```
# install packages and load libraries
library(broom)
```

```
## Warning: package 'broom' was built under R version 4.1.2
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

Step 1: Import data

```
hd_data <- read.csv("Cleveland_hd.csv")

# take a look at the first 5 rows of hd_data
head(hd_data, 5)
```

```
##   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
##   class
## 1     0
## 2     2
## 3     1
## 4     0
## 5     0
```

Step 2: Clean data

More information about variables in this data can be accessed here: https://www.kaggle.com/nareshbhat/health-care-data-set-on-heart-attack-possibility

We can see that "class" is a categorical variable converted into numerical values from 0 to 5. This will confuse our analysis. Therefore, we will convert it once again into binary values with 0 showing no presence of disease and 1 showing the presence of disease.

```
# Get a new column showing binary class outcomes
hd_data <- hd_data %>% mutate(hd = ifelse(class > 0, 1, 0))

# Recode strings
hd_data <- hd_data%>%mutate(hd_labelled = ifelse(hd == 0, "No Disease", "Disease"))

# View data
head(hd_data)
```

```
##   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
##   class hd hd_labelled
## 1     0  0  No Disease
## 2     2  1     Disease
## 3     1  1     Disease
## 4     0  0  No Disease
## 5     0  0  No Disease
## 6     0  0  No Disease
```

Step 3: Understand individual predictors' influence

We first use statistical test to examine the individual relationship between one single independent variable (age, sex and heart rate) and the dependent variable (hd). Because age and heart rate are continuous variables, we will use a t-test which is suited for difference of means test. Meanwhile, because sex is a categorical variable, we choose a chi-squared test.

```
# between sex and presence of heart disease
hd_sex <- chisq.test(hd_data$sex, hd_data$hd)

# age
hd_age <- t.test(hd_data$age ~ hd_data$hd)

# max heart rate
hd_heartrate <- t.test(hd_data$thalach ~ hd_data$hd)

hd_sex
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  hd_data$sex and hd_data$hd
## X-squared = 22.043, df = 1, p-value = 2.667e-06
```

```
hd_age
```

```
##
##  Welch Two Sample t-test
```

```
##
## data:  hd_data$age by hd_data$hd
## t = -4.0303, df = 300.93, p-value = 7.061e-05
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -6.013385 -2.067682
## sample estimates:
## mean in group 0 mean in group 1
##        52.58537        56.62590
```

