Report for Choose Your Own project submission Project HarvardX PH125.9x- Heart Disease Risk

2022-12-18

Introduction

In this report i decided to implement Machine Learning Ensemble to Make Heart Disease Prediction. In Mongolia where i live, heart disease is number one death cause and Mongolia is ranked 9th in the World of heart disease death rate per 100000 as stated in World Health Rankings website https://www.worldlifeexpectancy.com/cause-of-death/coronary-heart-disease/by-country/. Since we do not have available dataset of our country i used heart disease data set from University of California Irvine machine learning repository. This data set consist fo 14 different features and 303 observations. The description of the features from the website is the following:. *age: age in years* sex: sex (1 = male; 0 = female) *cp: chest pain type Value 1: typical angina Value 2: atypical angina Value 3: non-anginal pain* trestbps: resting blood pressure (in mm Hg on admission to the hospital) *chol: serum cholestoral in mg/dl* fbs: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false) *restecg: resting electrocardiographic results Value 0: normal Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV) Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria* thalach: maximum heart rate achieved *exang: exercise induced angina (1 = yes; 0 = no)* oldpeak = ST depression induced by exercise relative to rest *slope: the slope of the peak exercise ST segment Value 1: upsloping Value 2: flat Value 3: downsloping* ca: number of major vessels (0-3) colored by flourosopy *thal: 3 = normal; 6 = fixed defect; 7 = reversable defect* target: diagnosis of heart disease (angiographic disease status) Value 0: < 50% diameter narrowing Value 1: > 50% diameter narrowing The target feature is the feature we will be trying to predict for this project.

Load libraries

- tidyverse: For data cleaning, sorting, and visualization
- DataExplorer: For Exploratory Data Analysis
- gridExtra: To plot several plots in one figure
- ggpubr: To prepare publication-ready plots
- GGally: For correlations
- caTools: For classification model
- rpart: For classification model
- rattle: Plot nicer descision trees
- randomForest: For Random Forest model
- library(caret)
- library(dplyr)

- library(matrixStats)
- library(gam)
- library(evtree)
- library(knitr)

```
if(!require(tidyverse)) install.packages("tidyverse")
if(!require(caret)) install.packages("dplyr")
if(!require(dplyr)) install.packages("dplyr")
if(!require(matrixStats)) install.packages("matrixStats")
if(!require(gam)) install.packages("gam")
if(!require(evtree)) install.packages("evtree")
if(!require(knitr)) install.packages("knitr")

library(tidyverse)# For data cleaning, sorting, and visualization
library(caret)
library(dplyr)
library(matrixStats)
library(gam)
library(evtree)
library(knitr)
library(DataExplorer) # For Exploratory Data Analysis
library(gridExtra) # To plot several plots in one figure
library(ggpubr) # To prepare publication-ready plots
library(GGally) # For correlations
library(caTools) # For classification model
library(rpart) # For classification model
library(rattle) # Plot nicer descision trees
library(randomForest) # For Random Forest model
```

I Data

Link to the UCI heart disease data: https://archive.ics.uci.edu/ml/datasets/heart+disease Kaggle heart dataset: https://www.kaggle.com/datasets/zhaoyingzhu/heartcsv

As explained on the links above, it is essential to note that on this dataset, the target value 0 indicates that the patient has heart disease.

Attribute Information:
age: age in years
sex: (1 = male; 0 = female)
cp: chest pain type (typical angina, atypical angina, non-angina, or asymptomatic angina)
trestbps: resting blood pressure (in mm Hg on admission to the hospital)
chol: serum cholestoral in mg/dl
fbs: Fasting blood sugar ($<$ 120 mg/dl or $>$ 120 mg/dl) (1 = true; 0 = false)
restecg: resting electrocardiographic results (normal, ST-T wave abnormality, or left ventricular hypertrophy)
thalach: Max. heart rate achieved during thalium stress test
exang: Exercise induced angina (1 = yes; 0 = no)
oldpeak: ST depression induced by exercise relative to rest
slope: Slope of peak exercise ST segment (0 = upsloping, 1 = flat, or 2 = downsloping)
ca: number of major vessels (0-3) colored by flourosopy 4 = NA
thal: Thalium stress test result 3 = normal; 6 = fixed defect; 7 = reversable defect 0 = NA
target: Heart disease status 1 or 0 (0 = heart disease 1 = asymptomatic)

```
heart <- read.csv("C:/Users/Hitech/Downloads/Heart.csv")
```

Methods & Analysis

To achieve this goal we will be creating 9 different models and an ensemble and comparing their results. Because the nature of the problem is to determine if a patient is negative or positive, i.e 0 or 1, this is a binary classification problem and we will pick 10 algorithms that work well with binary classification. The algorithms we will be using are the following:

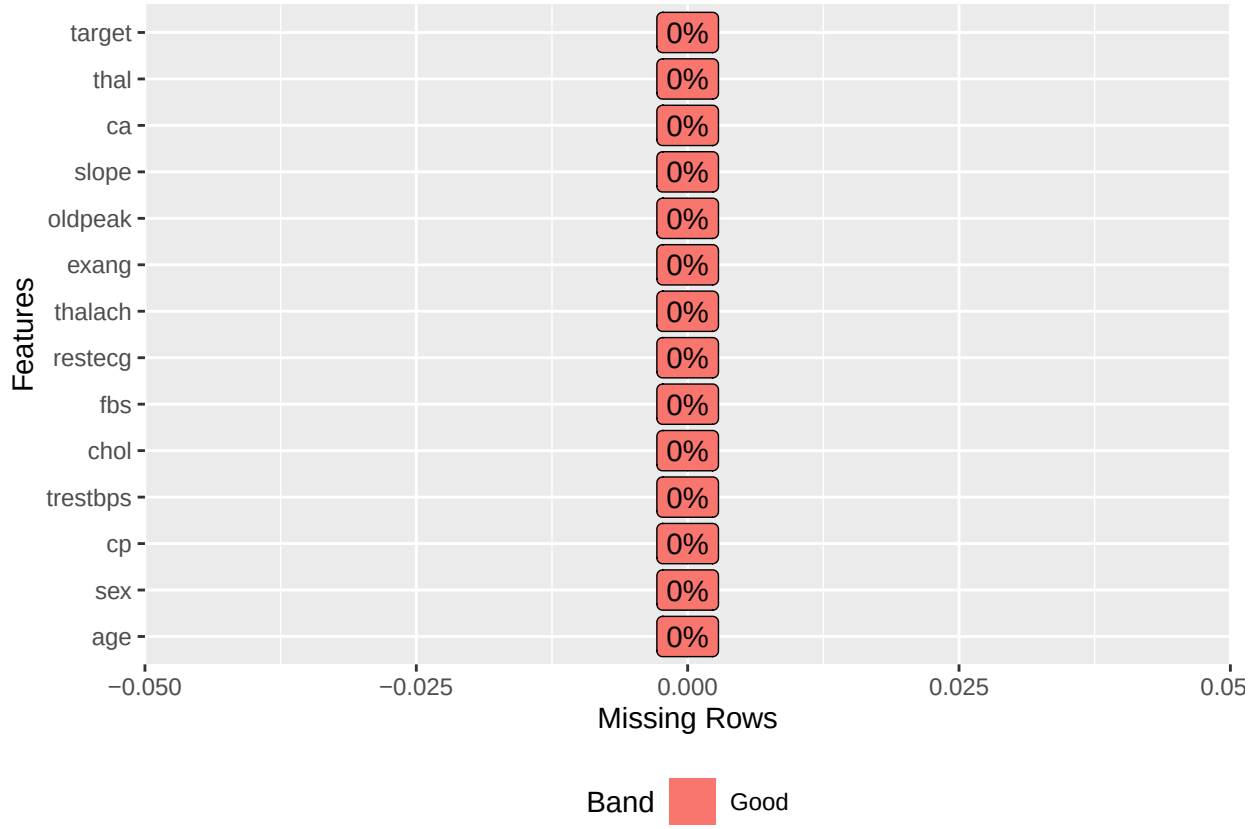
Data Preparation

```
copy <- heart

heart2 <- heart %>%
  filter(
    thal != 0 & ca != 4 # remove values correspondind to NA in original dataset
  ) %>%
# Recode the categorical variables as factors using the dplyr library.
  mutate(
    sex = case_when(
      sex == 0 ~ "female",
      sex == 1 ~ "male"
           ),
    fbs = case_when(
      fbs == 0 ~ "<=120",
      fbs == 1 ~ ">120"
           ),
    exang = case_when(
      exang == 0 ~ "no",
      exang == 1 ~ "yes"
           ),
    cp = case_when(
      cp == 3 ~ "typical angina",
      cp == 1 ~ "atypical angina",
      cp == 2 ~ "non-anginal",
      cp == 0 ~ "asymptomatic angina"
         ),
    restecg = case_when(
      restecg == 0 ~ "hypertrophy",
      restecg == 1 ~ "normal",
      restecg == 2 ~ "wave abnormality"
           ),
    target = case_when(
      target == 1 ~ "asymptomatic",
      target == 0 ~ "heart-disease"
           ),
    slope = case_when(
      slope == 2 ~ "upsloping",
      slope == 1 ~ "flat",
      slope == 0 ~ "downsloping"
    ),
```

```
    thal = case_when(
      thal == 1 ~ "fixed defect",
      thal == 2 ~ "normal",
      thal == 3 ~ "reversable defect"
    ),
    sex = as.factor(sex),
    fbs = as.factor(fbs),
    exang = as.factor(exang),
    cp = as.factor(cp),
    slope = as.factor(slope),
    ca = as.factor(ca),
    thal = as.factor(thal)
  )

glimpse(heart2) # Check that the transformation worked
```

```
## Rows: 296
## Columns: 14
## $ age      <int> 63, 37, 41, 56, 57, 57, 56, 44, 52, 57, 54, 48, 49, 64, 58, 5~
## $ sex      <fct> male, male, female, male, female, male, female, male, male, m~
## $ cp       <fct> typical angina, non-anginal, atypical angina, atypical angina~
## $ trestbps <int> 145, 130, 130, 120, 120, 140, 140, 120, 172, 150, 140, 130, 1~
## $ chol     <int> 233, 250, 204, 236, 354, 192, 294, 263, 199, 168, 239, 275, 2~
## $ fbs      <fct> >120, <=120, <=120, <=120, <=120, <=120, <=120, <=120, >120, ~
## $ restecg  <chr> "hypertrophy", "normal", "hypertrophy", "normal", "normal", "~
## $ thalach  <int> 150, 187, 172, 178, 163, 148, 153, 173, 162, 174, 160, 139, 1~
## $ exang    <fct> no, no, no, no, yes, no, no, no, no, no, no, no, no, yes, no,~
## $ oldpeak  <dbl> 2.3, 3.5, 1.4, 0.8, 0.6, 0.4, 1.3, 0.0, 0.5, 1.6, 1.2, 0.2, 0~
## $ slope    <fct> downsloping, downsloping, upsloping, upsloping, upsloping, fl~
## $ ca       <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0~
## $ thal     <fct> fixed defect, normal, normal, normal, normal, fixed defect, n~
## $ target   <chr> "asymptomatic", "asymptomatic", "asymptomatic", "asymptomatic~
```

```
plot_missing(heart2) # Check that the transformation did not induce NA values
```




```
heart <- heart2 # Replace the heart dataset by the tidy dataset
```

Data exploration

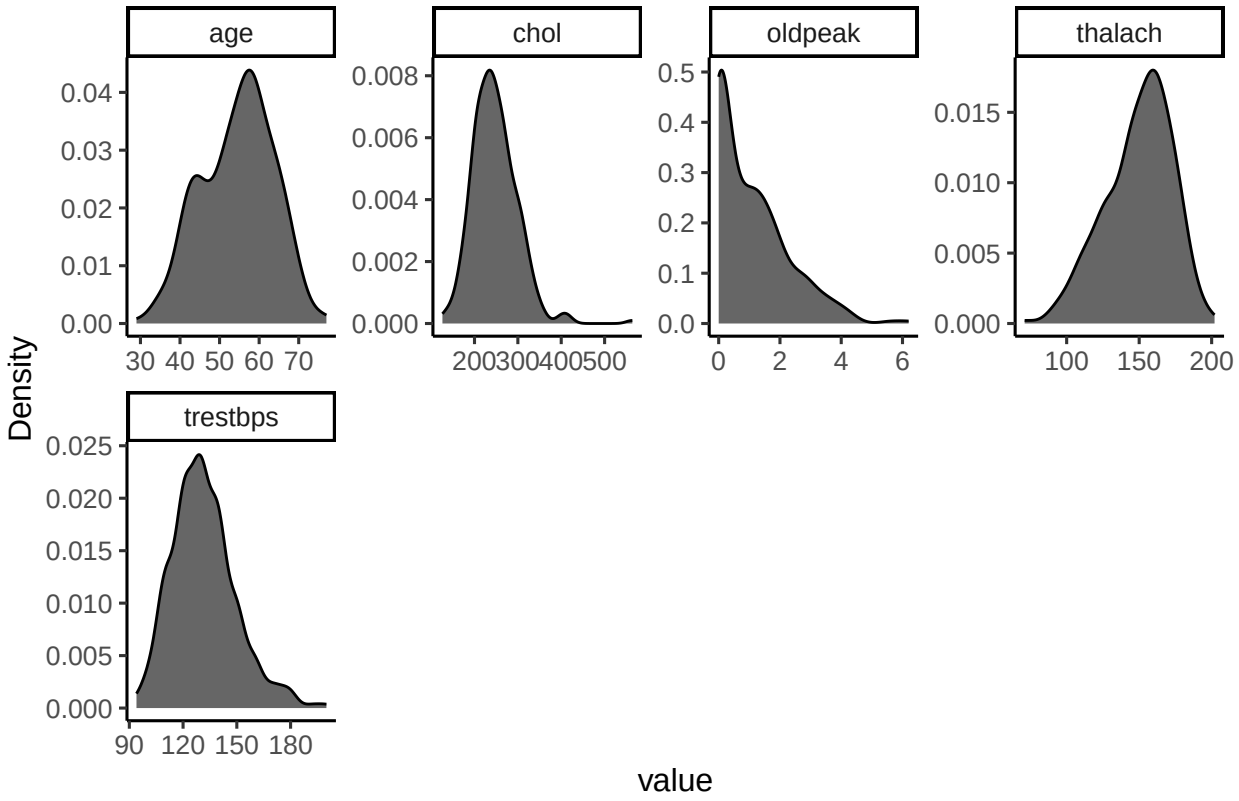
```
heart %>%
  summary()
```

```
##       age            sex                         cp         trestbps
##  Min.   :29.00   female: 95   asymptomatic angina:141   Min.   : 94.0
##  1st Qu.:48.00   male  :201   atypical angina    : 49   1st Qu.:120.0
##  Median :56.00                non-anginal        : 83   Median :130.0
##  Mean   :54.52                typical angina     : 23   Mean   :131.6
##  3rd Qu.:61.00                                          3rd Qu.:140.0
##  Max.   :77.00                                          Max.   :200.0
##       chol          fbs         restecg             thalach      exang
##  Min.   :126.0   <=120:253   Length:296         Min.   : 71.0   no :199
##  1st Qu.:211.0   >120 : 43   Class :character   1st Qu.:133.0   yes: 97
##  Median :242.5               Mode  :character   Median :152.5
##  Mean   :247.2                                  Mean   :149.6
##  3rd Qu.:275.2                                  3rd Qu.:166.0
##  Max.   :564.0                                  Max.   :202.0
##     oldpeak             slope     ca                     thal
##  Min.   :0.000   downsloping: 21   0:173   fixed defect     : 18
```

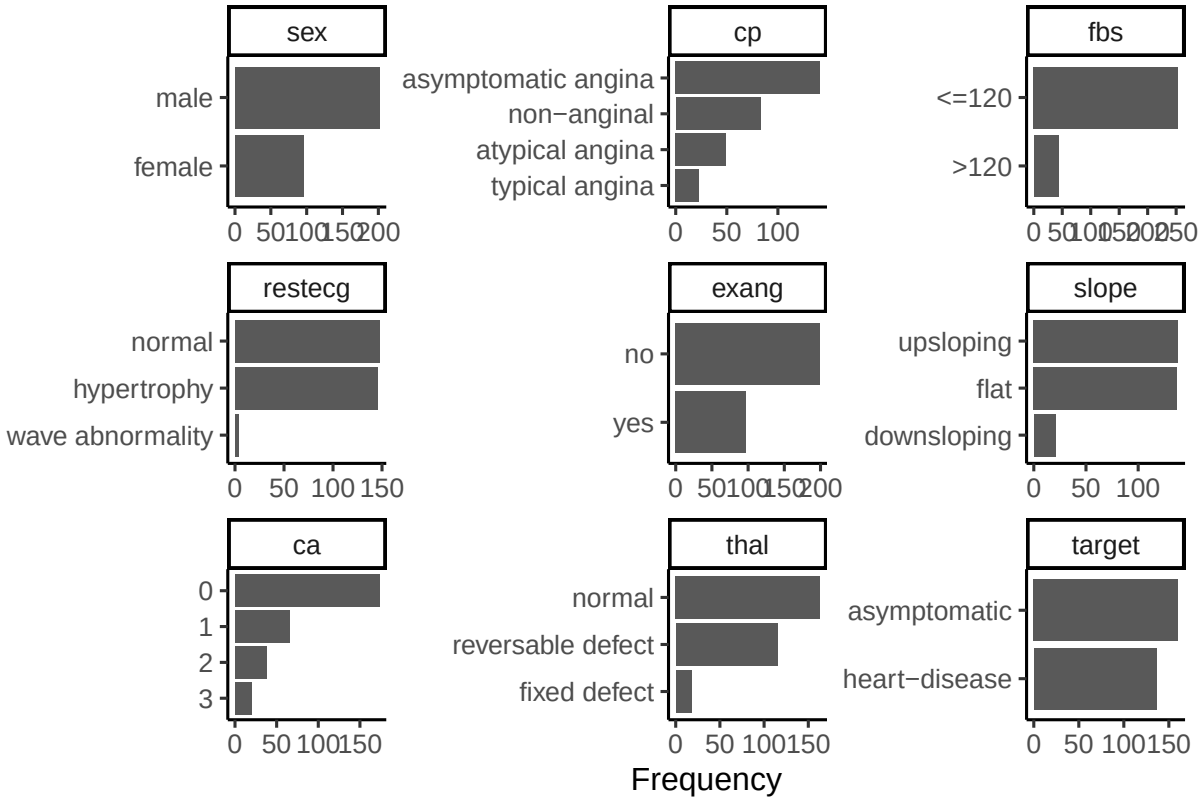
```
##  1st Qu.:0.000   flat       :137   1: 65   normal           :163
##  Median :0.800   upsloping  :138   2: 38   reversable defect:115
##  Mean   :1.059                      3: 20
##  3rd Qu.:1.650
##  Max.   :6.200
##     target
##  Length:296
##  Class :character
##  Mode  :character
##
##
##
```

