PSTAT 131- FINAL PROJECT _YIFAN XU

Yifan Xu

2022-11-27

Contents

INTRODUCTION

The purpose of this project is to generate a model that will predict the possibility of people have heart disease according to his physical data.

What is heart failure?

Heart failure is a heart condition in which the heart cannot pump sufficient blood around the body. This may be due to the heart not filling with enough blood or it is too weak to pump correctly. Despite the name, it does not refer to the heart stopping.

Heart failure usually develops gradually. The heart muscle becomes weaker and has trouble pumping blood to nourish the cells in your body. It is a chronic condition that gradually gets worse. Sometimes, heart failure comes on suddenly after a heart attack as it weakens the heart's pumping ability. In acute heart failure, the symptoms are usually severe at first. Heart failure symptoms may include coughing up white, pink, or foamy mucus, fatigue and weakness, irregular heartbeat, nausea, lack of appetite, shortness of breath when lying down or exerting energy or fluid retention in the abdomen or extremities. Symptoms may come and go, or they may persist over a period. If new symptoms develop or existing symptoms worsen, it may mean that heart failure is getting worse or that treatment isn't effective.

Why might this model be useful?

People rarely ever worry about heart disease in their lives, especially young people. But in fact, heart disease is one of the major diseases that cause human death in the world today. As a student, it was always essential for me to stay up late whenever the end of the semester came. But after each late night, my heart would always feel uncomfortable. This made me start to pay attention to the problem of heart disease. This mod can make us understand the connection between our body's information and heart disease, but certain abnormalities in our body's data can alert us to the presence of heart disease.

Loading Data and Packages

This project uses on a data set of the Heart Failure Prediction, it records information of 5 heart data sets are combined over 11 common features which makes it the largest heart disease dataset available so far for research purposes.

I get this data set from kaggle and the website ishttps://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction. Here are some of the key variables that are helpful to be aware of for this report:

1.Age: age of the patient [years]

2.Sex: sex of the patient [M: Male, F: Female]

3.ChestPainType: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]

4.RestingBP: resting blood pressure [mm Hg]

5.Cholesterol: serum cholesterol [mm/dl]

6.FastingBS: fasting blood sugar [1: if FastingBS $>$ 120 mg/dl, 0: otherwise]

7.RestingECG: resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of $>$ 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]

8.MaxHR: maximum heart rate achieved [Numeric value between 60 and 202]

9ExerciseAngina: exercise-induced angina [Y: Yes, N: No]

10.Oldpeak: oldpeak = ST [Numeric value measured in depression]

11.ST_Slope: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]

12.HeartDisease: output class [1: heart disease, 0: Normal]

```
Heart_origin <- read.csv("data/unprocessed/heart.csv")
head(Heart_origin)
```

```
##   Age Sex ChestPainType RestingBP Cholesterol FastingBS RestingECG MaxHR
## 1  40   M           ATA       140         289         0     Normal   172
## 2  49   F           NAP       160         180         0     Normal   156
## 3  37   M           ATA       130         283         0         ST    98
## 4  48   F           ASY       138         214         0     Normal   108
## 5  54   M           NAP       150         195         0     Normal   122
## 6  39   M           NAP       120         339         0     Normal   170
##   ExerciseAngina Oldpeak ST_Slope HeartDisease
## 1              N     0.0       Up            0
## 2              N     1.0     Flat            1
## 3              N     0.0       Up            0
## 4              Y     1.5     Flat            1
## 5              N     0.0       Up            0
## 6              N     0.0       Up            0
```

DATA CLEANING

The data set of heart failure that I chosen is tidy, a few different cleaning steps were necessary before the split occurred: there are some observation that do not have Cholesterol values and Restign BP values that shows 0.These skew our results, and thus we are removing them.

```
Heart_origin = Heart_origin %>%
  filter(Cholesterol != 0) %>%
  filter(RestingBP != 0 )
head(Heart_origin)
```

```
##   Age Sex ChestPainType RestingBP Cholesterol FastingBS RestingECG MaxHR
## 1  40   M           ATA       140         289         0     Normal   172
## 2  49   F           NAP       160         180         0     Normal   156
## 3  37   M           ATA       130         283         0         ST    98
## 4  48   F           ASY       138         214         0     Normal   108
## 5  54   M           NAP       150         195         0     Normal   122
## 6  39   M           NAP       120         339         0     Normal   170
##   ExerciseAngina Oldpeak ST_Slope HeartDisease
## 1              N     0.0       Up            0
## 2              N     1.0     Flat            1
## 3              N     0.0       Up            0
## 4              Y     1.5     Flat            1
## 5              N     0.0       Up            0
## 6              N     0.0       Up            0
```

After data cleaning, the total number of observations change from 918 observations to 746 observation.

Also, we need to change those character variables to factor variables.

```
Heart = Heart_origin %>%
  mutate(Sex = factor(Sex),
         ChestPainType = factor(ChestPainType),
         FastingBS = factor (FastingBS),
         RestingECG = factor(RestingECG),
         ExerciseAngina = factor(ExerciseAngina),
         ST_Slope = factor(ST_Slope),
         HeartDisease = factor(HeartDisease))
head(Heart)
```

```
##   Age Sex ChestPainType RestingBP Cholesterol FastingBS RestingECG MaxHR
## 1  40   M           ATA       140         289         0     Normal   172
## 2  49   F           NAP       160         180         0     Normal   156
## 3  37   M           ATA       130         283         0         ST    98
## 4  48   F           ASY       138         214         0     Normal   108
## 5  54   M           NAP       150         195         0     Normal   122
## 6  39   M           NAP       120         339         0     Normal   170
##   ExerciseAngina Oldpeak ST_Slope HeartDisease
## 1              N     0.0       Up            0
## 2              N     1.0     Flat            1
## 3              N     0.0       Up            0
## 4              Y     1.5     Flat            1
## 5              N     0.0       Up            0
## 6              N     0.0       Up            0
```

After data cleaning, all the types of variable are appropriate.

```
write_csv(Heart, file = "data/processed/processed_data.csv")
```

DATA SPLIT

The data was split in a 80% training, 20% testing split. Stratified sampling was used as the difficulty distribution was skewed.

```
set.seed(100)
Heart_split <- Heart %>%
  initial_split(prop = 0.8, strata = HeartDisease)

Heart_train <- training(Heart_split)
Heart_test <- testing(Heart_split)
```

After data split, the training data set has about 596 observations and the testing data set has just under 150 observations.

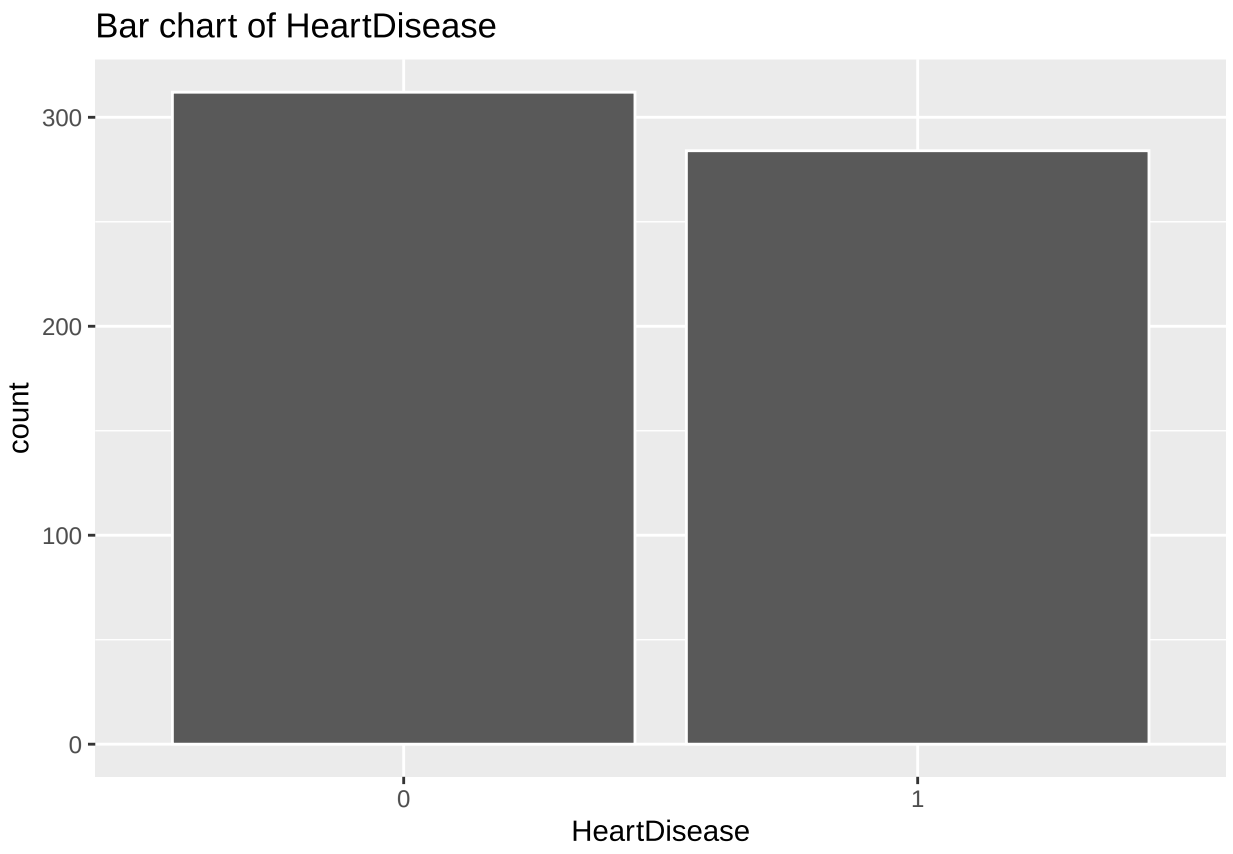
Exploratory Data Analysis

This entire exploratory data analysis will be based only on the training set, which has 596 observations. Each observation represents a single class. I will disscuss some inporant predictors in this data set and find out how they relate to the response.

HeartDisease

First of all, let's see response: HeaerDisease first.

