FORECASTING HEART DISEASE RISKS

Seif Kungulio

2025-10-20

Contents

Introduction

Business Understanding

Problem statement

To develop models for an insurance company using the Heart Disease dataset from the UCI Machine Learning Repository. The goal is to predict the likelihood of a person developing heart disease, which would help the insurance company estimate health risks and adjust premiums accordingly.

Data Understanding

The dataset contains various features related to patients' health and demographic information. We will explore the dataset to understand its structure and relationships between variables.

Data description

The Heart Disease dataset from the UCI Machine Learning Repository contains 303 instances and 14 attributes. These attributes include both numerical and categorical variables related to patients' health metrics and demographic information. The target variable indicates the presence or absence of heart disease. These attributes are:

1. `age`: Age of the patient (numeric)
2. `sex`: Gender of the patient (1 = male, 0 = female)
3. `cp`: Chest pain type (categorical: 1-4)
4. `trestbps`: Resting blood pressure (numeric)
5. `chol`: Serum cholesterol (numeric)
6. `fbs`: Fasting blood sugar (1 = true, 0 = false)
7. `restecg`: Resting electrocardiographic results (categorical)
8. `thalach`: Maximum heart rate achieved (numeric)
9. `exang`: Exercise-induced angina (1 = yes, 0 = no)
10. `oldpeak`: ST depression induced by exercise (numeric)
11. `slope`: The slope of the peak exercise ST segment (categorical)
12. `ca`: Number of major vessels (0-3, numeric)
13. `thal`: Thalassemia (categorical: 1 = normal, 2 = fixed defect, 3 = reversible defect)
14. `target`: Heart disease (1 = disease, 0 = no disease)

Data dictionary

The dataset contains 14 key attributes that are either numerical or categorical.

Attribute	Type	Description	Constraints/ Rules
`age`	Numerical	The age of the patient in years	Range: 29-77 (based on dataset statistics)
`sex`	Categorical	The gender of the patient	Values: 1 = Male, 0 = Female
`cp`	Categorical	Type of chest pain experienced by the patient	Values: 1 = Typical angina, 2 = Atypical angina, 3 = Non-anginal pain, 4 = Asymptomatic
`trestbps`	Numerical	Resting blood pressure of the patient, measured in mmHg	Range: Typically, between 94 and 200 mmHg
`chol`	Numerical	Serum cholesterol level in mg/dl	Range: Typically, between 126 and 564 mg/dl
`fbs`	Categorical	Fasting blood sugar level > 120 mg/dl	Values: 1 = True, 0 = False
`restecg`	Categorical	Results of the patient's resting electrocardiogram	Values: 0 = Normal, 1 = ST-T wave abnormality, 2 = Probable or definite left ventricular hypertrophy
`thalach`	Numerical	Maximum heart rate achieved during a stress test	Range: Typically, between 71 and 202 bpm
`exang`	Categorical	Whether the patient experiences exercise-induced angina	Values: 1 = Yes, 0 = No
`oldpeak`	Numerical	ST depression induced by exercise relative to rest (an ECG measure)	Range: 0.0 to 6.2 (higher values indicate more severe abnormalities)
`slope`	Categorical	Slope of the peak exercise ST segment	Values: 1 = Upsloping, 2 = Flat, 3 = Downsloping

Attribute	Type	Description	Constraints/ Rules
`ca`	Numerical	Number of major vessels colored by fluoroscopy	Range: 0-3
`thal`	Categorical	Blood disorder variable related to thalassemia	Values: 3 = Normal, 6 = Fixed defect, 7 = Reversible defect
`target`	Categorical	Diagnosis of heart disease	Values: 0 = No heart disease, 1 = Presence of heart disease

Initial observations

- The dataset contains a mix of numerical and categorical variables.
- Some variables may require preprocessing, such as handling missing values and encoding categorical variables.
- Missing Values: Some fields like ca and thal may have missing values or unknown entries ('?').
- Data Types: Some categorical variables are encoded numerically and will need to be interpreted correctly during analysis.
- Class Imbalance: Preliminary checks suggest the dataset is relatively balanced between presence and absence of disease, but this will be verified.
- Outliers: Numerical fields such as chol (cholesterol) and trestbps (blood pressure) may have outliers that need to be detected and considered in analysis.

Data Preparation

Data loading

Load the dataset from the UCI website to memory

```r
# Load the dataset
url <- "https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/processed.cleveland.data

# Read the dataset into a dataframe
Heart.df <- read.csv(text = getURL(url), header = FALSE, na.strings = "?")
```

Rename the columns into a meaningful column names

```r
colnames(Heart.df) <- c("age", "sex", "cp", "trestbps", "chol", "fbs",
                        "restecg", "thalach", "exang", "oldpeak",
                        "slope", "ca", "thal", "target")
```

Display dimensions of the dataset

```r
dim(Heart.df)
```

```
## [1] 303  14
```

Display the first six rows of the dataset

```r
head(Heart.df)
```

```
##   age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
## 1  63   1  1      145  233   1       2     150     0     2.3     3  0    6
## 2  67   1  4      160  286   0       2     108     1     1.5     2  3    3
## 3  67   1  4      120  229   0       2     129     1     2.6     2  2    7
## 4  37   1  3      130  250   0       0     187     0     3.5     3  0    3
## 5  41   0  2      130  204   0       2     172     0     1.4     1  0    3
## 6  56   1  2      120  236   0       0     178     0     0.8     1  0    3
##   target
## 1      0
## 2      2
## 3      1
## 4      0
## 5      0
## 6      0
```

Display the structure of the dataframe

```
glimpse(Heart.df)
```

```
## Rows: 303
## Columns: 14
## $ age      <dbl> 63, 67, 67, 37, 41, 56, 62, 57, 63, 53, 57, 56, 56, 44, 52, 5~
## $ sex      <dbl> 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1~
## $ cp       <dbl> 1, 4, 4, 3, 2, 2, 4, 4, 4, 4, 4, 2, 3, 2, 3, 3, 2, 4, 3, 2, 1~
## $ trestbps <dbl> 145, 160, 120, 130, 130, 120, 140, 120, 130, 140, 140, 140, 1~
## $ chol     <dbl> 233, 286, 229, 250, 204, 236, 268, 354, 254, 203, 192, 294, 2~
## $ fbs      <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0~
## $ restecg  <dbl> 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 0, 2, 2, 0, 0, 0, 0, 0, 0, 0, 2~
## $ thalach  <dbl> 150, 108, 129, 187, 172, 178, 160, 163, 147, 155, 148, 153, 1~
## $ exang    <dbl> 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1~
## $ oldpeak  <dbl> 2.3, 1.5, 2.6, 3.5, 1.4, 0.8, 3.6, 0.6, 1.4, 3.1, 0.4, 1.3, 0~
## $ slope    <dbl> 3, 2, 2, 3, 1, 1, 3, 1, 2, 3, 2, 2, 2, 1, 1, 1, 3, 1, 1, 1, 2~
## $ ca       <dbl> 0, 3, 2, 0, 0, 0, 2, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0~
## $ thal     <dbl> 6, 3, 7, 3, 3, 3, 3, 3, 7, 7, 6, 3, 6, 7, 7, 3, 7, 3, 3, 3, 3~
## $ target   <int> 0, 2, 1, 0, 0, 0, 3, 0, 2, 1, 0, 0, 2, 0, 0, 0, 1, 0, 0, 0, 0~
```

Display the statistical summary of the dataframe

```
summary(Heart.df)
```

```
##       age             sex               cp           trestbps
##  Min.   :29.00   Min.   :0.0000   Min.   :1.000   Min.   : 94.0
##  1st Qu.:48.00   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:120.0
##  Median :56.00   Median :1.0000   Median :3.000   Median :130.0
##  Mean   :54.44   Mean   :0.6799   Mean   :3.158   Mean   :131.7
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000   Max.   :4.000   Max.   :200.0
##
##       chol            fbs            restecg          thalach
##  Min.   :126.0   Min.   :0.0000   Min.   :0.0000   Min.   : 71.0
##  1st Qu.:211.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:133.5
##  Median :241.0   Median :0.0000   Median :1.0000   Median :153.0
##  Mean   :246.7   Mean   :0.1485   Mean   :0.9901   Mean   :149.6
##  3rd Qu.:275.0   3rd Qu.:0.0000   3rd Qu.:2.0000   3rd Qu.:166.0
##  Max.   :564.0   Max.   :1.0000   Max.   :2.0000   Max.   :202.0
##
##      exang           oldpeak         slope             ca
##  Min.   :0.0000   Min.   :0.00   Min.   :1.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000
##  Median :0.0000   Median :0.80   Median :2.000   Median :0.0000
##  Mean   :0.3267   Mean   :1.04   Mean   :1.601   Mean   :0.6722
##  3rd Qu.:1.0000   3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :6.20   Max.   :3.000   Max.   :3.0000
##                                                   NA's   :4
##       thal           target
##  Min.   :3.000   Min.   :0.0000
##  1st Qu.:3.000   1st Qu.:0.0000
##  Median :3.000   Median :0.0000
##  Mean   :4.734   Mean   :0.9373
##  3rd Qu.:7.000   3rd Qu.:2.0000
##  Max.   :7.000   Max.   :4.0000
##  NA's   :2
```

Data preprocessing

We will preprocess the data by handling missing values, encoding categorical variables, and scaling numerical features.

According to the data dictionary, the following attributes should be have binary variables: `sex`, `fbs`, `exang`, and `target`. But, some shows to have values besides 0's and 1's. Let's convert binary variables to (0, 1)

```
Heart.df$target <- ifelse(Heart.df$target > 0, 1, 0)
Heart.df$sex <- ifelse(Heart.df$sex > 0, 1, 0)
Heart.df$fbs <- ifelse(Heart.df$fbs > 0, 1, 0)
Heart.df$exang <- ifelse(Heart.df$exang > 0, 1, 0)
```

Handle missing values in `ca` and `thal` variables using mean/mode imputation.

```
Heart.df$ca[is.na(Heart.df$ca)] <- median(Heart.df$ca, na.rm = TRUE)
Heart.df$ca[Heart.df$ca == "?"] <- median(Heart.df$ca, na.rm = TRUE)
#Heart.df$thal[is.na(Heart.df$thal)] <- median(Heart.df$thal, na.rm = TRUE)
Heart.df$ca[Heart.df$thal == "?"] <- median(Heart.df$thal, na.rm = TRUE)
```

Check for missing values if still exist

```
sapply(Heart.df, function(x) sum(is.na(x)))
```

```
##      age      sex       cp trestbps     chol      fbs  restecg  thalach
##        0        0        0        0        0        0        0        0
##    exang  oldpeak    slope       ca     thal   target
##        0        0        0        0        2        0
```

Check for duplicate entries and print them if they exist.

```
dupes <- Heart.df[duplicated(Heart.df) | duplicated(Heart.df, fromLast = TRUE), ]
print(dupes)
```

```
##  [1] age      sex      cp       trestbps chol     fbs      restecg  thalach
##  [9] exang    oldpeak  slope    ca       thal     target
## <0 rows> (or 0-length row.names)
```