Use the DataExplorer library to get a sense of the distribution of the continuous and categorical variables.

```
plot_density(heart, ggtheme = theme_classic2(), geom_density_args = list("fill" = "black", "alpha" = 0.
```



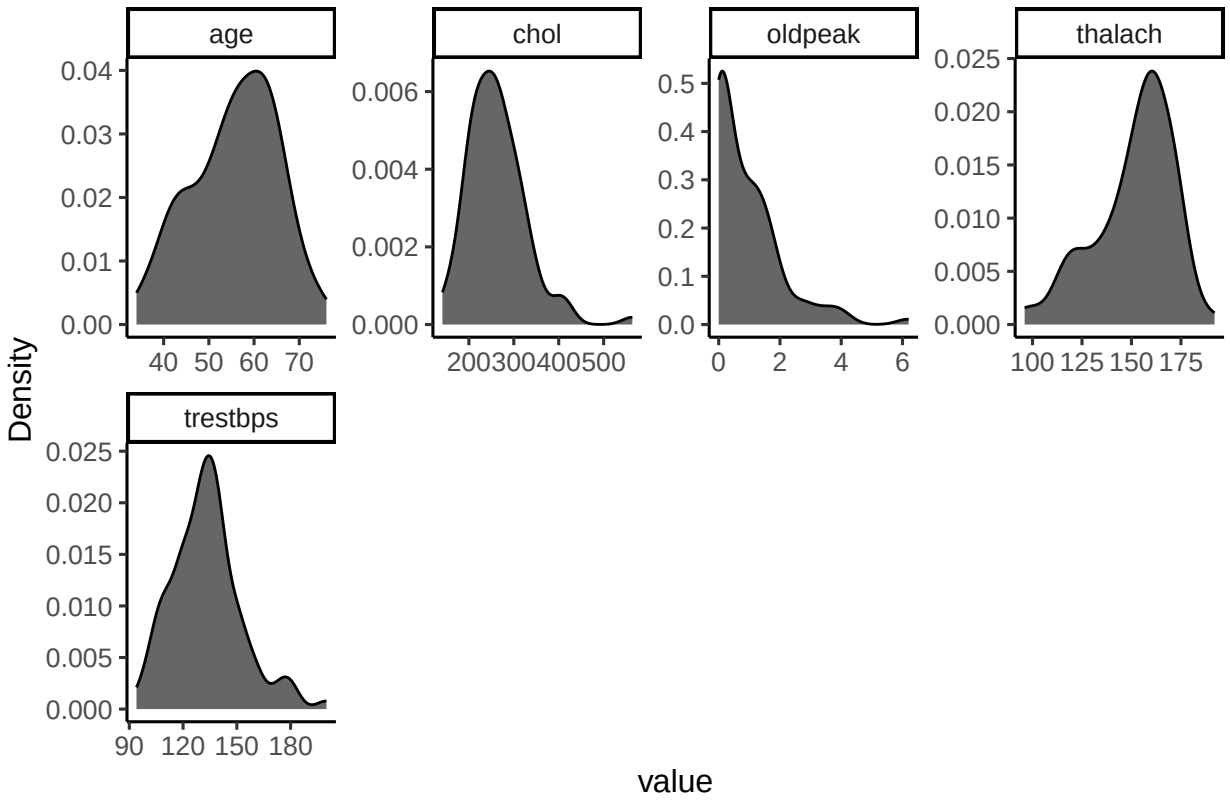

```
plot_bar(heart, ggtheme = theme_classic2())
```



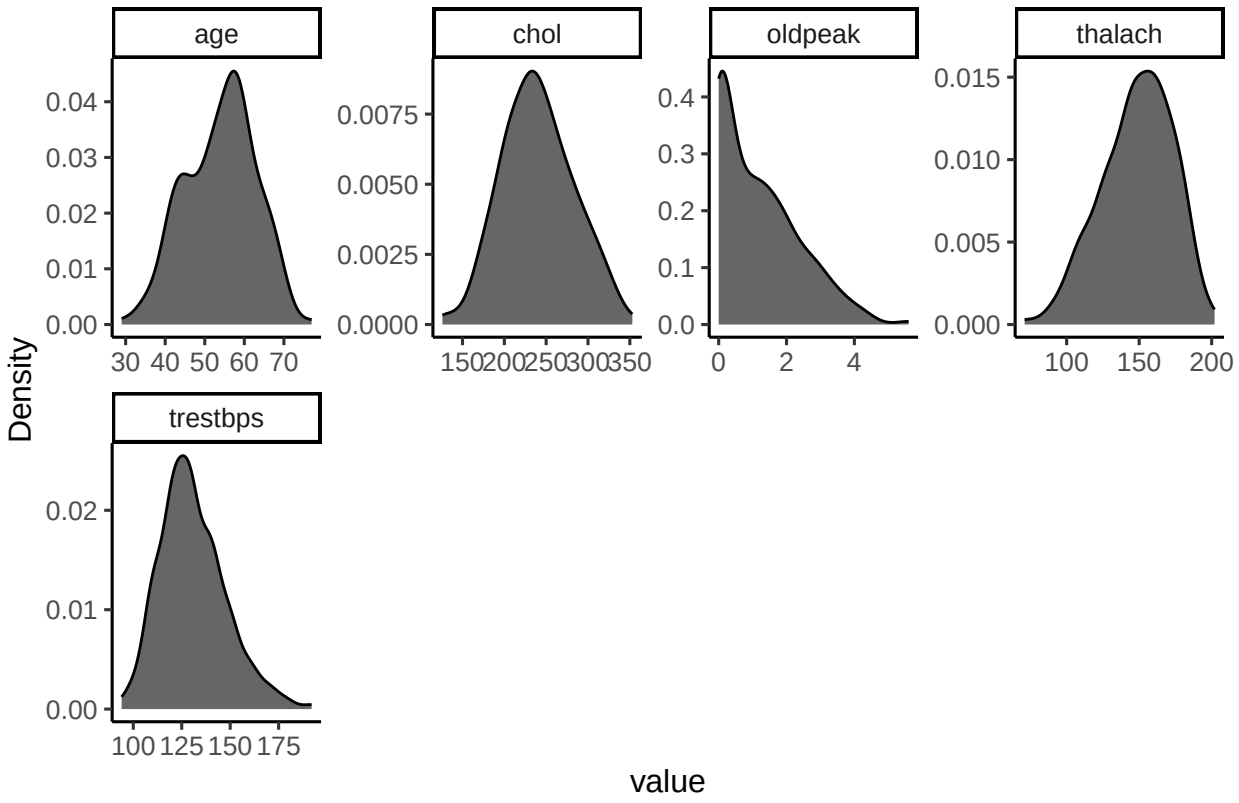
The next step is to combine dplyr and Data Explorer libraries to visualize the variables according to gender and disease.

B Analyze each variable per gender

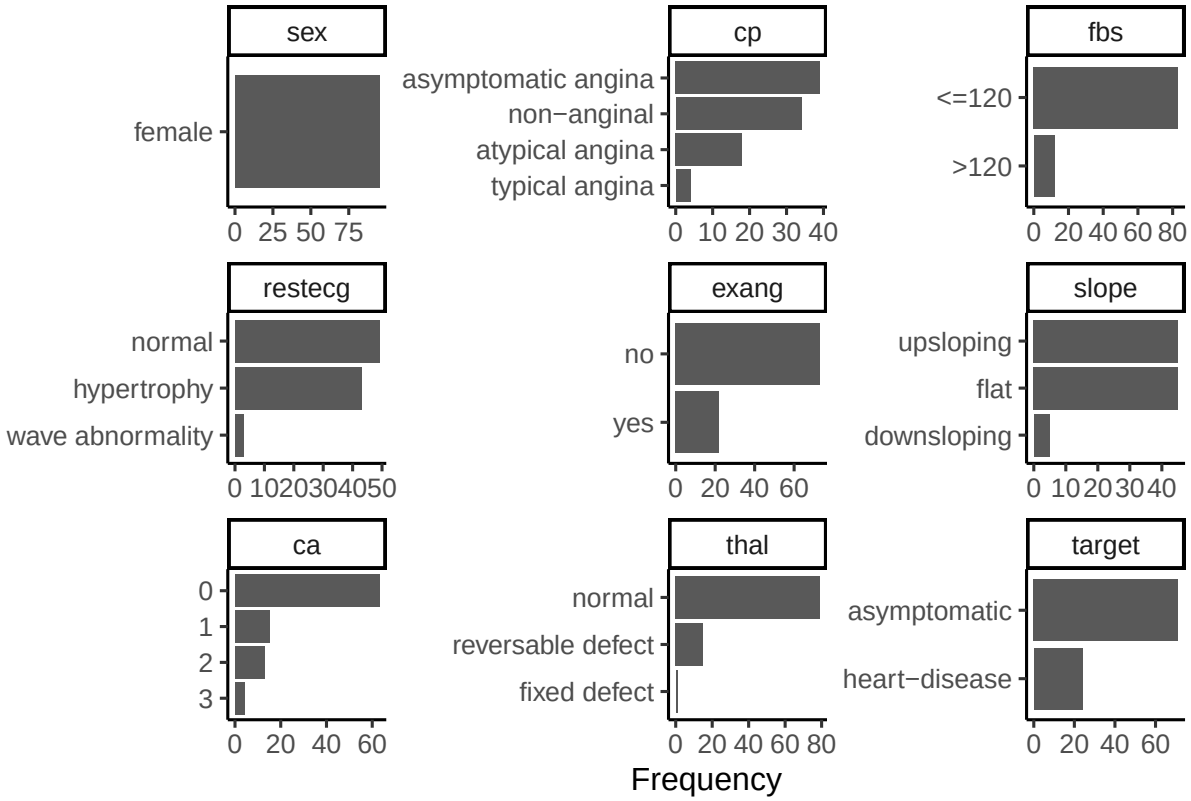
```
heart %>%
  filter(sex == "female") %>%
  plot_density(ggtheme = theme_classic2(), geom_density_args = list("fill" = "black", "alpha" = 0.6))
```



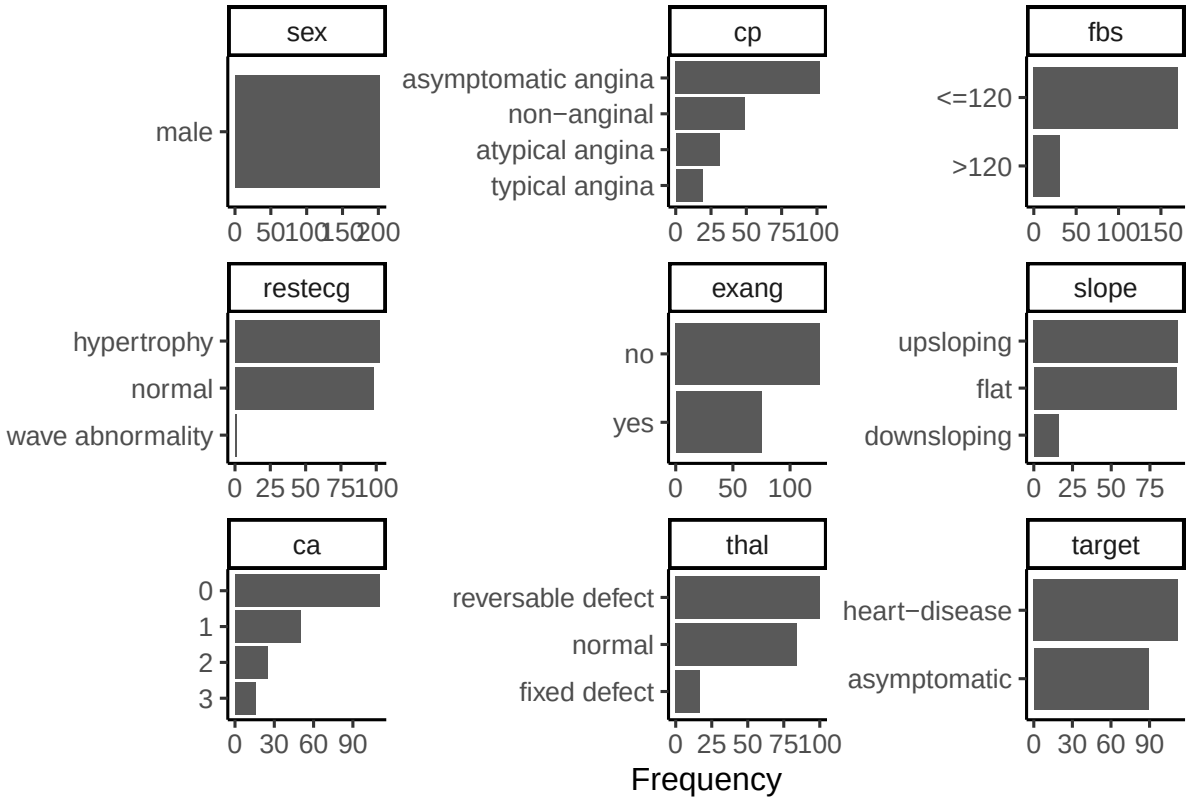

```
heart %>%
  filter(sex == "male") %>%
  plot_density(ggtheme = theme_classic2(), geom_density_args = list("fill" = "black", "alpha" = 0.6))
```




```
heart %>%
  filter(sex == "female") %>%
  plot_bar(ggtheme = theme_classic2())
```

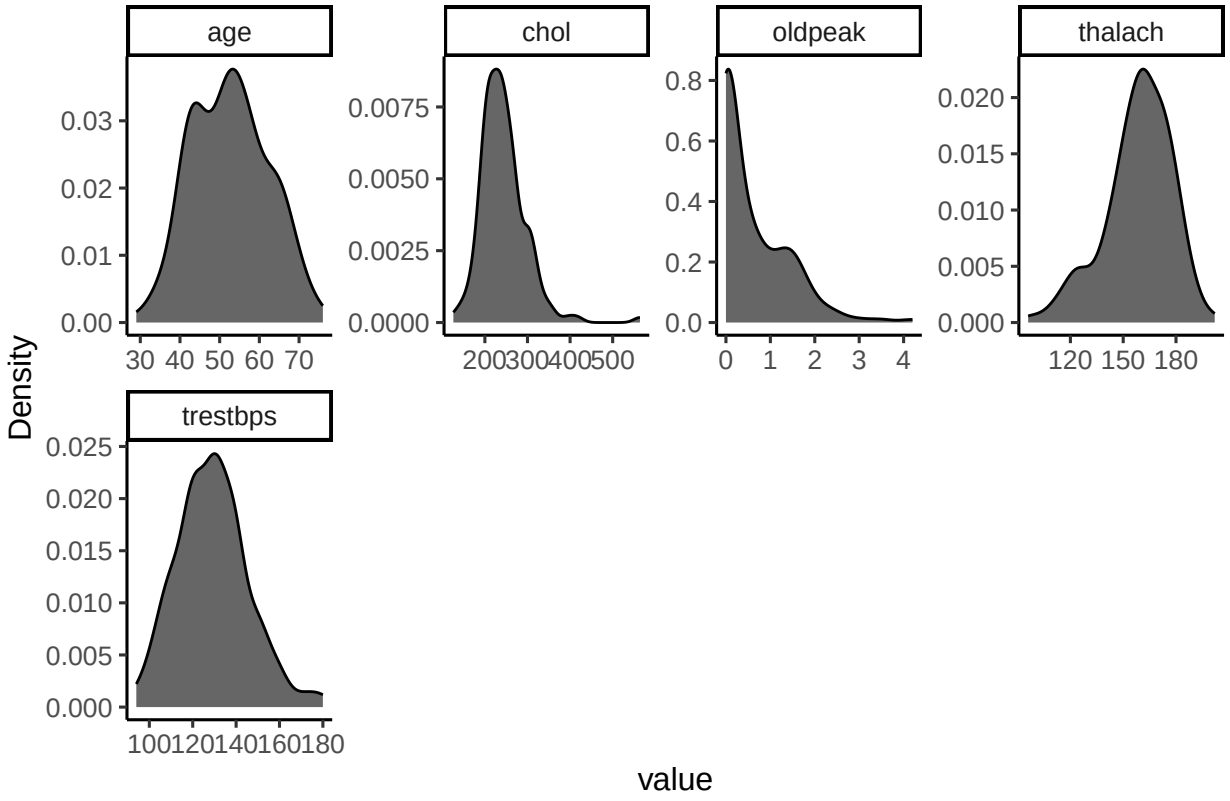



```
heart %>%
  filter(sex == "male") %>%
  plot_bar(ggtheme = theme_classic2())
```

