Heart Disease Prediction

Karthik Badiganti

07/23/2023

Loading Packages

```
library(kableExtra)
```

```
## Warning in !is.null(rmarkdown::metadata$output) && rmarkdown::metadata$output
## %in% : 'length(x) = 2 > 1' in coercion to 'logical(1)'
```

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()     masks stats::filter()
## x dplyr::group_rows() masks kableExtra::group_rows()
## x dplyr::lag()        masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(readr)
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(gbm)
```

```
## Loaded gbm 2.1.8.1
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(ggcorrplot)
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(rpart)
library(caret)
library(naivebayes)
```

```
## naivebayes 0.9.7 loaded
```

```
library(class)
library(randomForest)
```

```
## randomForest 4.7-1.1
## 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
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(cluster)
library(plyr)
```

```
## ------------------------------------------------------------------------------
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## ------------------------------------------------------------------------------
##
## Attaching package: 'plyr'
##
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
##
## The following object is masked from 'package:purrr':
```

```
##
##     compact
```

```
library(ClustOfVar)
library(dplyr)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:randomForest':
##
##     combine
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(grid)
library(lattice)
library(rpart.plot)
library(DataExplorer)
library(adabag)
```

```
## Loading required package: foreach
##
## Attaching package: 'foreach'
##
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
##
## Loading required package: doParallel
## Loading required package: iterators
## Loading required package: parallel
```

```
## Warning in rgl.init(initValue, onlyNULL): RGL: unable to open X11 display
```

```
## Warning: 'rgl.init' failed, running with 'rgl.useNULL = TRUE'.
```

```
library(e1071)
library(tidyr)
library(ClustOfVar)
library(gbm)
```

Importing Data

```
# loading data downloaded from kaggle
heart_disease_data <-read.csv("heart.csv")



colnames(heart_disease_data) <- c("Age", "Sex", "Chest_Pain_Type",
                          "Resting_Blood_Pressure", "Cholesterol",
                          "Fasting_Blood_Sugar", "Resting_ECG",
                          "Maximum_Heart_Rate", "Exercise_Angina",
```

```
                         "Old_peak", "Slope_HR", "No_MV",
                         "Thallium", "Heart_Disease_Indicator")
head(heart_disease_data)
```

```
##   Age Sex Chest_Pain_Type Resting_Blood_Pressure Cholesterol
## 1  63   1               3                    145         233
## 2  37   1               2                    130         250
## 3  41   0               1                    130         204
## 4  56   1               1                    120         236
## 5  57   0               0                    120         354
## 6  57   1               0                    140         192
##   Fasting_Blood_Sugar Resting_ECG Maximum_Heart_Rate Exercise_Angina Old_peak
## 1                   1           0                150               0      2.3
## 2                   0           1                187               0      3.5
## 3                   0           0                172               0      1.4
## 4                   0           1                178               0      0.8
## 5                   0           1                163               1      0.6
## 6                   0           1                148               0      0.4
##   Slope_HR No_MV Thallium Heart_Disease_Indicator
## 1        0     0        1                       1
## 2        0     0        2                       1
## 3        2     0        2                       1
## 4        2     0        2                       1
## 5        2     0        2                       1
## 6        1     0        1                       1
```

Checking Missing Values

```
colSums(is.na(heart_disease_data))
```

```
##                     Age                     Sex          Chest_Pain_Type
##                       0                       0                        0
##   Resting_Blood_Pressure             Cholesterol     Fasting_Blood_Sugar
##                       0                       0                        0
##              Resting_ECG      Maximum_Heart_Rate         Exercise_Angina
##                       0                       0                        0
##                Old_peak                Slope_HR                   No_MV
##                       0                       0                        0
##                Thallium Heart_Disease_Indicator
##                       0                       0
```

The Data is clean and can proceed to exploratory Data Analysis

Transforming Variables

Renaming Values based on the dataset reference to better understand the data

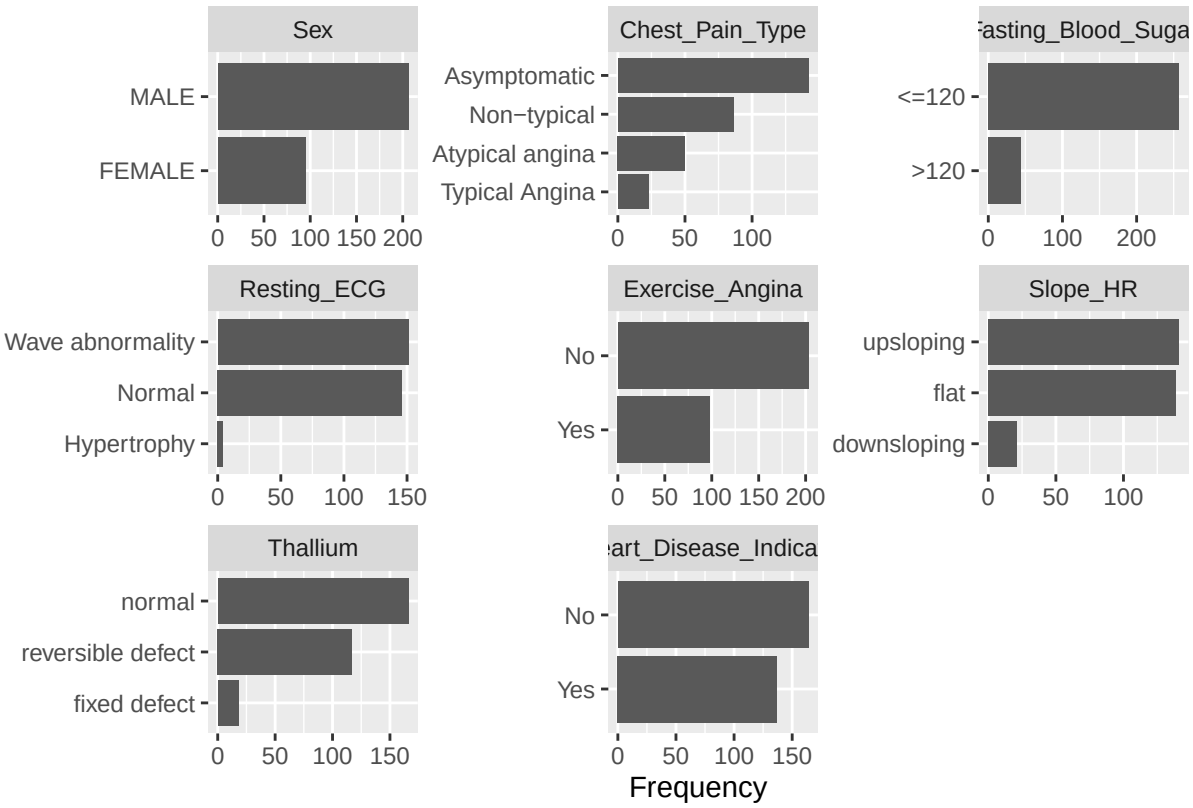
```
heart_disease_data <- heart_disease_data %>% filter(Thallium!=0)%>%
  mutate(Resting_ECG = if_else(Resting_ECG == 0, "Normal",
                               if_else(Resting_ECG == 1,
                                       "Wave abnormality",
                                       "Hypertrophy")),
  Heart_Disease_Indicator = if_else(Heart_Disease_Indicator == 1, "No", "Yes"),
```

```
  Exercise_Angina = if_else(Exercise_Angina == 1,
                                "Yes" ,"No"),
  Sex = if_else(Sex == 1, "MALE", "FEMALE"),
        Fasting_Blood_Sugar = if_else(Fasting_Blood_Sugar == 1,
                                ">120",  "<=120"),

  Chest_Pain_Type = if_else(Chest_Pain_Type == 1, "Atypical angina",
                            if_else(Chest_Pain_Type == 2,
                                    "Non-typical",
                                    if_else(Chest_Pain_Type == 0,
                                            "Asymptomatic",
                                            "Typical Angina"))),
  Slope_HR = if_else(Slope_HR == 2, "upsloping",
                        if_else(Slope_HR == 1, "flat","downsloping")),

        Thallium = case_when(
          Thallium == 1 ~ "fixed defect",
          Thallium == 2 ~ "normal",
          Thallium == 3 ~ "reversible defect"
        ),
        No_MV = as.numeric(No_MV)) %>%
  mutate_if(is.character, as.factor)
```

Dataset OverView