Convert categorical variables to factor. Define a list of categorical columns with their levels and labels

```
categorical_columns <- list(
  sex = list(levels = c(0, 1), labels = c("Female", "Male")),
  cp = list(levels = c(1, 2, 3, 4), labels = c("Typical Angina",
                                               "Atypical Angina", "Non-Angina",
                                               "Asymptomatic")),
  fbs = list(levels = c(0, 1), labels = c("False", "True")),
  restecg = list(levels = c(0, 1, 2), labels = c("Normal", "Wave-abnormality", "Probable")),
  exang = list(levels = c(0, 1), labels = c("No", "Yes")),
  slope = list(levels = c(1, 2, 3), labels = c("Upsloping", "Flat",
```

```r
                                               "Downsloping")),
  thal = list(levels = c(3, 6, 7), labels = c("Normal", "Fixed Defect", "Reversible")),
  target = list(levels = c(1, 0), labels = c("Yes", "No"))
)
```

Apply the factor transformation using a for-loop.

```r
for (col in names(categorical_columns)) {
  Heart.df[[col]] <- factor(Heart.df[[col]],
                            levels = categorical_columns[[col]]$levels,
                            labels = categorical_columns[[col]]$labels)
}
```

Helper functions

Function to create Box plots

```r
HeartDiseaseBoxplot <- function(var1, var2) {
  ggplot(Heart.df, aes(x = .data[[var1]],
                       y = .data[[var2]],
                       fill = .data[[var1]])) +
    geom_boxplot() +
    labs(title = paste("Boxplot of", var2, "by", var1),
         x = var1, y = var2, fill = "Heart Disease")
}
```

Function to create Bar plots

```r
HeartDiseaseBar <- function(var) {
  ggplot(Heart.df, aes(x = .data[[var]], fill = target)) +
    geom_bar(position = "dodge") +
    labs(title = paste("Distribution of Heart Disease by", var),
         x = var, fill = "Heart Disease")
}
```

Function to create Histograms

```r
HeartDiseaseHist <- function(var1) {
  ggplot(Heart.df, aes(x = .data[[var1]], fill = target)) +
    geom_histogram(bins = 15) +
    labs(title = paste("Distribution of", var1),
         x = var1, fill = "Heart Disease")
}
```

Function to create Scatter plots

```r
HeartDiseaseScatter <- function(point1, point2){
  ggplot(Heart.df, aes(x = .data[[point1]],
                       y = .data[[point2]],
                       color = target)) +
    geom_point(size = 2) +
    geom_smooth(method = "lm", se = FALSE, color = "blue", formula = y ~ x) +
    labs(title = paste("Scatterplot of", point1, "by", point2),
       x = point1, y = point2, color = "Heart Disease")
}
```

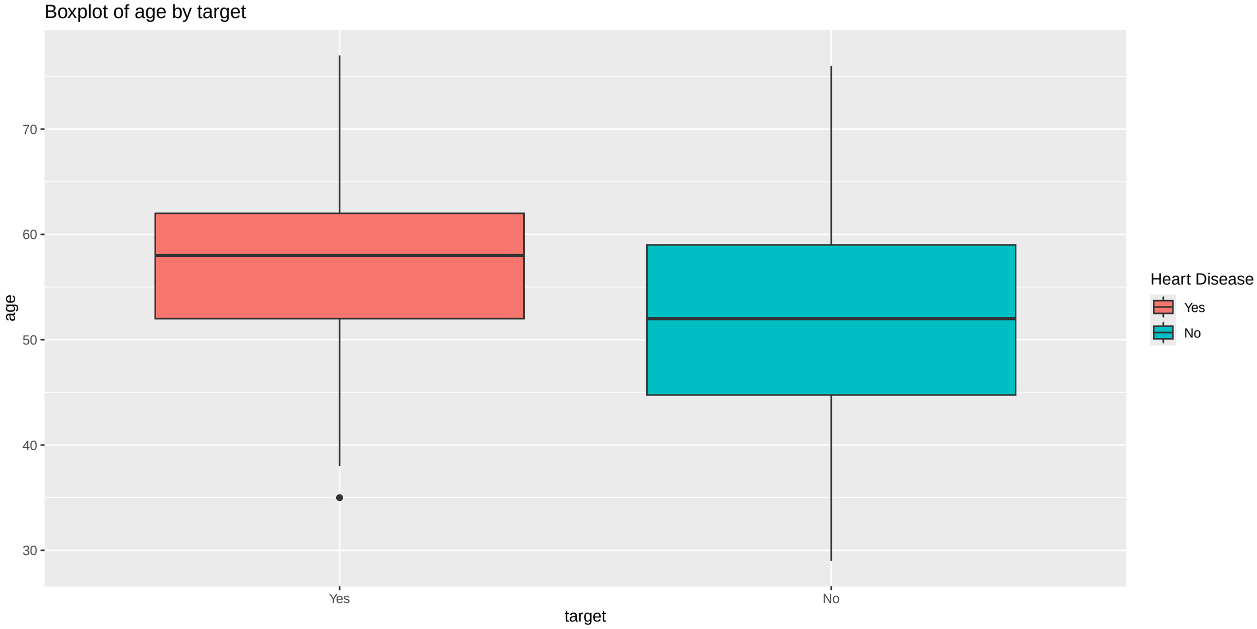
Exploratory data analysis

Boxplots for numerical variables

I used boxplots to visually examine the distribution of key continuous health indicators — such as age, resting blood pressure (trestbps), cholesterol (chol), maximum heart rate (thalach), and ST depression (oldpeak) — across the binary target variable (Heart Disease: Yes / No). Boxplots were chosen because they efficiently highlight differences in central tendency (median), variability (IQR), and the presence of potential outliers between patients with and without heart disease.

Boxplot of Age by Heart Disease

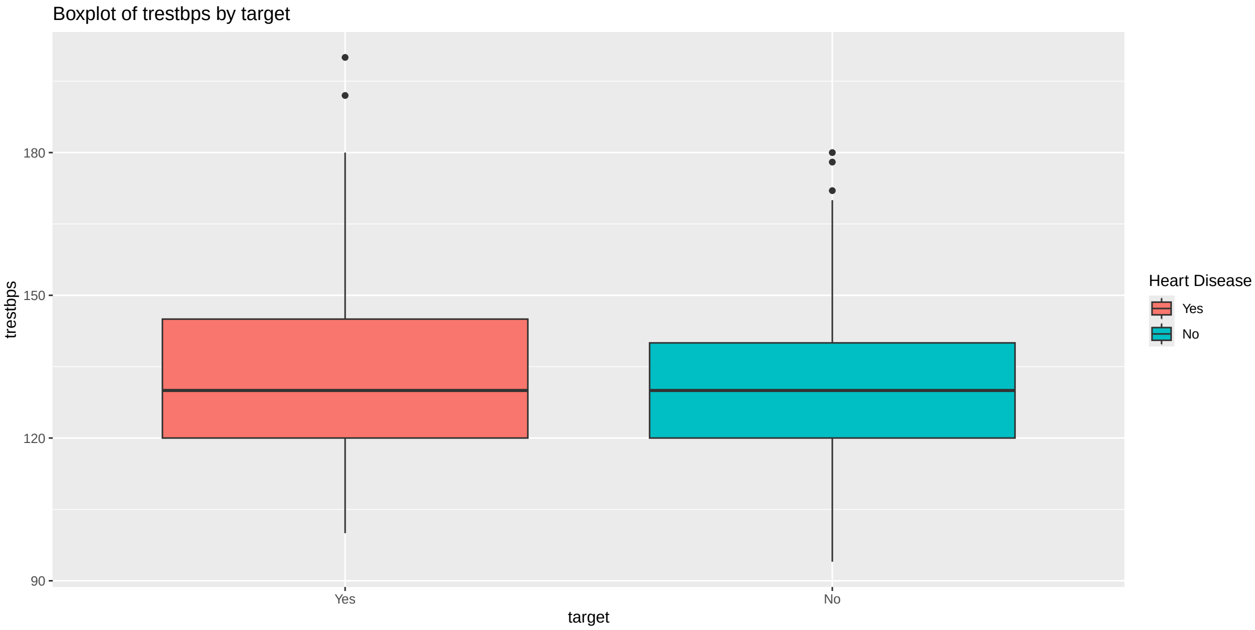
```
HeartDiseaseBoxplot("target", "age")
```



This boxplot compares patients' ages across those with and without heart disease. The median age of patients diagnosed with heart disease is slightly higher than that of those without the condition. The interquartile range (IQR) indicates moderate variability in both groups, though patients above 60 years tend to be more represented among the "Yes" category. This supports the well-established medical understanding that age is a key risk factor — as individuals grow older, arterial stiffening, accumulated cholesterol, and other degenerative processes increase the probability of heart disease onset.

Boxplot of Resting Blood Pressure (trestbps) by Heart Disease

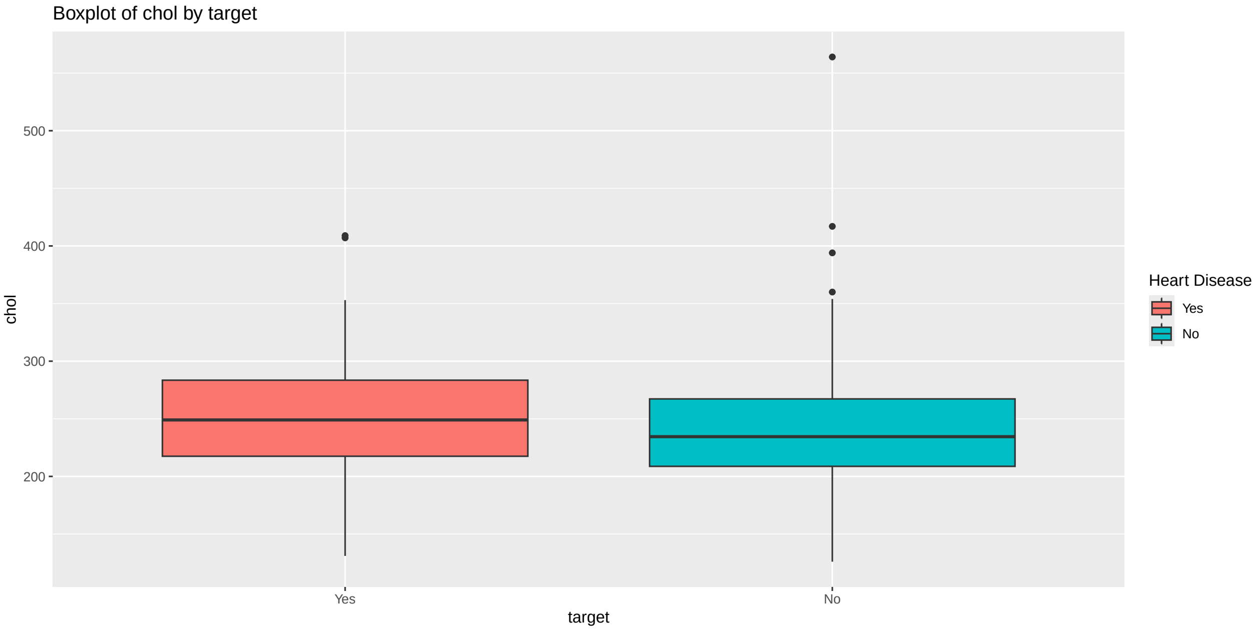
```
HeartDiseaseBoxplot("target", "trestbps")
```



The resting blood pressure distribution reveals higher median values among patients with heart disease. Several outliers are visible in both groups, but those diagnosed with heart disease show a wider spread and higher upper quartile range. This pattern suggests that elevated blood pressure (hypertension) is correlated with higher cardiac risk, consistent with clinical evidence linking high resting systolic pressure to cardiovascular complications. However, some overlap between groups also indicates that blood pressure alone may not be sufficient for classification without considering other risk variables.

