.2               Mean   :1.053
##                         3rd Qu.:165.0               3rd Qu.:1.600
##                         Max.   :202.0               Max.   :6.200
##          slope           ca                        thal          num
##             :  2   Min.   :0.0000                   :  4   Min.   :0.0000
##  downsloping: 21   1st Qu.:0.0000   fixed defect     : 18   1st Qu.:0.0000
##  flat       :140   Median :0.0000   normal           :164   Median :0.0000
##  upsloping  :140   Mean   :0.6634   reversable defect:117   Mean   :0.9406
##                    3rd Qu.:1.0000                           3rd Qu.:2.0000
##                    Max.   :3.0000                           Max.   :4.0000
```

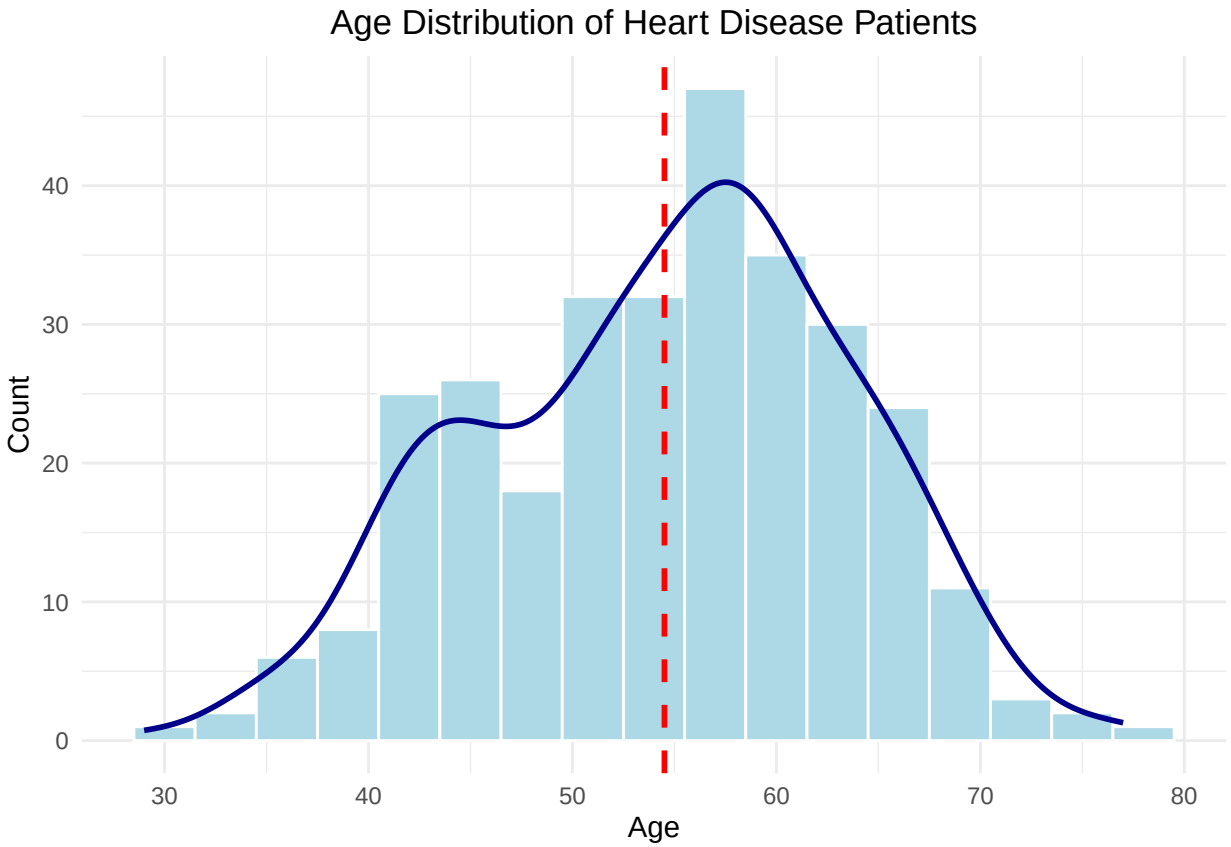
1.3 Visualization

a. Age analysis

```r
ggplot(cleaned_heart_data, aes(x = age)) +
  geom_histogram(binwidth = 3, fill = "lightblue", color = "white") +
  geom_density(aes(y = ..count.. * 3), color = "darkblue", size = 1) +
  geom_vline(aes(xintercept = mean(age)), color = "red", linetype = "dashed", size = 1) +
  labs(title = "Age Distribution of Heart Disease Patients",
       x = "Age", y = "Count") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