```
## Rows: 301
## Columns: 14
## $ Age                    <int> 63, 37, 41, 56, 57, 57, 56, 44, 52, 57, 54, 48~
## $ Sex                    <fct> MALE, MALE, FEMALE, MALE, FEMALE, MALE, FEMALE~
## $ Chest_Pain_Type        <fct> Typical Angina, Non-typical, Atypical angina, ~
## $ Resting_Blood_Pressure <int> 145, 130, 130, 120, 120, 140, 140, 120, 172, 1~
## $ Cholesterol            <int> 233, 250, 204, 236, 354, 192, 294, 263, 199, 1~
## $ Fasting_Blood_Sugar    <fct> >120, <=120, <=120, <=120, <=120, <=120, <=120~
## $ Resting_ECG            <fct> Normal, Wave abnormality, Normal, Wave abnorma~
## $ Maximum_Heart_Rate     <int> 150, 187, 172, 178, 163, 148, 153, 173, 162, 1~
## $ Exercise_Angina        <fct> No, No, No, No, Yes, No, No, No, No, No, No, N~
## $ Old_peak               <dbl> 2.3, 3.5, 1.4, 0.8, 0.6, 0.4, 1.3, 0.0, 0.5, 1~
## $ Slope_HR               <fct> downsloping, downsloping, upsloping, upsloping~
## $ No_MV                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ Thallium               <fct> fixed defect, normal, normal, normal, normal, ~
## $ Heart_Disease_Indicator <fct> No, No, No, No, No, No, No, No, No, No, No, No~
```

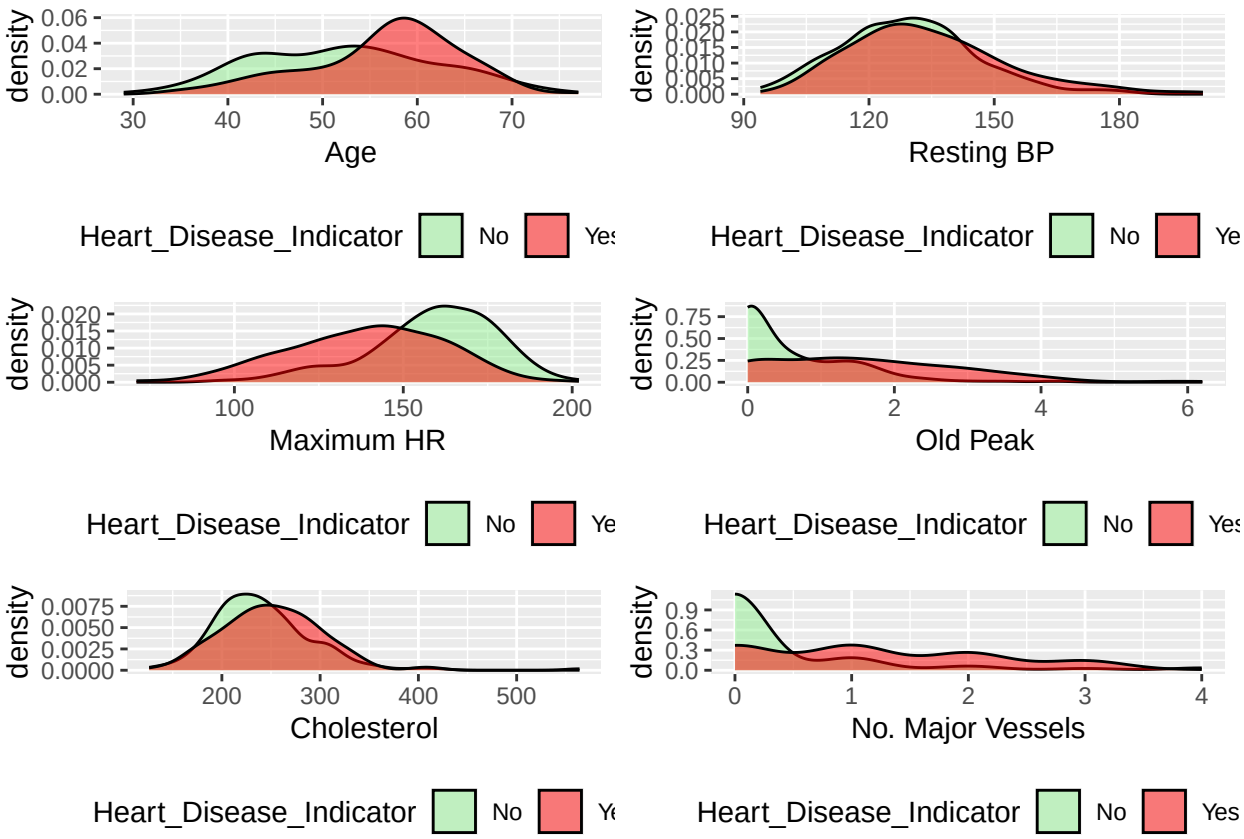
```
plot_bar(heart_disease_data)
```



Exploratory Data Analysis

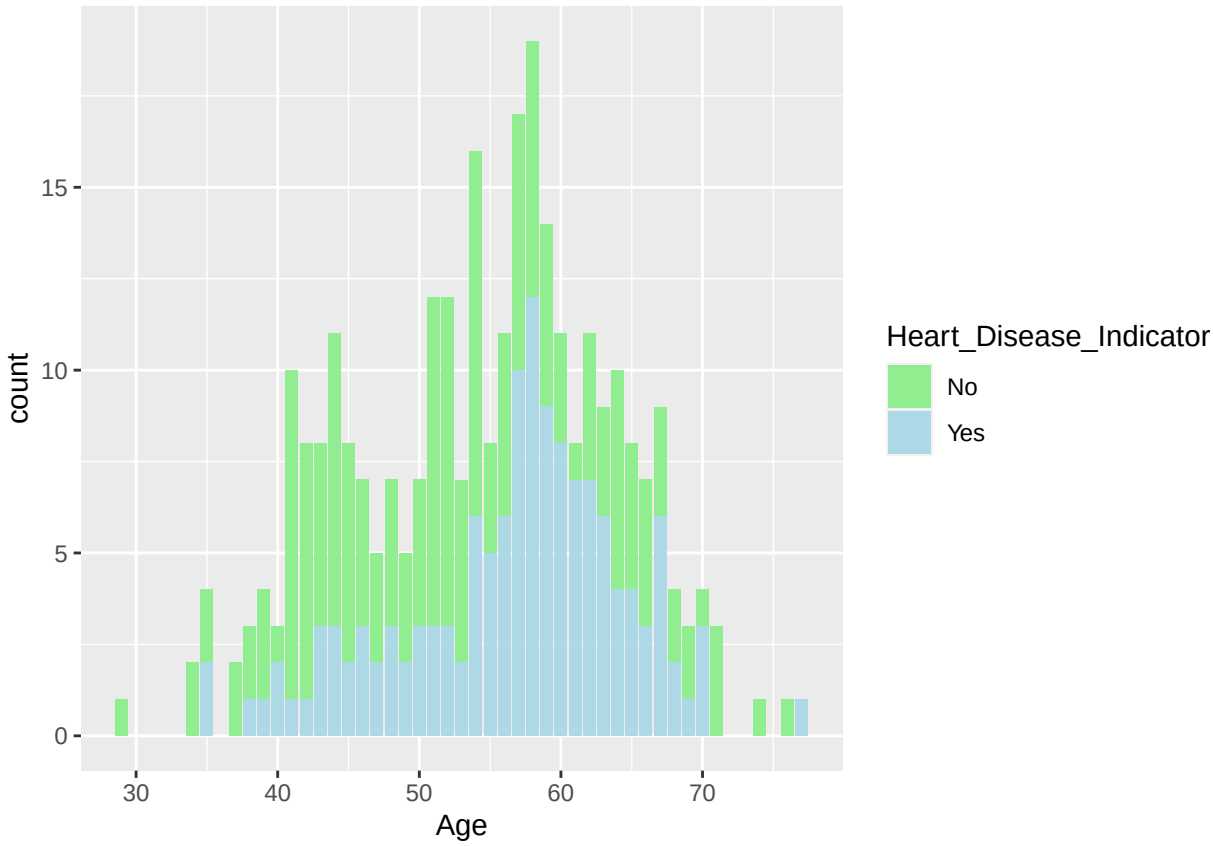
Density Plots with Heart Disease Indicator

```
dp <- function(col,P){
  ggplot(heart_disease_data, aes(x = col, fill = Heart_Disease_Indicator))+
    geom_density(alpha = 0.5)+
    theme(legend.position = "bottom")+
    scale_fill_manual(values=c("lightgreen", "red", "#56B4E9"))+
    scale_x_continuous(name = P)

}

Age_dp <- dp(heart_disease_data$Age, "Age")

bp_dp <- dp(heart_disease_data$Resting_Blood_Pressure, "Resting BP")

old_peak_dp <- dp(heart_disease_data$Old_peak, "Old Peak")

max_HR_dp <- dp(heart_disease_data$Maximum_Heart_Rate, "Maximum HR")

cholesterol_dp <- dp(heart_disease_data$Cholesterol, "Cholesterol")