Boxplot of Cholesterol (chol) by Heart Disease

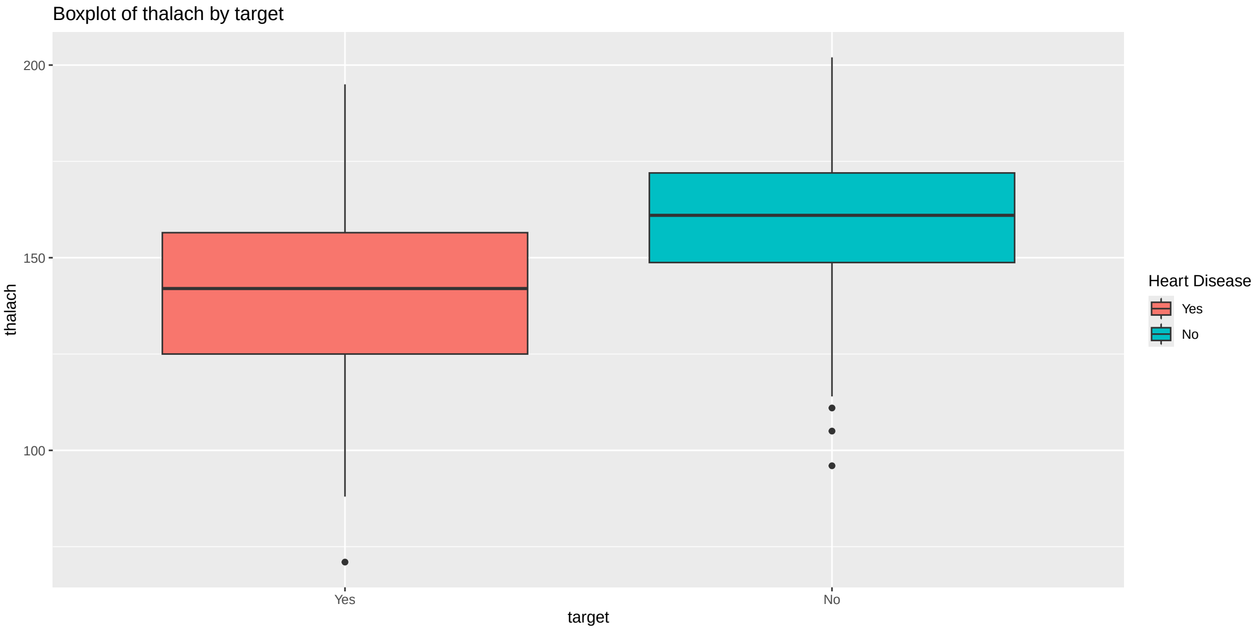
```
HeartDiseaseBoxplot("target", "chol")
```



Serum cholesterol levels demonstrate a wide distribution, with notable outliers at very high cholesterol values (above 400 mg/dl). Patients with heart disease show a slightly higher median cholesterol level, though the difference between the two groups is less pronounced compared to blood pressure. This indicates that while cholesterol is an important indicator, it may not linearly predict disease presence in isolation. The large spread also suggests variability due to genetic factors, diet, or lifestyle influences among patients.

Boxplot of Maximum Heart Rate Achieved (thalach) by Heart Disease

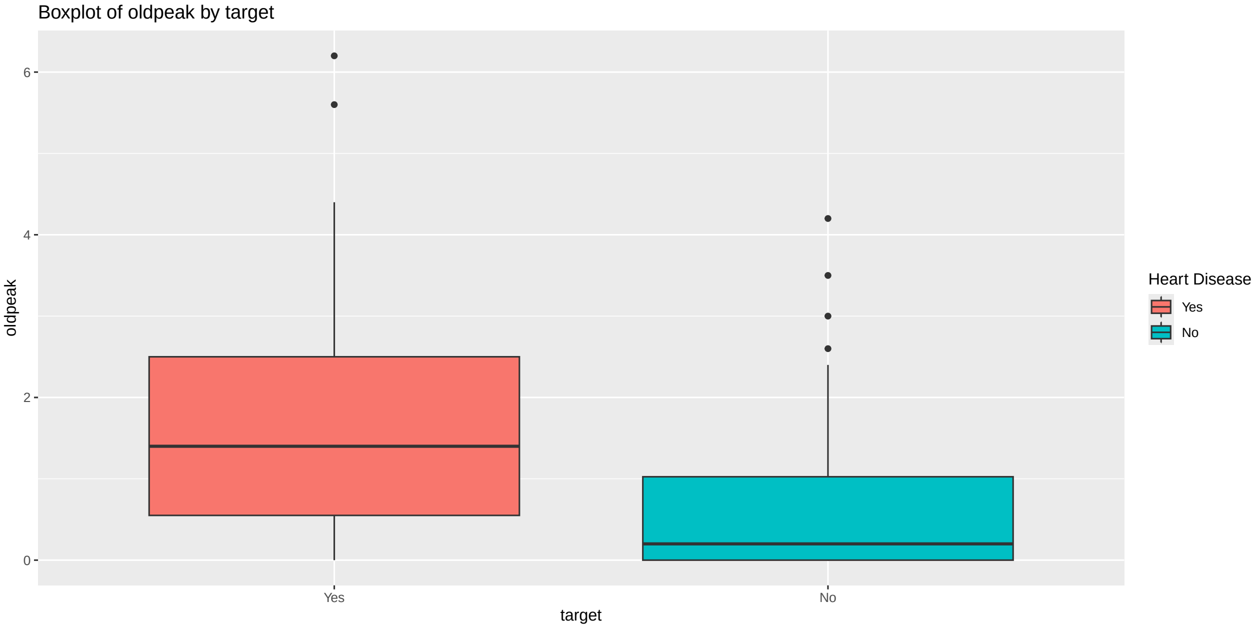
```
HeartDiseaseBoxplot("target", "thalach")
```



This boxplot displays an inverse relationship between maximum heart rate and heart disease presence. Patients without heart disease generally achieve higher maximum heart rates, whereas those with the condition tend to have lower values and a tighter IQR. This observation aligns with medical reasoning — reduced heart rate response during exercise can indicate compromised cardiovascular function or reduced cardiac efficiency. It reinforces the role of thalach (maximum heart rate achieved) as a strong negative predictor of heart disease.

Boxplot of ST Depression (oldpeak) by Heart Disease

```
HeartDiseaseBoxplot("target", "oldpeak")
```



The ST depression variable (oldpeak) shows a distinct separation between the two groups. Patients with heart disease exhibit higher median oldpeak values, suggesting greater ST depression during exercise testing.

This is clinically significant — elevated oldpeak values typically reflect myocardial ischemia (reduced blood flow to the heart muscle during stress). The distribution for the disease-positive group also displays a wider range and several high-value outliers, further underscoring oldpeak as a powerful diagnostic indicator in detecting ischemic patterns associated with heart disease.

Overall boxplots observations:

Across all boxplots, the EDA reveals that: * Age, resting blood pressure, and cholesterol are positively associated with heart disease risk. * Maximum heart rate (thalach) and ST depression (oldpeak) are particularly differentiating indicators — the former being lower and the latter being higher among heart disease patients. * The presence of outliers in several metrics (especially cholesterol and blood pressure) suggests that individual variability or external lifestyle factors play a non-negligible role. * Collectively, these patterns highlight the multifactorial nature of cardiovascular disease, where no single biomarker suffices — but a combination provides stronger predictive power for modeling.

Handle outliers

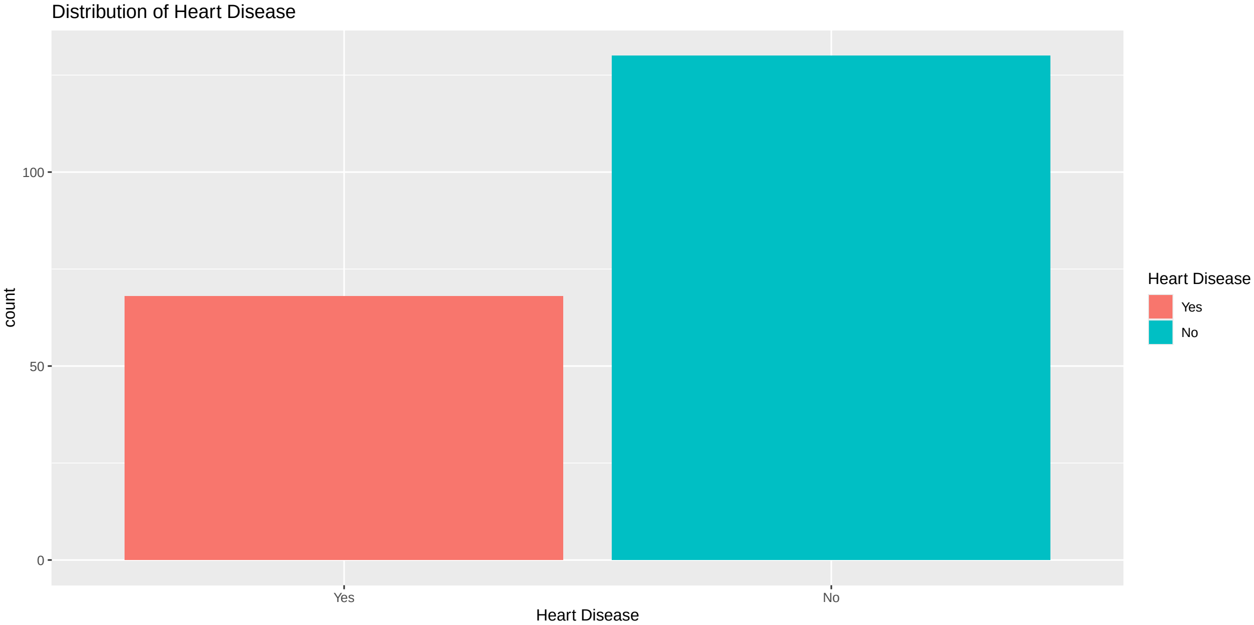
Apply multiple filters to identify and handle outliers in numerical variables.

```
Heart.df <- Heart.df[Heart.df$age > 40 &
                       Heart.df$trestbps < 170 &
                       Heart.df$chol < 340 &
                       Heart.df$chol > 150 &
                       Heart.df$thalach > 115 &
                       Heart.df$oldpeak < 2.4, ]
```