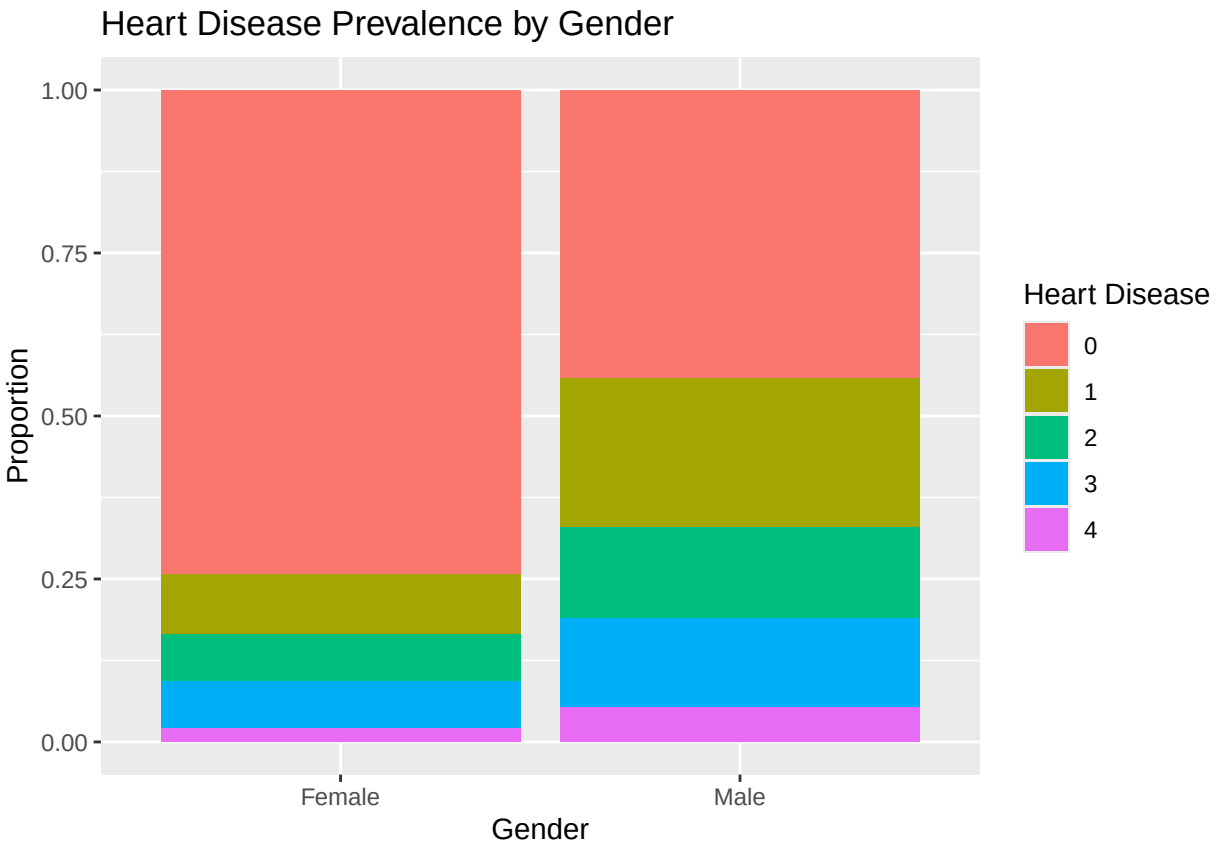
```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



This graph illustrates the age distribution of heart disease patients, ranging from 30 to 80 years old. - The distribution approximates a normal curve, with a peak around 55-60 years of age. - The average age, indicated by the red dashed line, is approximately 55 years old. - While the data spans from 30 to 80 years, it is predominantly concentrated in the 40-70 age range. - There's a notable increase in the frequency of heart disease cases starting from age 40, with the highest concentration between 50 and 65 years. - The occurrence of heart disease is relatively low for individuals under 40 and over 70 years old. This distribution highlights age as a crucial factor in heart disease risk, suggesting that middle-aged to older adults are at higher risk. This insight is valuable for developing targeted prevention strategies and risk assessment models.

b. Gender analysis