MV_dp <- dp(heart_disease_data$No_MV, "No. Major Vessels")

#using grid function to display them
grid.arrange(Age_dp, bp_dp, max_HR_dp, old_peak_dp, cholesterol_dp,MV_dp, ncol = 2, nrow = 3)
```



Age vs HDI

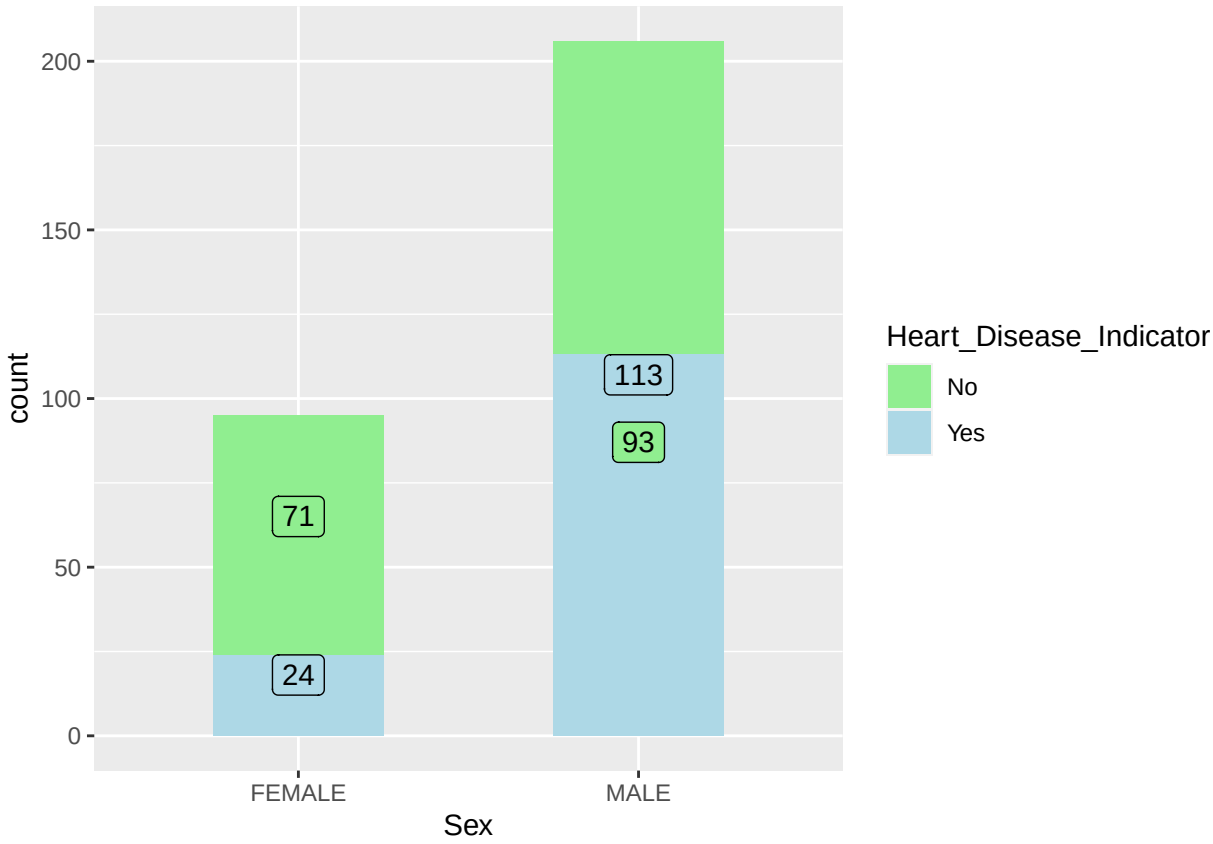
```
heart_disease_data %>% ggplot(aes(Age, fill = Heart_Disease_Indicator))+
  geom_bar() +scale_fill_manual(values = c("lightgreen", "lightblue"))
```



Sex vs HDI

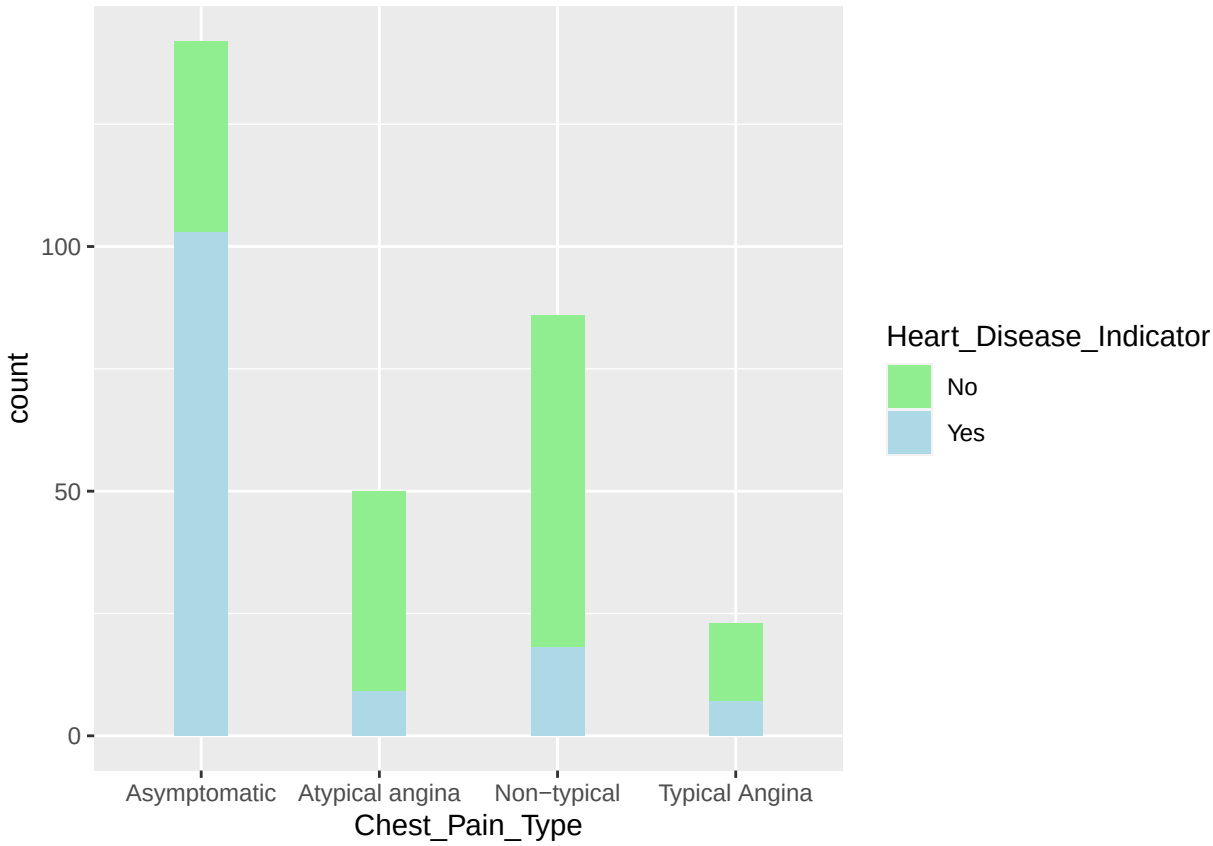
```
heart_disease_data %>% ggplot(aes(Sex, fill = Heart_Disease_Indicator))+
  geom_bar(width = 0.5)+
  geom_label(stat = "Count",aes(label = ..count..),show.legend = FALSE, vjust = 1)+
  scale_fill_manual(values =  c("lightgreen","lightblue"))
```