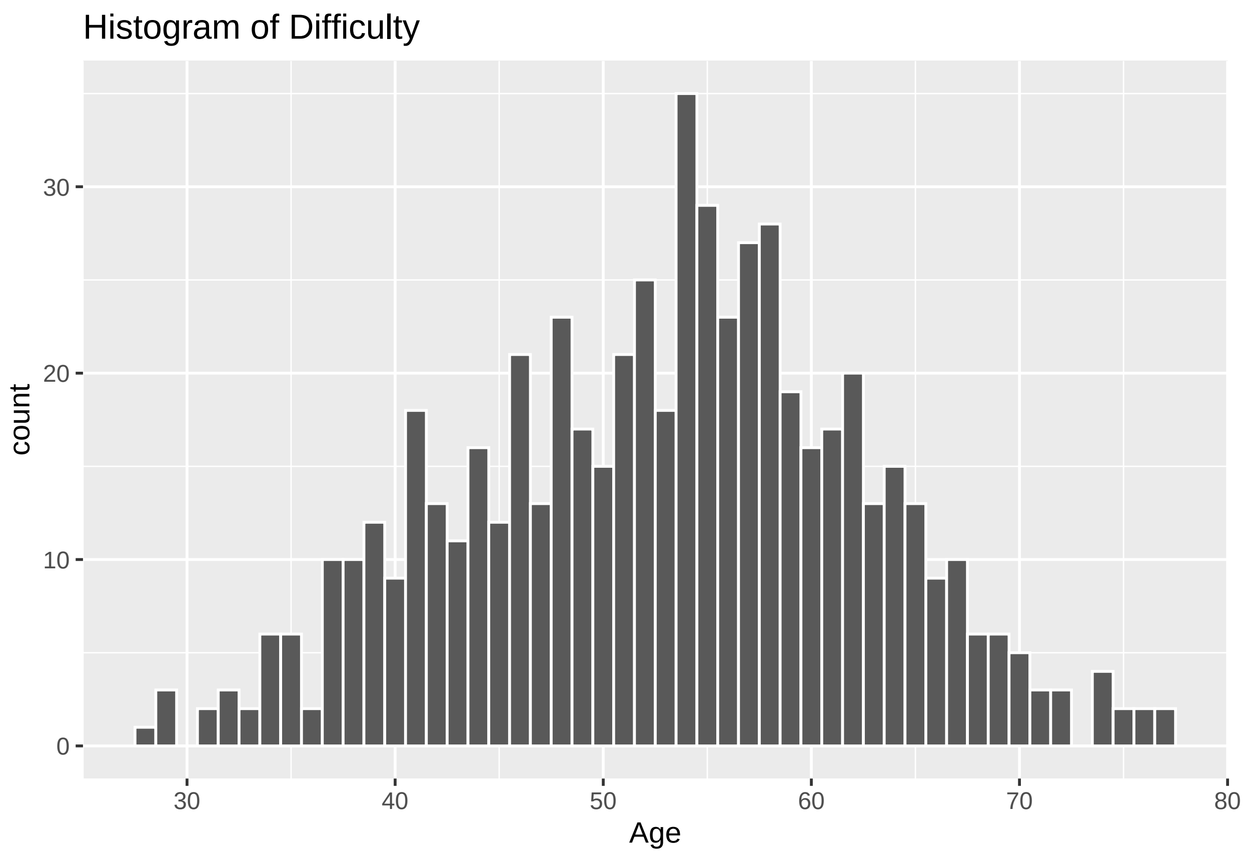
```
ggplot(Heart_train, aes(HeartDisease)) +
  geom_bar( color = "white") +
  labs(
    title = "Bar chart of HeartDisease "
  )
```



According to result of Bar chart of HeartDiseas, we can see that the percentage of people with or without heart disease was very close to 50, which is very good because it makes our analysis more convincing.

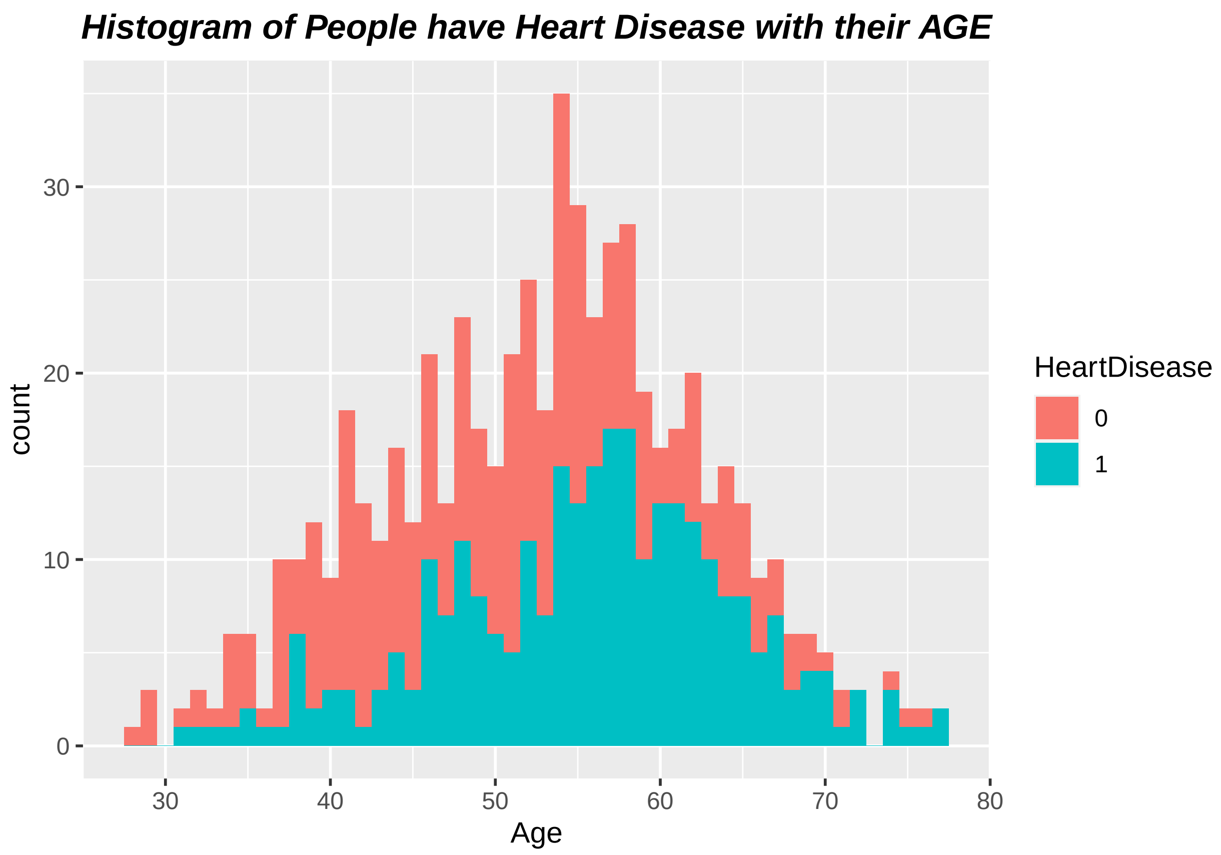
Then, I will discuss the predictors. ### AGE

```
ggplot(Heart_train, aes(Age)) +
  geom_histogram(bins = 50, color = "white") +
  labs(
    title = "Histogram of Difficulty"
  )
```



This histogram looks like a normal distribution which is the result i want. Because it shows that all people in different age period.

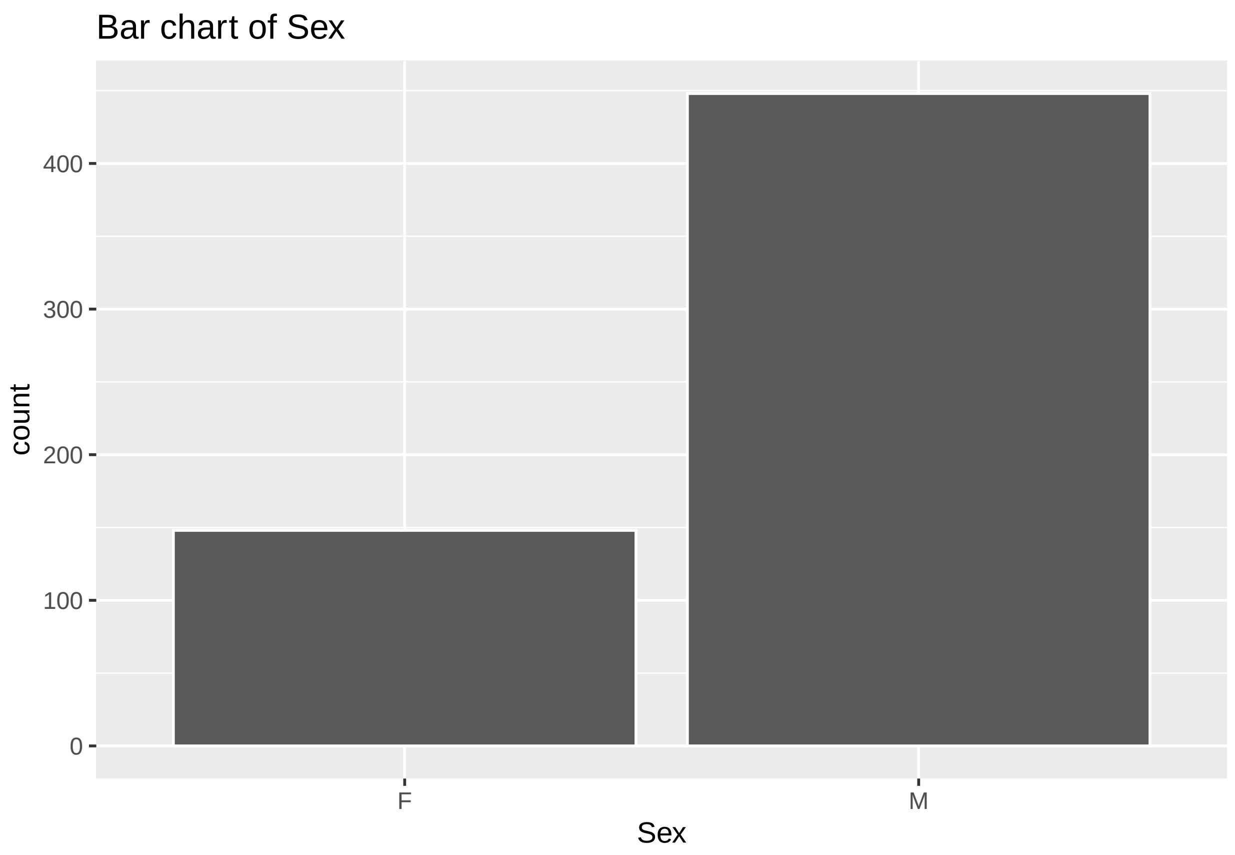
```r
ggplot(Heart_train, aes(x = Age, fill = HeartDisease)) +
  geom_histogram(bins = 50) +
  labs(
    title = "Histogram of People have Heart Disease with their AGE") +
    theme(plot.title = element_text(face = "bold.italic", hjust=0.5))
```



From the result of Histogram of People have Heart Disease with their AGE, we can find that older people have higher rates of heart disease. Especiall when people are older that 60.