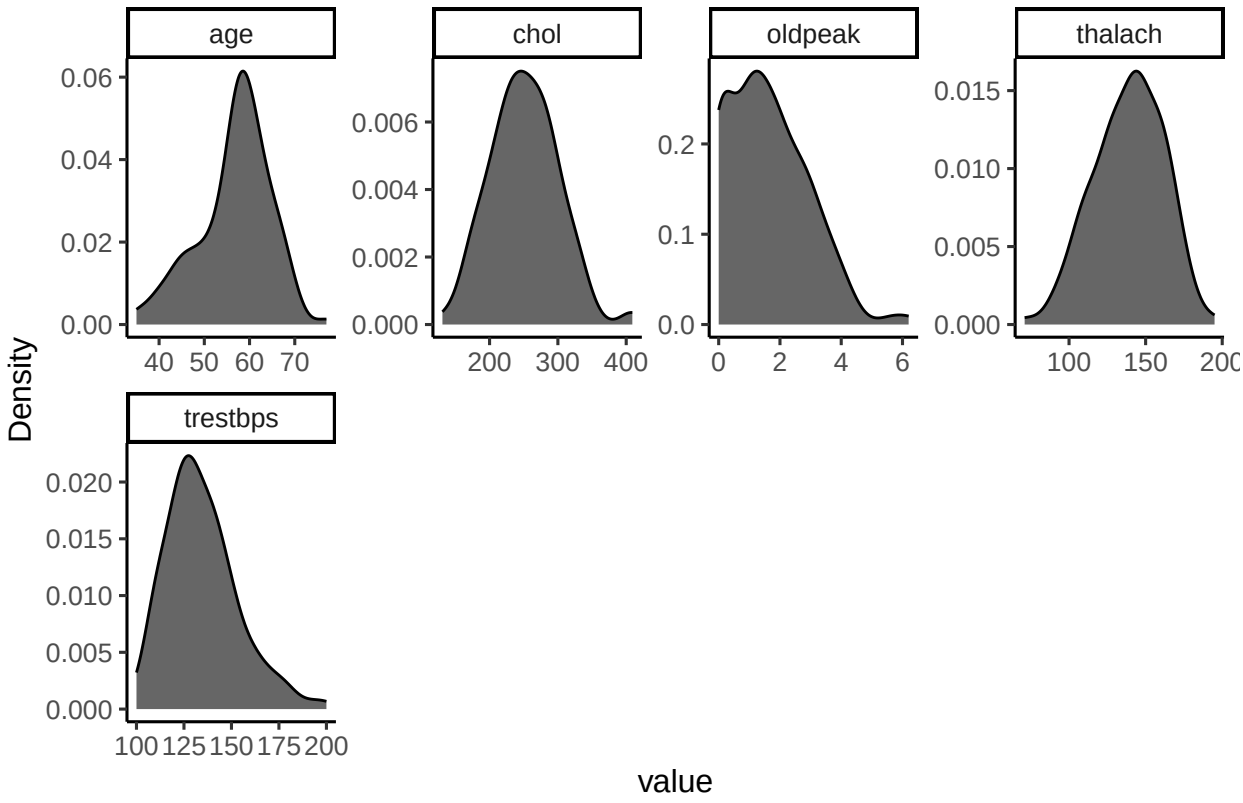


C Visualize variables per disease status

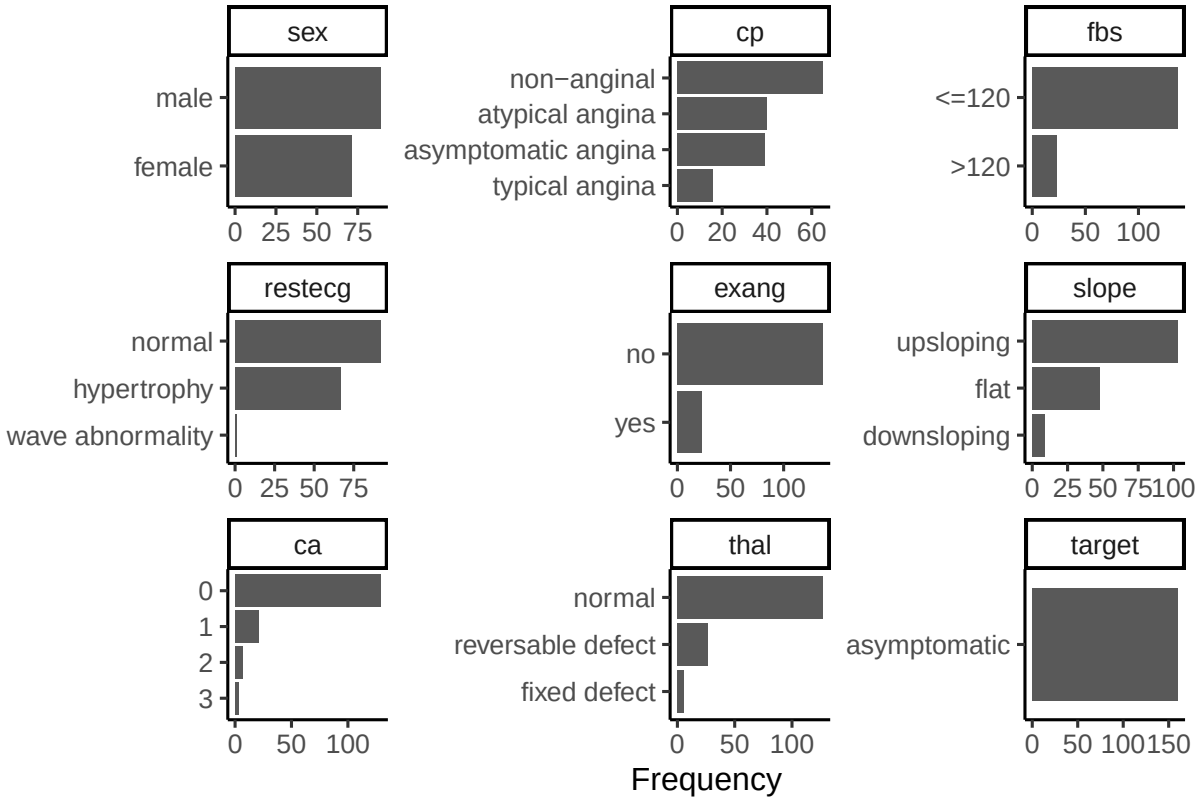
```
heart %>%
  filter(target == "asymptomatic") %>%
  plot_density(ggtheme = theme_classic2(), geom_density_args = list("fill" = "black", "alpha" = 0.6))
```



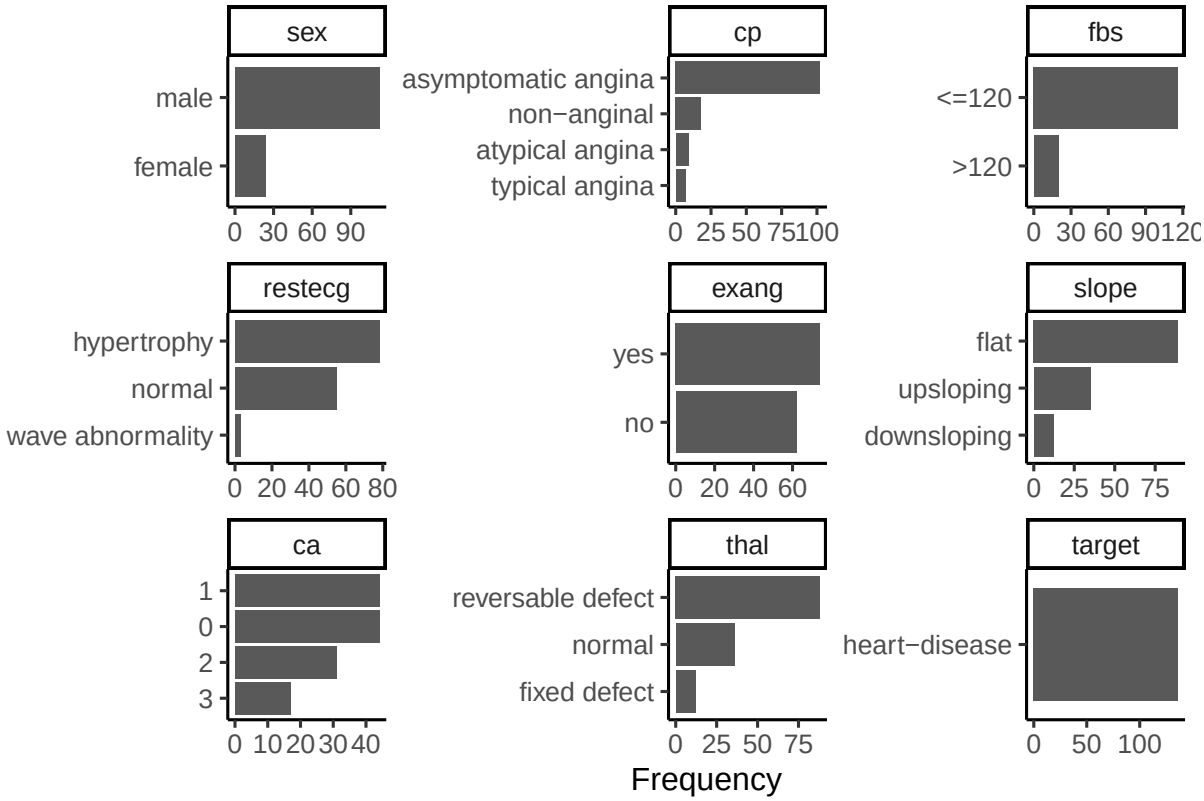

```
heart %>%
  filter(target == "heart-disease") %>%
  plot_density(ggtheme = theme_classic2(), geom_density_args = list("fill" = "black", "alpha" = 0.6))
```




```
heart %>%
  filter(target == "asymptomatic") %>%
  plot_bar(ggtheme = theme_classic2())
```

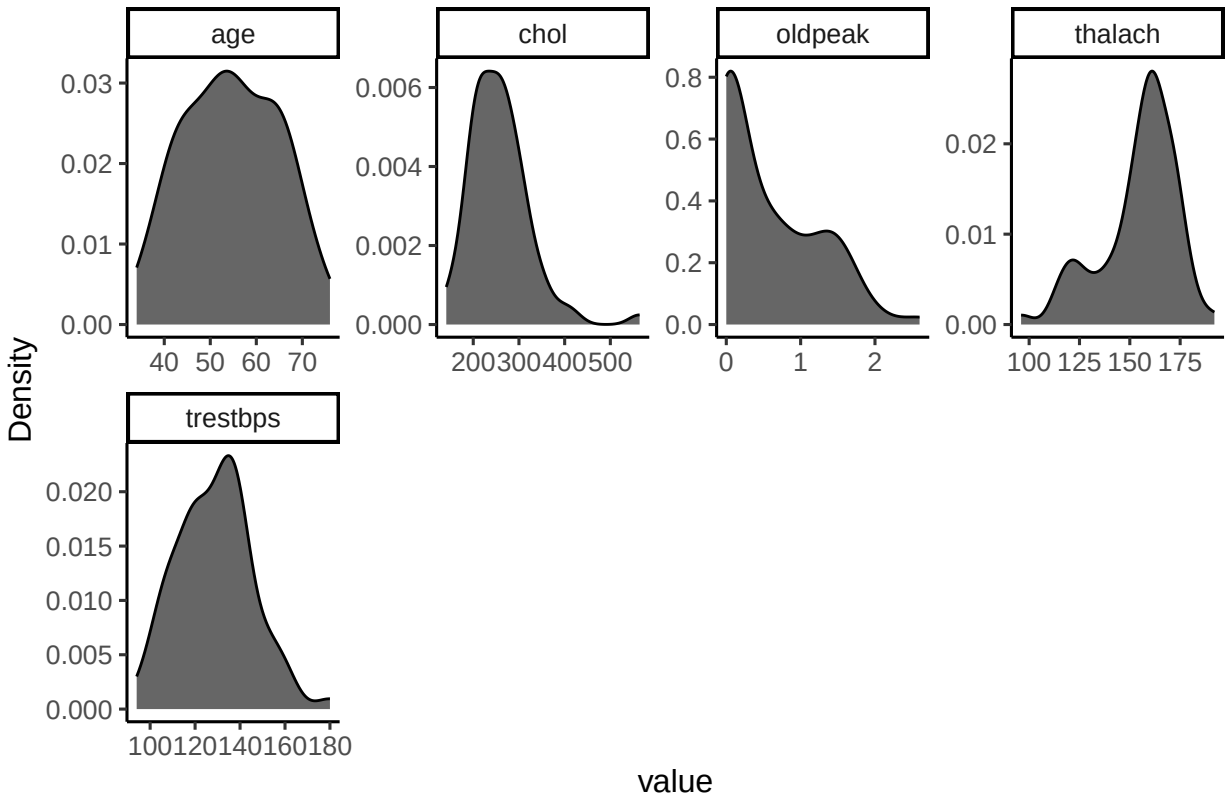



```
heart %>%
  filter(target == "heart-disease") %>%
  plot_bar(ggtheme = theme_classic2())
```

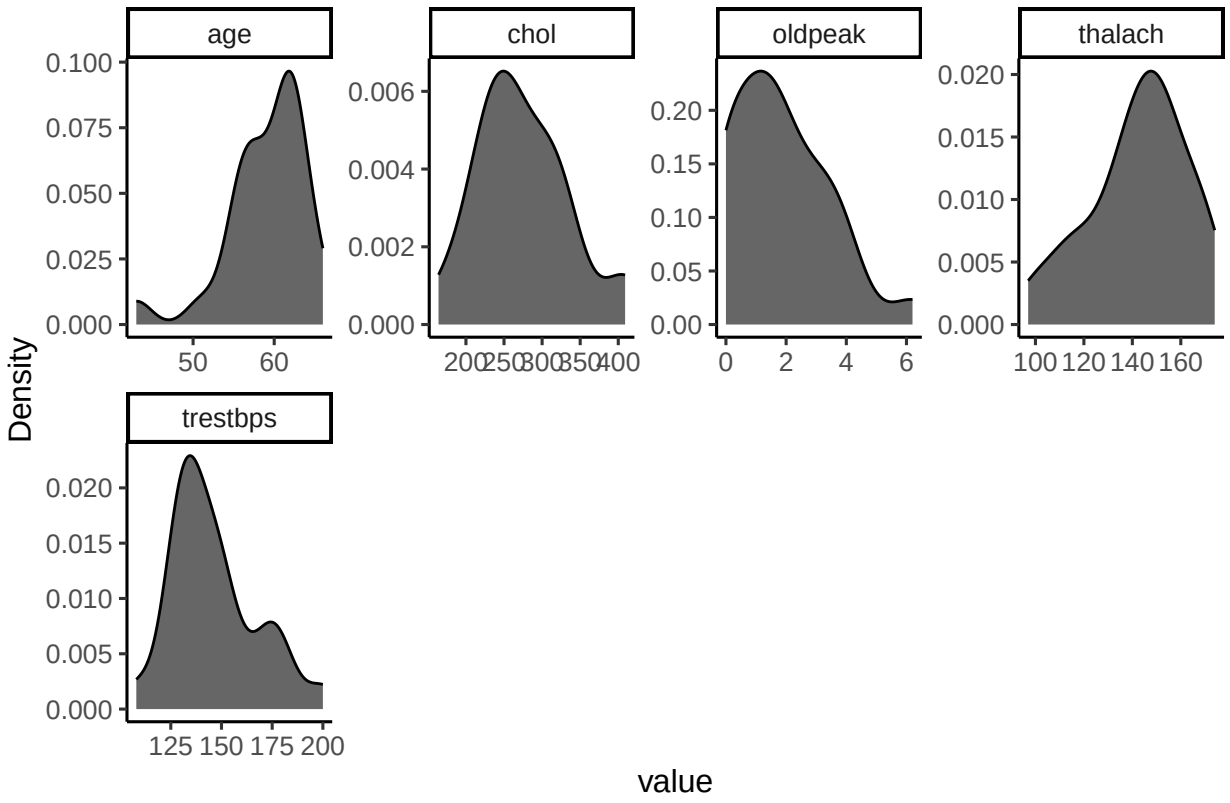


Visualize the data per gender and disease status

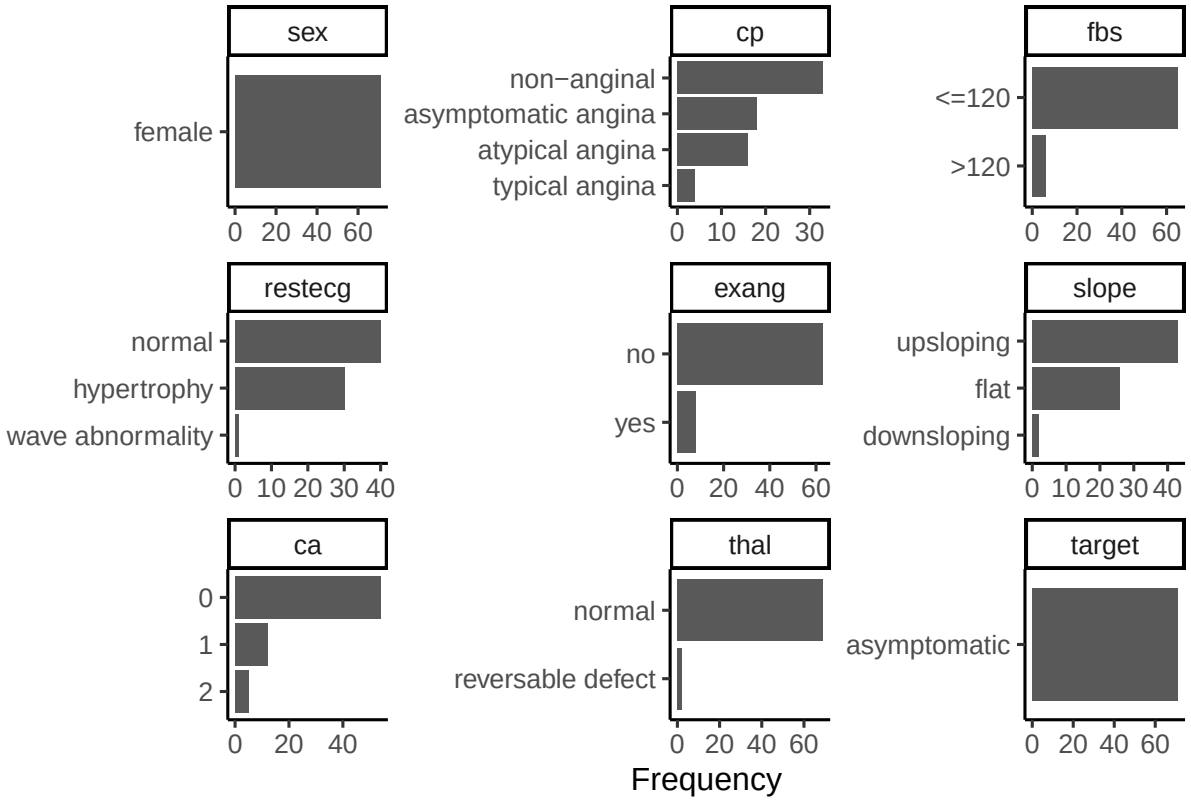
```
heart %>%
  filter(sex == "female", target == "asymptomatic") %>%
  plot_density(ggtheme = theme_classic2(), geom_density_args = list("fill" = "black", "alpha" = 0.6))
```