```
hd_heartrate
```

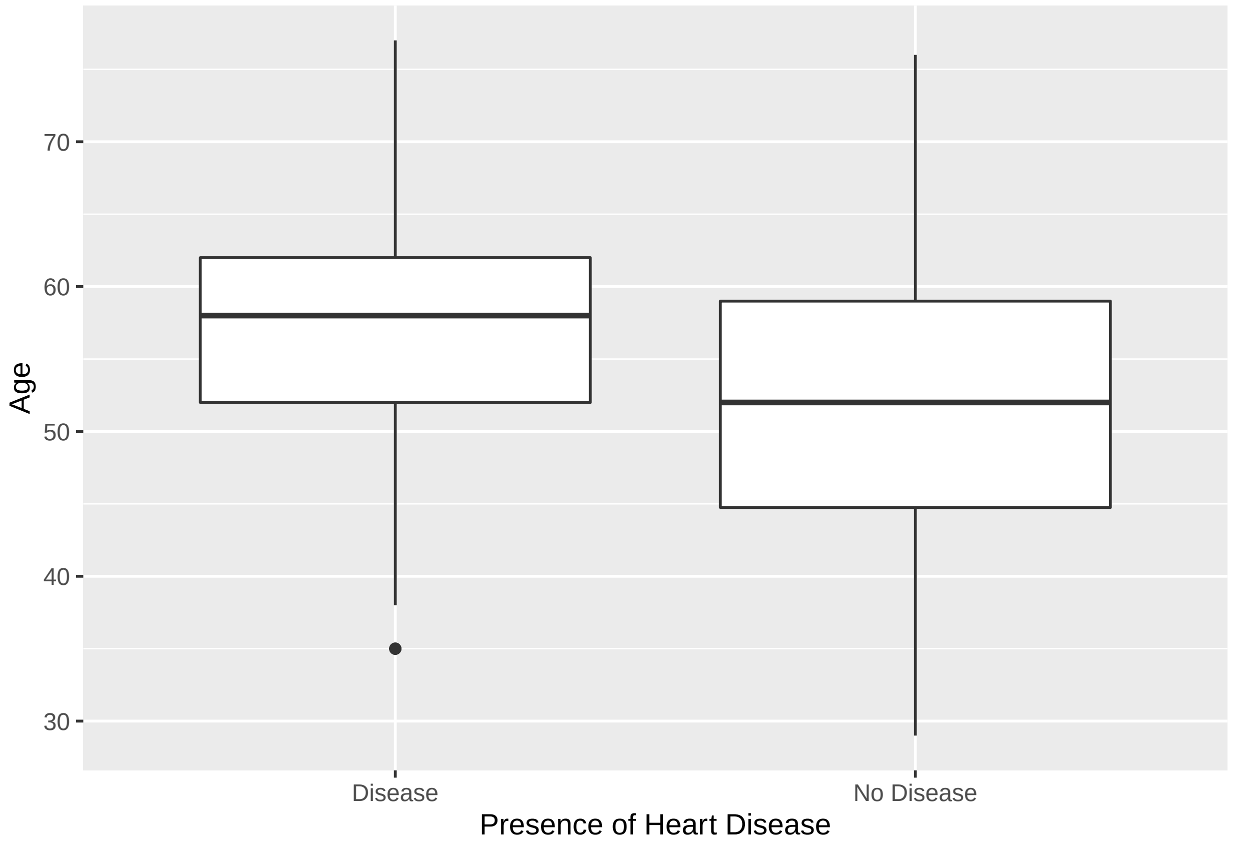
```
##
##  Welch Two Sample t-test
##
## data:  hd_data$thalach by hd_data$hd
## t = 7.8579, df = 272.27, p-value = 9.106e-14
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  14.32900 23.90912
## sample estimates:
## mean in group 0 mean in group 1
##         158.378         139.259
```

We can see that all these three variables are very significantly associated with the outcome because p-values in all tests are very small.