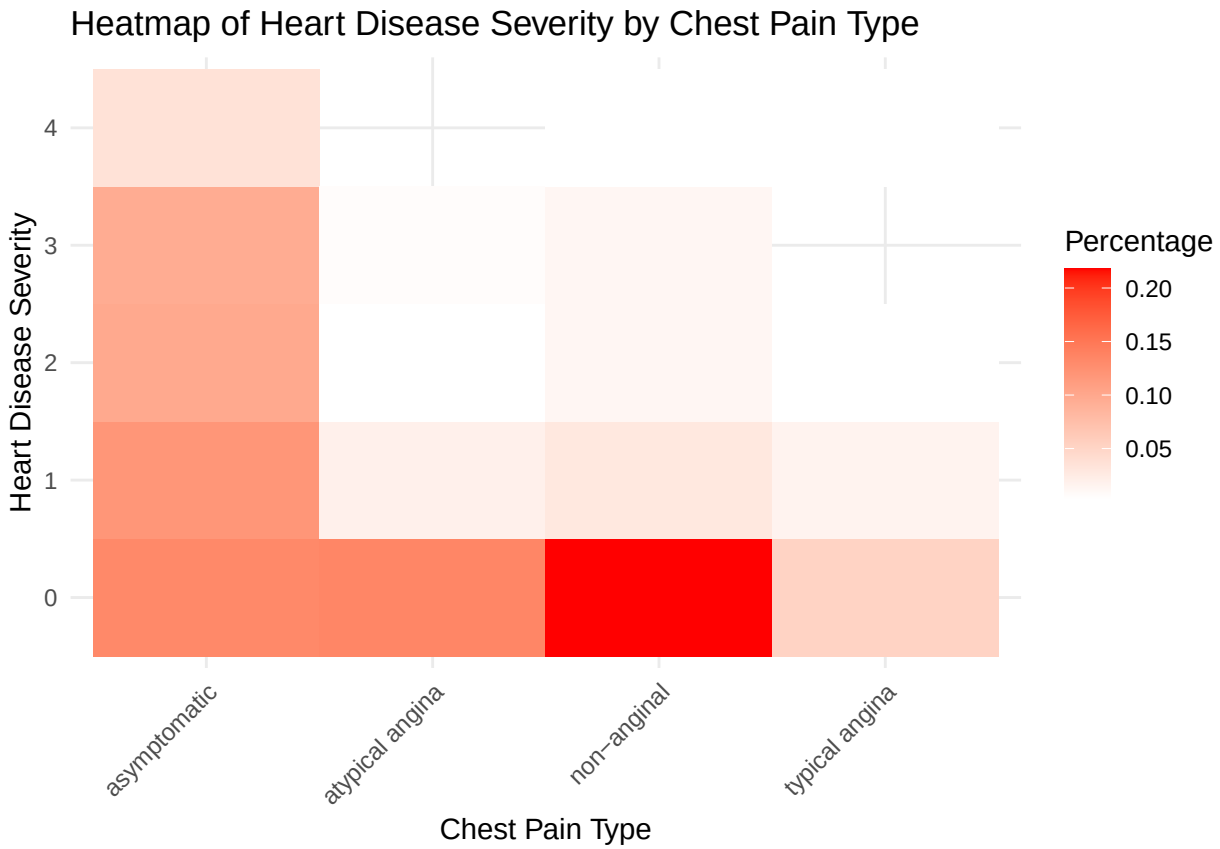
```
ggplot(cleaned_heart_data, aes(x = factor(sex), fill = factor(num))) +
  geom_bar(position = "fill") +
  labs(title = "Heart Disease Prevalence by Gender",
       x = "Gender", y = "Proportion", fill = "Heart Disease")
```



This graph shows the prevalence of heart disease by gender. - Males show a significantly higher overall prevalence of heart disease than females. - About 75% of females have no heart disease (level 0), compared to only 35% of males. - Males have higher proportions across all heart disease severity levels (1-4). This visualization clearly demonstrates that gender is a significant factor in heart disease risk, with males being at higher risk across all severity levels.

c. Chest Pain Type analysis

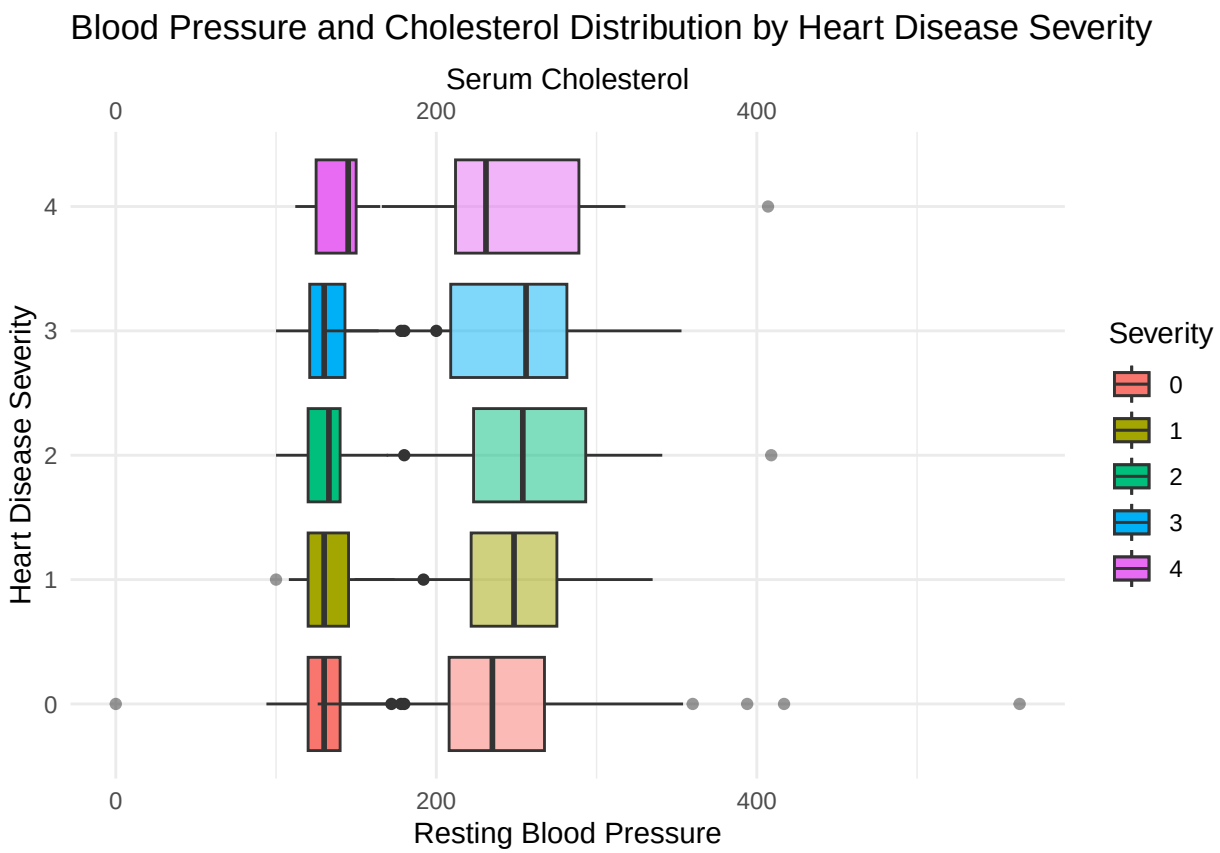
```
# Create a contingency table
heatmap_data <- cleaned_heart_data %>%
  group_by(cp, num) %>%
  summarise(count = n(), .groups = 'drop') %>%
  mutate(percentage = count / sum(count))