Barplots for categorical variables

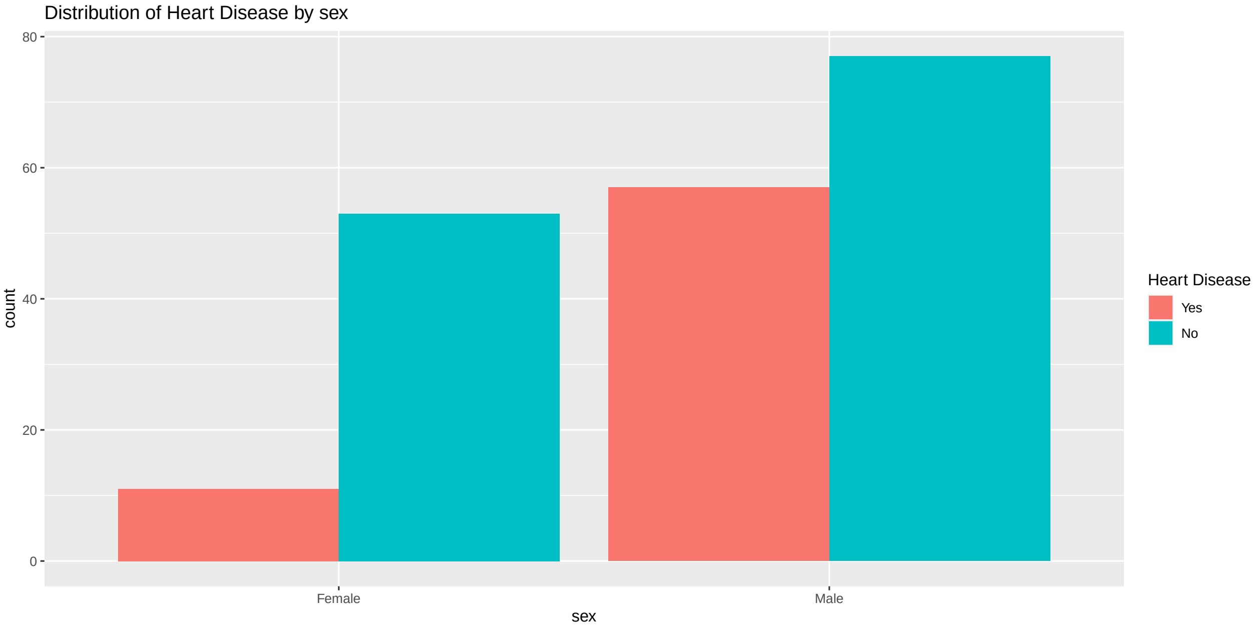
Heart disease distribution

```
ggplot(Heart.df, aes(x=target, fill=target))+
  geom_bar() +
  ggtitle("Distribution of Heart Disease") +
  labs(x = "Heart Disease", fill = "Heart Disease")
```

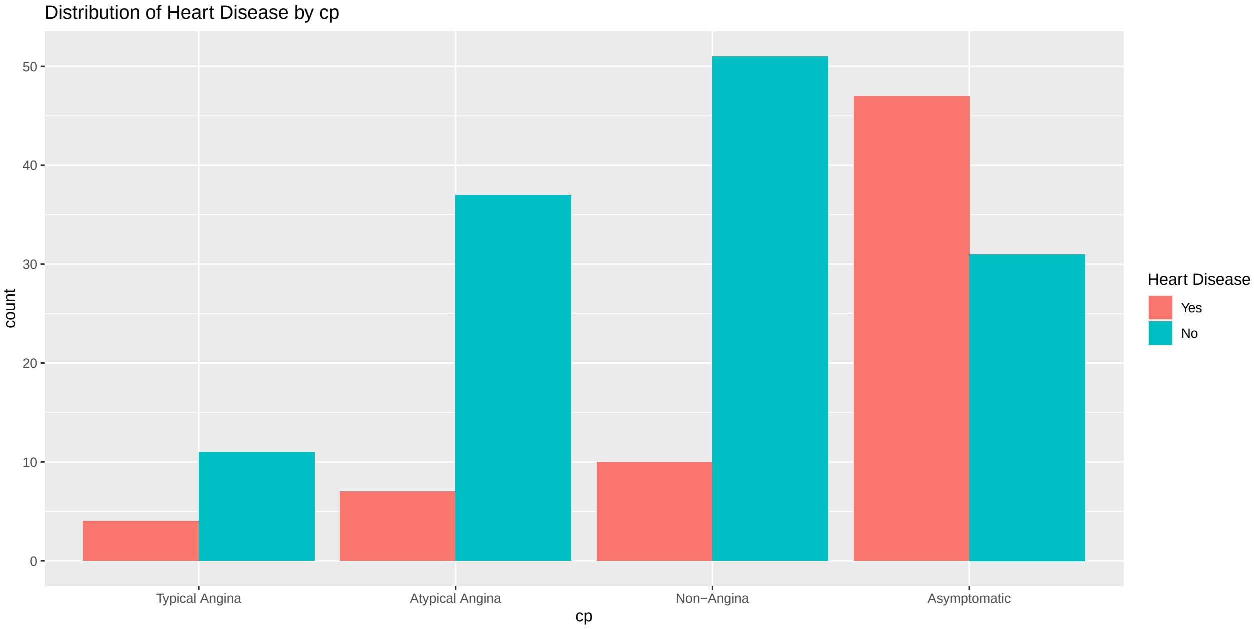


Visualize distribution of categorical variables by heart disease presence.