SEX

```
ggplot(Heart_train, aes(Sex)) +
  geom_bar( color = "white") +
  labs(
    title = "Bar chart of Sex"
  )
```

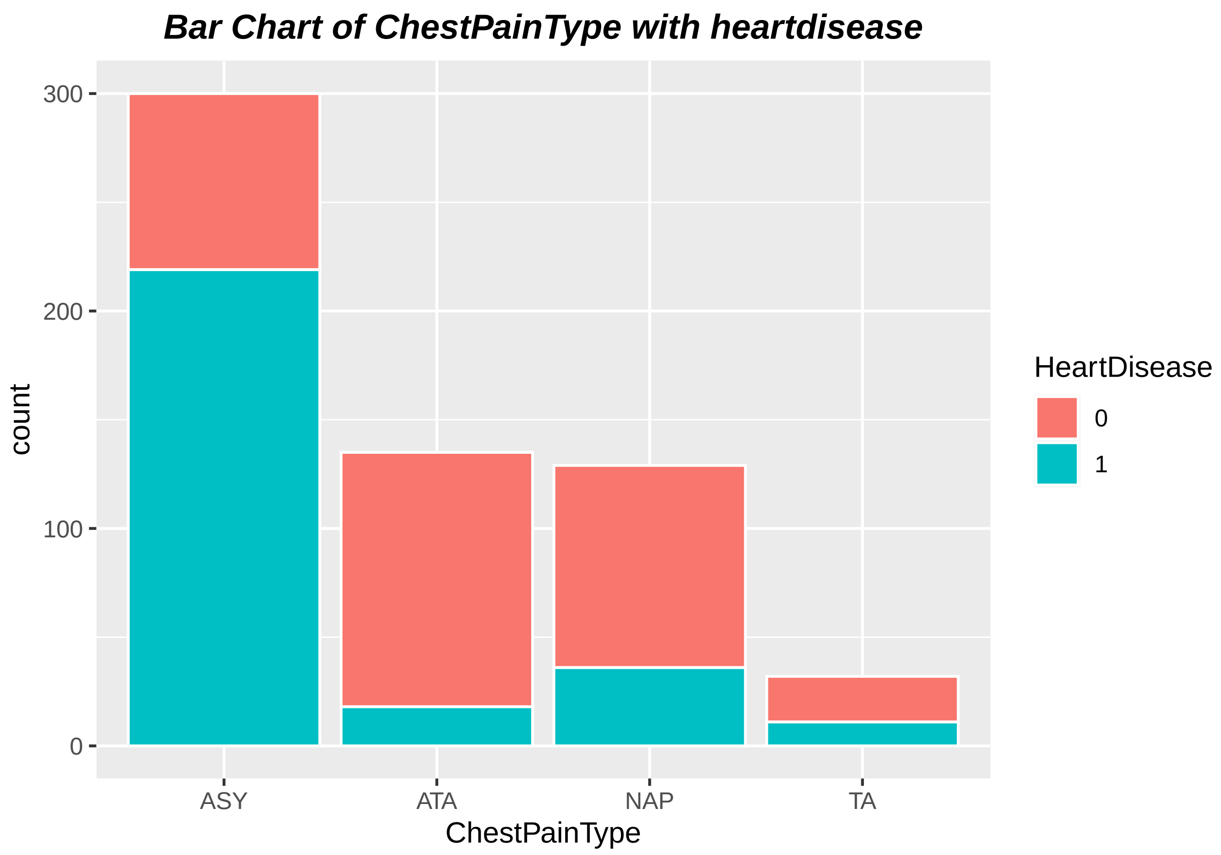


In the training data set, the distribution of Male and Female are not equal or close, so we can not analyse the effect of sex on heart disease.

ChestPainType

For ChestPainType data, there 4 different types: [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic].

```
ggplot(Heart_train, aes(x = ChestPainType, fill = HeartDisease)) +
  geom_bar( color = "white") +
  labs(
    title = "Bar Chart of ChestPainType with heartdisease")+
  theme(plot.title = element_text(face = "bold.italic", hjust=0.5))
```

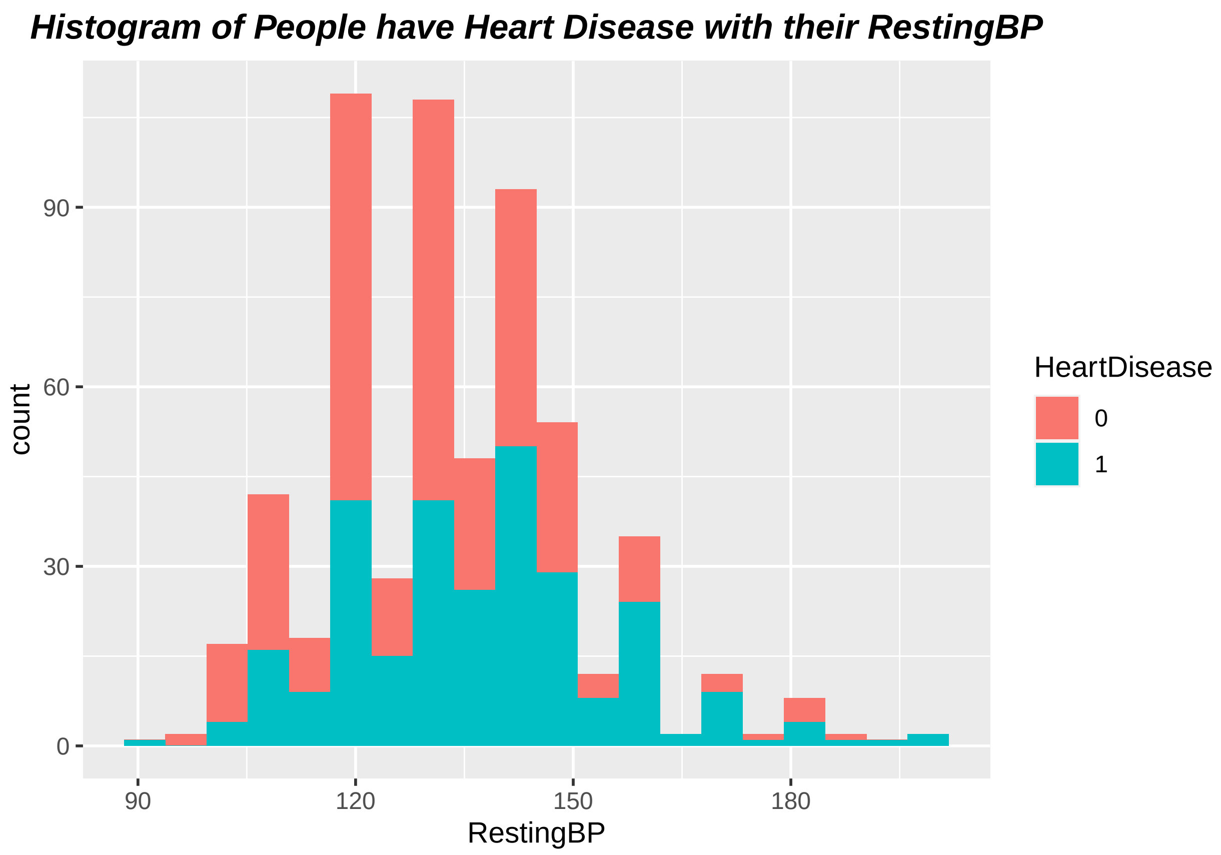


According to the Bar Chart of ChestPainType have heart disease, Only Asymptomatic type shows that has more than 50 percent have heart disease. This data value is far more than the other three types of chest pain.

Resting BP

Resting BP means resting blood pressure [mm Hg].

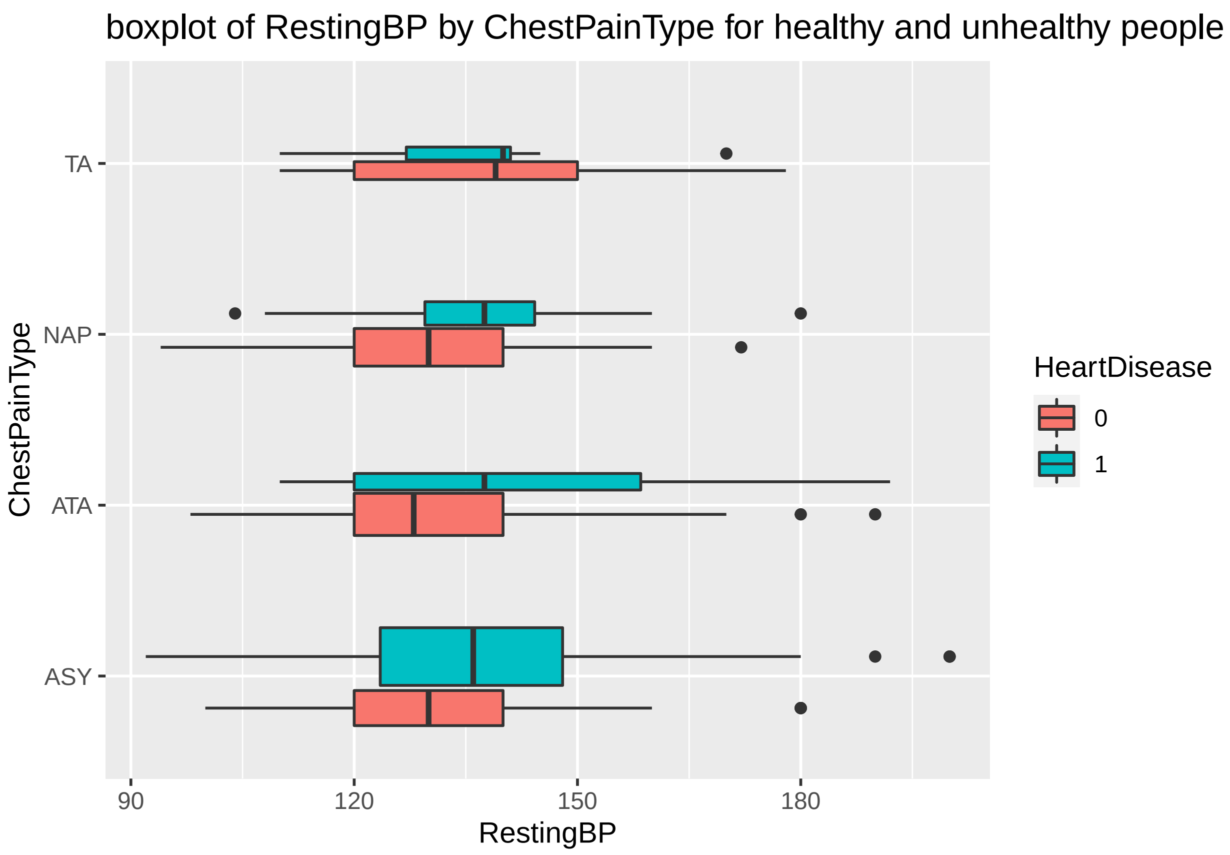
```r
ggplot(Heart_train, aes(x = RestingBP, fill = HeartDisease)) +
  geom_histogram(bins = 20) +
  labs(
    title = "Histogram of People have Heart Disease with their RestingBP") +
    theme(plot.title = element_text(face = "bold.italic", hjust=0.5))
```



According to Histogram of People have Heart Disease with their RestingBP, we noticed that the higher the Restign BP, the higher rate of have HeartDisease.