# Create the heat map
ggplot(heatmap_data, aes(x = cp, y = factor(num), fill = percentage)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "red") +
  labs(title = "Heatmap of Heart Disease Severity by Chest Pain Type",
       x = "Chest Pain Type",
       y = "Heart Disease Severity",
       fill = "Percentage") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



This heat map reveals the complex relationship between chest pain types and heart disease severity: - Asymptomatic patients show a significant distribution across all severity levels, particularly level 1, indicating that lack of chest pain doesn't rule out heart disease. - Atypical angina is strongly associated with no heart disease (level 0). - Non-anginal pain mostly indicates no heart disease, but has some cases across other severity levels. - Typical angina shows a strong correlation with heart disease, especially at severity levels 1 and 2. These insights highlight that chest pain type, while crucial, is not a definitive indicator of heart disease. The presence of heart disease in asymptomatic patients and the strong association of typical angina with heart disease are particularly noteworthy for risk assessment models.

d. Blood Pressure and Cholesterol analysis

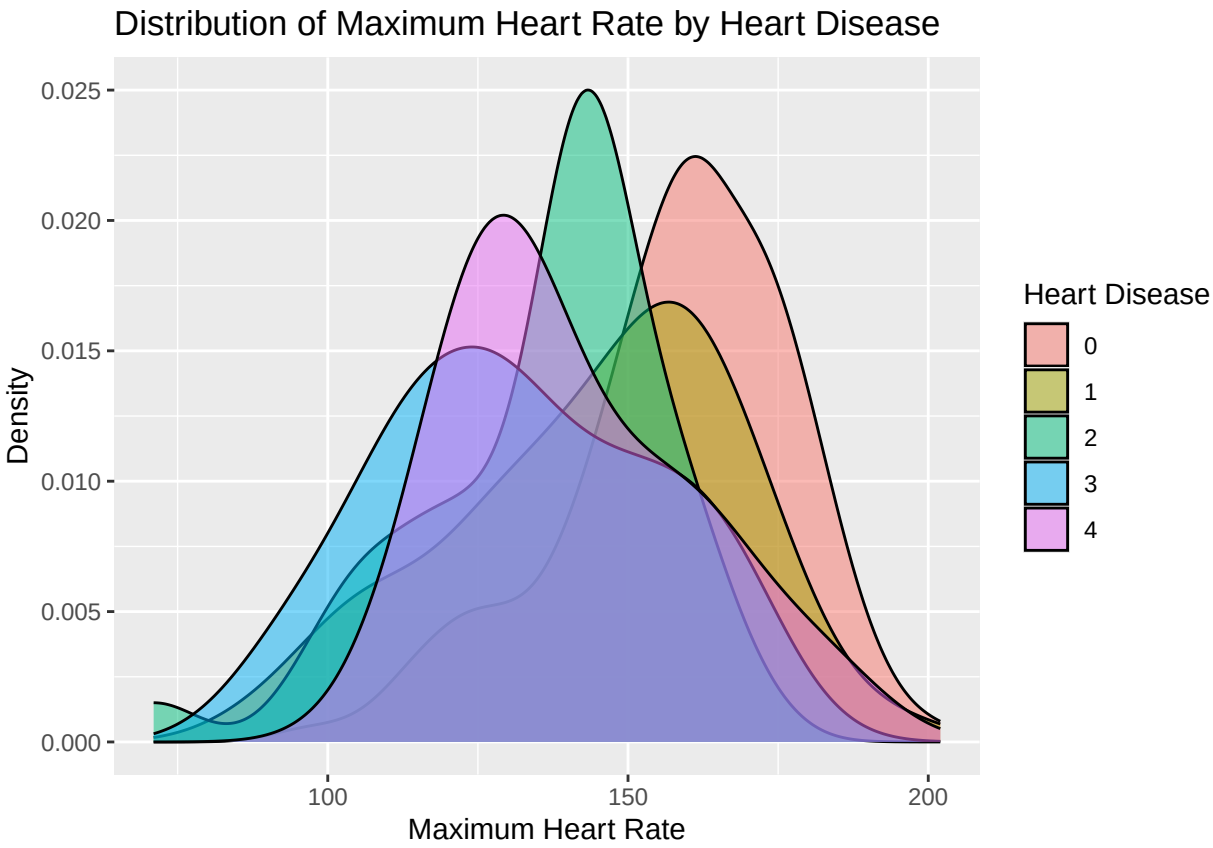
```
# Box plots
ggplot(cleaned_heart_data, aes(x = factor(num), y = trestbps, fill = factor(num))) +
  geom_boxplot() +
  labs(title = "Blood Pressure Distribution by Heart Disease Severity",
       x = "Heart Disease Severity", y = "Resting Blood Pressure", fill = "Severity") +
  theme_minimal() +
  coord_flip() +
  geom_boxplot(aes(y = chol), alpha = 0.5) +
  scale_y_continuous(sec.axis = sec_axis(~., name = "Serum Cholesterol")) +
  ggtitle("Blood Pressure and Cholesterol Distribution by Heart Disease Severity")
```



This box plot shows that while blood pressure remains relatively consistent across heart disease severity levels, cholesterol levels tend to increase with disease severity. However, the considerable overlap in distributions for both measures across severity levels indicates that neither factor alone is a definitive predictor of heart disease. The presence of outliers and varying ranges, particularly in the no-disease group, emphasizes the complex, multifacorial nature of heart disease risk.