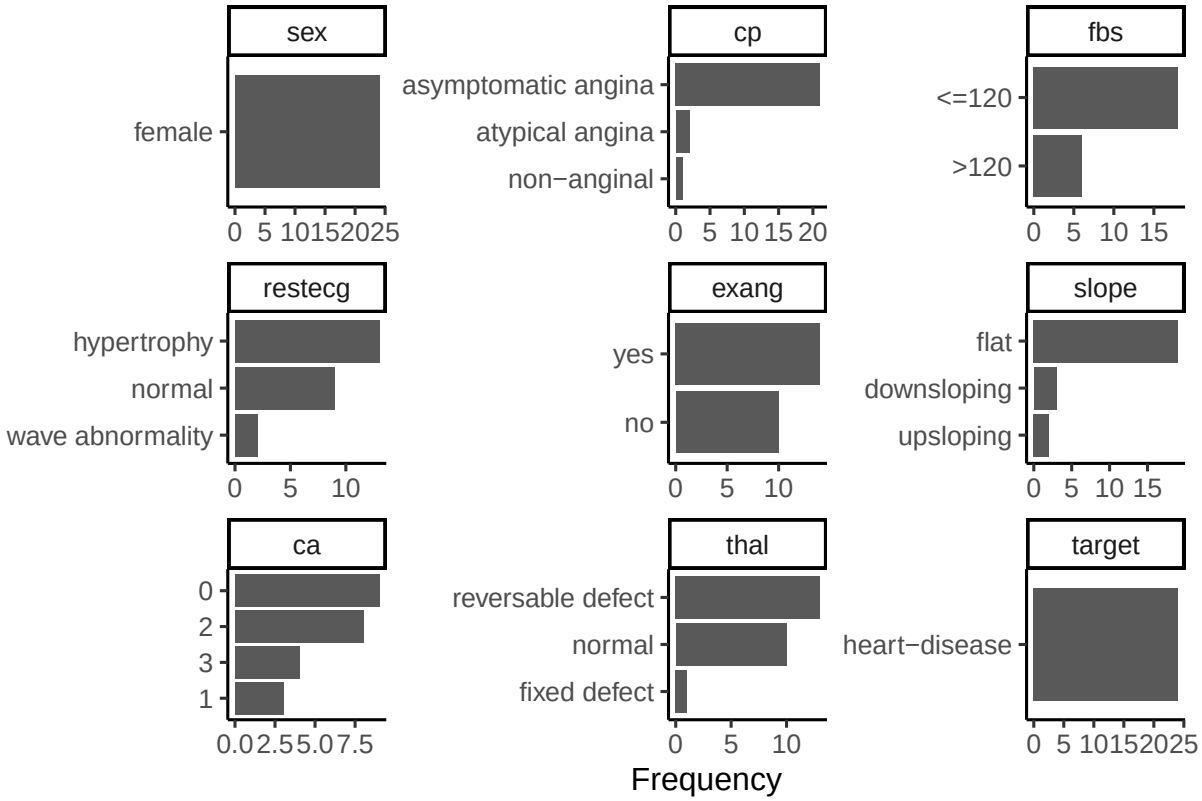

```
heart %>%
  filter(sex == "female", target == "heart-disease") %>%
  plot_density(ggtheme = theme_classic2(), geom_density_args = list("fill" = "black", "alpha" = 0.6))
```



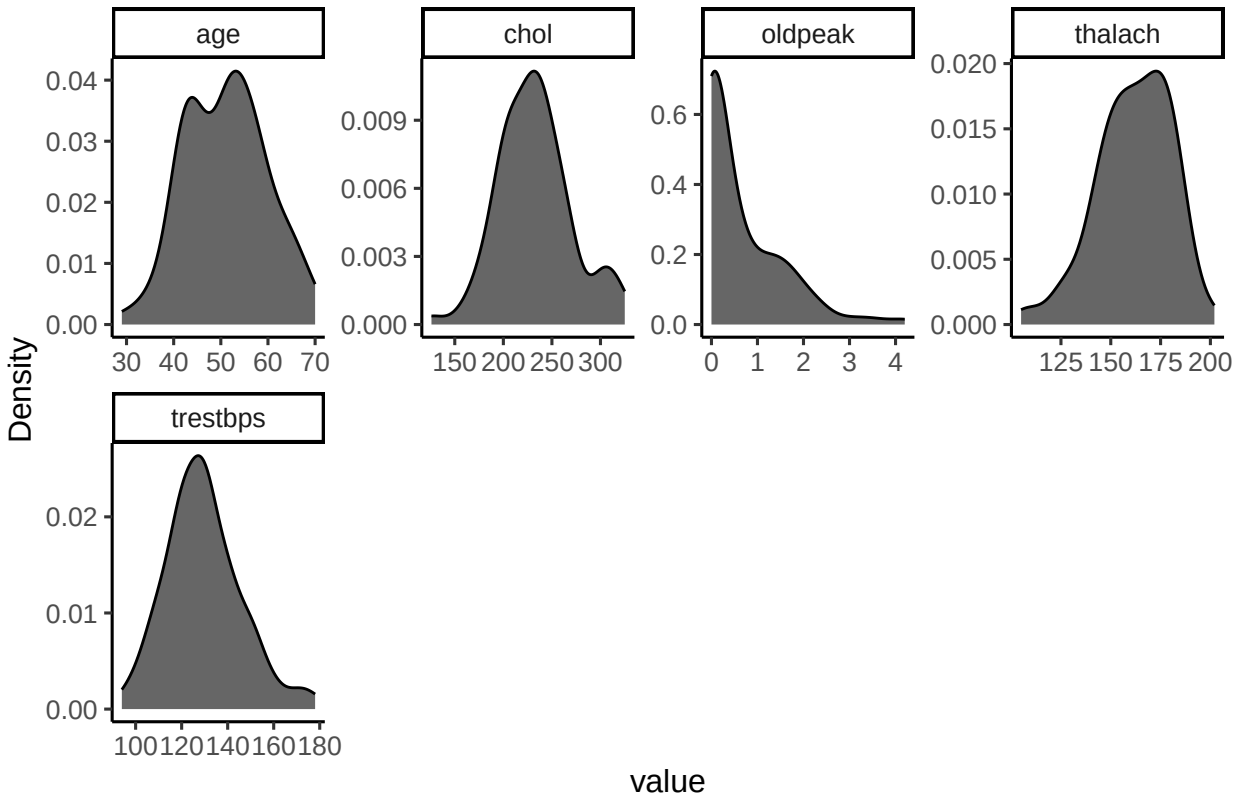

```
heart %>%
  filter(sex == "female", target == "asymptomatic") %>%
  plot_bar(ggtheme = theme_classic2())
```



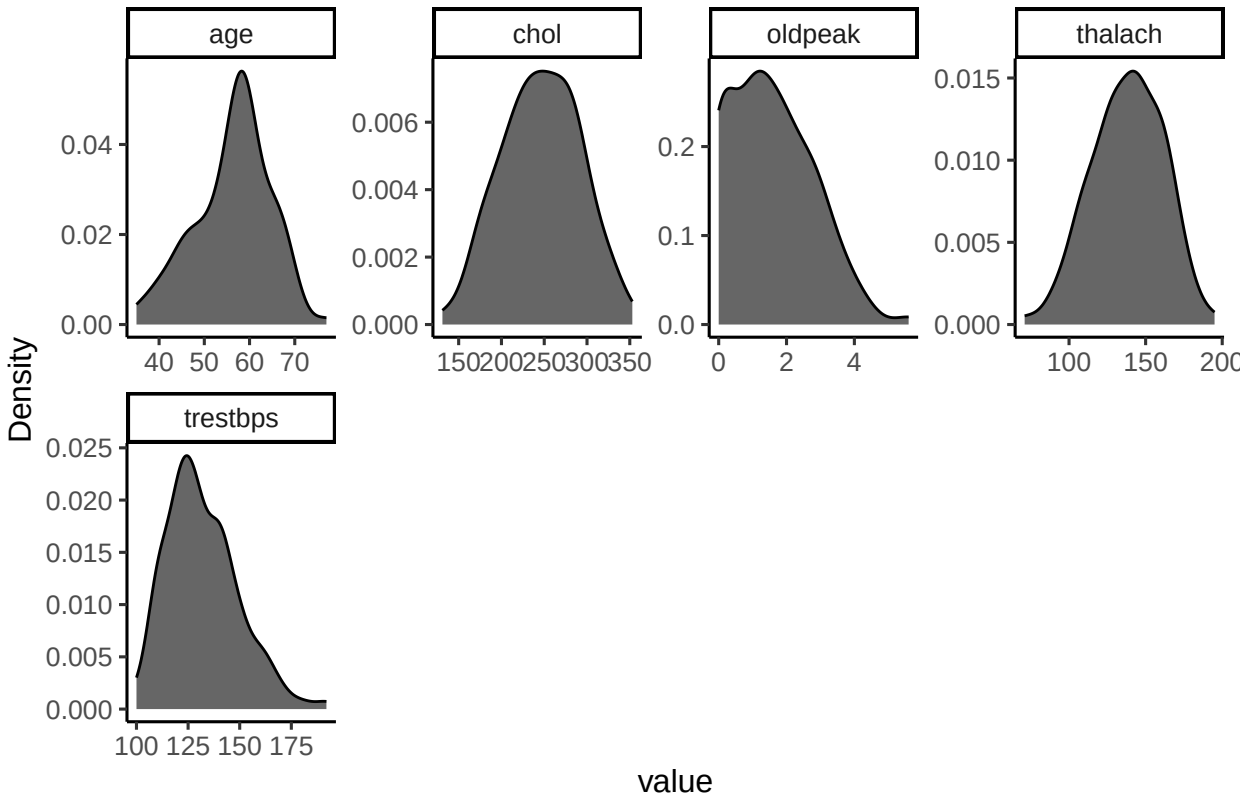

```
heart %>%
  filter(sex == "female", target == "heart-disease") %>%
  plot_bar(ggtheme = theme_classic2())
```



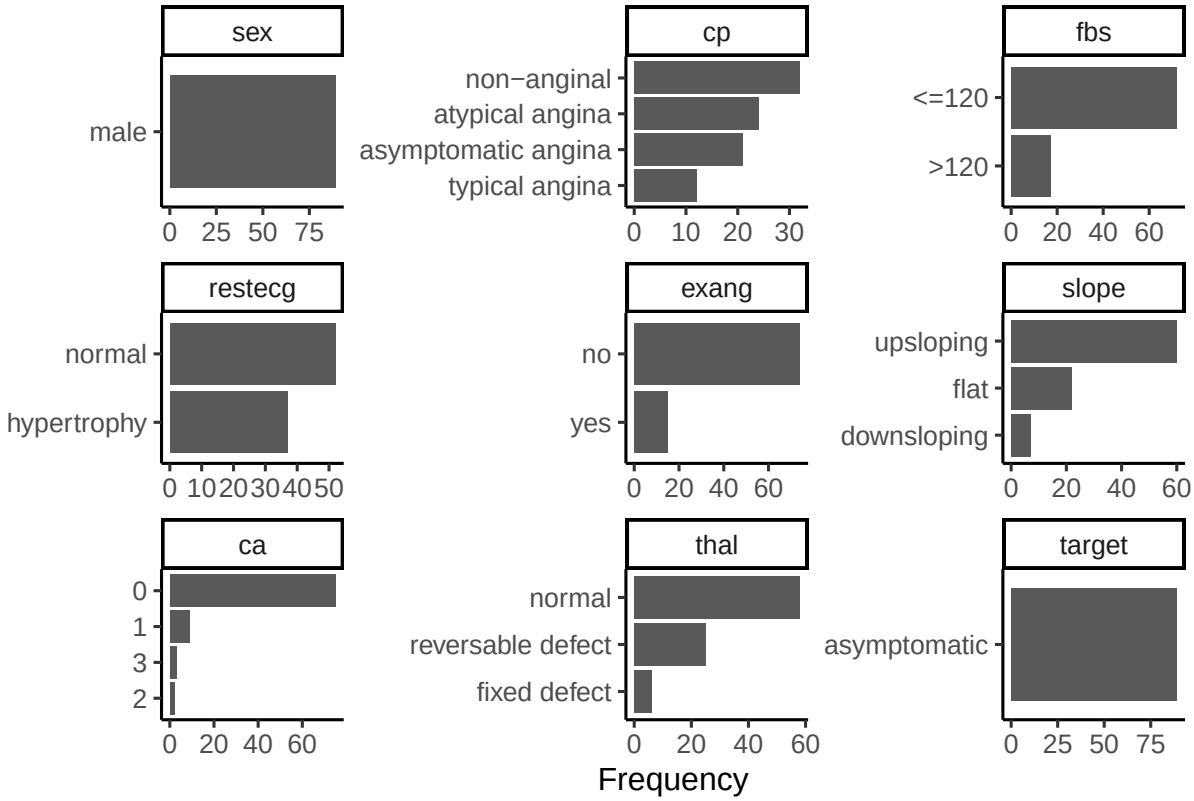

```
heart %>%
  filter(sex == "male", target == "asymptomatic") %>%
  plot_density(ggtheme = theme_classic2(), geom_density_args = list("fill" = "black", "alpha" = 0.6))
```



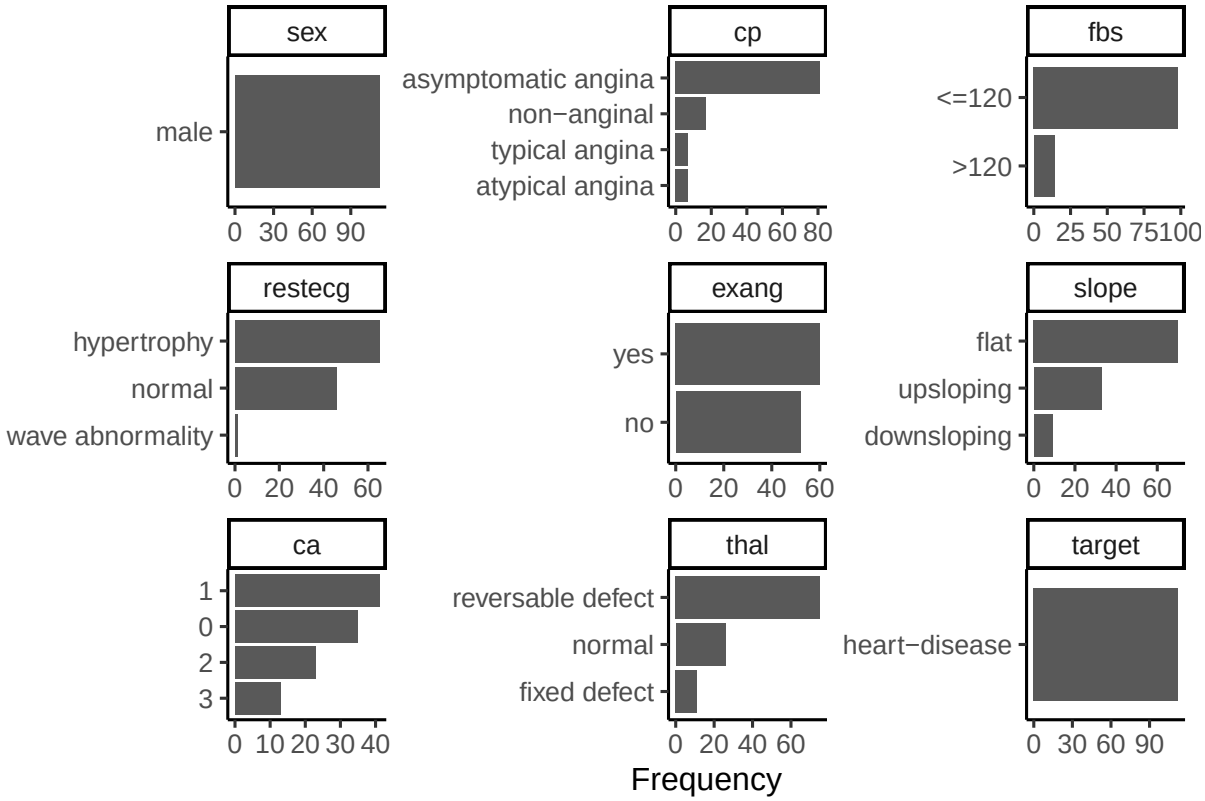

```
heart %>%
  filter(sex == "male", target == "heart-disease") %>%
  plot_density(ggtheme = theme_classic2(), geom_density_args = list("fill" = "black", "alpha" = 0.6))
```




```
heart %>%
  filter(sex == "male", target == "asymptomatic") %>%
  plot_bar(ggtheme = theme_classic2())
```




```
heart %>%
  filter(sex == "male", target == "heart-disease") %>%
  plot_bar(ggtheme = theme_classic2())
```