Then we connect RestingBP and ChestPainType

```
ggplot(Heart_train, aes(ChestPainType, RestingBP,fill = HeartDisease)) +
  geom_boxplot(varwidth = TRUE) +
  coord_flip() +
  labs(
    title = "boxplot of RestingBP by ChestPainType for healthy and unhealthy people ",
    y = "RestingBP",
    x = "ChestPainType"
  )
```

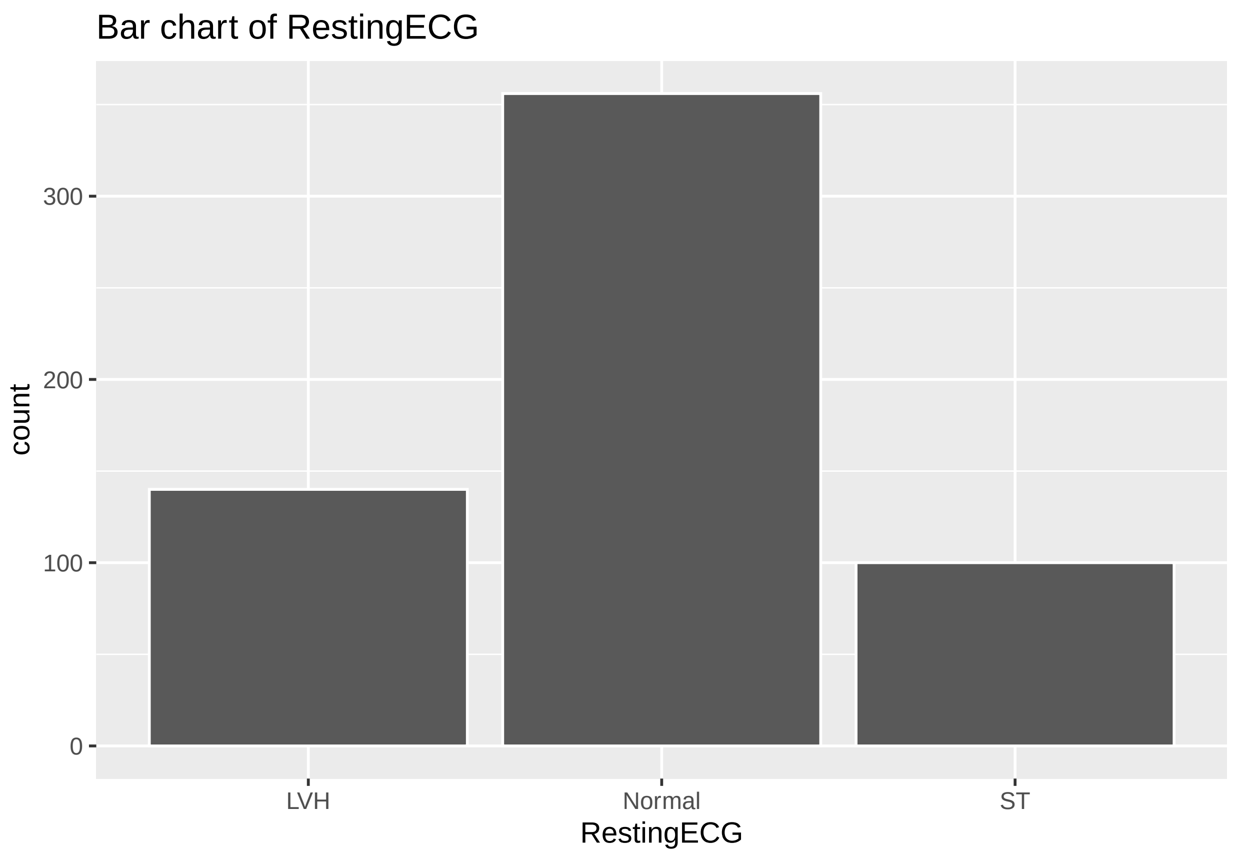


By the box plot above, we find that for all 4 types of chest pain, the mean of people with HeartDisease all higehr than people without HeartDisease. This is the same with our observation that the higher the Restign BP, the higher rate of have HeartDisease. So we can conclude that RestingBP variable is a imporatnt factor of HeartDisease.

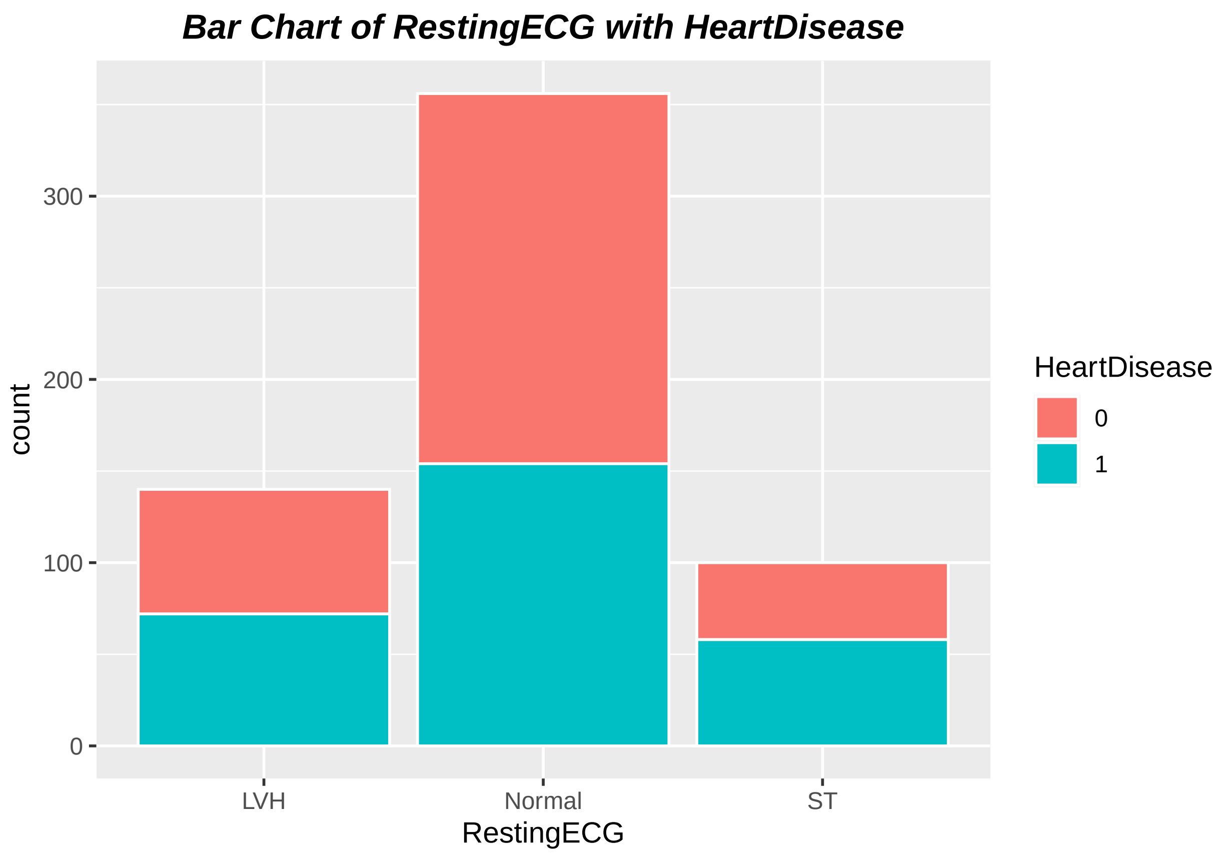
RestingECG

For RestingECG, it means resting electrocardiogram results. It has three kinds of data: [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST 8.elevation or depression of > 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria].

```r
ggplot(Heart_train, aes(RestingECG)) +
  geom_bar( color = "white") +
  labs(
    title = "Bar chart of RestingECG"
  )
```



```r
ggplot(Heart_train, aes(x = RestingECG, fill = HeartDisease)) +
  geom_bar( color = "white") +
  labs(
    title = "Bar Chart of RestingECG with HeartDisease")+
  theme(plot.title = element_text(face = "bold.italic", hjust=0.5))
```



According to Bar Chart of RestingECG with HeartDisease, We see that LVH and ST have higher percentage of having heart disease in data of Resting ECG. Also, for Normal people, they also have probability to have heart disease.

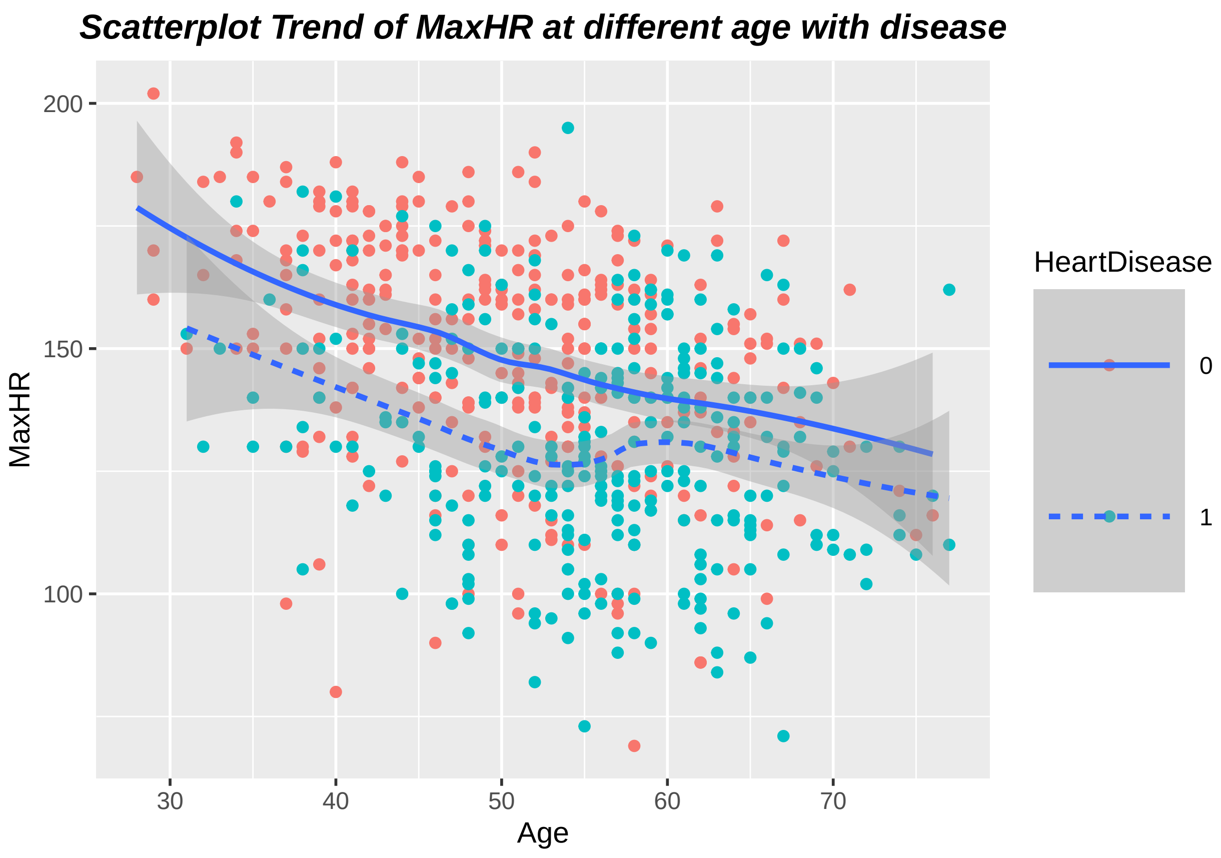
MAXHR

MaxHR means maximum heart rate achieved [Numeric value between 60 and 202]. I think MaxHR is a very important predictor in this data set. So I will associate it with other predictors to analyze.