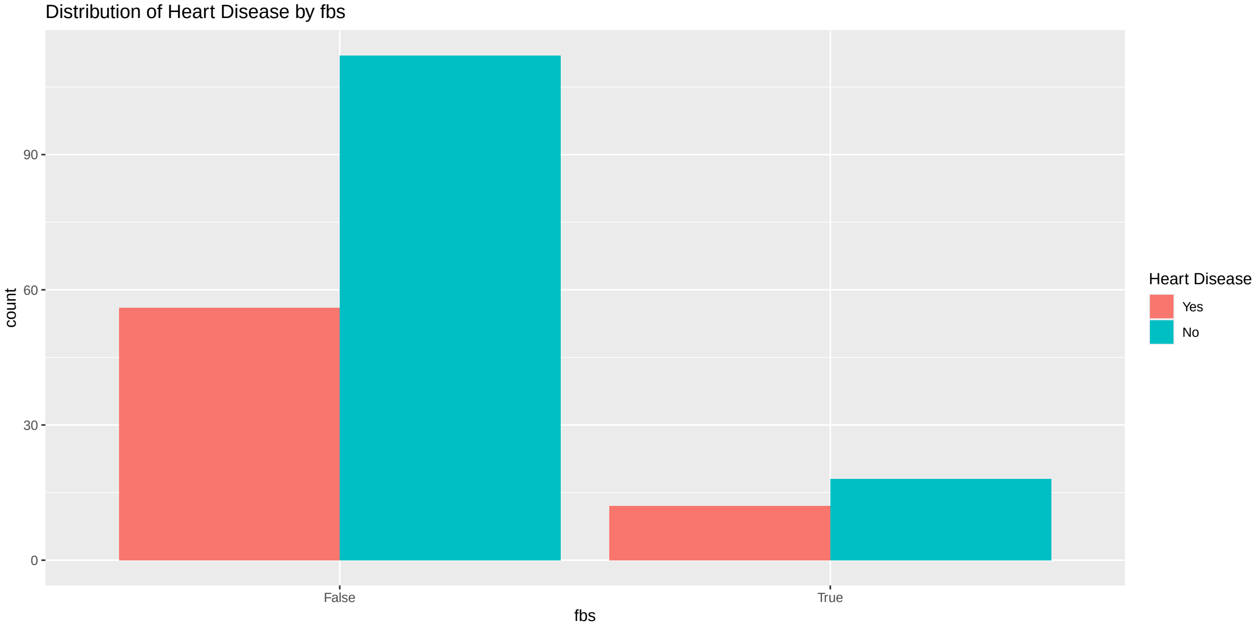
```r
HeartDiseaseBar("sex")
```



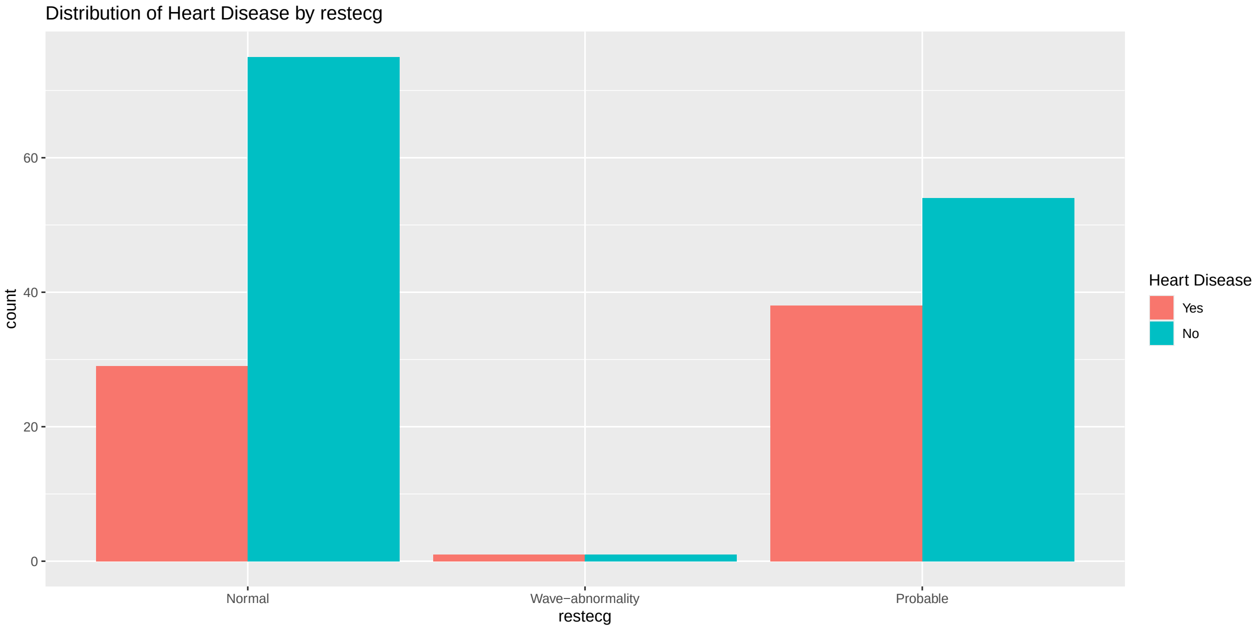
```r
HeartDiseaseBar("cp")
```



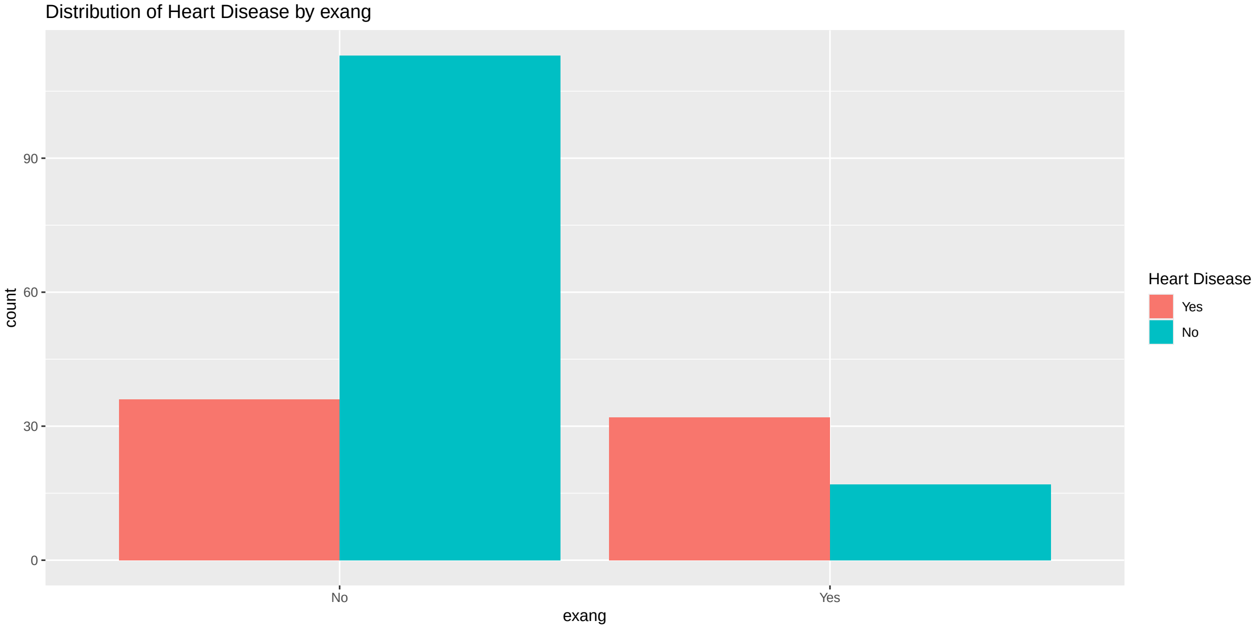
```r
HeartDiseaseBar("fbs")
```



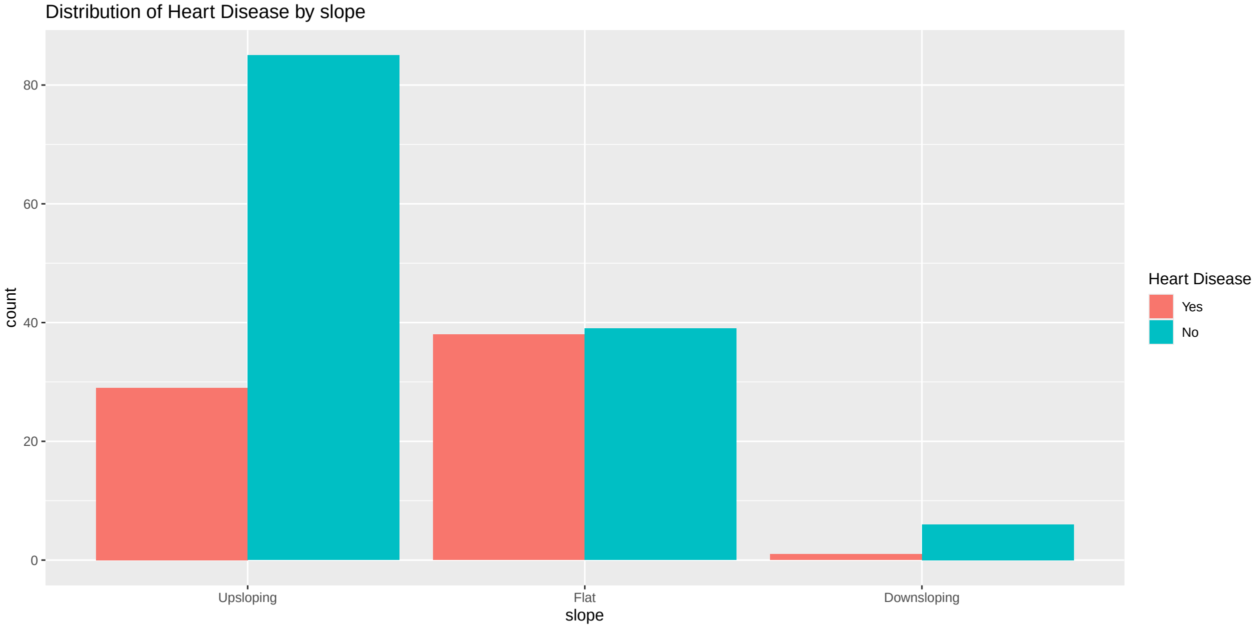
```
HeartDiseaseBar("restecg")
```



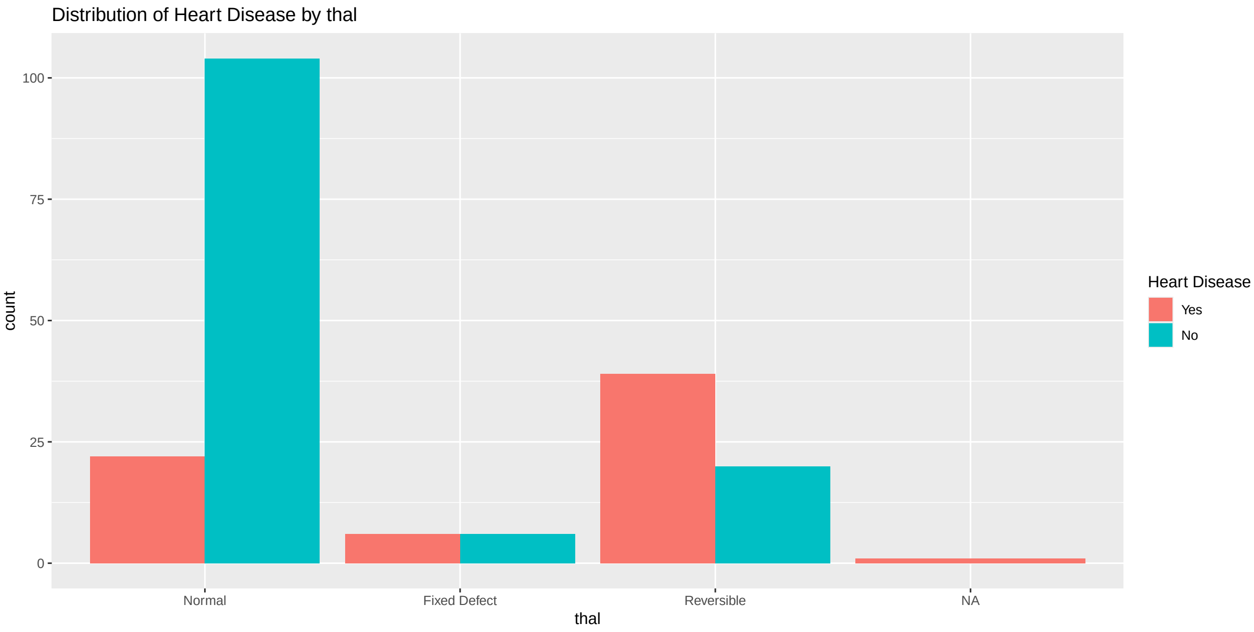
```
HeartDiseaseBar("exang")
```



```r
HeartDiseaseBar("slope")
```