```
## 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.
```



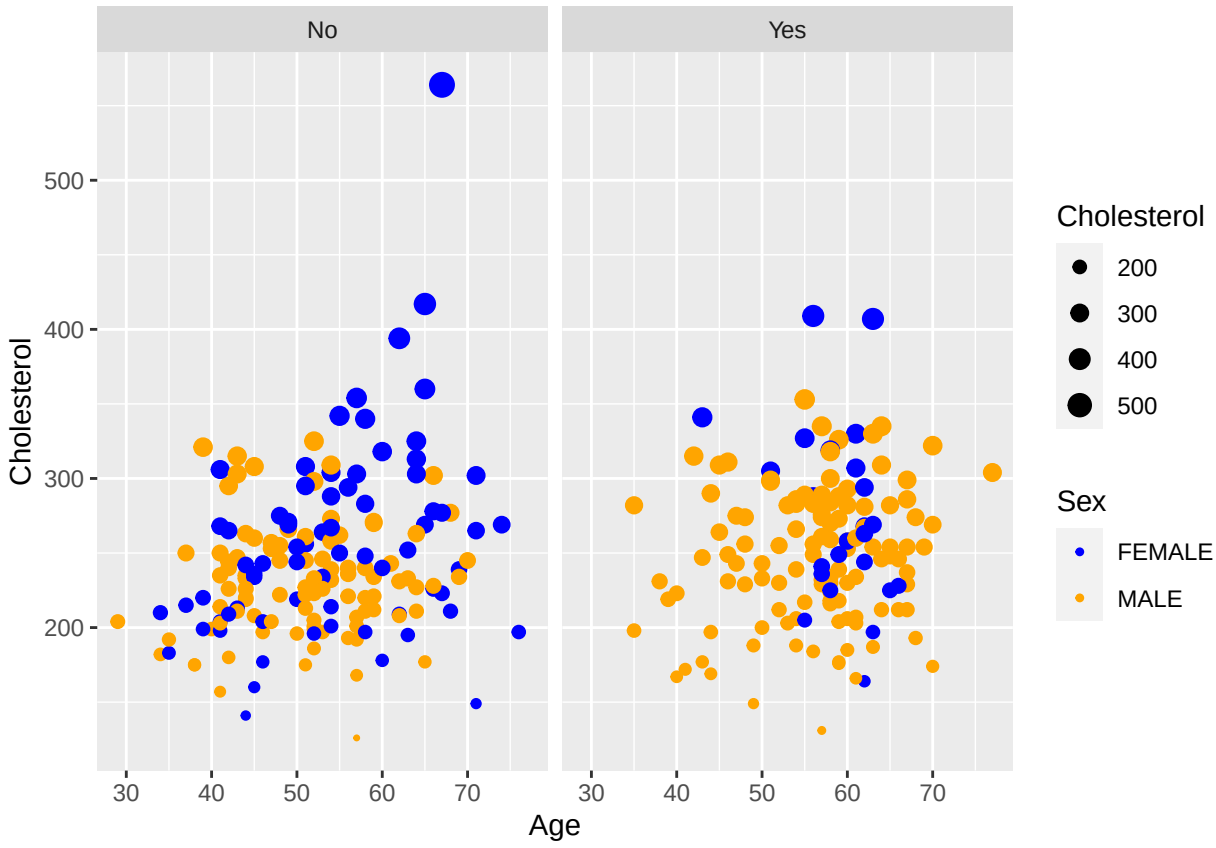
Chest pain type vs HDI

```
heart_disease_data %>% ggplot(aes(Chest_Pain_Type,fill = Heart_Disease_Indicator))+
  geom_bar(width = 0.3)+
  scale_fill_manual(values = c("lightgreen","lightblue"))
```



Cholestrol vs Age vs Sex

```
heart_disease_data %>% ggplot(aes(Age,Cholesterol, color = Sex, size= Cholesterol))+
  geom_point(shape = 20)+ scale_color_manual(values = c("blue","orange"))+
  facet_grid(~Heart_Disease_Indicator)
```



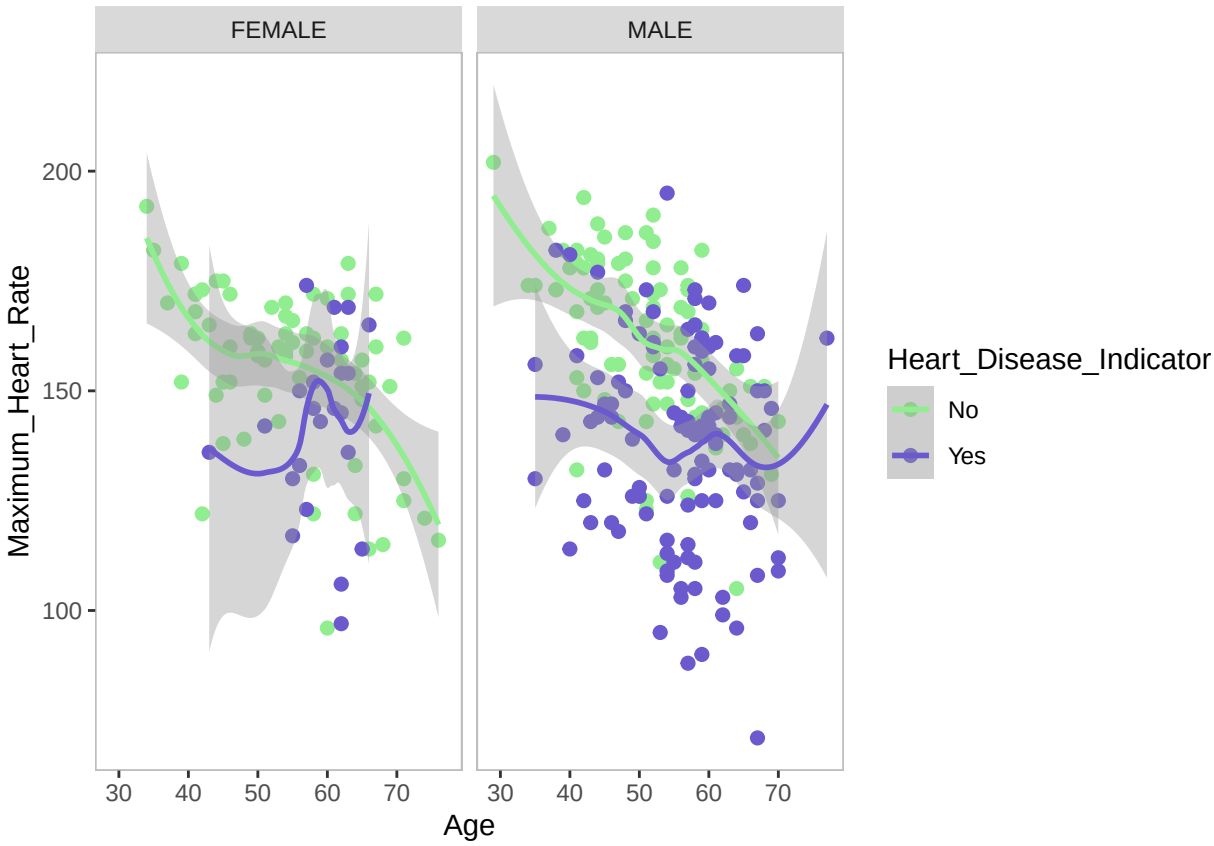
Age vs Heart Rate

```
heart_disease_data %>% ggplot(aes(Age, Maximum_Heart_Rate, color = Heart_Disease_Indicator)) +
  geom_point(size = 2) +
  geom_smooth(method = "loess",size = 1)+
  scale_color_manual(values = c("lightgreen","slateblue"))+
  theme(panel.background = element_rect(fill = "white"),
  panel.border = element_rect(colour = "gray", fill=NA, size=0.5))+
  facet_grid(~Sex)
```

```
## 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.
```

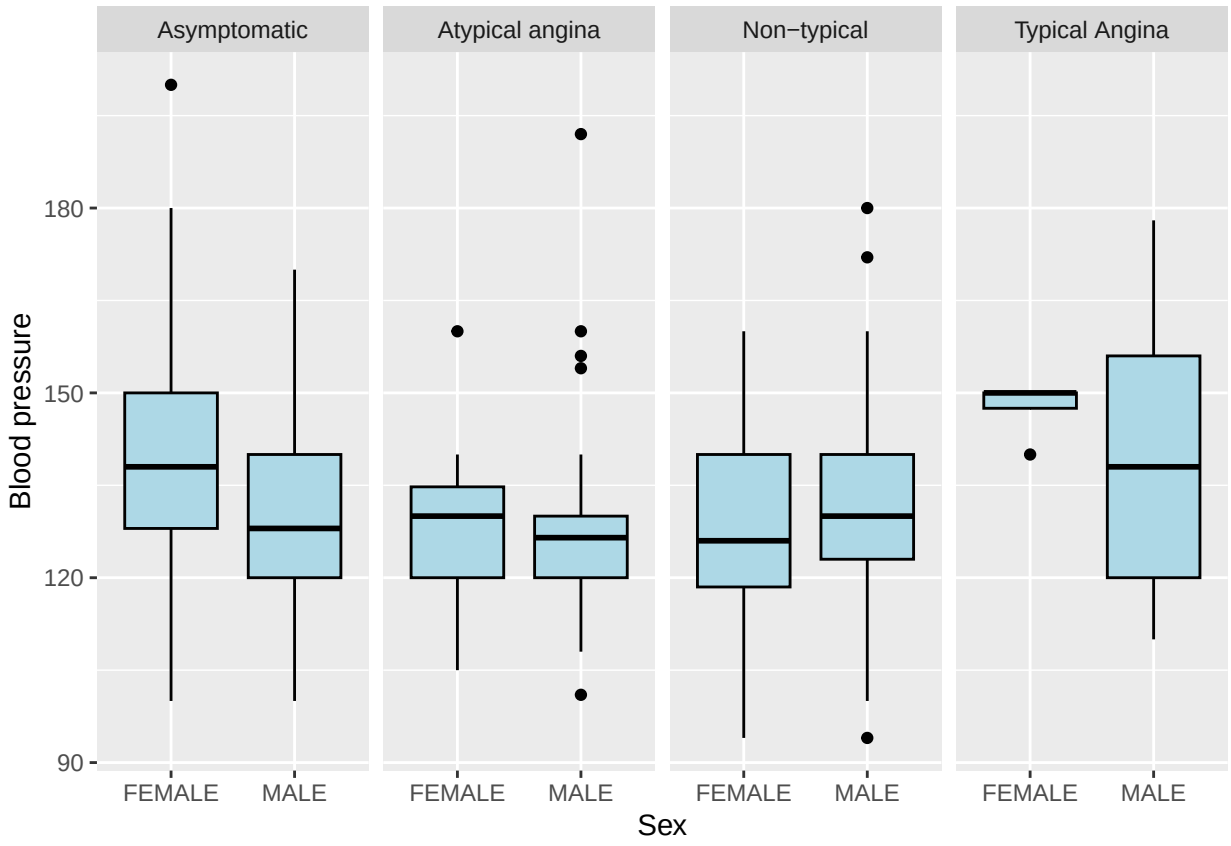
```
## Warning: The `size` argument of `element_rect()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