Prepare a summary table per disease and gender

```
heart %>%
  group_by(target, sex) %>%
  summarise(
    n_disease = n(),
    mean_age = round(mean(age), digits=2),
    sd_age = round(sd(age), digits=2),
    mean_trestbps = round(mean(trestbps), digits=2),
    sd_trestbps = round(sd(trestbps), digits=2),
    mean_chol = round(mean(chol), digits=2),
    sd_chol = round(sd(chol), digits=2),
    mean_thalach = round(mean(thalach), digits=2),
    sd_thalach = round(sd(thalach), digits=2),
    mean_oldpeak = round(mean(oldpeak), digits=2),
    sd_oldpeak = round(sd(oldpeak), digits=2)
  )
```

```
## `summarise()` has grouped output by 'target'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 4 x 13
## # Groups:   target [2]
```

```
##   target    sex   n_dis~1 mean_~2 sd_age mean_~3 sd_tr~4 mean_~5 sd_chol mean_~6
##   <chr>     <fct>   <int>   <dbl>  <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 asymptom~ fema~      71    54.6  10.3     129.    16.6    257.    66.5    155.
## 2 asymptom~ male       89    51.1   8.63    130.    16.2    232.    37.8    162.
## 3 heart-di~ fema~      24    59.0   4.96    146.    21.4    275.    60.9    142.
## 4 heart-di~ male      112    56.2   8.36    132.    17.4    246.    45.7    138.
## # ... with 3 more variables: sd_thalach <dbl>, mean_oldpeak <dbl>,
## #   sd_oldpeak <dbl>, and abbreviated variable names 1: n_disease, 2: mean_age,
## #   3: mean_trestbps, 4: sd_trestbps, 5: mean_chol, 6: mean_thalach
```

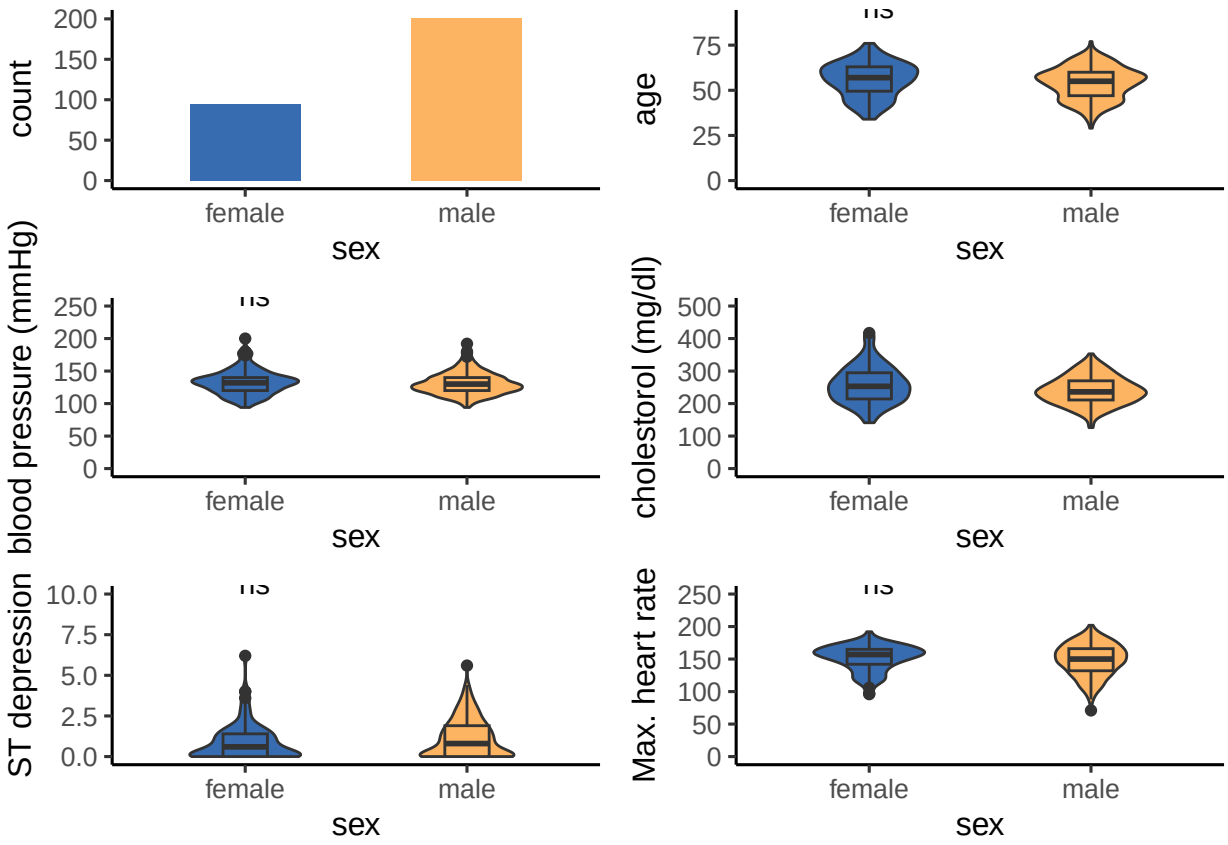
Visualization

From the Exploratory Data analysis, it seems that several differences are statistically significant according to gender and health status.

A Visualization of variables per gender

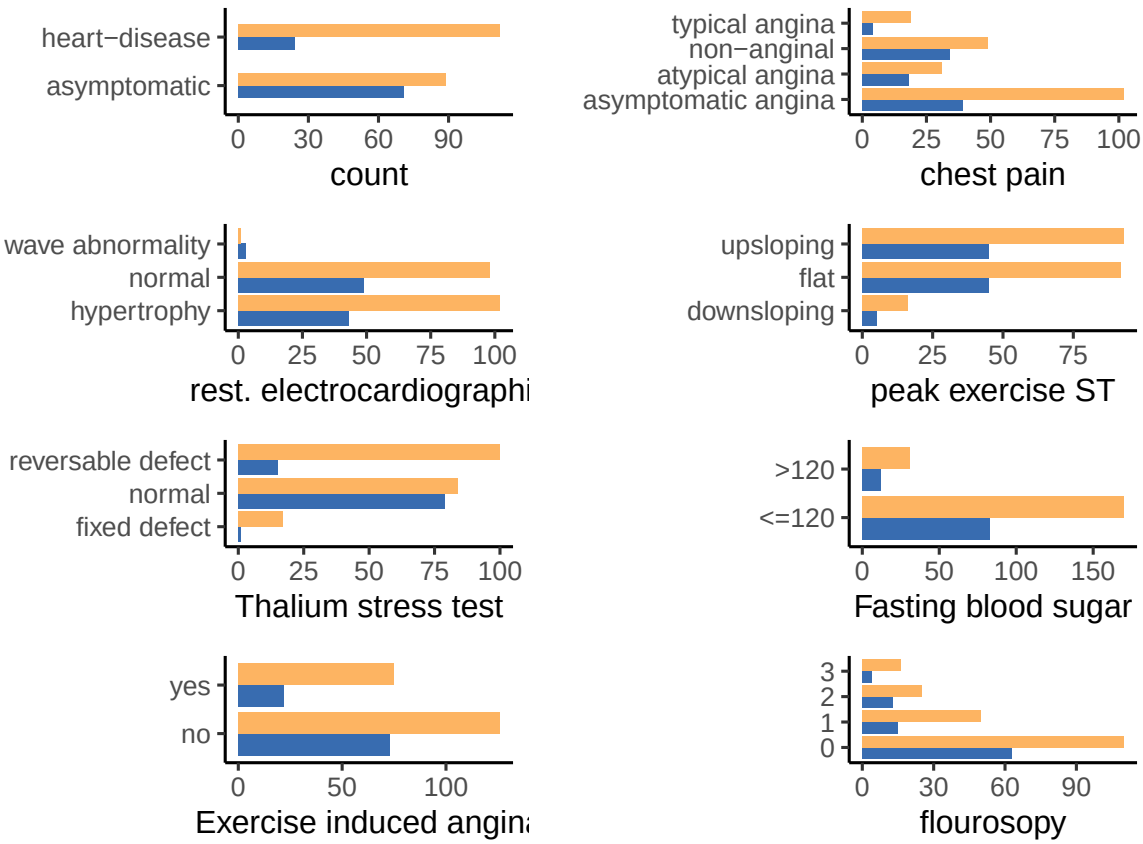
```
# Male and Female count
a1 <- ggplot(heart, aes(x = sex, fill = sex)) +
  geom_bar(width = 0.5) +
  scale_fill_manual(values = c("#386cb0","#fdb462"))+
  theme_classic2() +
  theme(legend.position='none')

# Age per gender
b1 <- ggplot(heart, aes(x= sex, y = age, fill = sex)) +
  geom_violin(width = 0.5) +
  geom_boxplot(width = 0.2) +
  ylim(0, 90) +
  stat_compare_means(aes(label = ..p.signif..), method = "t.test") +
  scale_fill_manual(values = c("#386cb0","#fdb462"))+
  theme_classic2() +
  theme(legend.position='none')

# trestbps
c1 <- ggplot(heart, aes(x = sex, y = trestbps, fill = sex)) +
  geom_violin(width = 0.5) +
  geom_boxplot(width = 0.2) +
  labs(y = "blood pressure (mmHg)") +
  ylim(0,250) +
  stat_compare_means(aes(label = ..p.signif..), method = "t.test") +
  scale_fill_manual(values = c("#386cb0","#fdb462"))+
  theme_classic2() +
  theme(legend.position='none')

# chol
d1 <- ggplot(heart, aes(x = sex, y = chol, fill = sex)) +
  geom_violin(width = 0.5) +
  geom_boxplot(width = 0.2) +
  labs(y = "cholestorol (mg/dl)") +
  ylim(0,500) +
```

```
  stat_compare_means(aes(label = ..p.signif..), method = "t.test") +
  scale_fill_manual(values = c("#386cb0","#fdb462"))+
  theme_classic2() +
  theme(legend.position='none')

# oldpeak
e1 <- ggplot(heart, aes(x = sex, y = oldpeak, fill = sex)) +
  geom_violin(width = 0.5) +
  geom_boxplot(width = 0.2) +
  labs(y = "ST depression") +
  ylim(0,10) +
  stat_compare_means(aes(label = ..p.signif..), method = "t.test") +
  scale_fill_manual(values = c("#386cb0","#fdb462"))+
  theme_classic2() +
  theme(legend.position='none')

# thalach
f1 <- ggplot(heart, aes(x = sex, y = thalach, fill = sex)) +
  geom_violin(width = 0.5) +
  geom_boxplot(width = 0.2) +
  labs(y = "Max. heart rate") +
  ylim(0,250) +
  stat_compare_means(aes(label = ..p.signif..), method = "t.test") +
  scale_fill_manual(values = c("#386cb0","#fdb462"))+
  theme_classic2() +
  theme(legend.position='none')

suppressWarnings(ggarrange(a1, b1, c1, d1, e1, f1,
          ncol = 2, nrow = 3,
            align = "v"))
```




```r
# Disease status
g1 <- ggplot(heart, aes(x = target, fill = sex)) +
  geom_bar(width = 0.5, position = 'dodge') +
  labs(x = "") +
  coord_flip() +
  scale_fill_manual(values = c("#386cb0","#fdb462"))+
  theme_classic2() +
  theme(legend.position='none')

# cp
h1 <- ggplot(heart, aes(cp, group = sex, fill = sex)) +
  geom_bar(position = "dodge") +
  labs(x = "", y = "chest pain") +
  coord_flip() +
  scale_fill_manual(values = c("#386cb0","#fdb462"))+
  theme_classic2() +
  theme(legend.position='none')

# restecg
i1 <- ggplot(heart, aes(restecg, group = sex, fill = sex)) +
  geom_bar(position = "dodge") +
  labs(x = "", y = "rest. electrocardiographic") +
  coord_flip() +
  scale_fill_manual(values = c("#386cb0","#fdb462"))+
  theme_classic2() +
  theme(legend.position='none')
```

```r
# slope
j1 <- ggplot(heart, aes(slope, group = sex, fill = sex)) +
  geom_bar(position = "dodge") +
  labs(x = "", y = "peak exercise ST") +
  coord_flip() +
  scale_fill_manual(values = c("#386cb0","#fdb462"))+
  theme_classic2() +
  theme(legend.position='none')

# thal
k1 <- ggplot(heart, aes(thal, group = sex, fill = sex)) +
  geom_bar(position = "dodge") +
  labs(x = "", y = "Thalium stress test") +
  coord_flip() +
  scale_fill_manual(values = c("#386cb0","#fdb462"))+
  theme_classic2() +
  theme(legend.position='none')

# fbp
l1 <- ggplot(heart, aes(fbs, group = sex, fill = sex)) +
  geom_bar(position = "dodge") +
  labs(x = "", y = "Fasting blood sugar") +
  coord_flip() +
  scale_fill_manual(values = c("#386cb0","#fdb462"))+
  theme_classic2() +
  theme(legend.position='none')

# exang
m1 <- ggplot(heart, aes(exang, group = sex, fill = sex)) +
  geom_bar(position = "dodge") +
  labs(x = "", y = "Exercise induced angina") +
  coord_flip() +
  scale_fill_manual(values = c("#386cb0","#fdb462"))+
  theme_classic2() +
  theme(legend.position='none')

# ca
n1 <- ggplot(heart, aes(ca, group = sex, fill = sex)) +
  geom_bar(position = "dodge") +
  labs(x = "", y = "flourosopy") +
  coord_flip() +
  scale_fill_manual(values = c("#386cb0","#fdb462"))+
  theme_classic2() +
  theme(legend.position='none')

ggarrange(g1, h1, i1, j1, k1, l1, m1, n1,
          ncol = 2, nrow = 4,
          align = "v")
```



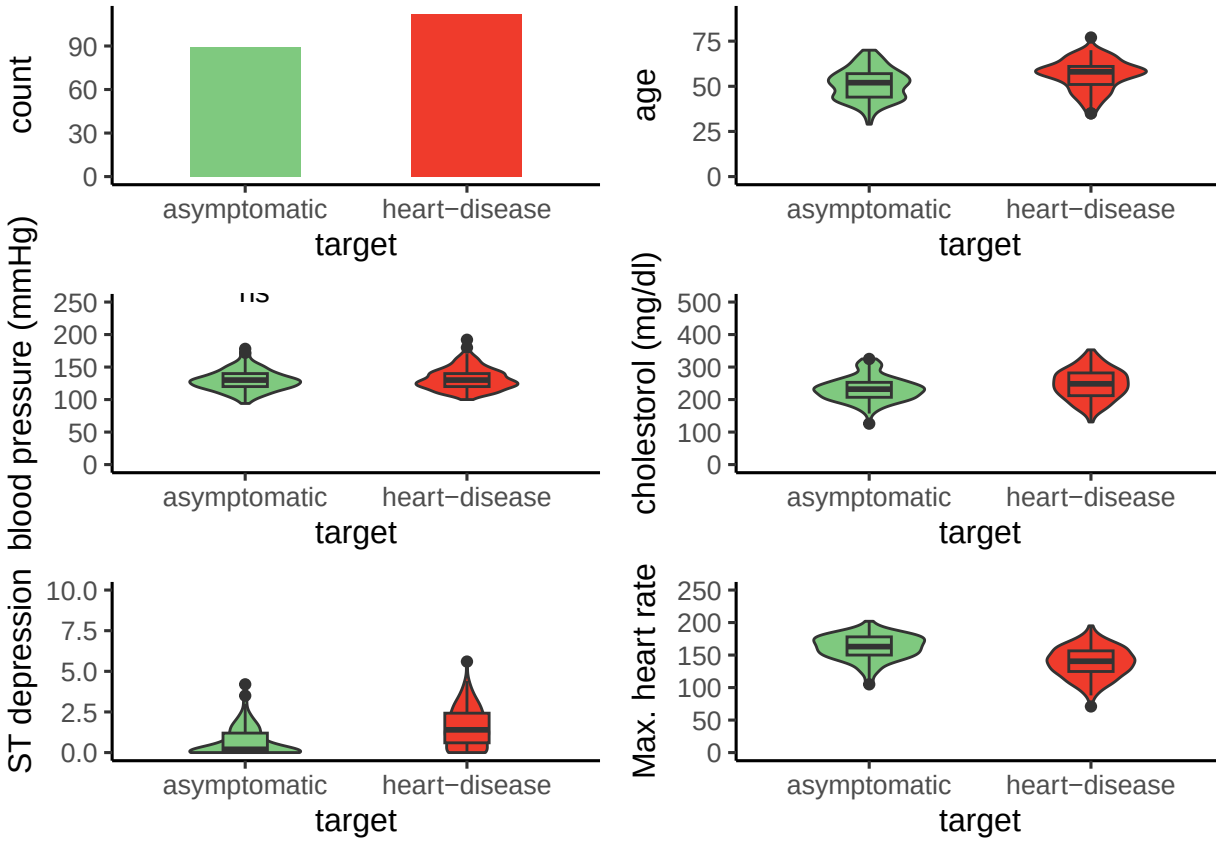
From this first plot, it appears that this dataset contains more males patients with a higher proportion of heart disease compared to female patients.