Firstly, I associate MaxHR with Age.

```
Heart_train %>%
  ggplot(mapping = aes(x = Age, y = MaxHR)) +
  ggtitle("Scatterplot Trend of MaxHR at different age with disease")+
  geom_point(aes(color = HeartDisease)) +
  geom_smooth(aes(linetype = HeartDisease), se = TRUE)+
  labs(x = "Age", y = "MaxHR")+
  theme( legend.key.size = unit(2, 'cm'),
        plot.title = element_text(face = "bold.italic", hjust=0.5))
```

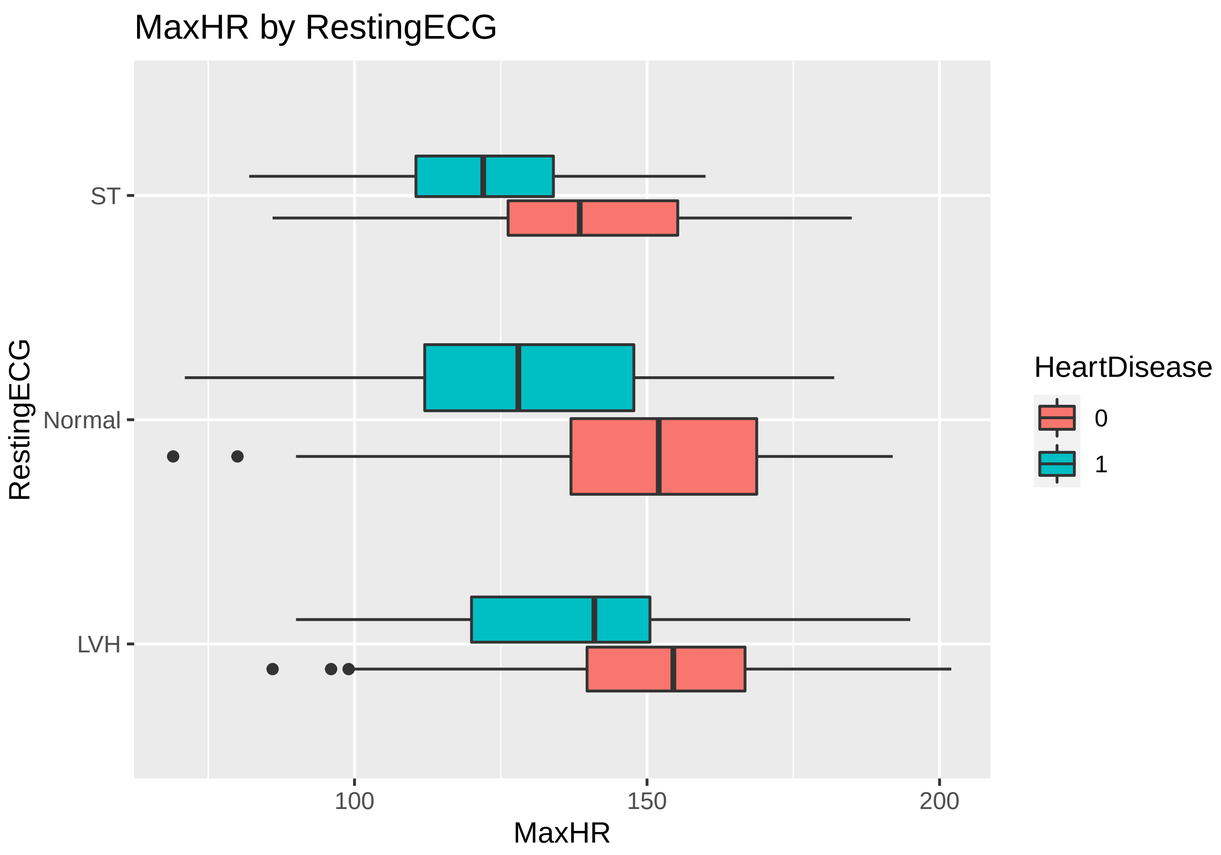
```
## ‘geom_smooth()‘ using method = ’loess’ and formula ’y ~ x’
```



According Scatterplot Trend of MaxHR at different age with disease, we find that for both with heart disease and without heart disease, as we get older, MaxHR decreases, with an inverse relationship between the two.

Then, I also want to know the connection between RestingECG and MaxHR.

```
ggplot(Heart_train, aes(RestingECG, MaxHR,fill = HeartDisease )) +
  geom_boxplot(varwidth = TRUE) +
  coord_flip() +
  labs(
    title = "MaxHR by RestingECG",
    x = "RestingECG",
    y="MaxHR"
  )
```



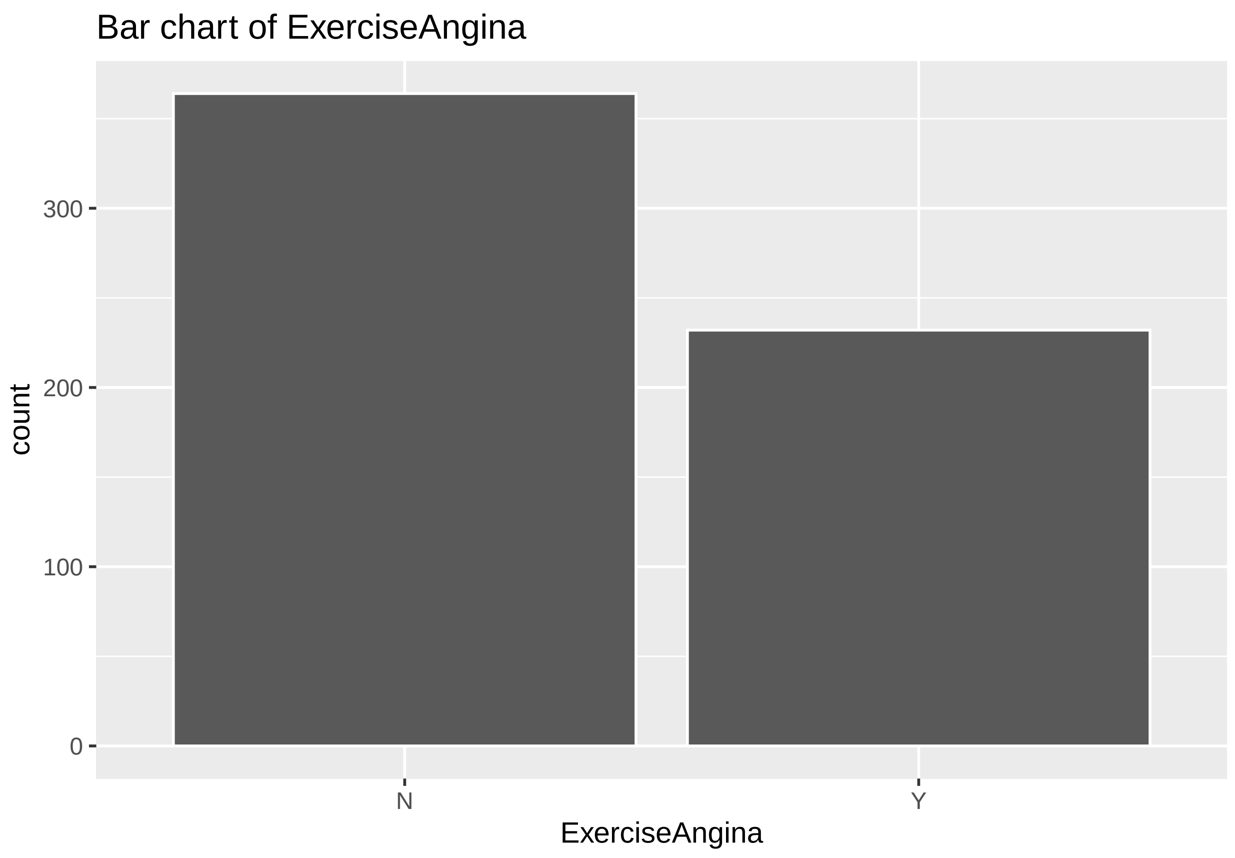
Accrording to boxplot of MaxHR by RestingECG, we find that for alll types of RestingECG, the MaxHR of people with heart disease all lower than people without heart disease. This indicates that the higeer MaxHR you have, the lower chance that you will have heart disease.