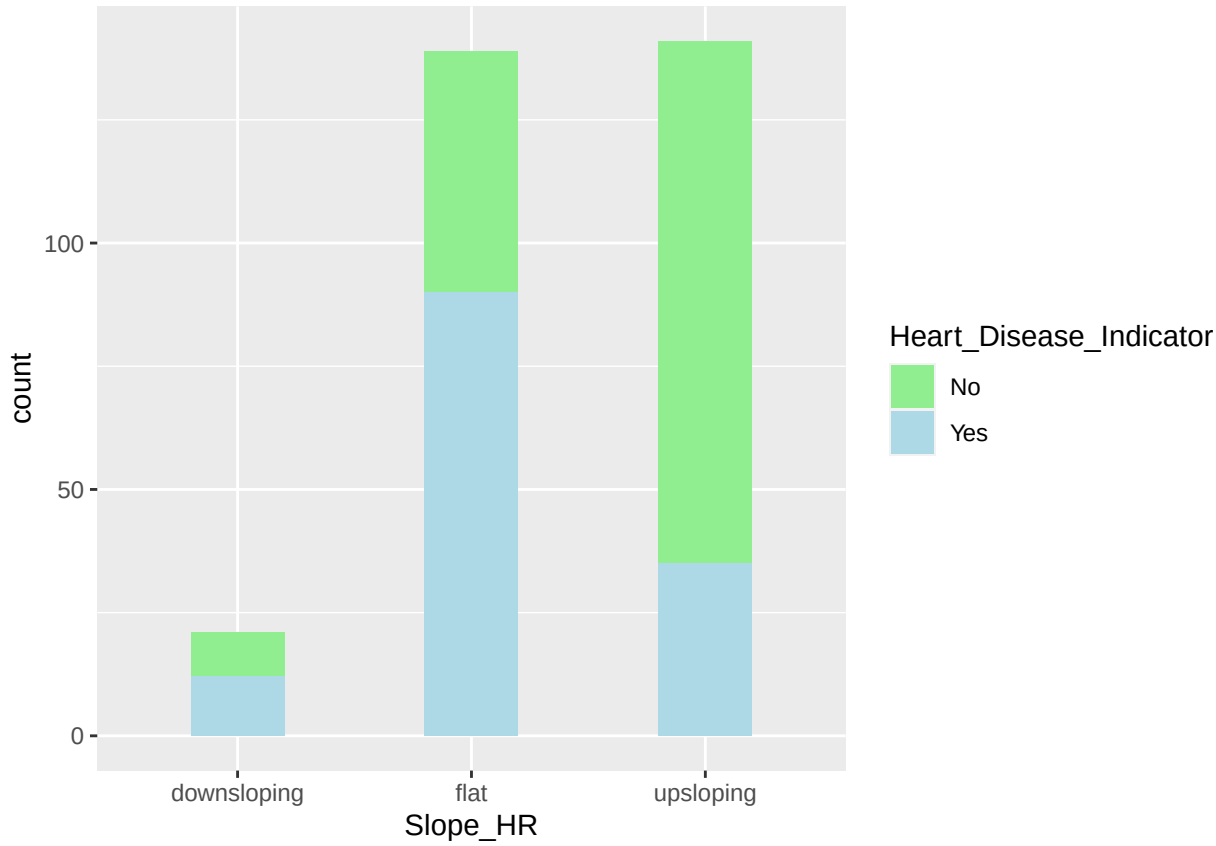
Resting BP vs chest pain Type vs sex

```
heart_disease_data %>% ggplot(aes(Sex,Resting_Blood_Pressure))+
  geom_boxplot(color = "black", fill = "lightblue")+
  labs(x = "Sex", y ="Blood pressure")+
  facet_grid(~Chest_Pain_Type)
```



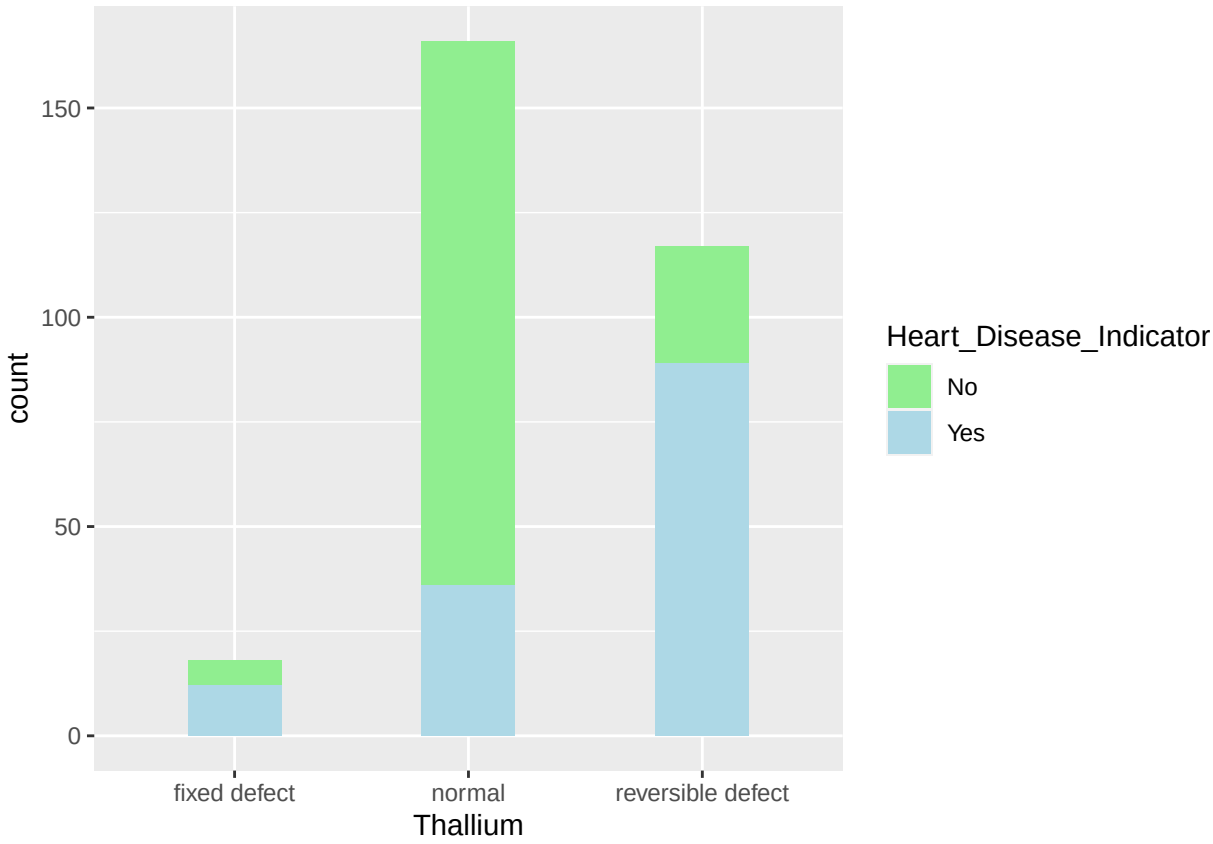
Slope_HR vs HDI

```
heart_disease_data %>% ggplot(aes(Slope_HR, fill=Heart_Disease_Indicator))+
  geom_bar(width = 0.4) + scale_fill_manual(values = c("lightgreen","lightblue"))
```