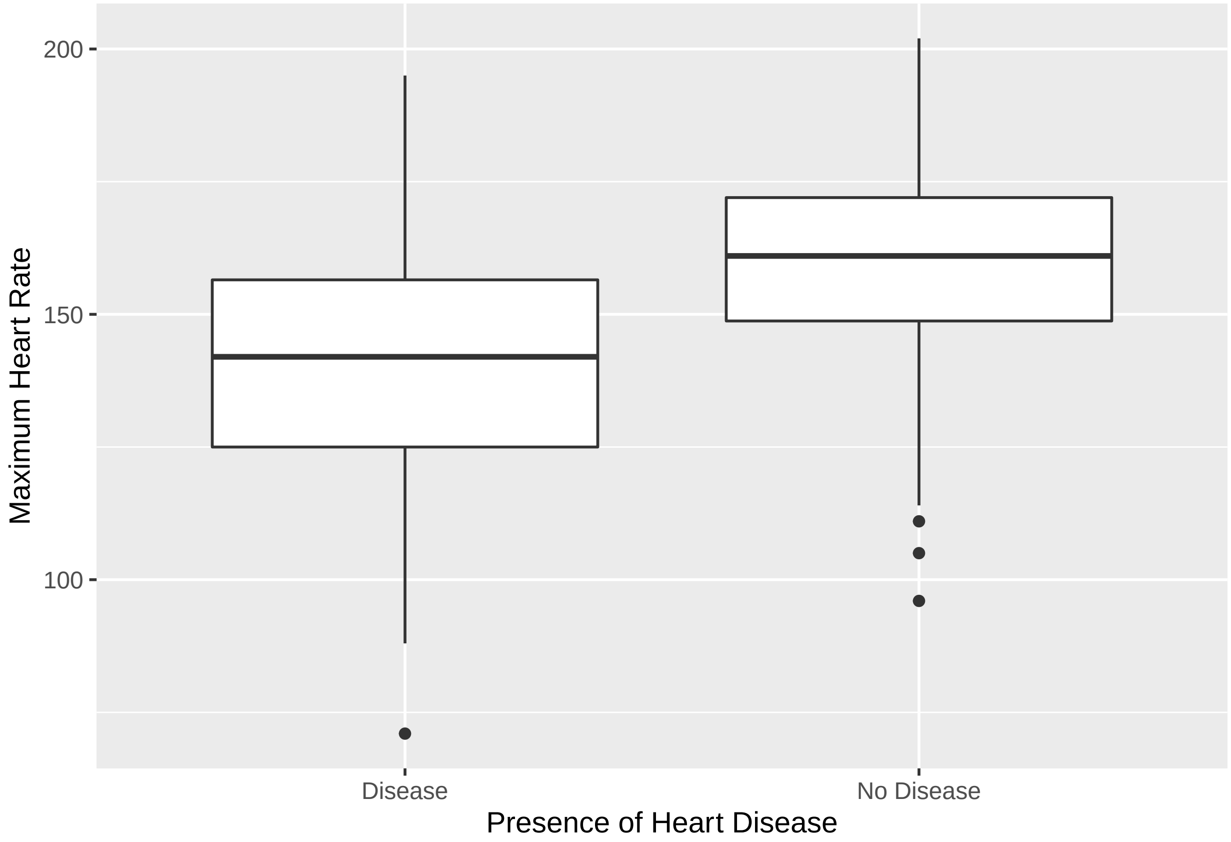
Step 4: Visualize the associations

We will now draw a boxplot with the calculated above confidence interval of the association between presence of heart disease and two independent variabes (age and heart rate). For sex, we will visually show the proportion of binary outcomes. Note that we still explore individual relationships.

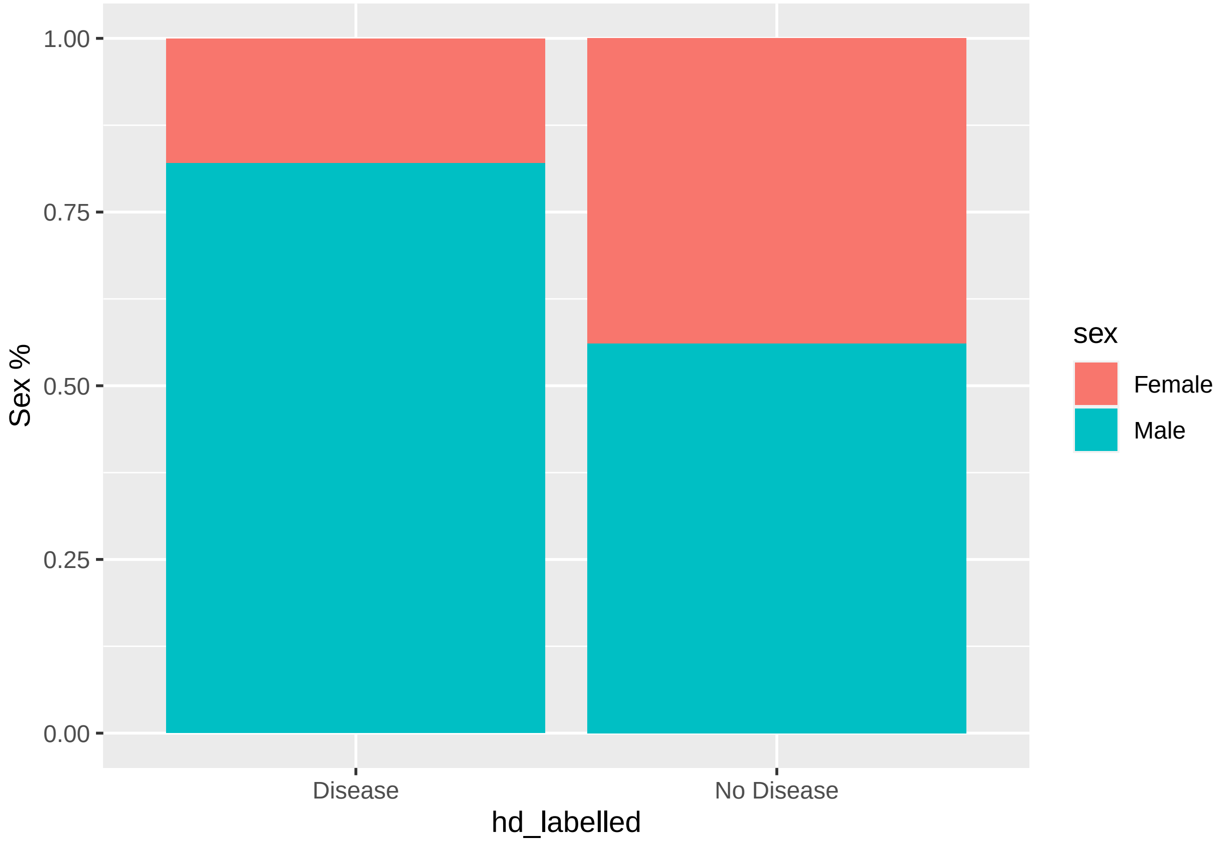
```
ggplot(data = hd_data, aes(x = hd_labelled,y = age)) + geom_boxplot() + labs(x = "Presence of Heart Dis
```