e. Maximum Heart Rate analysis

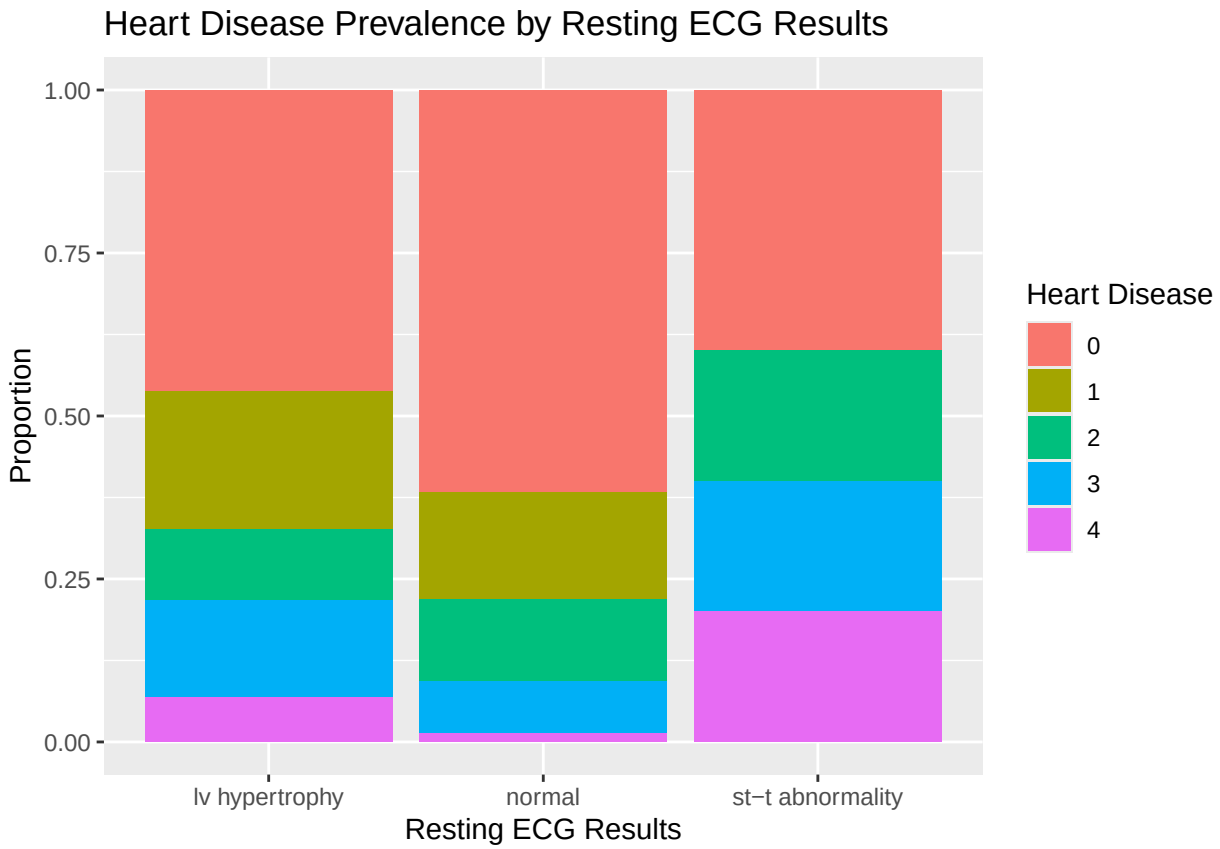
```
ggplot(cleaned_heart_data, aes(x = thalch, fill = factor(num))) +
  geom_density(alpha = 0.5) +
  labs(title = "Distribution of Maximum Heart Rate by Heart Disease",
       x = "Maximum Heart Rate", y = "Density", fill = "Heart Disease")
```



This density plot shows the distribution of maximum heart rates across heart disease severity levels. - As disease severity increases, peak maximum heart rates shift lower. - No-disease group has a higher, narrower peak, indicating more consistent heart rates. - Significant overlap exists between all distributions. - More severe cases tend towards lower maximum heart rates. - Higher severity levels show wider distributions, suggesting more variability. The plot reveals an inverse relationship between maximum heart rate and disease severity, while also demonstrating that heart rate alone is not a definitive indicator of heart disease due to distribution overlaps.

f. Resting ECG analysis

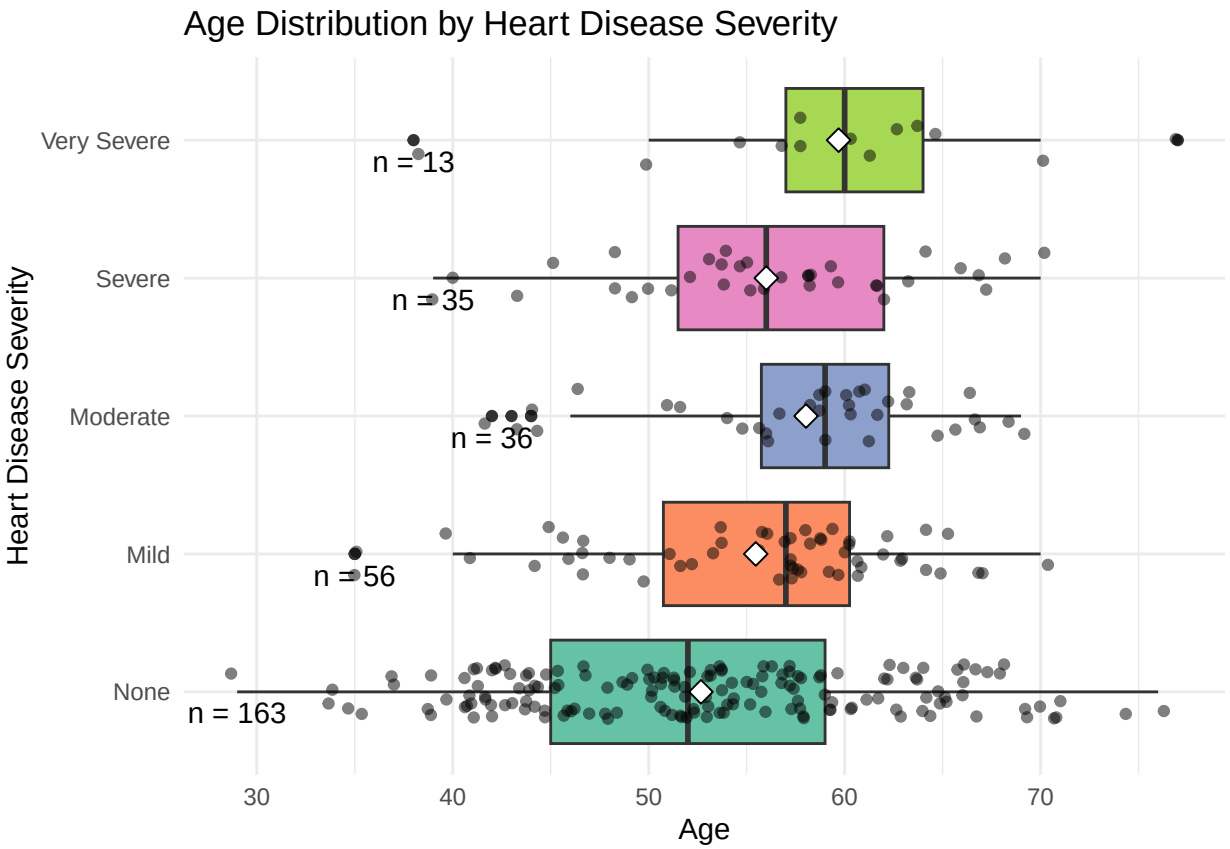
```
ggplot(cleaned_heart_data, aes(x = factor(restecg), fill = factor(num))) +
  geom_bar(position = "fill") +
  labs(title = "Heart Disease Prevalence by Resting ECG Results",
       x = "Resting ECG Results", y = "Proportion", fill = "Heart Disease")
```