ExerciseAngina

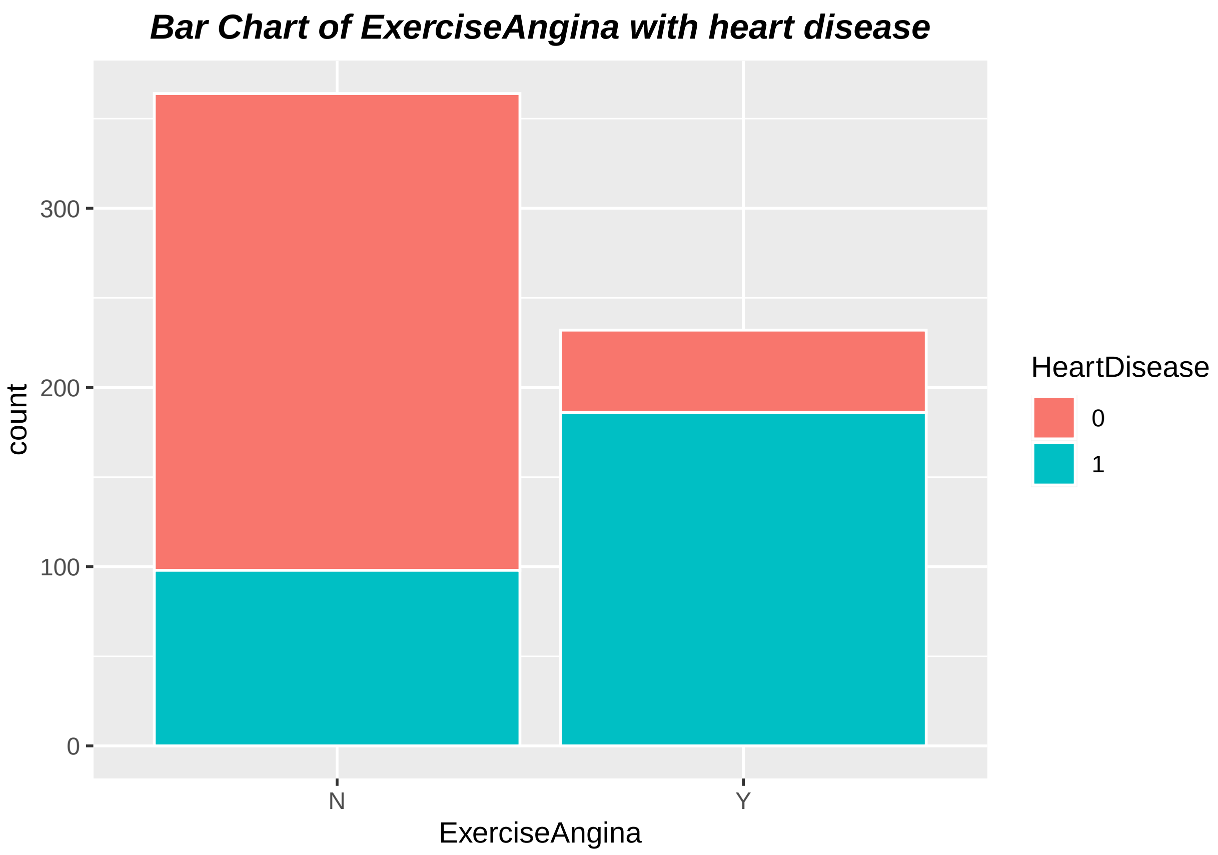
Predictor ExerciseAngina means exercise-induced angina.

We all know that people with heart disease are not suitable for high intensity exercise. So I think ExerciseAngina will be a very interesrting predictor.

```
ggplot(Heart_train, aes(ExerciseAngina)) +
  geom_bar( color = "white") +
  labs(
    title = "Bar chart of ExerciseAngina"
  )
```



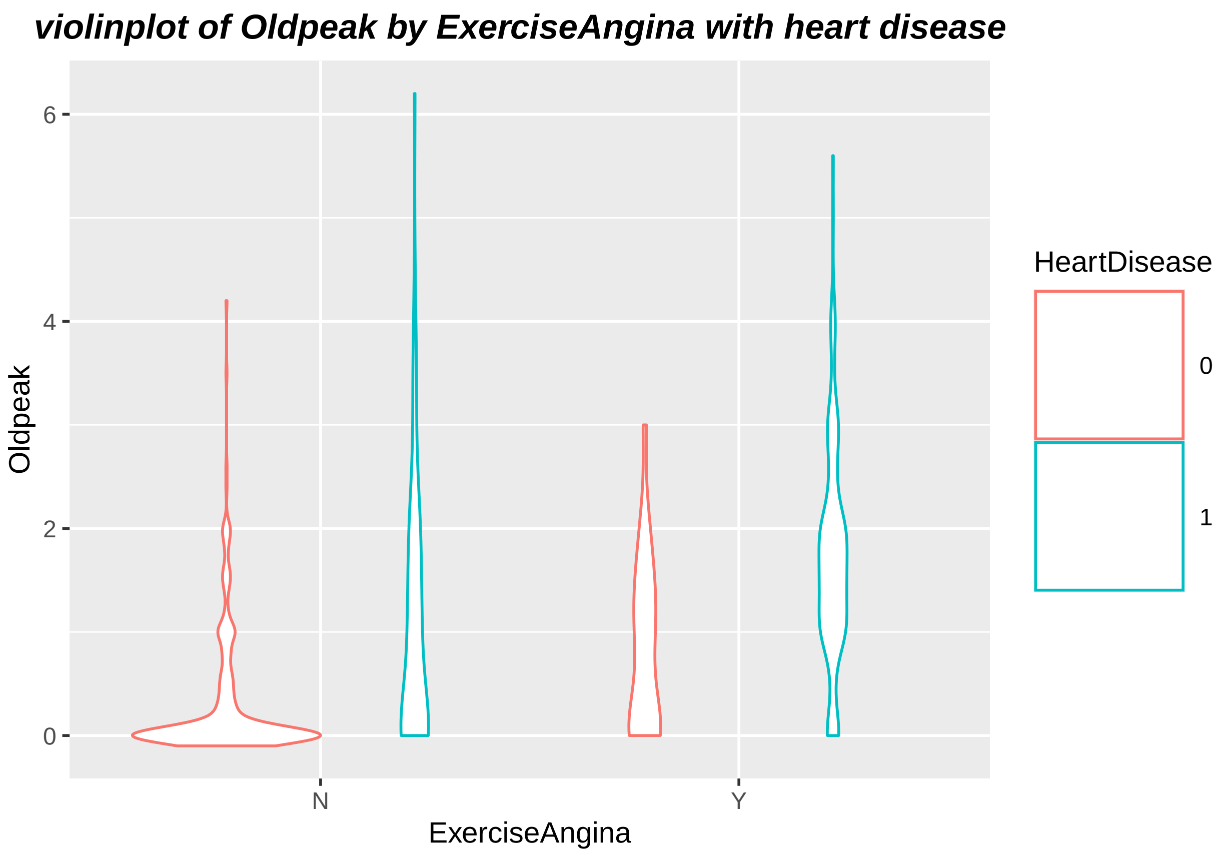
```
ggplot(Heart_train, aes(x = ExerciseAngina, fill = HeartDisease)) +
 geom_bar( color = "white") +
 labs(
   title = "Bar Chart of ExerciseAngina with heart disease")+
 theme(plot.title = element_text(face = "bold.italic", hjust=0.5))
```



According to Bar Chart of ExerciseAngina with heart disease, we can notice that most people with heart disease have situation of exercise-induced angina. On the other hand, the chance for people without heart disease to happen exercise-induced angin is small. This means people with heaer disease usually will have the situation of exercise-induced angina

Next, I am also interested in relationship of ExerciseAngina and Oldpeak. Oldpeak means the numeric value measured when people are doing exercise.

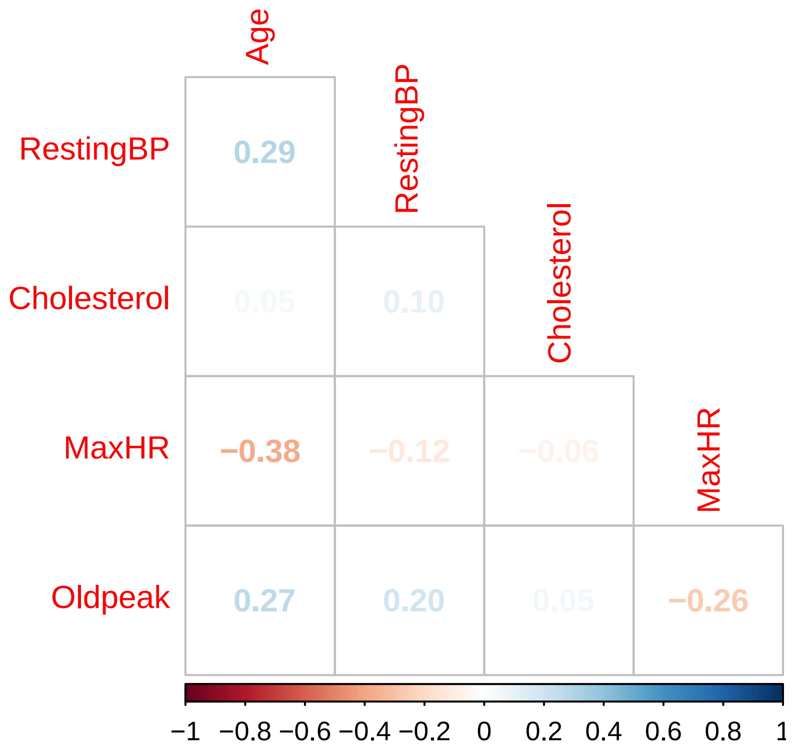
```
Heart_train %>%
  ggplot(mapping = aes(x = ExerciseAngina, y = Oldpeak)) +
  ggtitle("violinplot of Oldpeak by ExerciseAngina with heart disease  ")+
  geom_violin(aes(color = HeartDisease)) +
  labs(x = "ExerciseAngina", y = "Oldpeak")+
  theme( legend.key.size = unit(2, 'cm'),
        plot.title = element_text(face = "bold.italic", hjust=0.5))
```



According to the violinplot of Oldpeak by ExerciseAngina with heart disease, we find that for people with heart disease usually have a higher Oldpeak. Also, for people not have exercise angina and without heart disease, most of them have the value 0 of the Oldpeak. Otherwise, other popele all have a higehr Oldpeak value. This indicates the lower value fo oldpeak, the lower change that peopel will have heart disease.

```
Heart_train %>%

  select(-Sex,-ChestPainType,-FastingBS,-RestingECG ,-ExerciseAngina,-ST_Slope,-HeartDisease) %>%
  cor() %>%
  corrplot(type = "lower",method = "number",diag = FALSE)
```



According to the correlation matrix, we can see that this correlation matrix performed very well, which means there is no need to change.

Model Selection

I first create the recipe of the data. Because I use the factor varibales, I need to use step function: step_dummy() to all nominal predictors. Also use step function: step_center() and step_scale to all predictors.

```
Heart_recipe <- recipe(HeartDisease ~ Age + Sex + ChestPainType +
                       RestingBP + Cholesterol + FastingBS + RestingECG + MaxHR + ExerciseAngina + Oldp
  step_dummy(all_nominal_predictors()) %>%
  step_center(all_predictors()) %>%
  step_scale(all_predictors())
```

I applied cross validation by using funtion vfold_cv() with 5 folds.

```
set.seed(100)
Heart_folds <- vfold_cv(Heart_train, v = 5, strata = HeartDisease)
```

logistic regression

I created a logistic regression model to the data with glm engine, and I created a workflow for logistic regression called log_workflow.

```
log_reg <- logistic_reg() %>%
  set_engine("glm")
log_wkflow <- workflow() %>%
  add_model(log_reg) %>%
  add_recipe(Heart_recipe)
```

I fitted the logistic regression model with the cross validation folds.

```
glm_cv <- log_wkflow %>%
  fit_resamples(Heart_folds)
```

```
collect_metrics(glm_cv)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.874     5  0.0123 Preprocessor1_Model1
## 2 roc_auc  binary     0.926     5  0.0112 Preprocessor1_Model1
```

Accuracy and roc_auc are collected. The acuracy is 0.874, and the roc_auc is 0.926.