B Visualization of variables per disease status

```
heart <- heart2 %>%
    filter(sex == "male")
```

```
# Male and Female count
a2 <- ggplot(heart, aes(x = target, fill = target)) +
  geom_bar(width = 0.5, position = 'dodge') +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# Age per gender
b2 <- ggplot(heart, aes(x= target, y = age, fill = target)) +
  geom_violin(width = 0.5) +
  geom_boxplot(width = 0.2) +
  ylim(0, 90) +
  stat_compare_means(aes(label = ..p.signif..), method = "t.test") +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
```

```
  theme_classic2() +
  theme(legend.position='none')

# trestbps
c2 <- ggplot(heart, aes(x = target, y = trestbps, fill = target)) +
  geom_violin(width = 0.5) +
  geom_boxplot(width = 0.2) +
  labs(y = "blood pressure (mmHg)") +
  ylim(0,250) +
 stat_compare_means(aes(label = ..p.signif..), method = "t.test") +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# chol
d2 <- ggplot(heart, aes(x = target, y = chol, fill = target)) +
  geom_violin(width = 0.5) +
  geom_boxplot(width = 0.2) +
  labs(y = "cholestorol (mg/dl)") +
  ylim(0,500) +
  stat_compare_means(aes(label = ..p.signif..), method = "t.test") +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# oldpeak
e2 <- ggplot(heart, aes(x = target, y = oldpeak, fill = target)) +
  geom_violin(width = 0.5) +
  geom_boxplot(width = 0.2) +
  labs(y = "ST depression") +
  ylim(0,10) +
  stat_compare_means(aes(label = ..p.signif..), method = "t.test") +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# thalach
f2 <- ggplot(heart, aes(x = target, y = thalach, fill = target)) +
  geom_violin(width = 0.5) +
  geom_boxplot(width = 0.2) +
  labs(y = "Max. heart rate") +
  ylim(0,250) +
  stat_compare_means(aes(label = ..p.signif..), method = "t.test") +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

ggarrange(a2, b2, c2, d2, e2, f2,
          ncol = 2, nrow = 3,
            align = "v")
```

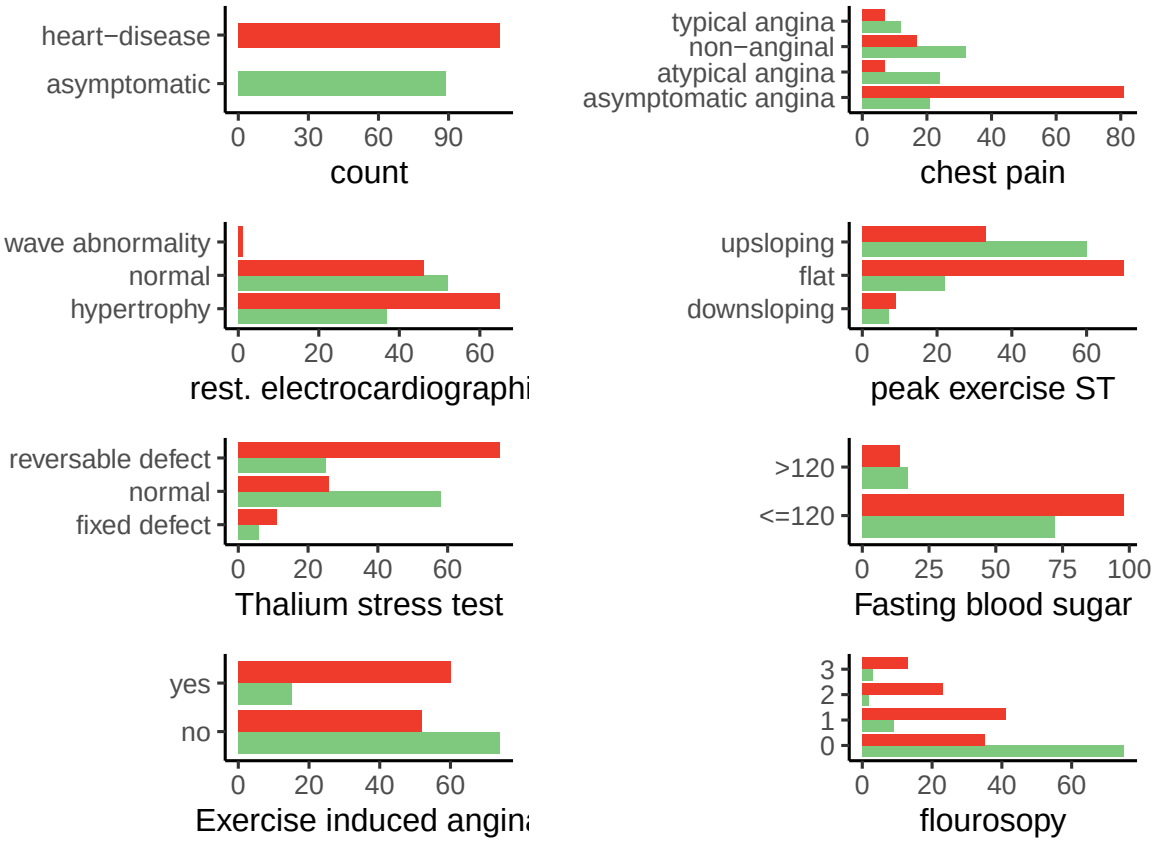


1 Male patient

Male patients with heart disease are significantly older, have higher cholesterol level, and reduced maximum heart rate response to the thallium test.

```
# Disease status
g2 <- ggplot(heart, aes(x = target, fill = target)) +
  geom_bar(width = 0.5, position = 'dodge') +
  labs(x = "") +
  coord_flip() +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# cp
h2 <- ggplot(heart, aes(cp, group = target, fill = target)) +
  geom_bar(position = "dodge") +
  labs(x = "", y = "chest pain") +
  coord_flip() +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# restecg
i2 <- ggplot(heart, aes(restecg, group = target, fill = target)) +
  geom_bar(position = "dodge") +
  labs(x = "", y = "rest. electrocardiographic") +
  coord_flip() +
```

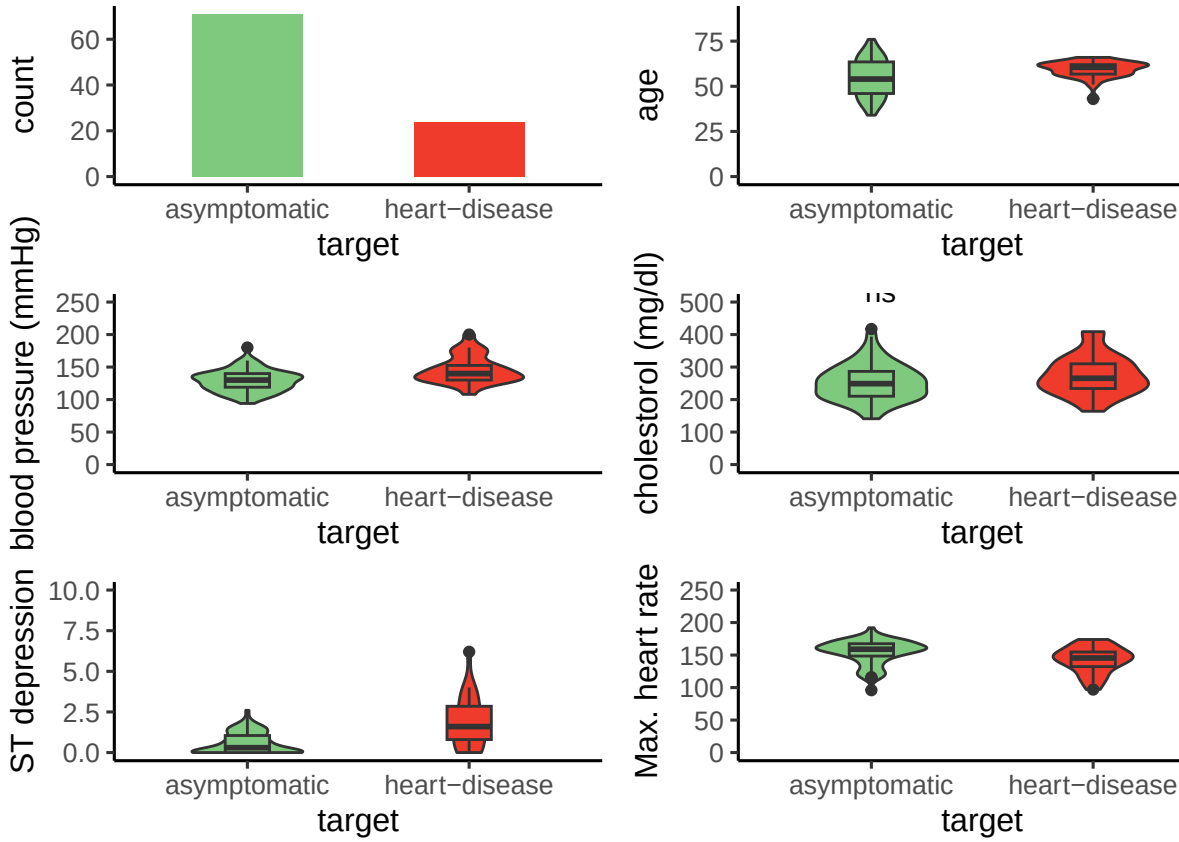
```
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# slope
j2 <- ggplot(heart, aes(slope, group = target, fill = target)) +
  geom_bar(position = "dodge") +
  labs(x = "", y = "peak exercise ST") +
  coord_flip() +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# thal
k2 <- ggplot(heart, aes(thal, group = target, fill = target)) +
  geom_bar(position = "dodge") +
  labs(x = "", y = "Thalium stress test") +
  coord_flip() +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# fbp
l2 <- ggplot(heart, aes(fbs, group = target, fill = target)) +
  geom_bar(position = "dodge") +
  labs(x = "", y = "Fasting blood sugar") +
  coord_flip() +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# exang
m2 <- ggplot(heart, aes(exang, group = target, fill = target)) +
  geom_bar(position = "dodge") +
  labs(x = "", y = "Exercise induced angina") +
  coord_flip() +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# ca
n2 <- ggplot(heart, aes(ca, group = target, fill = target)) +
  geom_bar(position = "dodge") +
  labs(x = "", y = "flourosopy") +
  coord_flip() +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

ggarrange(g2, h2, i2, j2, k2, l2, m2, n2,
          ncol = 2, nrow = 4,
          align = "v")
```




```
heart <- heart2 %>%
    filter(sex == "female")
```

```
# Male and Female count
a2 <- ggplot(heart, aes(x = target, fill = target)) +
  geom_bar(width = 0.5, position = 'dodge') +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# Age per gender
b2 <- ggplot(heart, aes(x= target, y = age, fill = target)) +
  geom_violin(width = 0.5) +
  geom_boxplot(width = 0.2) +
  ylim(0, 90) +
  stat_compare_means(aes(label = ..p.signif..), method = "t.test") +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# trestbps
```

```r
c2 <- ggplot(heart, aes(x = target, y = trestbps, fill = target)) +
  geom_violin(width = 0.5) +
  geom_boxplot(width = 0.2) +
  labs(y = "blood pressure (mmHg)") +
  ylim(0,250) +
  stat_compare_means(aes(label = ..p.signif..), method = "t.test") +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# chol
d2 <- ggplot(heart, aes(x = target, y = chol, fill = target)) +
  geom_violin(width = 0.5) +
  geom_boxplot(width = 0.2) +
  labs(y = "cholestorol (mg/dl)") +
  ylim(0,500) +
  stat_compare_means(aes(label = ..p.signif..), method = "t.test") +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# oldpeak
e2 <- ggplot(heart, aes(x = target, y = oldpeak, fill = target)) +
  geom_violin(width = 0.5) +
  geom_boxplot(width = 0.2) +
  labs(y = "ST depression") +
  ylim(0,10) +
  stat_compare_means(aes(label = ..p.signif..), method = "t.test") +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# thalach
f2 <- ggplot(heart, aes(x = target, y = thalach, fill = target)) +
  geom_violin(width = 0.5) +
  geom_boxplot(width = 0.2) +
  labs(y = "Max. heart rate") +
  ylim(0,250) +
  stat_compare_means(aes(label = ..p.signif..), method = "t.test") +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

suppressWarnings(ggarrange(a2, b2, c2, d2, e2, f2,
          ncol = 2, nrow = 3,
            align = "v"))
```



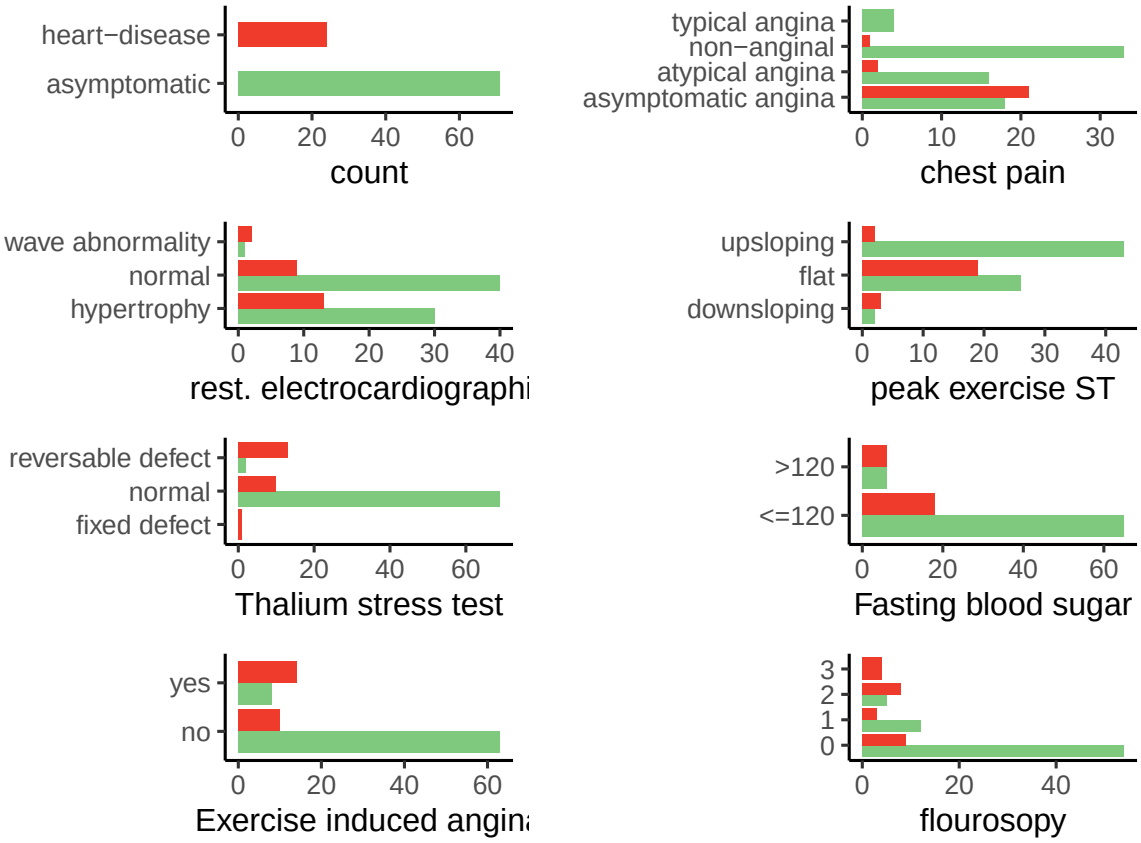
2 Female patients

There is less woman with heart disease on this data set. Women with heart disease have a significantly higher resting blood presure contrary to male with heart disease. Similarly to men, women with heart disease have a lower maximum heart rate in response to the thallium test.

```
# Disease status
g2 <- ggplot(heart, aes(x = target, fill = target)) +
  geom_bar(width = 0.5, position = 'dodge') +
  labs(x = "") +
  coord_flip() +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# cp
h2 <- ggplot(heart, aes(cp, group = target, fill = target)) +
  geom_bar(position = "dodge") +
  labs(x = "", y = "chest pain") +
  coord_flip() +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# restecg
i2 <- ggplot(heart, aes(restecg, group = target, fill = target)) +
  geom_bar(position = "dodge") +
```

```
  labs(x = "", y = "rest. electrocardiographic") +
  coord_flip() +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# slope
j2 <- ggplot(heart, aes(slope, group = target, fill = target)) +
  geom_bar(position = "dodge") +
  labs(x = "", y = "peak exercise ST") +
  coord_flip() +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# thal
k2 <- ggplot(heart, aes(thal, group = target, fill = target)) +
  geom_bar(position = "dodge") +
  labs(x = "", y = "Thalium stress test") +
  coord_flip() +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# fbp
l2 <- ggplot(heart, aes(fbs, group = target, fill = target)) +
  geom_bar(position = "dodge") +
  labs(x = "", y = "Fasting blood sugar") +
  coord_flip() +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# exang
m2 <- ggplot(heart, aes(exang, group = target, fill = target)) +
  geom_bar(position = "dodge") +
  labs(x = "", y = "Exercise induced angina") +
  coord_flip() +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

# ca
n2 <- ggplot(heart, aes(ca, group = target, fill = target)) +
  geom_bar(position = "dodge") +
  labs(x = "", y = "flourosopy") +
  coord_flip() +
  scale_fill_manual(values = c("#7fc97f","#ef3b2c"))+
  theme_classic2() +
  theme(legend.position='none')

ggarrange(g2, h2, i2, j2, k2, l2, m2, n2,
          ncol = 2, nrow = 4,
```

```
          align = "v")
```

