Heart Disease Predictor

Kaustav Pal

18/08/2021

First, we load the required libraries

```
library(dplyr)
library(ggplot2)
library(caTools)
library(randomForest)
```

Data:

This database contains 14 physical attributes based on physical testing of a patient. Blood samples are taken and the patient also conducts a brief exercise test. The “goal” field refers to the presence of heart disease in the patient. It is integer (0 for no presence, 1 for presence). In general, to confirm 100% if a patient has heart disease can be quite an invasive process, so if we can create a model that accurately predicts the likelihood of heart disease, we can help avoid expensive and invasive procedures.

```
data<-read.csv('heart.csv')
head(data)
```

```
##   ï..age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
## 1     63   1  3      145  233   1       0     150     0     2.3     0  0    1
## 2     37   1  2      130  250   0       1     187     0     3.5     0  0    2
## 3     41   0  1      130  204   0       0     172     0     1.4     2  0    2
## 4     56   1  1      120  236   0       1     178     0     0.8     2  0    2
## 5     57   0  0      120  354   0       1     163     1     0.6     2  0    2
## 6     57   1  0      140  192   0       1     148     0     0.4     1  0    1
##   target
## 1      1
## 2      1
## 3      1
## 4      1
## 5      1
## 6      1
```

Some information about the variables:

age age in years

sex (1 = male; 0 = female)

cp: chest pain type

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

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

ca: number of major vessels (0-3) colored by flourosopy

thal: 3 = normal; 6 = fixed defect; 7 = reversable defect

target: 1 or 0

Exploratory Data Analysis

We check for any missing values, the structure of the data, and then convert some of the variables into factors for better results.

```
any(is.na(data))
```