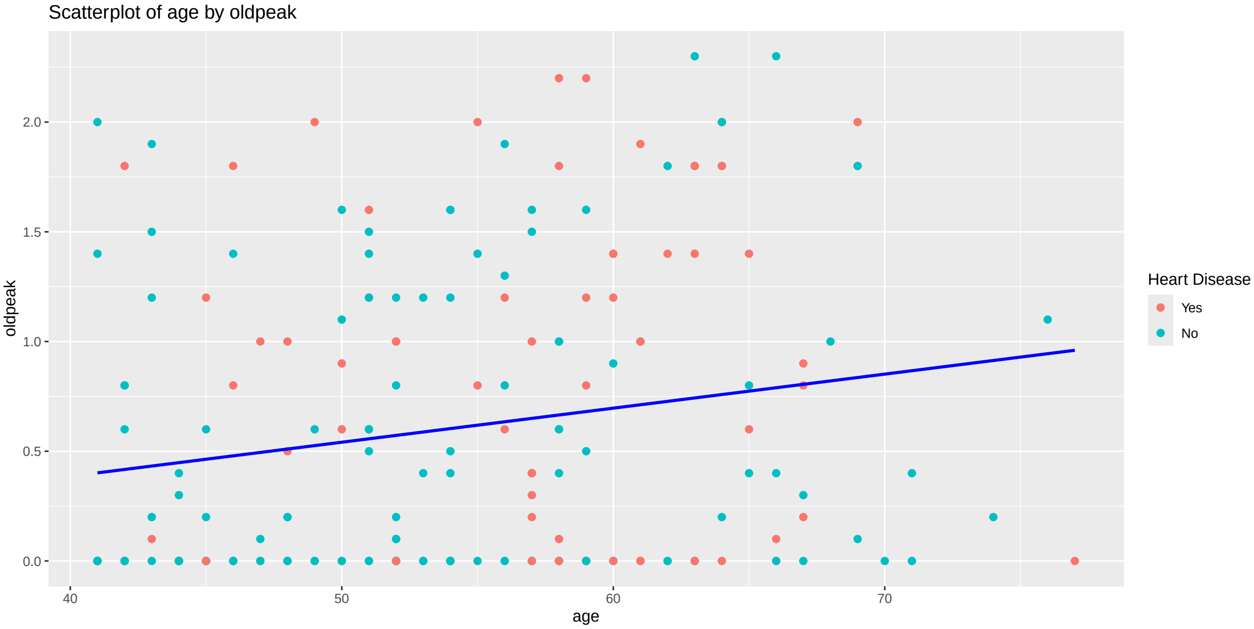


```r
HeartDiseaseBar("thal")
```

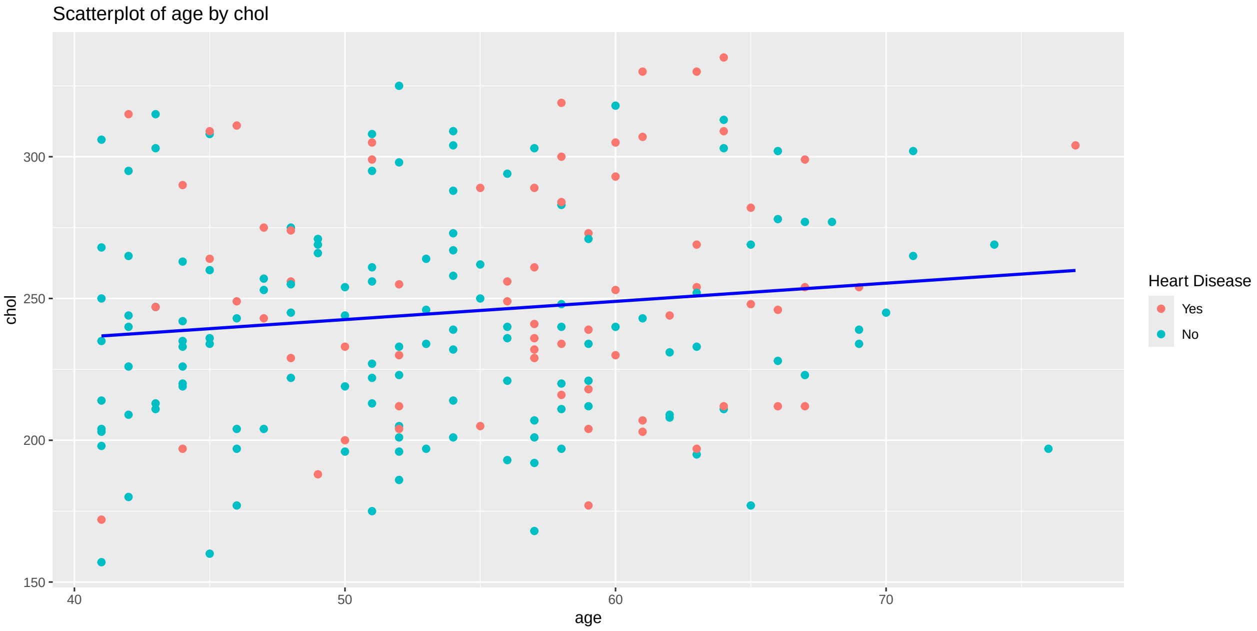


Scatterplots for Numerical Variables

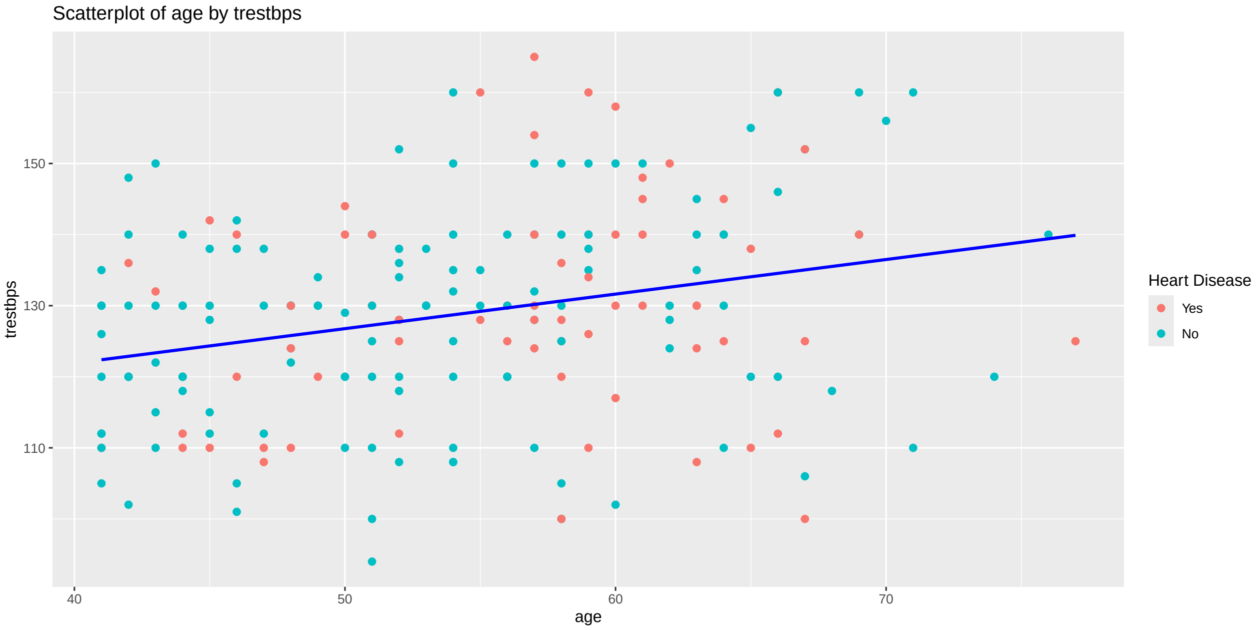
```r
HeartDiseaseScatter("age", "oldpeak")
```



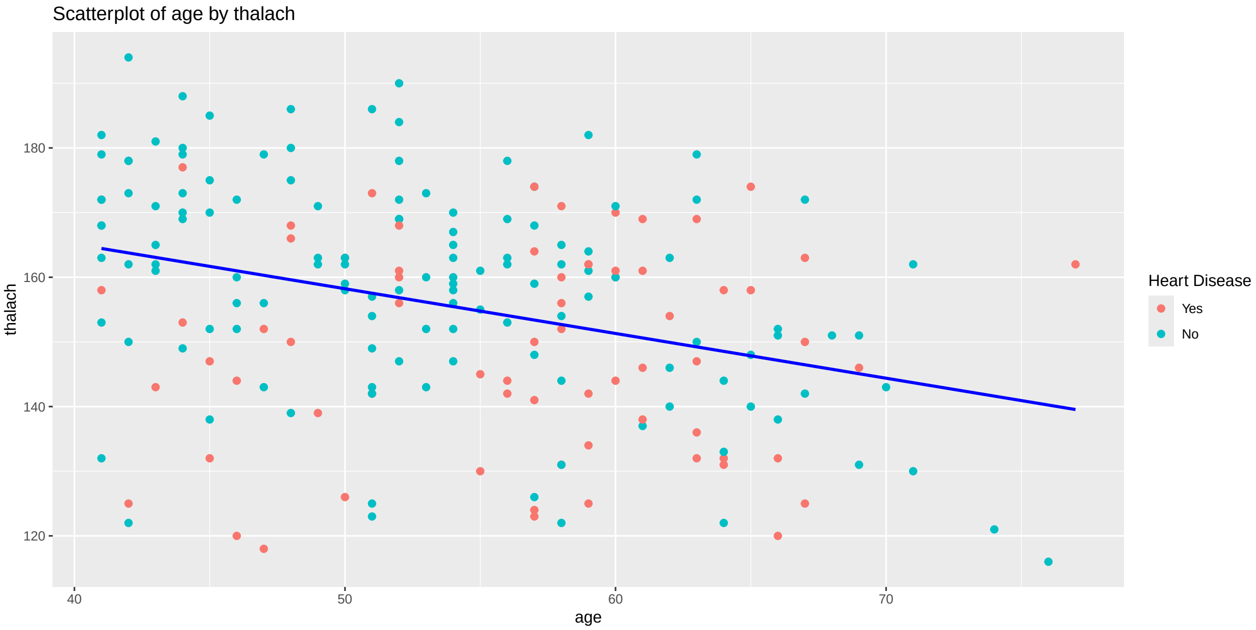
```r
HeartDiseaseScatter("age", "chol")
```



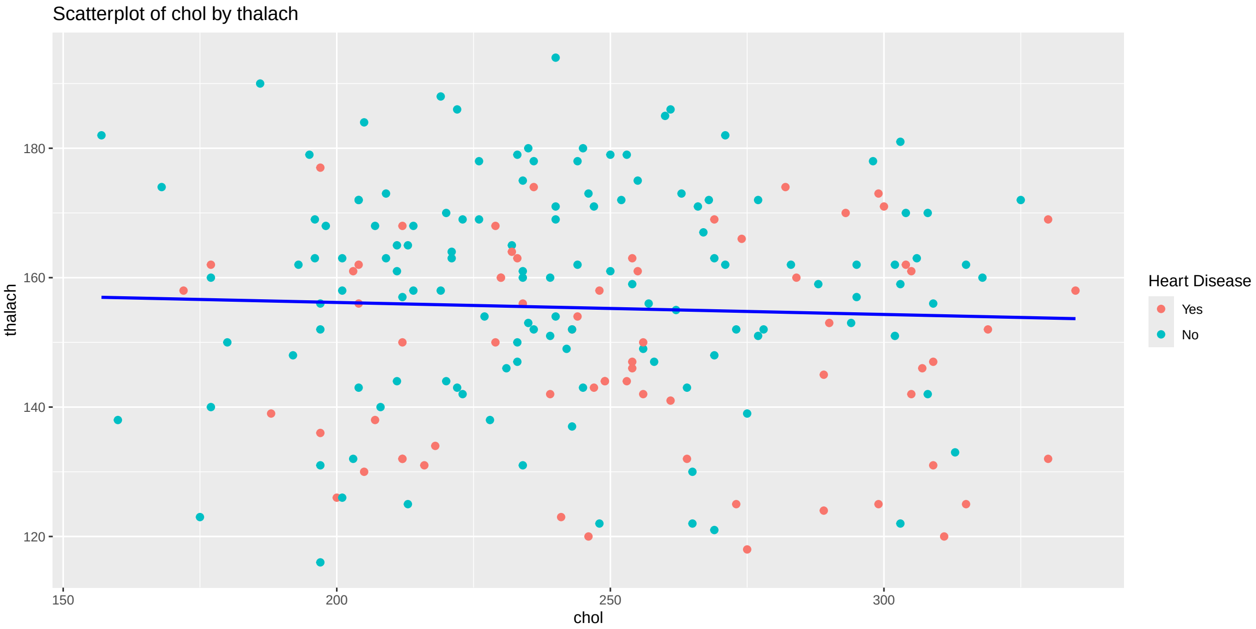
```
HeartDiseaseScatter("age", "trestbps")
```



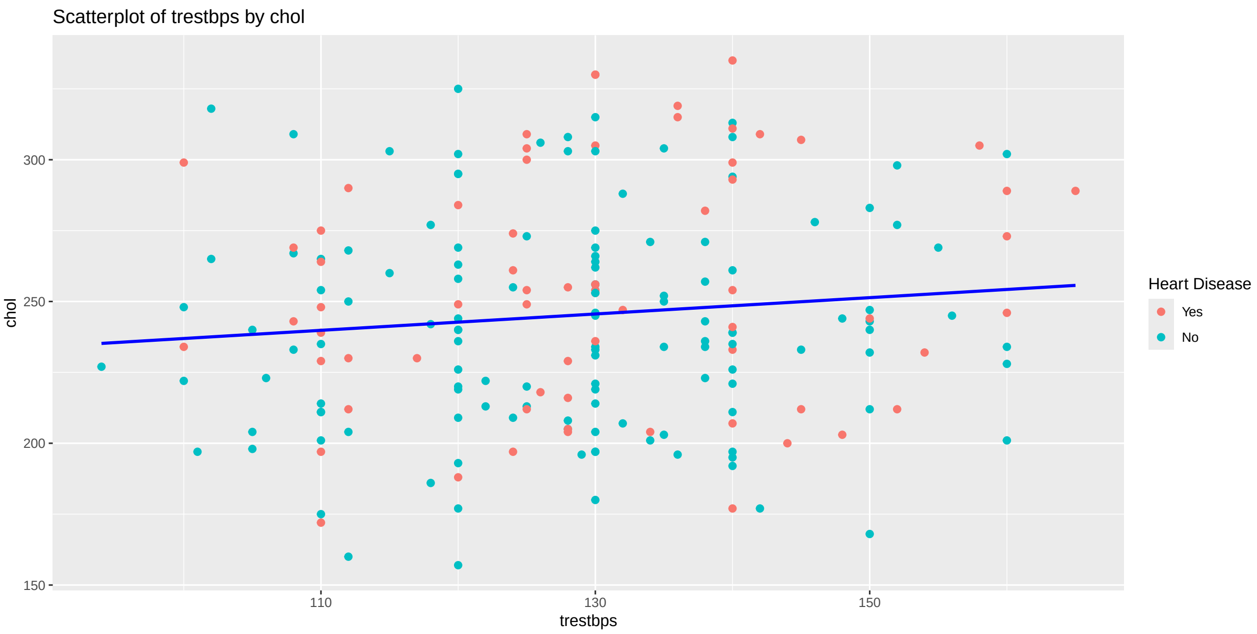
```
HeartDiseaseScatter("age", "thalach")
```