This stacked bar chart shows heart disease severity distribution across different resting ECG results: - Normal ECG correlates with lower heart disease risk. - ST-T abnormality strongly links to increased disease severity. - LV hypertrophy shows a mixed distribution across all severity levels. - All ECG categories include some level of heart disease, indicating ECG alone isn't definitive for diagnosis. - Varying distributions emphasize ECG's importance in risk assessment, while suggesting the need for additional diagnostic factors.

h. Target variable distribution

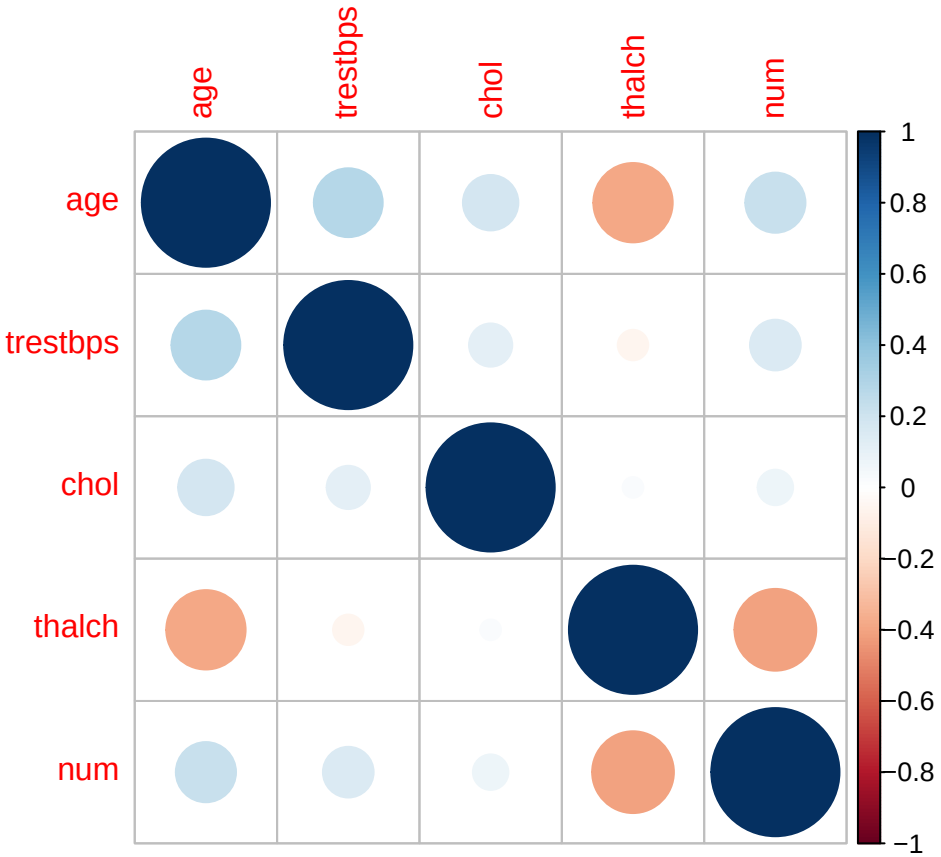
```
ggplot(cleaned_heart_data, aes(x = factor(num), y = age, fill = factor(num))) +
  geom_boxplot() +
  geom_jitter(width = 0.2, alpha = 0.5) +
  stat_summary(fun = mean, geom = "point", shape = 23, size = 3, fill = "white") +
  labs(title = "Age Distribution by Heart Disease Severity",
       x = "Heart Disease Severity",
       y = "Age",
       fill = "Severity") +
  scale_x_discrete(labels = c("None", "Mild", "Moderate", "Severe", "Very Severe")) +
  scale_fill_brewer(palette = "Set2") +
  theme_minimal() +
  theme(legend.position = "none") +
  coord_flip() +
  geom_text(data = cleaned_heart_data %>% group_by(num) %>% summarise(n = n(), age = min(age)),
            aes(label = paste("n =", n), y = age), vjust = 1.5)
```