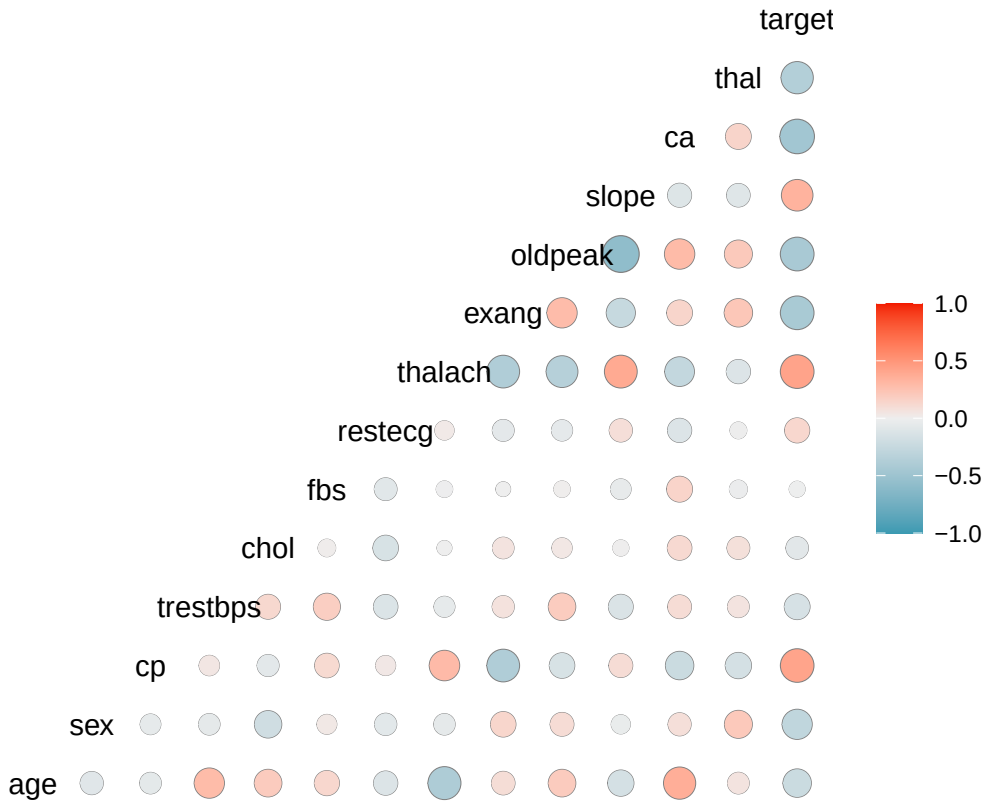


Results

Setting up the models

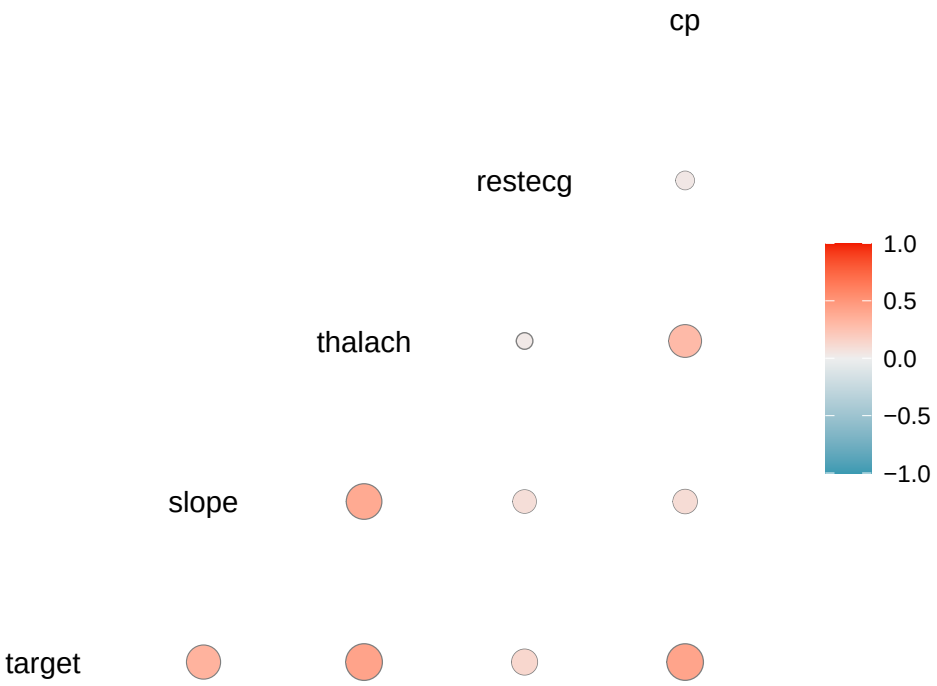
```
heart <- copy %>%
  filter(
    thal != 0 & ca != 4 # remove values correspondind to NA in original dataset
    )
```

```
# ggcorr(heart, palette = "RdBu")
GGally::ggcorr(heart, geom = "circle")
```

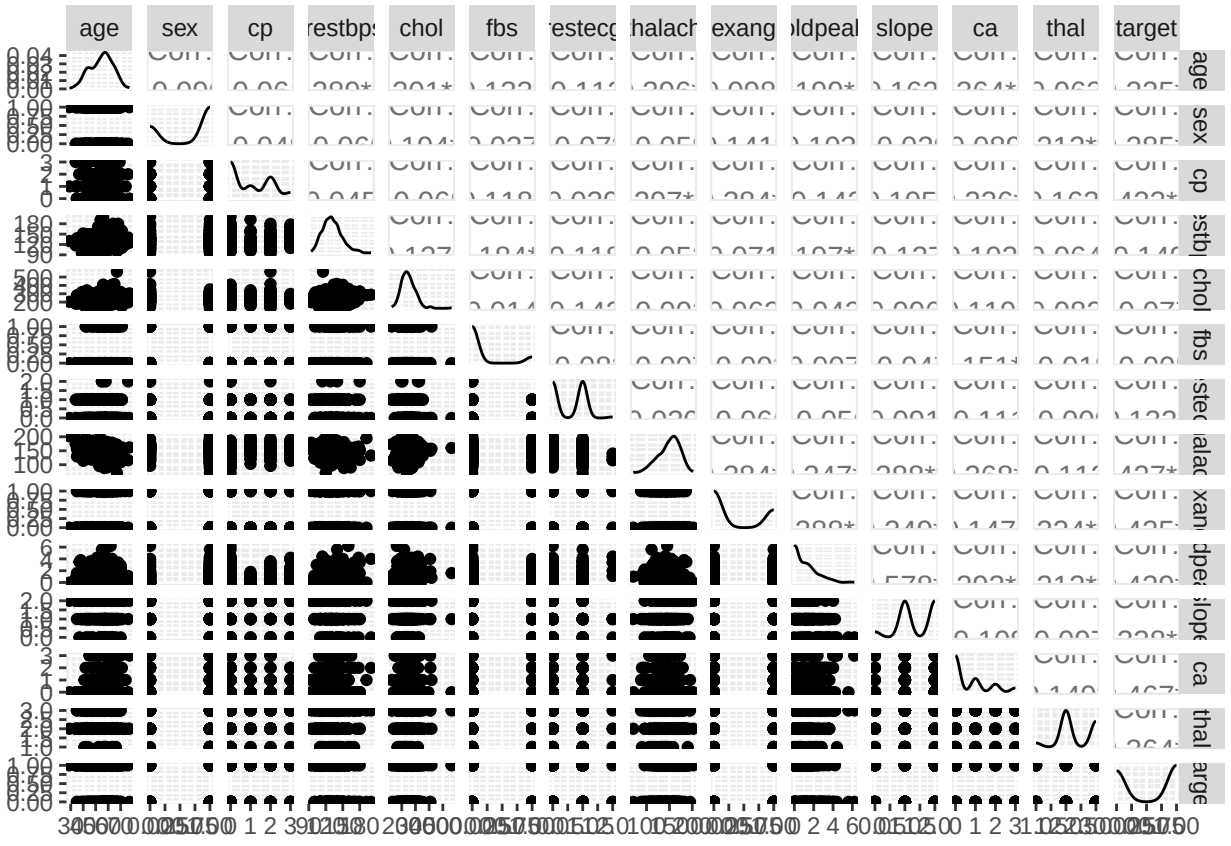



```
select2 <- heart %>%
  dplyr::select(
    target,
    slope,
    thalach,
    restecg,
    cp
  )
```

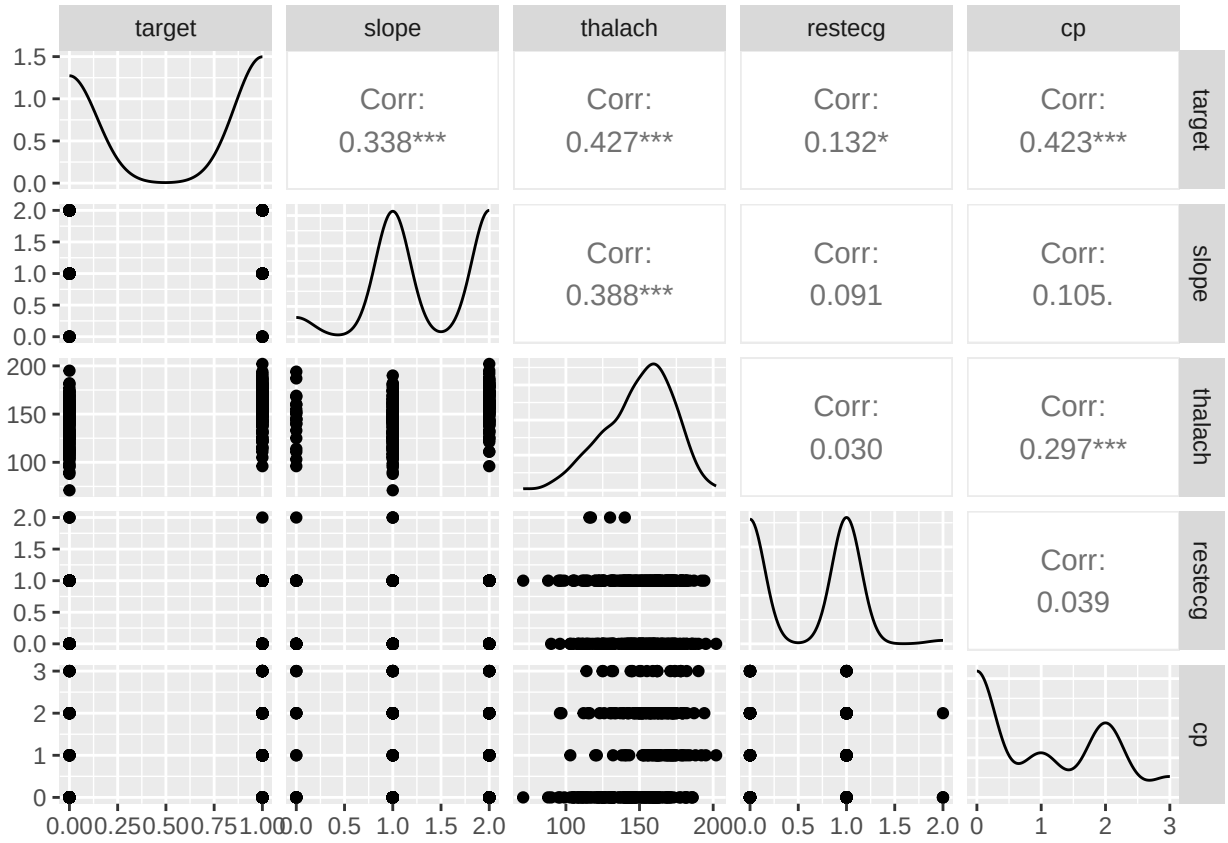
```
ggcorr(select2, geom = "circle")
```




```
ggpairs(heart)
```




```
ggpairs(select2)
```



From the correlation study it seems that the parameters
* cp
* restecg
* thalach
* slope
are the most usefull to predict the risk of heart disease

From the EDA anlysis it semms that
* age
* sex
* cholesterol
* restecg
are also usefull

For prediction the following variables seems the most usefull
* age
* sex
* cholesterol
* restecg
* cp
* thalach
* slope

VI Machine Learning: classification model with rpart and random forest packages

1. Select the columns usefull for prediction according to the EDA analysis.
2. Separate the data set in a train and test subsets.
3. Build a classification tree model with rpart.
4. Print model accuracy and descision tree.

A Use select columns for classification

```
# glimpse(heart)

heart_select <- heart %>%
  dplyr::select( #because of conflict between MASS and dplyr select need to use dplyr::select
    target,
    age,
    sex,
    chol,
    restecg,
    cp,
    thalach,
    slope
  )
```

```
heart_select$target <- factor(heart_select$target) # Define target as a factor. rpart classification wo

accuracy <- 0

# Build a simple classification desicion tree with rpart. Run the model until the accuracy reach the se
while(accuracy <= 0.88) {
  split_values <- sample.split(heart_select$target, SplitRatio = 0.65)
  train_set <- subset(heart_select, split_values == T)
  test_set <- subset(heart_select, split_values == F)
  mod_class <- rpart(target~. , data=train_set)
  result_class <- predict(mod_class, test_set, type = "class")
  table <- table(test_set$target, result_class)
  accuracy <- (table["0","0"] + table["1","1"])/sum(table)
  # cat("accuracy = ", round(accuracy, digits = 2)*100, "%")
}
```

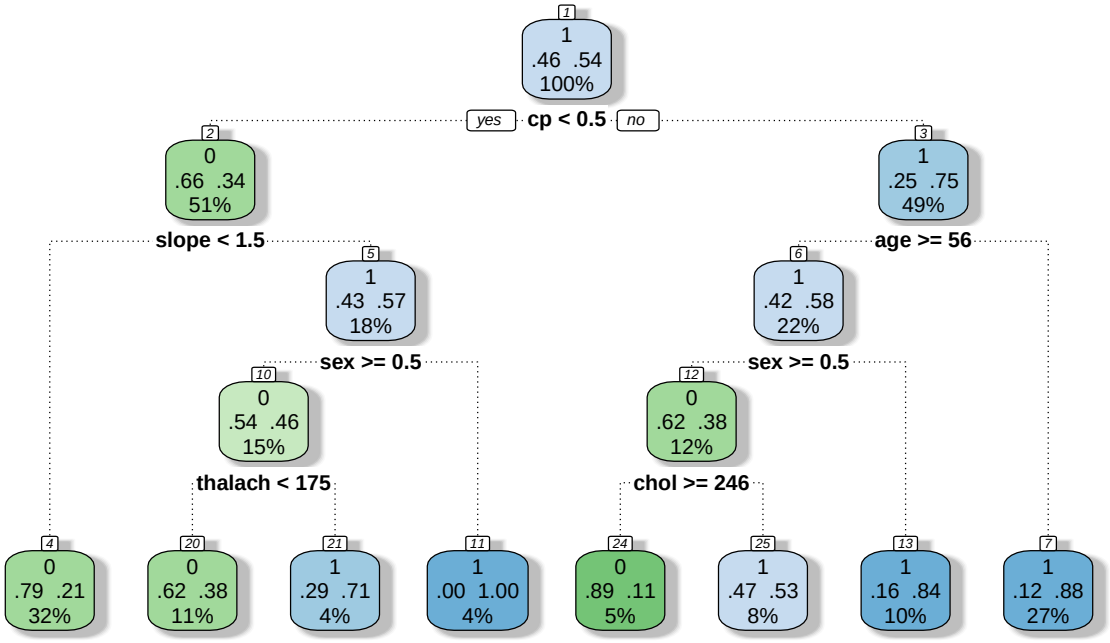
Print model accuracy.
According to parameters the model should be at least 88% accurate.

```
cat("Model accuracy", round(accuracy, digits = 2)*100, "%")
```

```
## Model accuracy 89 %
```

Print the desicion tree.

```
# par(mfrow = c(1,2), xpd = NA) # otherwise on some devices the text is clipped
fancyRpartPlot(mod_class, , caption = NULL)
```




```
# plot(mod_class)
# text(mod_class, use.n = TRUE)
```

B Use the full dataset for classification

```
copy2 <- heart
heart$target <- factor(heart$target)
accuracy <- 0

# Build a simple classification desicion tree with rpart. Run the model until the accuracy reach the se
while(accuracy <= 0.88) {
  split_values <- sample.split(heart_select$target, SplitRatio = 0.65)
  train_set <- subset(heart, split_values == T)
  test_set <- subset(heart, split_values == F)
  mod_class <- rpart(target~. , data=train_set)
  result_class <- predict(mod_class, test_set, type = "class")
  table <- table(test_set$target, result_class)
  accuracy <- (table["0","0"] + table["1","1"])/sum(table)
  # cat("accuracy = ", round(accuracy, digits = 2)*100, "%")
}
```

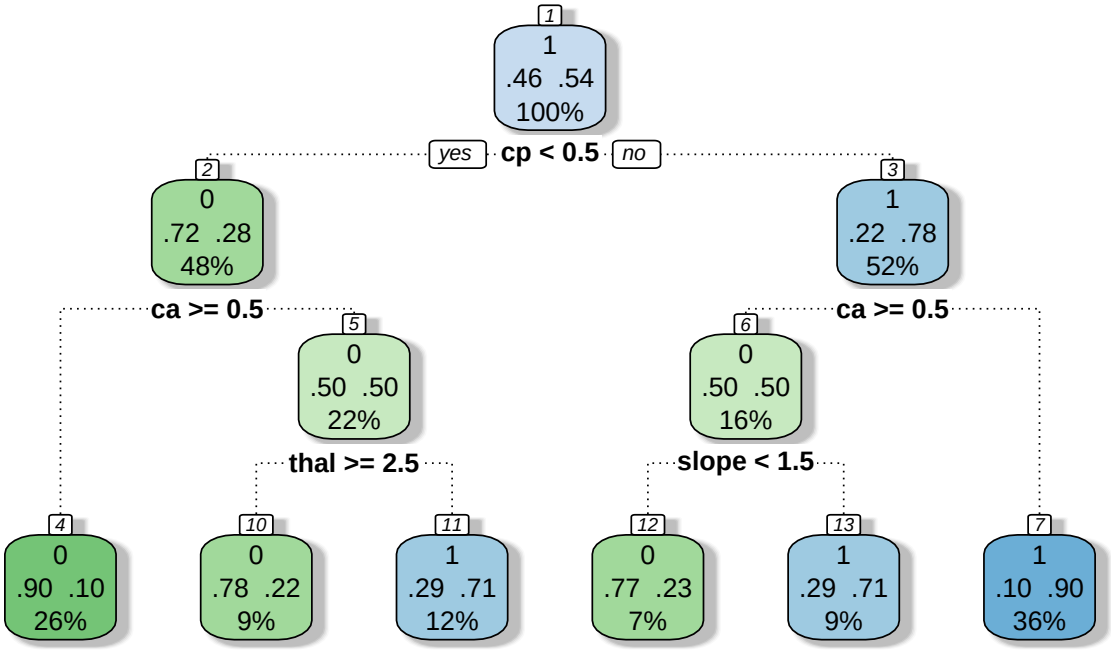
Print model accuracy.
According to parameters the model should be at least 88% accurate.

```
cat("Model accuracy", round(accuracy, digits = 2)*100, "%")
```

```
## Model accuracy 88 %
```

Print the desicion tree.

```
# par(mfrow = c(1,2), xpd = NA) # otherwise on some devices the text is clipped
fancyRpartPlot(mod_class, , caption = NULL)
```




```
# plot(mod_class)
# text(mod_class, use.n = TRUE)
```