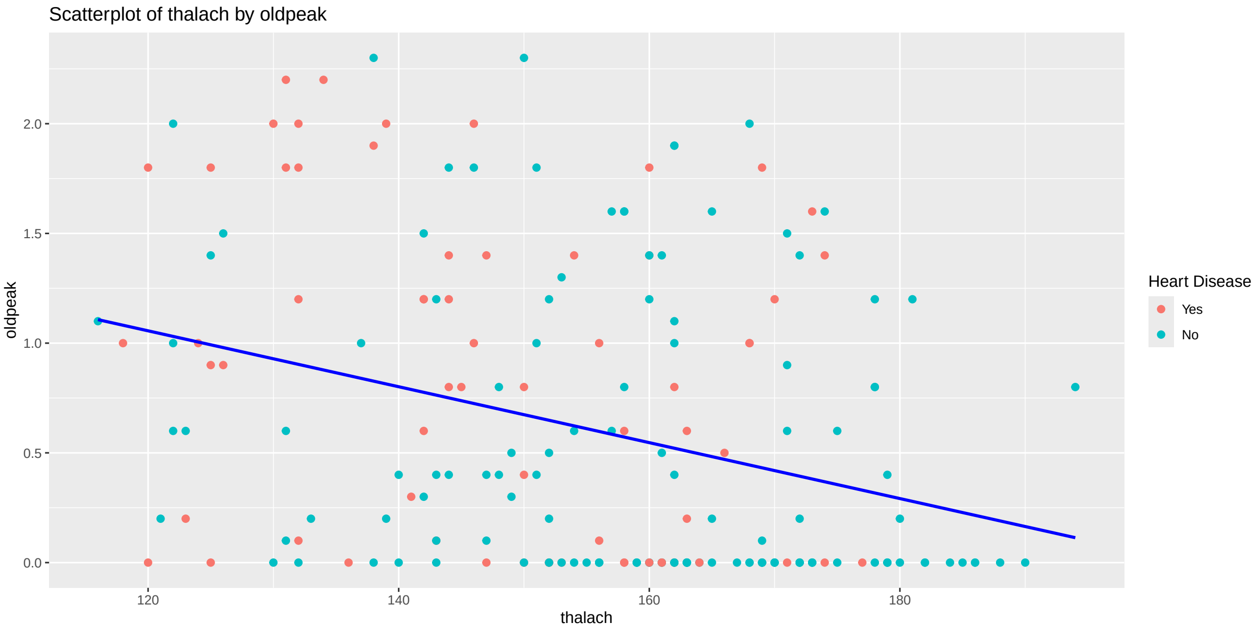
```
HeartDiseaseScatter("chol", "thalach")
```



```
HeartDiseaseScatter("trestbps", "chol")
```

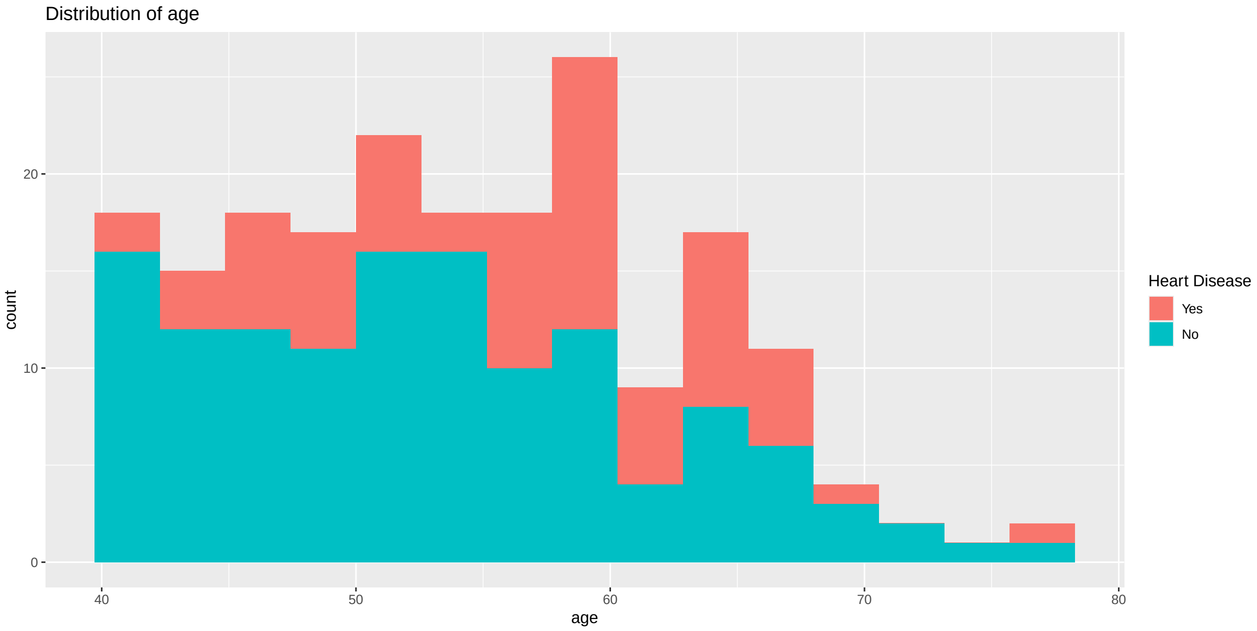


```
HeartDiseaseScatter("thalach", "oldpeak")
```