LDA

I created a logistic regression model, called lda_mod to the data using MASS engine, and I created a workflow for LDA called lda_wkflow.

```
lda_mod <- discrim_linear() %>%
  set_engine("MASS")
lda_wkflow = workflow() %>%
  add_model(lda_mod) %>%
  add_recipe(Heart_recipe)
```

LDA model is fitted to the Heart_folds, which is the cross validation folds I created. The fitted result is saved to a rds file in the R_script folder.

```
lda_cv <- lda_wkflow %>%
  fit_resamples(Heart_folds)
write_rds(lda_cv, file = "R_scripts/lda.rds")
```

```
lda_cv = read_rds(file = "R_scripts/lda.rds")
collect_metrics(lda_cv)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.861     5  0.0185 Preprocessor1_Model1
## 2 roc_auc  binary     0.926     5  0.0103 Preprocessor1_Model1
```

Accuracy and roc_auc are collected. The acuracy is 0.861, and the roc_auc is 0.926.

QDA

I created a QDA model, called qda_mod to the data using MASS engine, and I created a workflow for QDA called qda_wkflow.

```r
qda_mod <- discrim_quad() %>%
  set_engine("MASS")
qda_wkflow = workflow() %>%
  add_model(qda_mod) %>%
  add_recipe(Heart_recipe)
```

LDA model is fitted to the Heart_folds The fitted result is saved to a rds file in the R_script folder.

```r
qda_cv <- qda_wkflow %>%
  fit_resamples(Heart_folds)
write_rds(qda_cv, file = "R_scripts/qda.rds")
```

```r
qda = read_rds(file = "R_scripts/qda.rds")
collect_metrics(qda)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.856     5 0.0159  Preprocessor1_Model1
## 2 roc_auc  binary     0.918     5 0.00583 Preprocessor1_Model1
```

Accuracy and roc_auc are collected. The acuracy is 0.856, and the roc_auc is 0.918.

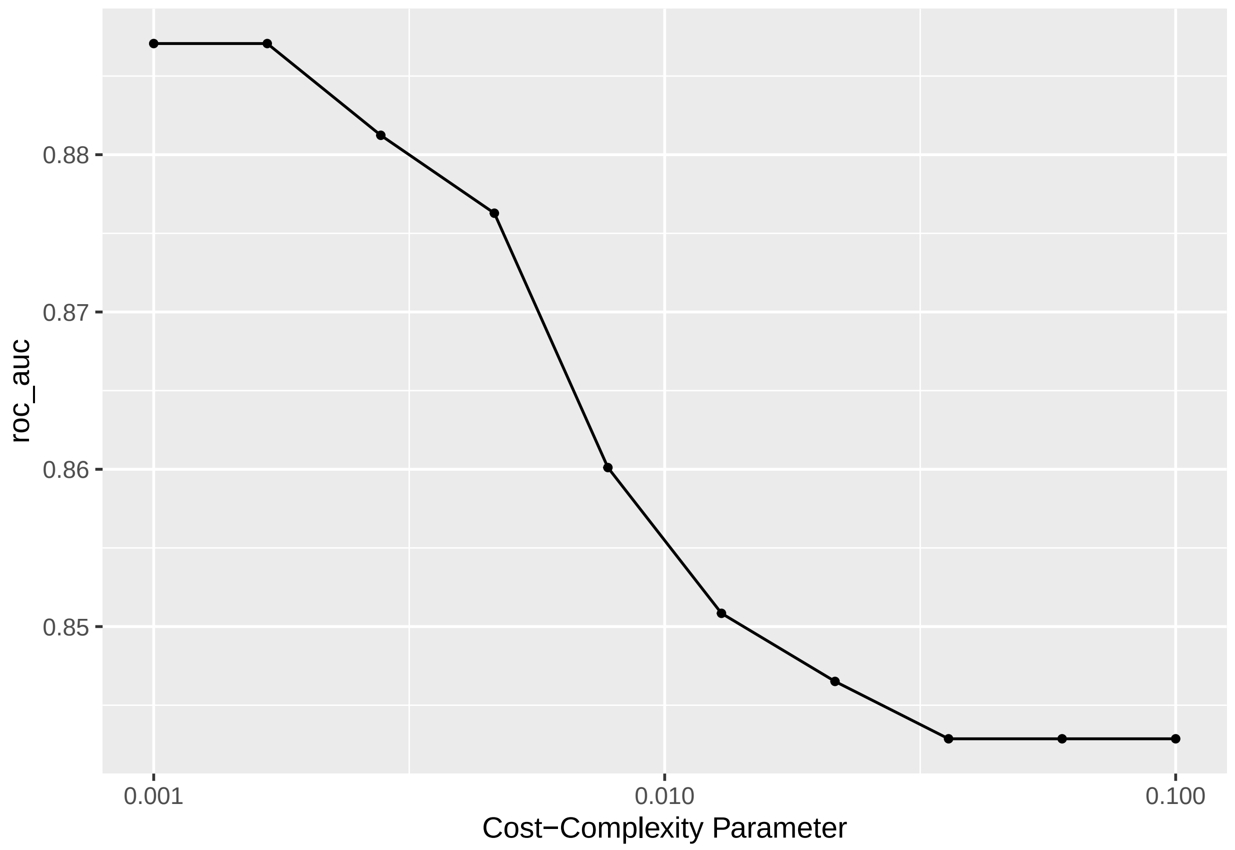
Decision tree model

I created a decision tree model, called tree_spec to the data using rpart engine, and I created a workflow for dicidion tree called tree_wkflow.

```r
tree_spec <- decision_tree(cost_complexity = tune()) %>%
  set_engine("rpart") %>%
  set_mode("classification")
tree_wkflow <- workflow() %>%
  add_recipe(Heart_recipe) %>%
  add_model(tree_spec)
```

I created a parameter grid and set the range to be between -3 and -1 with 10 levels.

```r
param_grid <- grid_regular(cost_complexity(range = c(-3, -1)), levels = 10)
dt_tune_res <- tune_grid(tree_wkflow,
                         resamples = Heart_folds,
                         grid = param_grid,
                         metrics = metric_set(roc_auc))
autoplot(dt_tune_res)
```



```
collect_metrics(dt_tune_res) %>% arrange(desc(mean)) %>% slice(1)
```

```
## # A tibble: 1 x 7
##   cost_complexity .metric .estimator  mean     n std_err .config
##             <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1           0.001 roc_auc binary     0.887     5  0.0168 Preprocessor1_Model01
```

The best tree is selected based on roc_auc. The best decision tree model has cost complexity of 0.001, and the roc_auc is 0.887 with standard error of 0.0168.

boosted tree model

I created a boosted tree model and set the number of trees to be the tuned, using xgboost engine and classificatio mode. A grid is created with number of trees to be between 10 and 2000 using 10 levels.

```
bt_spec <- boost_tree(trees = tune()) %>%
  set_engine("xgboost") %>%
  set_mode("classification")
bt_grid <- grid_regular(trees(c(10,2000)), levels = 10)
bt_wkflow <- workflow() %>%
  add_model(bt_spec) %>%
    add_recipe(Heart_recipe)
```

The tuned result is saved in the R_script folder called boost_res.rds.

```
bt_tune_res <- tune_grid(
  bt_wkflow,
  resamples = Heart_folds,
  grid = bt_grid,
  metrics = metric_set(roc_auc)
)
autoplot(bt_tune_res)
write_rds(bt_tune_res, file = "R_scripts/boost_res.rds")
```

I loaded the tuned result from the R_script folder, and found the best model.

```
bt_res = read_rds("R_scripts/boost_res.rds")
collect_metrics(bt_res) %>% arrange(desc(mean)) %>% slice(1)
```

```
## # A tibble: 1 x 7
##   trees .metric .estimator  mean     n std_err .config
##   <int> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1   231 roc_auc binary     0.906     5  0.0124 Preprocessor1_Model02
```

The best model has 231 trees and a roc_auc of 0.906.

Random forest

I tuned min_n and mtry, set mode to "classification" (because my outcome is discrete variable), and used the ranger engine. I stored this model and my recipe in a workflow.

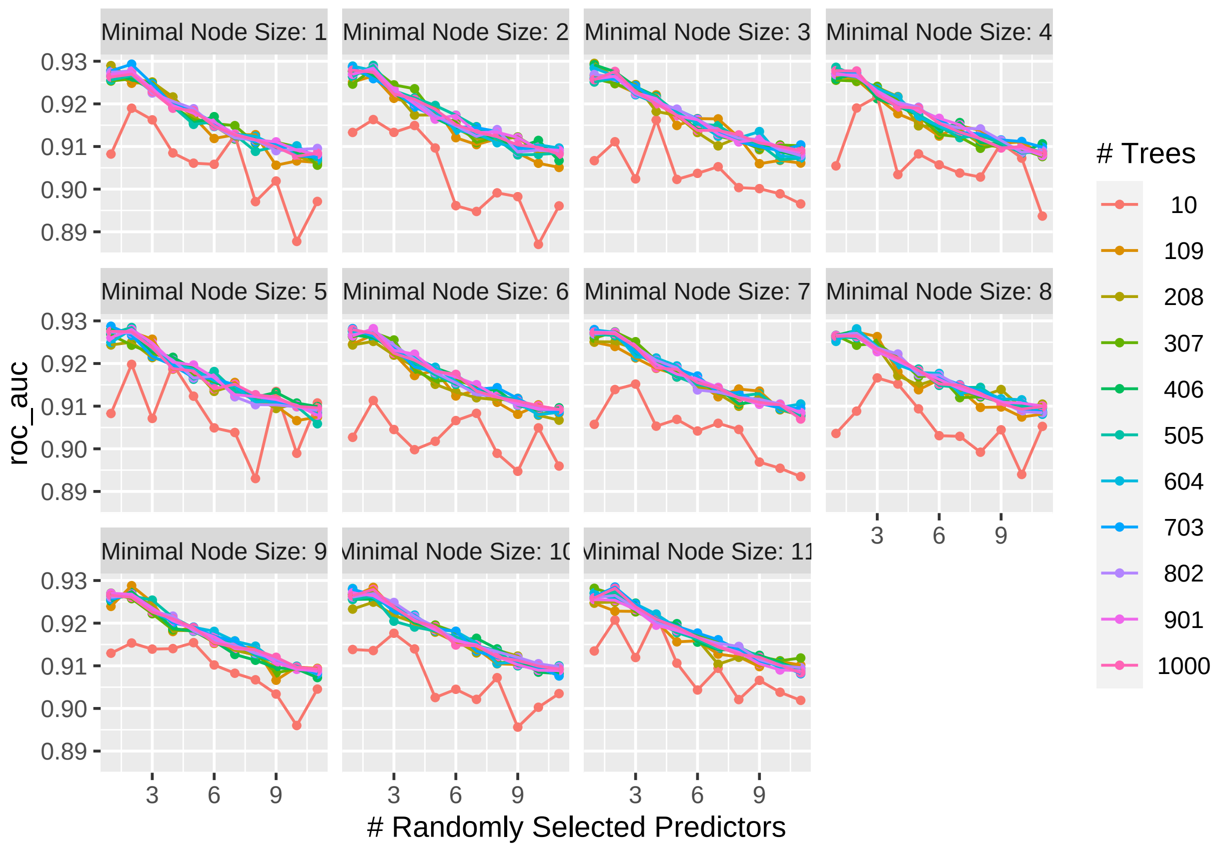
```
 rf_model <- rand_forest(mtry = tune(),trees = tune(),min_n = tune()) %>%
  set_engine("ranger",importance = "impurity") %>%
  set_mode("classification")
 rf_wkflow <- workflow() %>%
  add_recipe(Heart_recipe) %>%
  add_model(rf_model)
```