```
ggplot(data = hd_data,aes(x = hd_labelled, y = thalach)) + geom_boxplot() + labs(x = "Presence of Heart
```



```
# Convert sex into factor for visualization
hd_data <- hd_data %>% mutate(sex = factor(sex, levels = 0:1, labels = c("Female", "Male")))
ggplot(data = hd_data,aes(x = hd_labelled, fill = sex)) + geom_bar(position = "fill") + ylab("Sex %")
```



Step 5: Multiple logistic regression

Because we know all three independent variables are significantly associated with the outcome, we now conduct a multiple logistic regression by combining all three variables together to predict the presence of heart disease. We choose logistic regression because our response variable is a binary outcome, not a continuous numerical value. Therefore, linear regression will be less appropriate.

Before constructing the logistic regression model, we now split the data into the training set and test set.

```
# Get the indices for training data
train_ind <- sort(sample(nrow(hd_data), nrow(hd_data)*.5))

# Create the training set
hd_train <- hd_data[train_ind,]

head(hd_train)
```

```
##    age    sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
## 1   63   Male  1      145  233   1       2     150     0     2.3     3  0    6
## 3   67   Male  4      120  229   0       2     129     1     2.6     2  2    7
## 5   41 Female  2      130  204   0       2     172     0     1.4     1  0    3
## 12  56 Female  2      140  294   0       2     153     0     1.3     2  0    3
## 13  56   Male  3      130  256   1       2     142     1     0.6     2  1    6
## 15  52   Male  3      172  199   1       0     162     0     0.5     1  0    7
##    class hd hd_labelled
## 1      0  0  No Disease
## 3      1  1     Disease
## 5      0  0  No Disease
## 12     0  0  No Disease
## 13     2  1     Disease
## 15     0  0  No Disease
```

```
# Create the test set
hd_test <- hd_data[-train_ind,]

head(hd_test)
```

```
##   age    sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
## 2  67   Male  4      160  286   0       2     108     1     1.5     2  3    3
## 4  37   Male  3      130  250   0       0     187     0     3.5     3  0    3
## 6  56   Male  2      120  236   0       0     178     0     0.8     1  0    3
## 7  62 Female  4      140  268   0       2     160     0     3.6     3  2    3
## 8  57 Female  4      120  354   0       0     163     1     0.6     1  0    3
## 9  63   Male  4      130  254   0       2     147     0     1.4     2  1    7
##   class hd hd_labelled
## 2     2  1     Disease
## 4     0  0  No Disease
## 6     0  0  No Disease
## 7     3  1     Disease
## 8     0  0  No Disease
## 9     2  1     Disease
```