Thallium vs HDI

```
heart_disease_data %>% ggplot(aes(Thallium, fill = Heart_Disease_Indicator))+
  geom_bar(stat = "count", width = 0.4)+scale_fill_manual(values=c("lightgreen","lightblue"))
```

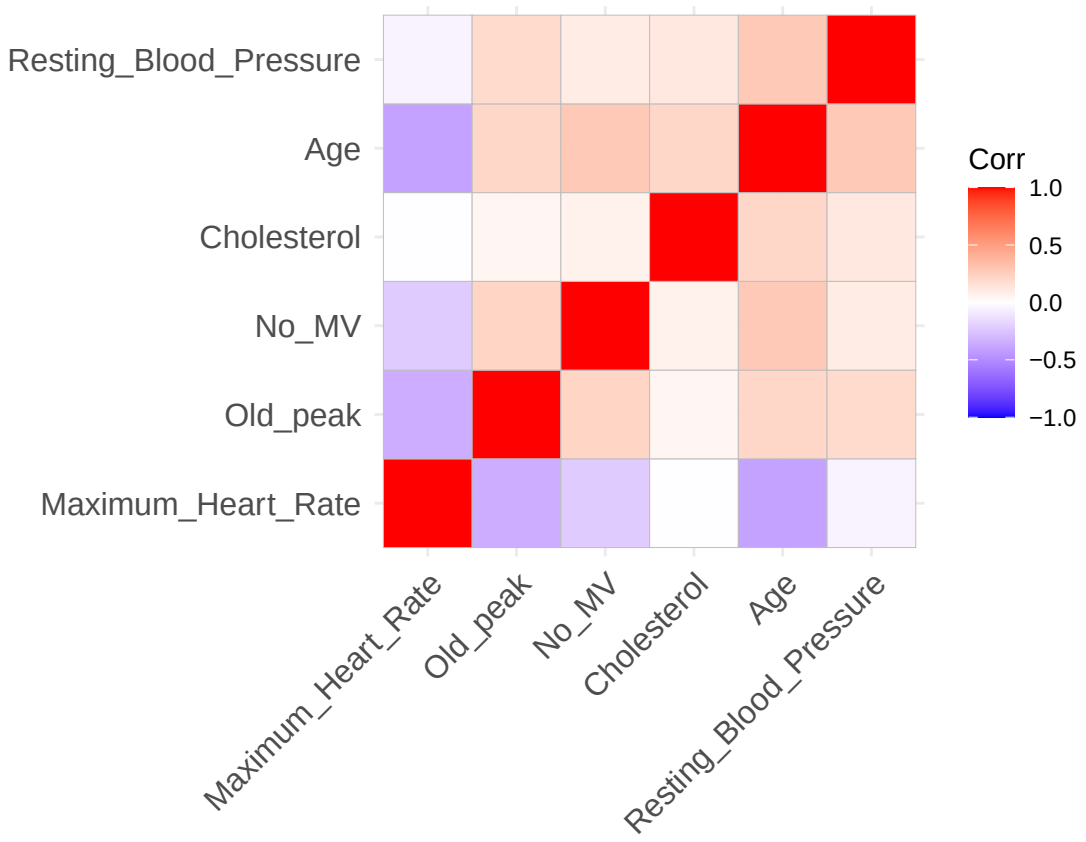


Correlation plot

```
# filtering columns that are factors and numbers
col_heart <- round(cor(heart_disease_data[c(1,4,5,8,10,12)]),3)
col_heart
```

```
##                          Age Resting_Blood_Pressure Cholesterol
## Age                    1.000                 0.279       0.213
## Resting_Blood_Pressure 0.279                 1.000       0.122
## Cholesterol            0.213                 0.122       1.000
## Maximum_Heart_Rate    -0.401                -0.048      -0.012
## Old_peak               0.210                 0.193       0.052
## No_MV                  0.276                 0.101       0.067
##                       Maximum_Heart_Rate Old_peak  No_MV
## Age                               -0.401    0.210  0.276
## Resting_Blood_Pressure            -0.048    0.193  0.101
## Cholesterol                       -0.012    0.052  0.067
## Maximum_Heart_Rate                 1.000   -0.350 -0.217
## Old_peak                          -0.350    1.000  0.221
## No_MV                             -0.217    0.221  1.000
```

```
#correlation plot
ggcorrplot(col_heart, hc.order = TRUE
)
```

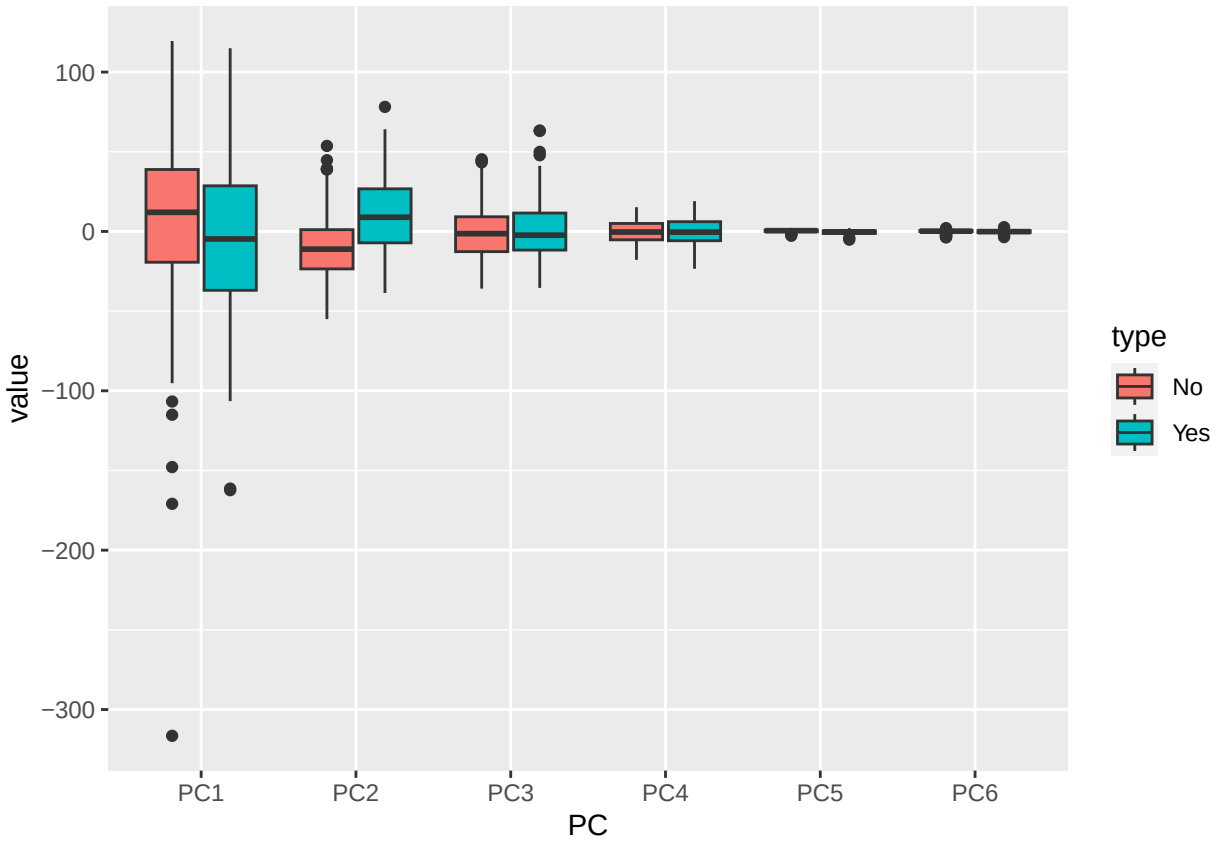


PCA

```
pca <- prcomp(heart_disease_data[c(1,4,5,8,10,12)])
summary(pca)
```

```
## Importance of components:
##                             PC1     PC2      PC3     PC4     PC5     PC6
## Standard deviation     52.0067 23.2756 17.51948 7.66468 1.10593 0.93430
## Proportion of Variance  0.7483  0.1499  0.08492 0.01625 0.00034 0.00024
## Cumulative Proportion   0.7483  0.8982  0.98317 0.99942 0.99976 1.00000
```