This boxplot depicts the age distribution across heart disease severity levels, primarily spanning 40-70 years with outliers from 30-75. While median age generally increases with disease severity, significant overlap exists across all levels. The "None" category (n=163) shows the widest age range and lowest median, whereas the "Very Severe" category (n=13) has the narrowest range and highest median. This visualization underscores age as a crucial factor in heart disease risk, but the overlaps suggest it's not the sole determinant. The chart implies that risk assessment models should consider multiple factors, especially for younger patients, and recommends increased vigilance for those over 40 while not neglecting younger individuals with other risk factors.

i. Correlation analysis of numerical variables

```
numerical_vars <- cleaned_heart_data %>% select(age, trestbps, chol, thalch, num)
cor_matrix <- cor(numerical_vars)
corrplot(cor_matrix, method = "circle")
```



This correlation matrix visualizes relationships between key numerical variables in the heart disease dataset. Age shows a moderate negative correlation with maximum heart rate (thalch), suggesting decreased heart rate capacity with age. Notably, thalch has a moderate negative correlation with heart disease diagnosis (num), indicating lower maximum heart rates may be associated with higher disease severity. Age has a weak positive correlation with num, slightly increasing heart disease risk. Resting blood pressure (trestbps) and cholesterol (chol) show weak correlations with other variables. These insights suggest that maximum heart rate and age are potentially significant predictors for the risk model, while blood pressure and cholesterol might need consideration in combination with other factors. The analysis underscores the importance of a multifaceted approach in heart disease risk prediction, accounting for complex interactions between variables.

j. Summary statistics

```
summary(cleaned_heart_data[c("age", "trestbps", "chol", "thalch", "num")])
```

```
##       age           trestbps          chol           thalch
##  Min.   :29.00   Min.   : 94.0   Min.   :  0.0   Min.   : 71.0
##  1st Qu.:48.00   1st Qu.:120.0   1st Qu.:211.0   1st Qu.:132.0
##  Median :56.00   Median :130.0   Median :240.0   Median :152.0
##  Mean   :54.51   Mean   :131.7   Mean   :245.5   Mean   :149.2
##  3rd Qu.:61.00   3rd Qu.:140.0   3rd Qu.:275.0   3rd Qu.:165.0
##  Max.   :77.00   Max.   :200.0   Max.   :564.0   Max.   :202.0
##       num
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.9406
##  3rd Qu.:2.0000
```

```
##  Max.   :4.0000
```

This process generates summary statistics for key numerical variables in the heart disease dataset. It provides a concise overview of age, resting blood pressure (trestbps), cholesterol (chol), maximum heart rate (thalch), and heart disease diagnosis (num). For each variable, it displays the minimum, first quartile, median, mean, third quartile, and maximum values. This summary offers crucial insights into data distribution, central tendencies, and potential outliers, serving as a foundation for understanding variable ranges and informing subsequent data preprocessing and model development stages in the heart disease risk prediction project. Such statistical summaries are essential for identifying significant patterns and guiding feature engineering in the creation of an effective predictive model.

2. Test with dataset

Step 1: Split the dataset into training dataset (80%) and testing dataset (20%)

```r
# Set the random seed to ensure reproducibility
set.seed(42)

# Calculate the number of rows in the original dataset
n <- nrow(cleaned_heart_data)

# Generate indices for the testing set
test_indices <- sample(n, size = round(0.2 * n))

# Split the data into training dataset and testing dataset
train_data <- cleaned_heart_data[-test_indices, ]
test_data <- cleaned_heart_data[test_indices, ]

# View the number of rows in the training dataset and testing dataset
cat("Training dataset sample size:", nrow(train_data), "\n")
```

```
## Training dataset sample size: 242
```

```r
cat("Testing dataset sample size:", nrow(test_data), "\n")
```

```
## Testing dataset sample size: 61
```