```
# construct the model
# use only three independent variables
model <- glm(data = hd_train, hd ~ age + sex + thalach, family = "binomial")

# extract the model summary
summary(model)
```

```
##
## Call:
## glm(formula = hd ~ age + sex + thalach, family = "binomial",
##     data = hd_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0976  -0.9186  -0.4910   0.9150   2.0353
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.041200   2.206261   2.285 0.022316 *
## age         -0.002289   0.023243  -0.098 0.921540
## sexMale      1.436437   0.407601   3.524 0.000425 ***
## thalach     -0.040462   0.009366  -4.320 1.56e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 208.79  on 150  degrees of freedom
## Residual deviance: 169.83  on 147  degrees of freedom
## AIC: 177.83
##
## Number of Fisher Scoring iterations: 4
```

From the summary table, we can see that age is no longer a statistically significant predictor. Meanwhile, sex and heart rate still have very small p-value.

Step 6: Odds Ratio and 95% Confidence Interval

```
# tidy up the coefficient table
tidy_m <- tidy(model)
tidy_m
```

```
## # A tibble: 4 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)  5.04      2.21       2.28   0.0223
## 2 age         -0.00229   0.0232    -0.0985 0.922
## 3 sexMale      1.44      0.408      3.52   0.000425
## 4 thalach     -0.0405    0.00937   -4.32   0.0000156
```

```
# calculate OR
tidy_m$OR <- tidy_m$estimate

# calculate 95% CI and two bounds
tidy_m$lower_CI <- exp(tidy_m$estimate - 1.96 * tidy_m$std.error)
tidy_m$upper_CI <- exp(tidy_m$estimate + 1.96 * tidy_m$std.error)

tidy_m
```

```
## # A tibble: 4 x 8
##   term        estimate std.error statistic   p.value       OR lower_CI  upper_CI
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept)  5.04      2.21       2.28   0.0223     5.04       2.05  11678.
## 2 age         -0.00229   0.0232    -0.0985 0.922     -0.00229    0.953     1.04
## 3 sexMale      1.44      0.408      3.52   0.000425   1.44       1.89      9.35
## 4 thalach     -0.0405    0.00937   -4.32   0.0000156 -0.0405     0.943     0.978
```

Step 7: Prediction

Now, we will input values into the model and make predictions. We also apply a decision rule to convert predicted probabilities into binary outcomes. Later on, we will calculate the misclassification error rate to evaluate model accuracy.

```
# apply the model to the testing data, predict the probability of the presence of heart disease
predicted_values <- predict(model, newdata = hd_test, type = "response")

# to convert predicted probabilities into binary outcomes, we apply an arbitrary decision rule
# we consider 0.5 as the threshold here
predicted_values_binary <- ifelse(predicted_values >= 0.5, 1, 0)

# create a table to easily compare observed and predicted outcomes
matrix <- table(predicted_values_binary, hd_test$hd)
matrix
```

```
##
## predicted_values_binary  0  1
##                       0 65 23
##                       1 19 45
```

From this matrix, we can easily calculate the sensitivity and specificity of our model. However, now we will just calculate the misclassification error of our model

Step 8: Evaluate model accuracy

```
accuracy <- sum(diag(matrix)) / sum(matrix)

misclassification_error <- 1 - accuracy

accuracy
```

```
## [1] 0.7236842
```

```
misclassification_error
```

```
## [1] 0.2763158
```

From results above, I conclude that our model can predict accurately 71.71% of all observations. In other words, out misclassification rate is 0.282 or 28.2%

Overall, this is not a pretty good model but there is still room for improvement. We can try different independent variables as well as include interaction terms to understand further the association between predictors and the outcome - the presence of heart disease.