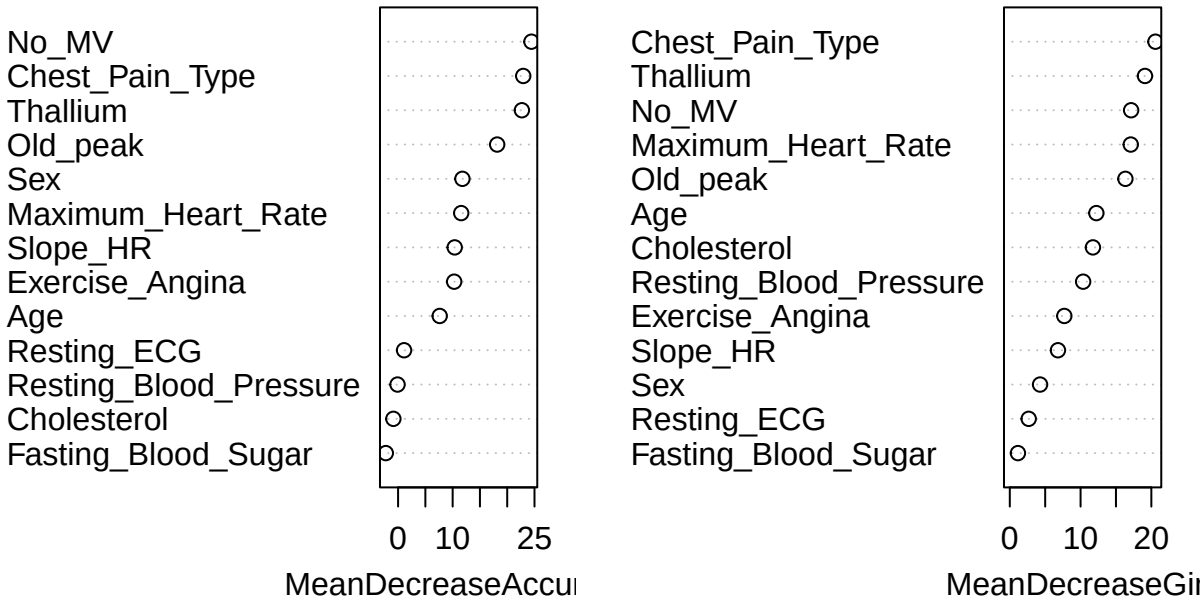
```
data.frame(type = heart_disease_data$Heart_Disease_Indicator, pca$x[,1:6]) %>%
  gather(key = "PC", value = "value", -type) %>%
  ggplot(aes(PC, value, fill = type)) +
  geom_boxplot()
```



Data pre-processing

Feature Selection

```
set.seed(1108)
# feature selection using random forest
var_imp_heart <- randomForest(Heart_Disease_Indicator~., data = heart_disease_data,
                              importance = TRUE)
varImpPlot(var_imp_heart)
```



Splitting dataset 75% Train and 25% Test

```
set.seed(1108)
index <- createDataPartition(y = heart_disease_data$Heart_Disease_Indicator,
                             p = 0.75, list = FALSE)
H_test <- heart_disease_data[-index,]
H_train <- heart_disease_data[index,]



# removing 4 columns from train and test
H_train<-subset(H_train,select = -c(Cholesterol,Fasting_Blood_Sugar,Resting_Blood_Pressure,Resting_ECG))
H_test<-subset(H_test,select = -c(Cholesterol,Fasting_Blood_Sugar,Resting_Blood_Pressure,Resting_ECG))
```

```
# setting control parameter for 10 fold cross validation
ctrl <- trainControl(method="cv", number=10)
```

Models

Logistic Regression

```
set.seed(0811)
glm_train <- train(Heart_Disease_Indicator ~.,
                   data = H_train,
                   method = "glm",
                   trControl = ctrl)
glm_predict <- predict(glm_train, H_test)
```

```r
#Confusion Matrix
glm_CM <- confusionMatrix(glm_predict,H_test$Heart_Disease_Indicator, positive = "Yes")
```

```r
accuracy_glm <- glm_CM$overall["Accuracy"]
sensitivity_glm <- glm_CM$byClass["Sensitivity"]
specificity_glm <- glm_CM$byClass["Specificity"]
accuracy_results <- data_frame(Method = "Logistic Regression", Accuracy = accuracy_glm)
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## i Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
accuracy_results
```

```
## # A tibble: 1 x 2
##   Method              Accuracy
##   <chr>                  <dbl>
## 1 Logistic Regression    0.893
```

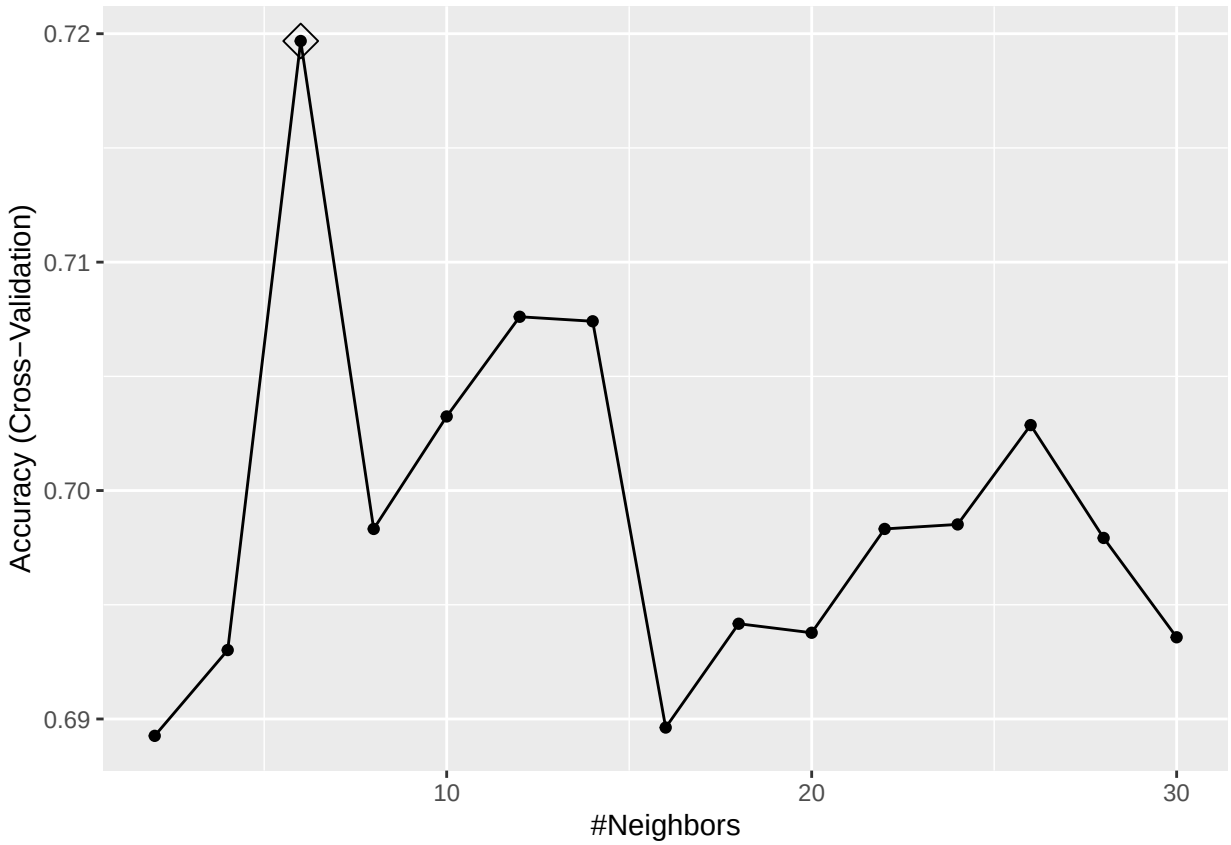
Accuracy of 0.8933333 which is a great start. Let's look into other models.