Next, I set up the tuning grid, and I updated the parameters to make the number of trees slightly below the number of variables I had. I set up the tuning grid with min_n to be between 1 and 10 (number of predictor variable I have) and 11 levels.

```
set.seed(100)
rf_grid <- grid_regular(
  mtry(range = c(1, 11)),
  trees(range = c(10,1000)),
  min_n(range = c(1, 11)),
  levels = 11)
rf_tune <- tune_grid(
  rf_wkflow,
  resamples = Heart_folds,
  grid = rf_grid,
  metrics = metric_set(roc_auc)
)

write_rds(rf_tune, file = "R_scripts/rf_tune.rds")
```

```r
rf = read_rds("R_scripts/rf_tune.rds")
autoplot(rf)
```

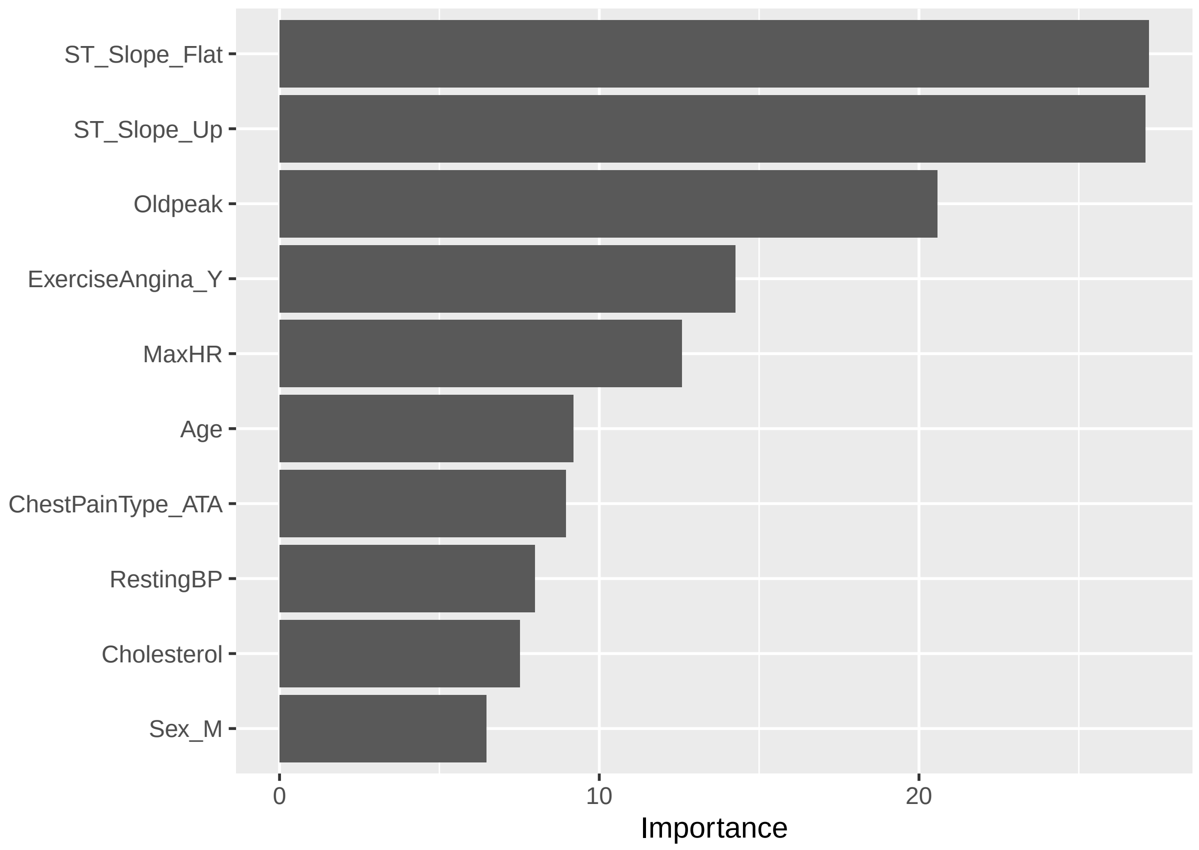


```r
collect_metrics(rf) %>% arrange(desc(mean)) %>% slice(1)
```

```
## # A tibble: 1 x 9
##    mtry trees min_n .metric .estimator  mean     n std_err .config
##   <int> <int> <int> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1     1   208     3 roc_auc binary     0.929     5  0.0109 Preprocessor1_Model0~
```

Then, we will create a variable importance plot, using vip(), with our best-performing random forest model fit on the training set.

```r
rf_final <- finalize_workflow(rf_wkflow,select_best(rf,"roc_auc"))
rf_fit <- fit(rf_final,Heart_train)
rf_fit %>%
  extract_fit_engine() %>%
  vip()
```



According to the variable importance plot, we find that ST_Slope_Flat and ST_Slope_UP are most useful, and Cholesterol and Sex-M are least useful.

Final Model Building

Find the best modle by selecting the best roc_auc of the different models.

```
roc_auc_log <-collect_metrics(glm_cv) %>% slice(2)
roc_auc_LDA <-collect_metrics(lda_cv) %>% slice(2)
roc_auc_QDA <-collect_metrics(qda_cv) %>% slice(2)
roc_auc_dt <-collect_metrics(dt_tune_res) %>% arrange(desc(mean)) %>% slice(1)
roc_auc_rf <-collect_metrics(rf) %>% arrange(desc(mean)) %>% slice(1)
roc_auc_bt <-collect_metrics(bt_res) %>% arrange(desc(mean)) %>% slice(1)
roc_auc_value = c(roc_auc_log[3],roc_auc_LDA[3],roc_auc_QDA[3],roc_auc_dt[4],roc_auc_bt[4],roc_auc_rf[6
model = c("logistic regression","LDA","QDA","decision tree","Boosted tree","Random forest")
cbind(model,roc_auc_value)
```

```
##      model                 roc_auc_value
## mean "logistic regression" 0.926248
## mean "LDA"                 0.9259851
## mean "QDA"                 0.9183263
## mean "decision tree"       0.8870665
## mean "Boosted tree"        0.9062141
## mean "Random forest"       0.929496
```

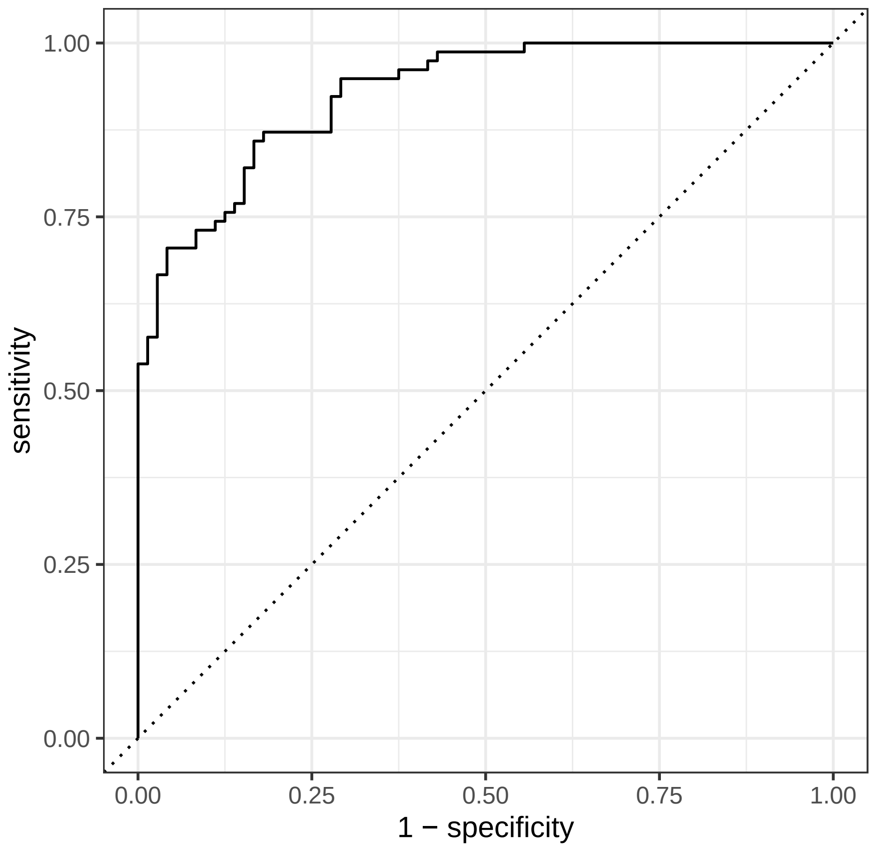
By the result, Random forest model has the highest roc_auc, so we will use Random forest model to fit our test data.

Analysis of The Test Set

```r
set.seed(100)
rf_final <- finalize_workflow(rf_wkflow,select_best(rf,"roc_auc"))
rf_fit <-fit(rf_final,Heart_train)
res = augment(rf_fit, new_data = Heart_test)
res %>%
  roc_auc(truth = HeartDisease, estimate = .pred_0)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.927
```

```r
res %>%
  roc_curve(truth = HeartDisease, .pred_0) %>%
  autoplot()
```



An ROC curve is a graph showing the performance of a classification model at all classification thresholds. The curve plots the true positive rate and false positive rate. Since the area under the ROC curve is the bigger the better, my ROC curve looks very good.

Conclusion

Modern people are becoming more and more conscious of the health of our bodies, of which the prevention and detection of heart disease I think is very important. In particular, the increasing number of sudden deaths nowadays indicates that we should pay more attention to heart disease.

In my project, I first found a very well-researched database.Then I try fit different models including logistic regression, LDA, QDA, Decision tree, boosted tree and random forest. Next, I found the best model is random forest with roc_auc equal to 0.929496. Finally, I Fitted the random forest to the testing dataset with roc_auc equal to 0.9273504. The model performed very well! Therefore, I think this project is meaningful.