Step 2: Train and test the model

```r
# Install and load the nnet package if not already installed
# install.packages("nnet")
library(nnet)

# Build the multinomial logistic regression model on the training set
multinom_model <- multinom(num ~ ., data = train_data)
```

```
## # weights:  135 (104 variable)
## initial  value 389.483975
## iter  10 value 298.342805
## iter  20 value 257.469128
## iter  30 value 198.146894
## iter  40 value 185.103585
## iter  50 value 182.412456
## iter  60 value 181.893184
## iter  70 value 181.833079
```

```
## iter  80 value 181.793490
## iter  90 value 181.771660
## iter 100 value 181.766697
## final  value 181.766697
## stopped after 100 iterations
```

```
# Print the model summary
summary(multinom_model)
```

```
## Warning in sqrt(diag(vc)): NaNs produced
```

```
## Call:
## multinom(formula = num ~ ., data = train_data)
##
## Coefficients:
##   (Intercept)            id          age   sexMale datasetHungary
## 1  -8.7892800 -0.0005865153 -0.042537781 1.4133786     12.4671598
## 2  -0.6726492  0.0023398605 -0.011506409 0.9429758     -4.9558066
## 3  -8.4261031  0.0028567635 -0.081010434 0.8682018     -5.5398162
## 4  -9.8142268 -0.0018554700  0.007259355 3.6873550     -0.3451015
##   datasetSwitzerland datasetVA Long Beach cpatypical angina cpnon-anginal
## 1                  0           15.5105058        -1.0136773     -1.949449
## 2                  0           -4.1909704        -1.5070803     -2.169019
## 3                  0           -3.9830489         0.2872784     -1.760835
## 4                  0           -0.9716086       -22.7818073     -3.541883
##   cptypical angina   trestbps         chol    fbsTRUE restecglv hypertrophy
## 1        -1.692278 0.03052029  0.005620777 -0.8108056             1.9862953
## 2        -2.665095 0.01234794  0.012666262  0.7533201            -0.5519798
## 3       -16.204527 0.02457162  0.005366036  0.4374756             5.0545674
## 4        -2.504197 0.04972558 -0.012086473 -1.0103624            -4.3720386
##   restecgnormal restecgst-t abnormality      thalch exangTRUE   oldpeak
## 1     1.3208279             -12.0964031 -0.01615605 0.4785430 0.3699541
## 2    -0.8271537               0.7064842 -0.02070213 0.4195653 0.8175190
## 3     3.8821349             -17.3628053 -0.02650292 1.3793886 0.7792263
## 4    -6.0772558               0.6350676  0.02422162 0.7752725 0.6331364
##   slopedownsloping  slopeflat slopeupsloping       ca thalfixed defect
## 1       -3.0334387 -2.3452105      -3.410631 1.127723         6.080653
## 2       -0.1426537  0.5811364      -1.111132 1.524639        -3.013020
## 3       -1.8586153 -2.4147970      -4.152691 1.934955       -15.676639
## 4       -2.3960814 -2.0771978      -5.340948 2.140955         3.108242
##   thalnormal thalreversable defect
## 1   7.032760              7.969236
## 2  -4.688857             -2.472940
## 3   4.792635              7.017974
## 4   2.472768              4.416286
##
## Std. Errors:
##   (Intercept)          id        age   sexMale datasetHungary
## 1    1.373744 0.002752178 0.03041992 0.6697033   6.023797e-07
## 2    1.882512 0.003769828 0.04255707 0.9013224   1.535429e-08
## 3    1.673147 0.004074275 0.04344994 0.9333744   2.377175e-08
## 4    1.087897 0.005173662 0.05745772 1.5591638   1.641191e-07
##   datasetSwitzerland datasetVA Long Beach cpatypical angina cpnon-anginal
## 1                NaN         1.900637e-07      7.226267e-01     0.6528963
## 2                NaN         4.343048e-09      1.588171e+00     0.8452027
```

```
## 3               NaN         1.490468e-08      1.395947e+00     0.9530595
## 4       1.574868e-26         5.765604e-08      2.583873e-09     1.3911266
##   cptypical angina   trestbps        chol   fbsTRUE restecglv hypertrophy
## 1     7.604816e-01 0.01398597 0.005675609 0.8317518             0.7429075
## 2     1.336905e+00 0.01952259 0.007421964 0.8847158             0.9663070
## 3     3.477556e-06 0.02048109 0.007639825 0.9509492             0.9400331
## 4     1.551794e+00 0.02564910 0.011976565 1.3308918             0.6462336
##   restecgnormal restecgst-t abnormality     thalch exangTRUE   oldpeak
## 1     0.7084465            1.744097e-06 0.01245466 0.5267392 0.2829648
## 2     0.8851754            1.013706e+00 0.01556011 0.6735093 0.3633907
## 3     0.8725304            7.308780e-10 0.01687026 0.7846352 0.3630634
## 4     0.7446591            1.085721e+00 0.02140538 0.9633551 0.4456183
##   slopedownsloping slopeflat slopeupsloping        ca thalfixed defect
## 1        0.9092847 0.5998925      0.6885620 0.3283380     8.666876e-01
## 2        1.0779295 0.7970149      0.9919641 0.3930724     9.773955e-01
## 3        1.0399698 0.7358707      0.9402403 0.4185793     3.912781e-08
## 4        1.2436448 0.7396926      1.1558252 0.5101666     1.011033e+00
##   thalnormal thalreversable defect
## 1  0.6185356             0.6386070
## 2  0.9241424             0.9004734
## 3  0.8894790             0.9722449
## 4  0.9038860             0.7729399
##
## Residual Deviance: 363.5334
## AIC: 547.5334
```

```r
# Make predictions on the test set
predicted_classes <- predict(multinom_model, newdata = test_data, type = "class")

# Evaluate the model's performance
confusion_matrix <- table(predicted_classes, test_data$num)
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)

cat("Confusion Matrix:\n")
```

```
## Confusion Matrix:
```

```r
print(confusion_matrix)
```

```
##
## predicted_classes  0  1  2  3  4
##                 0 33  8  3  0  0
##                 1  1  2  0  2  0
##                 2  1  0  1  3  1
##                 3  0  0  2  1  0
##                 4  1  1  0  1  0
```

```r
cat("\nAccuracy:", round(accuracy, 3), "\n")
```

```
##
## Accuracy: 0.607
```

Model performance description: 1. Model Convergence: The model converged after 100 iterations, reaching a final deviance value of 181.766697. 2. Model Fit: The model's Residual Deviance is 363.5334, and its AIC (Akaike Information Criterion) is 547.5334. Lower values indicate better fit, but without comparison models, it's hard to judge the absolute quality of fit. 3. Coefficients: The model produced coefficients for each predictor variable across the different outcome levels. Some coefficients show large values, which may

indicate strong predictive power or potential overfitting. 4. Standard Errors: Some standard errors are very small (close to zero), which could indicate perfect prediction in some cases or potential issues with the model. 5. Confusion Matrix: This shows the model's predictive performance: The model correctly classified 33 cases of no heart disease (class 0). It had mixed performance on other classes, with some misclassifications. 6. Accuracy: The overall accuracy of the model is 0.607 or 60.7%. This indicates that the model correctly predicted the heart disease class for about 61% of the cases in the test set.

In summary, the model shows some predictive power, especially for identifying cases with no heart disease, its performance on other classes is mixed. The accuracy of 60.7% suggests there's room for improvement. Further analysis and potentially trying other modeling approaches might be beneficial to enhance the predictive performance for all classes of heart disease severity.