C Prediction on selected column with random forest

```
set.seed(103)
train <- sample(nrow(heart_select), 0.7*nrow(heart_select), replace = FALSE)
TrainSet <- heart_select[train,]
ValidSet <- heart_select[-train,]
summary(TrainSet)
```

```
##  target       age              sex              chol        restecg
##  0: 93   Min.   :29.00   Min.   :0.0000   Min.   :126   Min.   :0.0000
##  1:114   1st Qu.:46.50   1st Qu.:0.0000   1st Qu.:211   1st Qu.:0.0000
##          Median :55.00   Median :1.0000   Median :239   Median :1.0000
##          Mean   :54.03   Mean   :0.6908   Mean   :246   Mean   :0.5411
##          3rd Qu.:60.00   3rd Qu.:1.0000   3rd Qu.:277   3rd Qu.:1.0000
##          Max.   :77.00   Max.   :1.0000   Max.   :564   Max.   :2.0000
##        cp            thalach          slope
##  Min.   :0.0000   Min.   : 88.0   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:132.0   1st Qu.:1.000
##  Median :1.0000   Median :152.0   Median :1.000
##  Mean   :0.9903   Mean   :149.3   Mean   :1.377
##  3rd Qu.:2.0000   3rd Qu.:167.5   3rd Qu.:2.000
##  Max.   :3.0000   Max.   :202.0   Max.   :2.000
```

```
summary(ValidSet)
```

```
##  target      age              sex              chol          restecg
##  0:43   Min.   :34.00   Min.   :0.0000   Min.   :131.0   Min.   :0.0000
##  1:46   1st Qu.:50.00   1st Qu.:0.0000   1st Qu.:213.0   1st Qu.:0.0000
##         Median :57.00   Median :1.0000   Median :245.0   Median :0.0000
##         Mean   :55.67   Mean   :0.6517   Mean   :249.9   Mean   :0.4831
##         3rd Qu.:62.00   3rd Qu.:1.0000   3rd Qu.:274.0   3rd Qu.:1.0000
##         Max.   :74.00   Max.   :1.0000   Max.   :417.0   Max.   :1.0000
##        cp            thalach          slope
##  Min.   :0.0000   Min.   : 71.0   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:142.0   1st Qu.:1.000
##  Median :0.0000   Median :154.0   Median :2.000
##  Mean   :0.8876   Mean   :150.2   Mean   :1.438
##  3rd Qu.:2.0000   3rd Qu.:163.0   3rd Qu.:2.000
##  Max.   :3.0000   Max.   :192.0   Max.   :2.000
```

```
# Create a Random Forest model with default parameters
model1 <- randomForest(target ~ ., data = TrainSet,  ntree = 1000, mtry = 1, importance = TRUE)
model1
```

```
##
## Call:
##  randomForest(formula = target ~ ., data = TrainSet, ntree = 1000,      mtry = 1, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 26.09%
## Confusion matrix:
##    0  1 class.error
## 0 66 27   0.2903226
## 1 27 87   0.2368421
```

```
# Predicting on train set
predTrain <- predict(model1, TrainSet, type = "class")
# Checking classification accuracy
table(predTrain, TrainSet$target)
```

```
##
## predTrain   0   1
##         0  84  10
##         1   9 104
```

```
# Predicting on Validation set
predValid <- predict(model1, ValidSet, type = "class")
# Checking classification accuracy
mean(predValid == ValidSet$target)
```

```
## [1] 0.8426966
```

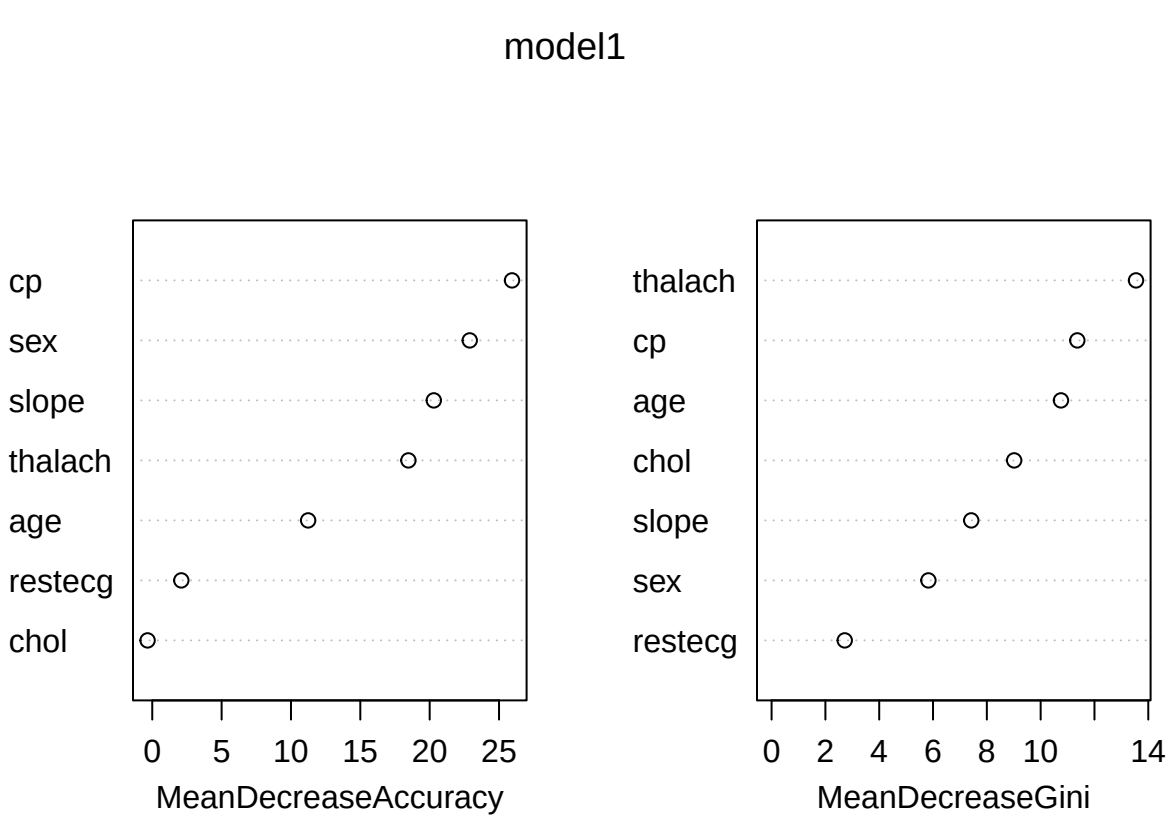
```
table(predValid,ValidSet$target)
```

```
##
## predValid  0  1
##         0 34  5
##         1  9 41
```

```
# To check important variables
importance(model1)
```

```
##                  0          1 MeanDecreaseAccuracy MeanDecreaseGini
## age      8.5635858  8.1274728           11.2387546        10.754847
## sex     20.7787044 16.4186188           22.8891913         5.828656
## chol    -0.6401581  0.1514547           -0.3373315         9.016257
## restecg  3.2825129 -0.2000304            2.0950618         2.718995
## cp      21.0887042 22.3577590           25.9377637        11.363132
## thalach 13.9497972 12.7792174           18.4671834        13.544435
## slope   16.3291696 14.9037870           20.2973077         7.420501
```

```
varImpPlot(model1)
```



D Use the full dataset for classification with random forest

```
set.seed(103)
train <- sample(nrow(heart), 0.7*nrow(heart_select), replace = FALSE)
TrainSet <- heart[train,]
ValidSet <- heart[-train,]
summary(TrainSet)
```

```
##       age             sex               cp            trestbps
##  Min.   :29.00   Min.   :0.0000   Min.   :0.0000   Min.   : 94.0
##  1st Qu.:46.50   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:120.0
##  Median :55.00   Median :1.0000   Median :1.0000   Median :130.0
##  Mean   :54.03   Mean   :0.6908   Mean   :0.9903   Mean   :130.6
##  3rd Qu.:60.00   3rd Qu.:1.0000   3rd Qu.:2.0000   3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000   Max.   :3.0000   Max.   :200.0
##       chol           fbs            restecg          thalach
##  Min.   :126   Min.   :0.0000   Min.   :0.0000   Min.   : 88.0
##  1st Qu.:211   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:132.0
##  Median :239   Median :0.0000   Median :1.0000   Median :152.0
##  Mean   :246   Mean   :0.1498   Mean   :0.5411   Mean   :149.3
##  3rd Qu.:277   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:167.5
##  Max.   :564   Max.   :1.0000   Max.   :2.0000   Max.   :202.0
##      exang           oldpeak          slope             ca
##  Min.   :0.0000   Min.   :0.000   Min.   :0.000   Min.   :0.0000
```

```
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:1.000   1st Qu.:0.0000
##  Median :0.0000   Median :1.000   Median :1.000   Median :0.0000
##  Mean   :0.3237   Mean   :1.146   Mean   :1.377   Mean   :0.7053
##  3rd Qu.:1.0000   3rd Qu.:1.800   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :5.600   Max.   :2.000   Max.   :3.0000
##       thal       target
##  Min.   :1.000   0: 93
##  1st Qu.:2.000   1:114
##  Median :2.000
##  Mean   :2.309
##  3rd Qu.:3.000
##  Max.   :3.000
```

```
summary(ValidSet)
```

```
##       age             sex               cp            trestbps
##  Min.   :34.00   Min.   :0.0000   Min.   :0.0000   Min.   : 94
##  1st Qu.:50.00   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:120
##  Median :57.00   Median :1.0000   Median :0.0000   Median :132
##  Mean   :55.67   Mean   :0.6517   Mean   :0.8876   Mean   :134
##  3rd Qu.:62.00   3rd Qu.:1.0000   3rd Qu.:2.0000   3rd Qu.:145
##  Max.   :74.00   Max.   :1.0000   Max.   :3.0000   Max.   :180
##       chol            fbs            restecg          thalach
##  Min.   :131.0   Min.   :0.0000   Min.   :0.0000   Min.   : 71.0
##  1st Qu.:213.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:142.0
##  Median :245.0   Median :0.0000   Median :0.0000   Median :154.0
##  Mean   :249.9   Mean   :0.1348   Mean   :0.4831   Mean   :150.2
##  3rd Qu.:274.0   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:163.0
##  Max.   :417.0   Max.   :1.0000   Max.   :1.0000   Max.   :192.0
##      exang           oldpeak           slope             ca
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:1.000   1st Qu.:0.000
##  Median :0.0000   Median :0.5000   Median :2.000   Median :0.000
##  Mean   :0.3371   Mean   :0.8573   Mean   :1.438   Mean   :0.618
##  3rd Qu.:1.0000   3rd Qu.:1.2000   3rd Qu.:2.000   3rd Qu.:1.000
##  Max.   :1.0000   Max.   :6.2000   Max.   :2.000   Max.   :3.000
##       thal       target
##  Min.   :1.000   0:43
##  1st Qu.:2.000   1:46
##  Median :2.000
##  Mean   :2.371
##  3rd Qu.:3.000
##  Max.   :3.000
```

```
# Create a Random Forest model with default parameters
model2 <- randomForest(target ~ ., data = TrainSet,  ntree = 1000, mtry = 2, importance = TRUE)
model2
```

```
##
## Call:
##  randomForest(formula = target ~ ., data = TrainSet, ntree = 1000,      mtry = 2, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
```

```
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 19.81%
## Confusion matrix:
##    0  1 class.error
## 0 71 22   0.2365591
## 1 19 95   0.1666667
```

```
# Predicting on train set
predTrain <- predict(model2, TrainSet, type = "class")
# Checking classification accuracy
table(predTrain, TrainSet$target)
```

```
##
## predTrain   0   1
##         0  93   0
##         1   0 114
```

```
# Predicting on Validation set
predValid <- predict(model2, ValidSet, type = "class")
# Checking classification accuracy
mean(predValid == ValidSet$target)
```

```
## [1] 0.8876404
```

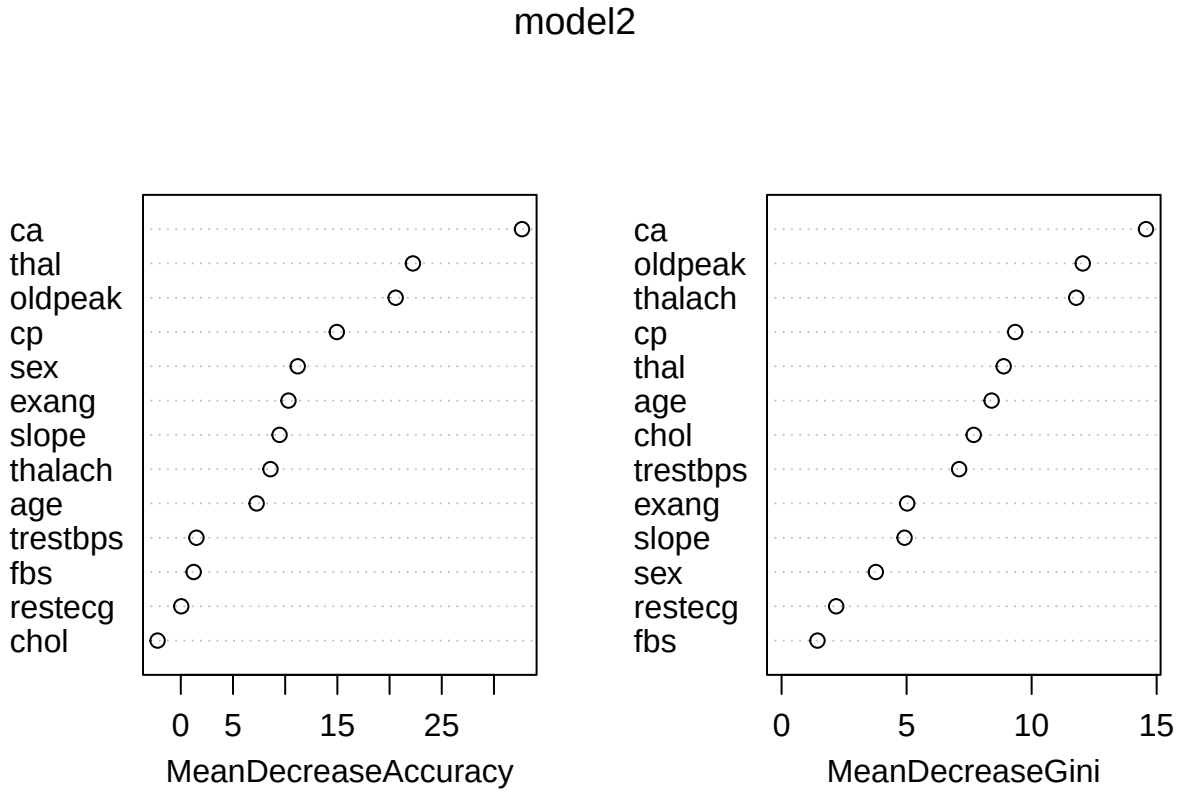
```
table(predValid,ValidSet$target)
```

```
##
## predValid  0  1
##         0 34  1
##         1  9 45
```

```
# To check important variables
importance(model2)
```

```
##                   0          1 MeanDecreaseAccuracy MeanDecreaseGini
## age       3.5531855  6.9841836           7.26413831         8.399466
## sex       5.4296622 10.6451739          11.20179489         3.769720
## cp       13.1854993  8.8365604          14.94704991         9.342946
## trestbps -0.7248403  2.6564089           1.49917817         7.100646
## chol     -1.4289861 -1.6216622          -2.22440917         7.682719
## fbs       0.1249613  1.5281060           1.23237510         1.435625
## restecg  -0.1085154  0.4856894           0.04541514         2.184212
## thalach   2.2711310  9.6719216           8.58938320        11.784905
## exang     7.4776841  7.2417434          10.31578378         5.023431
## oldpeak  16.7213070 13.5158911          20.58424679        12.044601
## slope     8.5126201  5.2090096           9.46218293         4.917241
## ca       23.4603133 28.5683118          32.68822956        14.574756
## thal     13.2246941 19.8485560          22.23692307         8.882171
```

```
varImpPlot(model2)
```



Conclusion

We set out to use the UCI data set on Heart Disease to create a model that could correctly predict Heart Disease diagnoses. We set a goal of achieving an validation set accuracy score of 0.85. We started by downloading the UCI data set on Heart Disease. We then cleaned the data set and prepared it for analysis. We split the data set into training and test sets. We found that having a heart defect, the number of major vessels that were working, and the type of chest pain we the most important factors in determining if you have heart disease or not, using this data set. We achieved our goal of creating a model with a validation set accuracy score of 0.88 with the Random Forest model!

Limitations

For me the biggest limitation in this project is the size of the data set. With only 303 observations this is a very small sample size. The other limitation is the data within the data set. 14 features is enough to achieve a high prediction accuracy, as we proved, but I think with more features we could achieve a score over 91%.

Future Work

For the future, I would be curious to see how these algorithms preform on a much larger data set of data of our country. Along with a data set that has more features such as: height, weight, if parents had heart disease, use of drugs and alcohol, exercise amount, etc. I would be curious to see which algorithms preform

better and if any preform worse. One final thing that I would include is adding more algorithms to this project. These 9 algorithms are not the only algorithms that work well with classification and they may produce a higher validation set accuracy score.

References

Data transformation
https://archive.ics.uci.edu/ml/datasets/Heart+Disease
https://lucdemortier.github.io/projects/3_mcnulty
https://www.kaggle.com/ronitf/heart-disease-uci/discussion/105877

Kaggles notebooks:
R notebooks:
https://www.kaggle.com/ekrembayar/heart-disease-uci-eda-models-with-r
https://www.kaggle.com/joemenifee/heart-disease-uci-data-exploratory

Data Processing
http://www.cookbook-r.com/ https://bookdown.org/rdpeng/exdata/managing-data-frames-with-the-dplyr-package.html#data-frames https://rpkgs.datanovia.com/ggpubr/reference/stat_compare_means.html
https://towardsdatascience.com/simple-fast-exploratory-data-analysis-in-r-with-dataexplorer-package-e055348d9619
http://www.sthda.com/english/articles/24-ggpubr-publication-ready-plots/81-ggplot2-easy-way-to-mix-multiple-graphs-on-the-same-page/
https://www.r-graph-gallery.com/267-reorder-a-variable-in-ggplot2
for categorical variable
https://www.analyticsvidhya.com/blog/2015/11/easy-methods-deal-categorical-variables-predictive-modeling/
for correlations
http://www.sthda.com/english/wiki/ggally-r-package-extension-to-ggplot2-for-correlation-matrix-and-survival-plots-r-software-and-data-visualization
https://www.kaggle.com/code/jarredpriester/heart-disease-predictions-using-a-ml-ensemble-in-r/notebook
https://www.kaggle.com/code/wguesdon/predicting-heart-disease-risk-with-random-forest/script