KNN

```r
set.seed(0811)

knn<- train(Heart_Disease_Indicator ~.,
                  data = H_train,
                  method = "knn",
                  trControl = ctrl,
                  tuneGrid = data.frame(k = seq(2, 30, 2)))

knn$bestTune #best tune
```

```
##   k
## 3 6
```

```r
knn_predict <- predict(knn, H_test)
ggplot(knn,highlight = TRUE)
```




```
#confusion MAtrix of Knn
knn_CM <- confusionMatrix(knn_predict,H_test$Heart_Disease_Indicator, positive = "Yes")
specificity_knn <- knn_CM$byClass["Specificity"]
accuracy_knn <- knn_CM$overall["Accuracy"]
sensitivity_knn <- knn_CM$byClass["Sensitivity"]
accuracy_results <- bind_rows(accuracy_results,
                              data_frame(Method = "KNN", Accuracy = accuracy_knn))

accuracy_results
```

```
## # A tibble: 2 x 2
##   Method              Accuracy
##   <chr>                  <dbl>
## 1 Logistic Regression    0.893
## 2 KNN                    0.68
```

Decision tree

```
set.seed(0811)
rpart <- train(Heart_Disease_Indicator ~.,
                  data = H_train,
                  method = "rpart")

rpart_predict <- predict(rpart, H_test)
```

```
rpart_CM <- confusionMatrix(rpart_predict,H_test$Heart_Disease_Indicator, positive = "Yes")
```

```r
sensitivity_rpart <- rpart_CM$byClass["Sensitivity"]
specificity_rpart <- rpart_CM$byClass["Specificity"]
accuracy_rpart <- rpart_CM$overall["Accuracy"]
pos_pred_rpart<- rpart_CM$byClass["Pos Pred Value"]
neg_pred_rpart <- rpart_CM$byClass["Neg Pred Value"]
accuracy_results <- bind_rows(accuracy_results,
                              data_frame(Method = "Decision Trees", Accuracy = accuracy_rpart))
accuracy_results
```

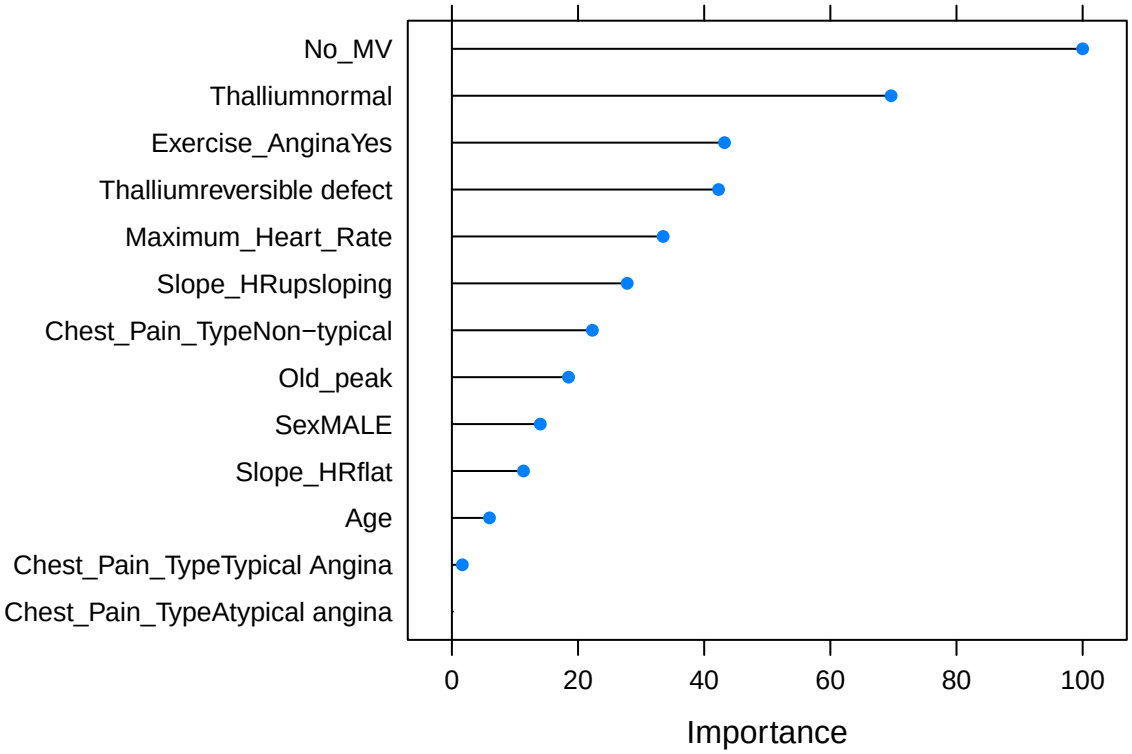
```
## # A tibble: 3 x 2
##   Method              Accuracy
##   <chr>                  <dbl>
## 1 Logistic Regression    0.893
## 2 KNN                    0.68
## 3 Decision Trees         0.76
```

Random forest model

```r
set.seed(0811)
rf <- train(Heart_Disease_Indicator ~.,
                  data = H_train,
                     method = "rf",
                  importance = TRUE)
rf$bestTune
```

```
##   mtry
## 1    2
```

```r
rf_predict <- predict(rf, H_test)

# plotting variable importance
plot(varImp(rf))
```




```
rf_CM <- confusionMatrix(rf_predict,H_test$Heart_Disease_Indicator, positive = "Yes")
sensitivity_rf <- rf_CM$byClass["Sensitivity"]
specificity_rf <- rf_CM$byClass["Specificity"]
accuracy_rf <- rf_CM$overall["Accuracy"]
accuracy_results <- bind_rows(accuracy_results,
                              data_frame(Method = "Random Forest", Accuracy = accuracy_rf))

accuracy_results
```

```
## # A tibble: 4 x 2
##   Method              Accuracy
##   <chr>                  <dbl>
## 1 Logistic Regression    0.893
## 2 KNN                    0.68
## 3 Decision Trees         0.76
## 4 Random Forest          0.813
```

Adaptive boosting

```
#adaptive boosting
set.seed(0811)

ada <- boosting(Heart_Disease_Indicator ~.,
                  data = H_train,mfinal = 70)
ada_predict <- predict(ada, H_test)
```

```
#confusion matrix for ada boost model
ada_CM <- confusionMatrix(as.factor(ada_predict$class),as.factor(H_test$Heart_Disease_Indicator), positi
sensitivity_ada <- ada_CM$byClass["Sensitivity"]
specificity_ada <- ada_CM$byClass["Specificity"]
```

```
accuracy_ada <- ada_CM$overall["Accuracy"]
accuracy_results <- bind_rows(accuracy_results,
                              data_frame(Method = "Ada Boost", Accuracy = accuracy_ada))
accuracy_results
```

```
## # A tibble: 5 x 2
##   Method              Accuracy
##   <chr>                  <dbl>
## 1 Logistic Regression    0.893
## 2 KNN                    0.68
## 3 Decision Trees         0.76
## 4 Random Forest          0.813
## 5 Ada Boost              0.773
```

Results

```
##                      Logistic_Regression       KNN Regression_Trees
## Sensitivity                    0.8529412 0.6470588        0.7941176
## Specificity                    0.9268293 0.7073171        0.7317073
## Pos Pred Value                 0.9062500 0.6470588        0.7105263
## Neg Pred Value                 0.8837209 0.7073171        0.8108108
## Precision                      0.9062500 0.6470588        0.7105263
## Recall                         0.8529412 0.6470588        0.7941176
## F1                             0.8787879 0.6470588        0.7500000
## Prevalence                     0.4533333 0.4533333        0.4533333
## Detection Rate                 0.3866667 0.2933333        0.3600000
## Detection Prevalence           0.4266667 0.4533333        0.5066667
## Balanced Accuracy              0.8898852 0.6771879        0.7629125
##                      Random_Forest Ada_boost
## Sensitivity              0.7647059 0.7941176
## Specificity              0.8536585 0.7560976
## Pos Pred Value           0.8125000 0.7297297
## Neg Pred Value           0.8139535 0.8157895
## Precision                0.8125000 0.7297297
## Recall                   0.7647059 0.7941176
## F1                       0.7878788 0.7605634
## Prevalence               0.4533333 0.4533333
## Detection Rate           0.3466667 0.3600000
## Detection Prevalence     0.4266667 0.4933333
## Balanced Accuracy        0.8091822 0.7751076
```

Conclusion

The objective of this project is to use the Cleveland heart disease data set to correctly diagnose people with heart diseases. An explanatory data analysis was done and it revealed how different variables in the dataset help us predict the disease. It also revealed how some factors don't directly influence the results and those factors were later removed to improve our model.

Different machine learning models were built to optimize the accuracy of the prediction and the ones that proved most successful were Logistic Regression model and the Random forest model. The least successful one is the KNN model. Although our accuracy was on an acceptable level, our sensitivity and specificity were still below 90% which is concerning. But with the given set of data this is an efficient outcome.

Many other models were trained but they dint quite improve on the accuracy and hence weren't included in

the report. Having more volume of data will enable an improvement in the model with much higher sample set. Also, using feature selection might also improve in a much accurate model.

References

https://www.heartandstroke.ca/heart-disease/what-is-heart-disease/types-of-heart-disease

https://archive.ics.uci.edu/ml/datasets/heart+disease

https://uc-r.github.io/regression_trees

https://towardsdatascience.com/random-forest-in-r-f66adf80ec9 http://finzi.psych.upenn.edu/R/library/caret/html/sensitivity.html

https://rafalab.github.io/dsbook/machine-learning-in-practice.html