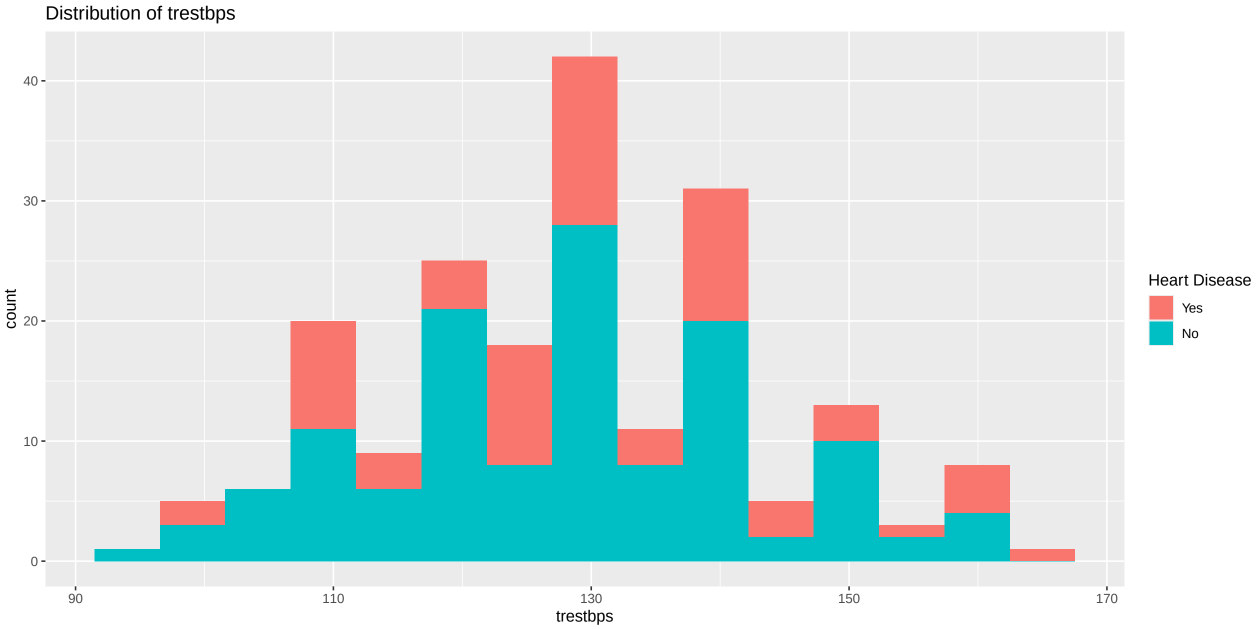


Histograms for Numerical Variables

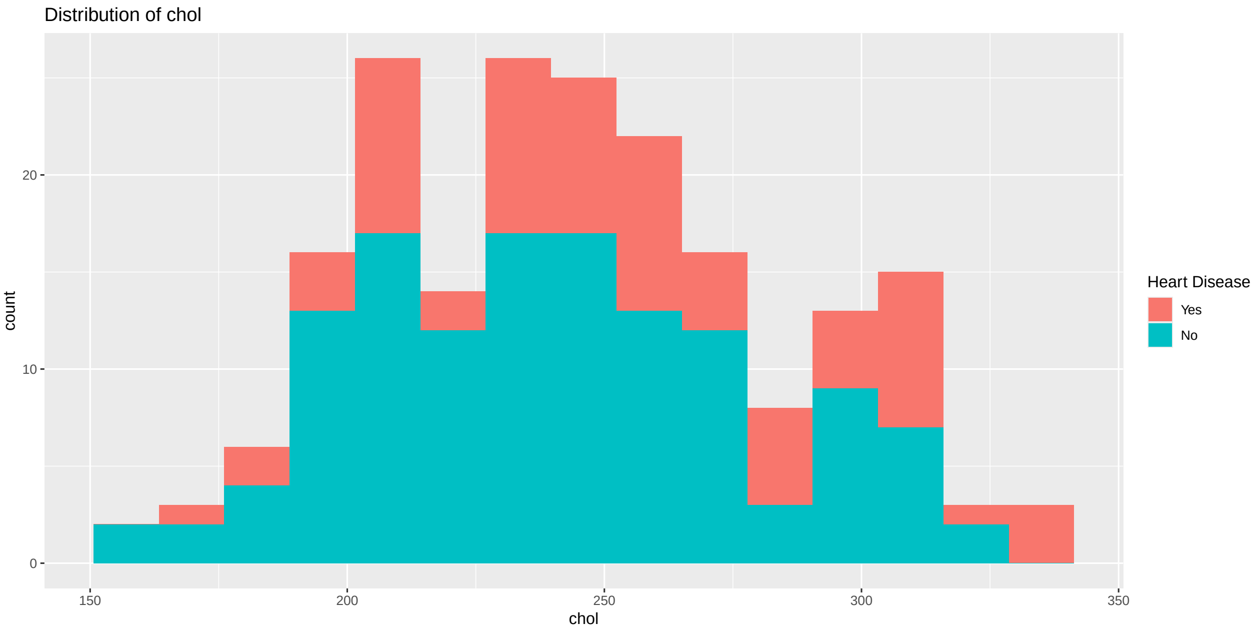
```
HeartDiseaseHist("age")
```



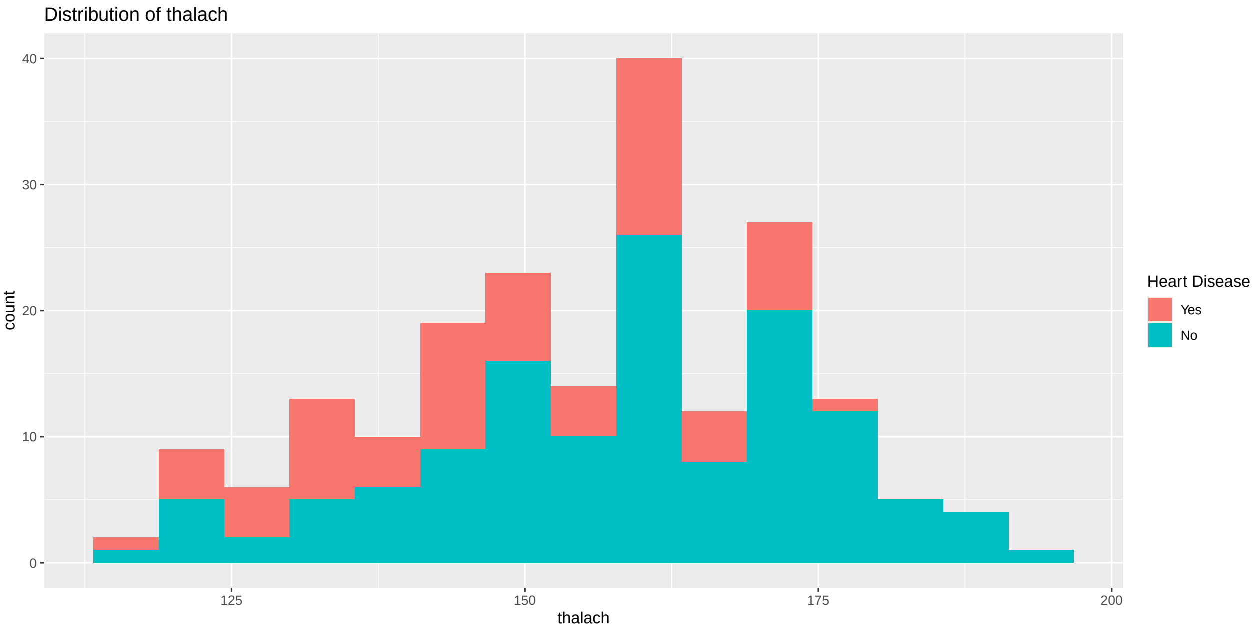
```r
HeartDiseaseHist("trestbps")
```



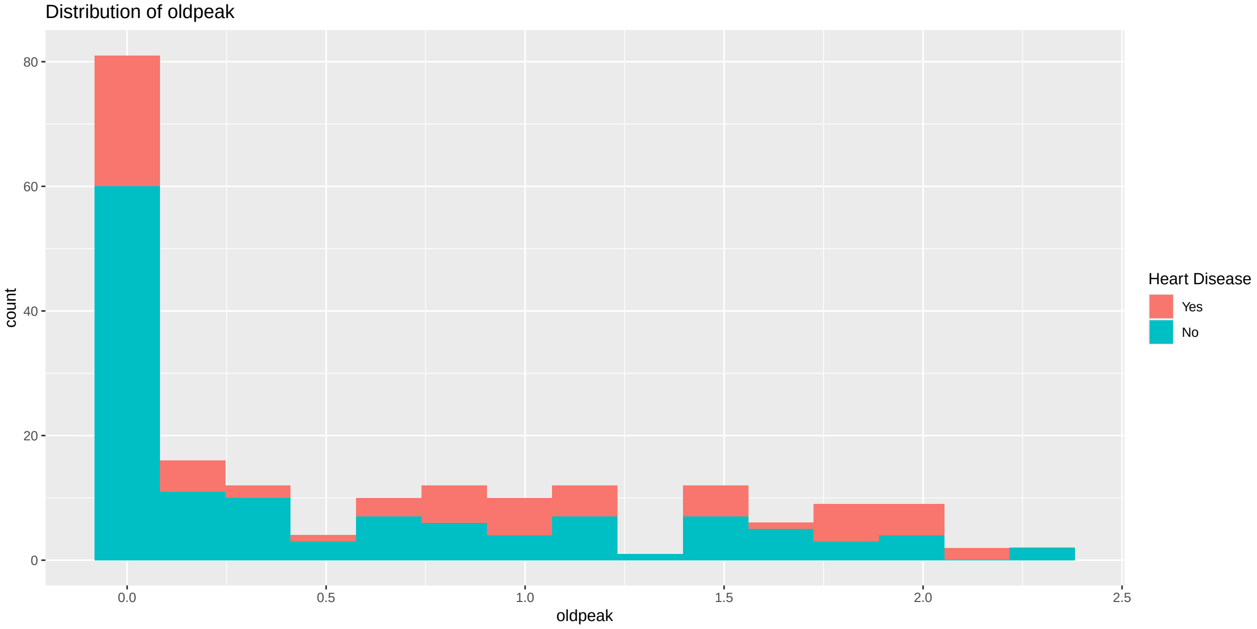
```r
HeartDiseaseHist("chol")
```



```
HeartDiseaseHist("thalach")
```



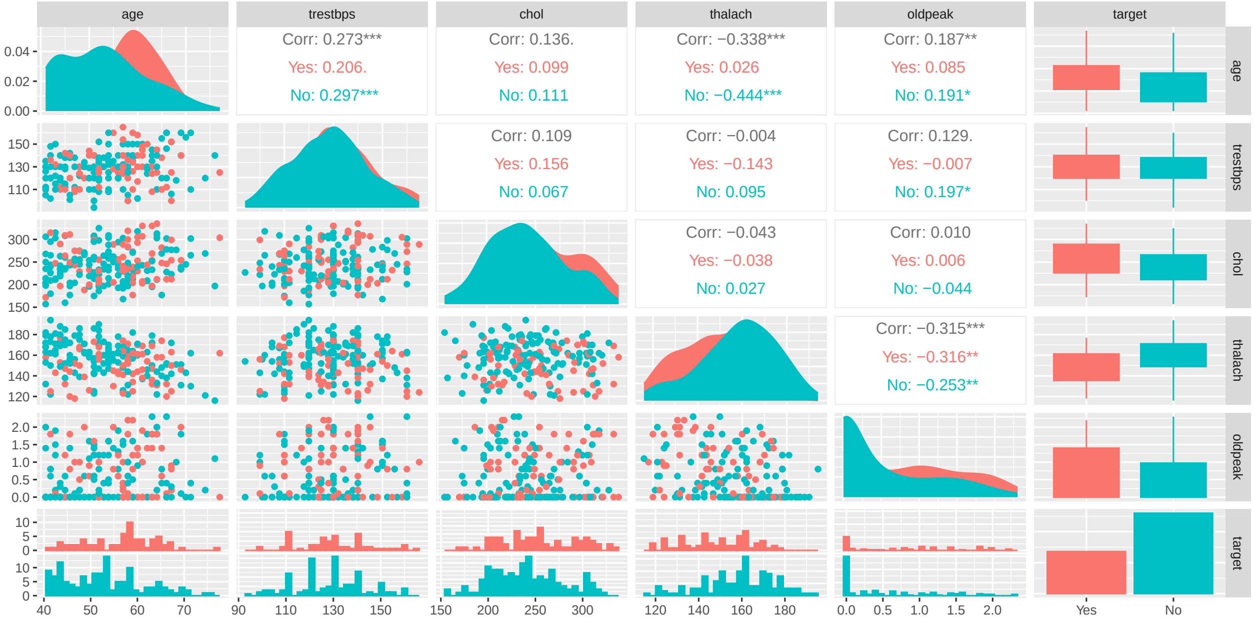
```
HeartDiseaseHist("oldpeak")
```



Pairwise correlation plot for numerical variables

```r
ggpairs(Heart.df[, c("age", "trestbps", "chol",
                     "thalach", "oldpeak", "target")],
        aes(color = target, fill = target))
```

```
## `stat_bin()` using `bins = 30`. Pick better value `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```

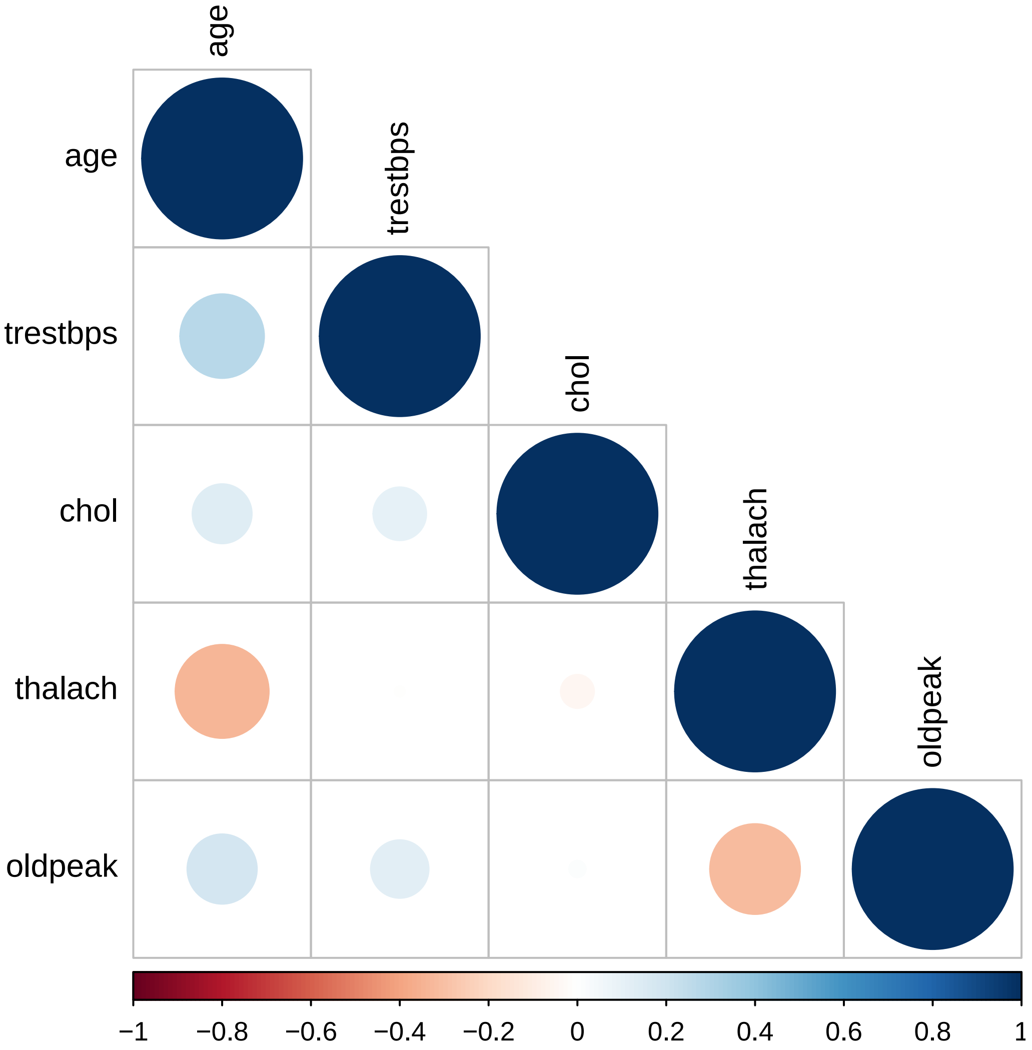


Correlation matrix for numerical variables

```r
# Selecting only continuous variables
continuous_vars <- c("age", "trestbps", "chol", "thalach", "oldpeak")
continuous_data <- Heart.df %>% select(all_of(continuous_vars))

# Calculating correlation matrix
correlation_matrix <- cor(continuous_data)

# Plotting the correlation matrix
corrplot(correlation_matrix, method = "circle",
         type = "lower", tl.col = "black")
```



Modeling

Evaluation

Deployment

Conclusion