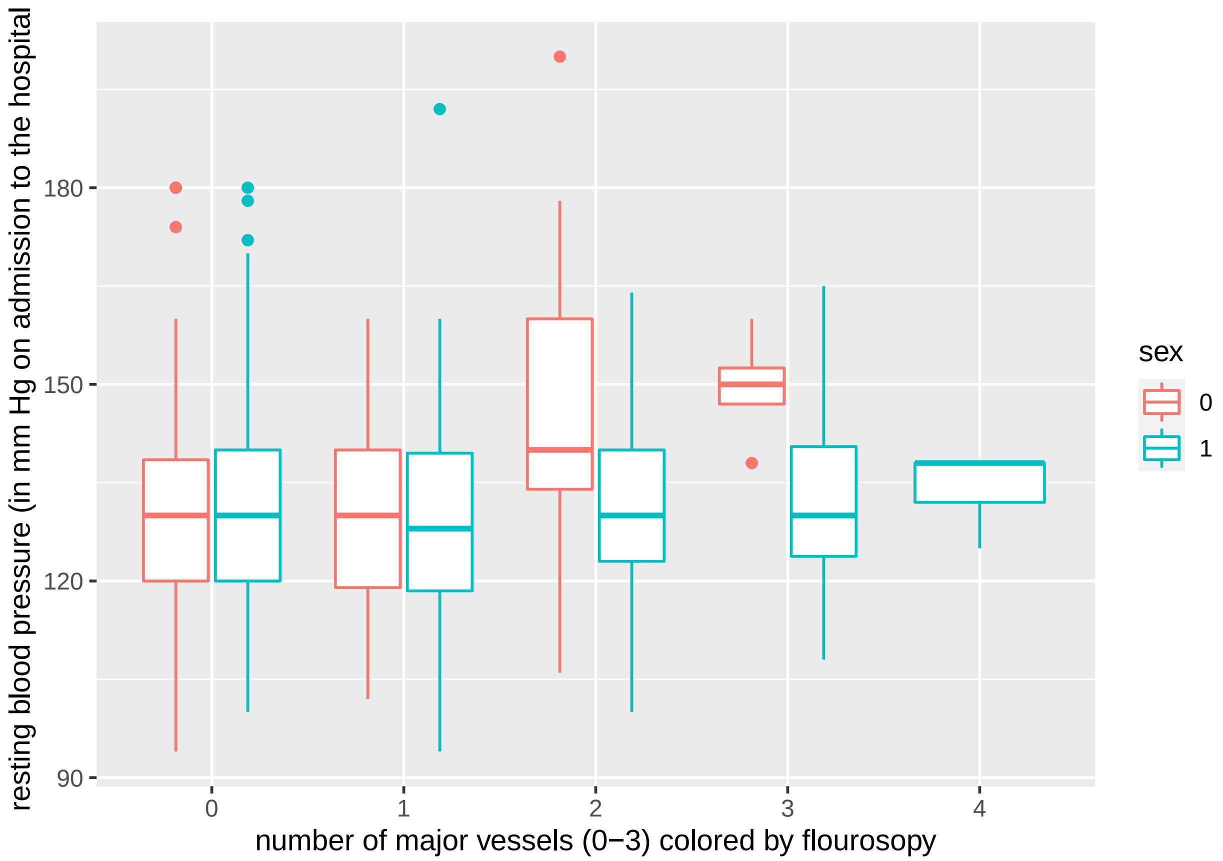
```
## [1] FALSE
```

```
str(data)
```

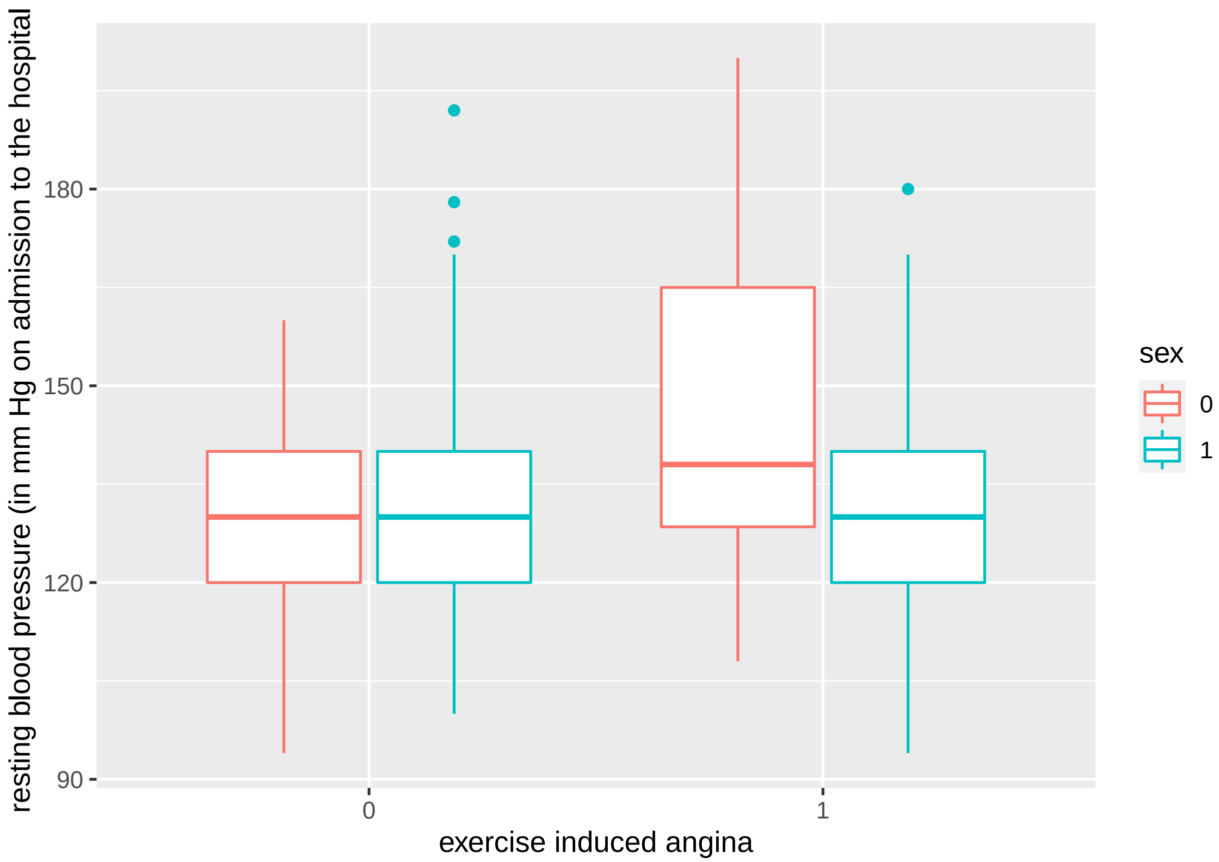
```
## ’data.frame’:    303 obs. of  14 variables:
##  $ ï..age  : int  63 37 41 56 57 57 56 44 52 57 ...
##  $ sex     : int  1 1 0 1 0 1 0 1 1 1 ...
##  $ cp      : int  3 2 1 1 0 0 1 1 2 2 ...
##  $ trestbps: int  145 130 130 120 120 140 140 120 172 150 ...
##  $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
##  $ fbs     : int  1 0 0 0 0 0 0 0 1 0 ...
##  $ restecg : int  0 1 0 1 1 1 0 1 1 1 ...
##  $ thalach : int  150 187 172 178 163 148 153 173 162 174 ...
##  $ exang   : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
##  $ slope   : int  0 0 2 2 2 1 1 2 2 2 ...
##  $ ca      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ thal    : int  1 2 2 2 2 1 2 3 3 2 ...
##  $ target  : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
data$sex<-factor(data$sex)
data$fbs<-factor(data$fbs)
data$restecg<-factor(data$restecg)
data$exang<-factor(data$exang)
data$slope<-factor(data$slope)
data$ca<-factor(data$ca)
data$target<-factor(data$target)
data$thal<-factor(data$thal)
```

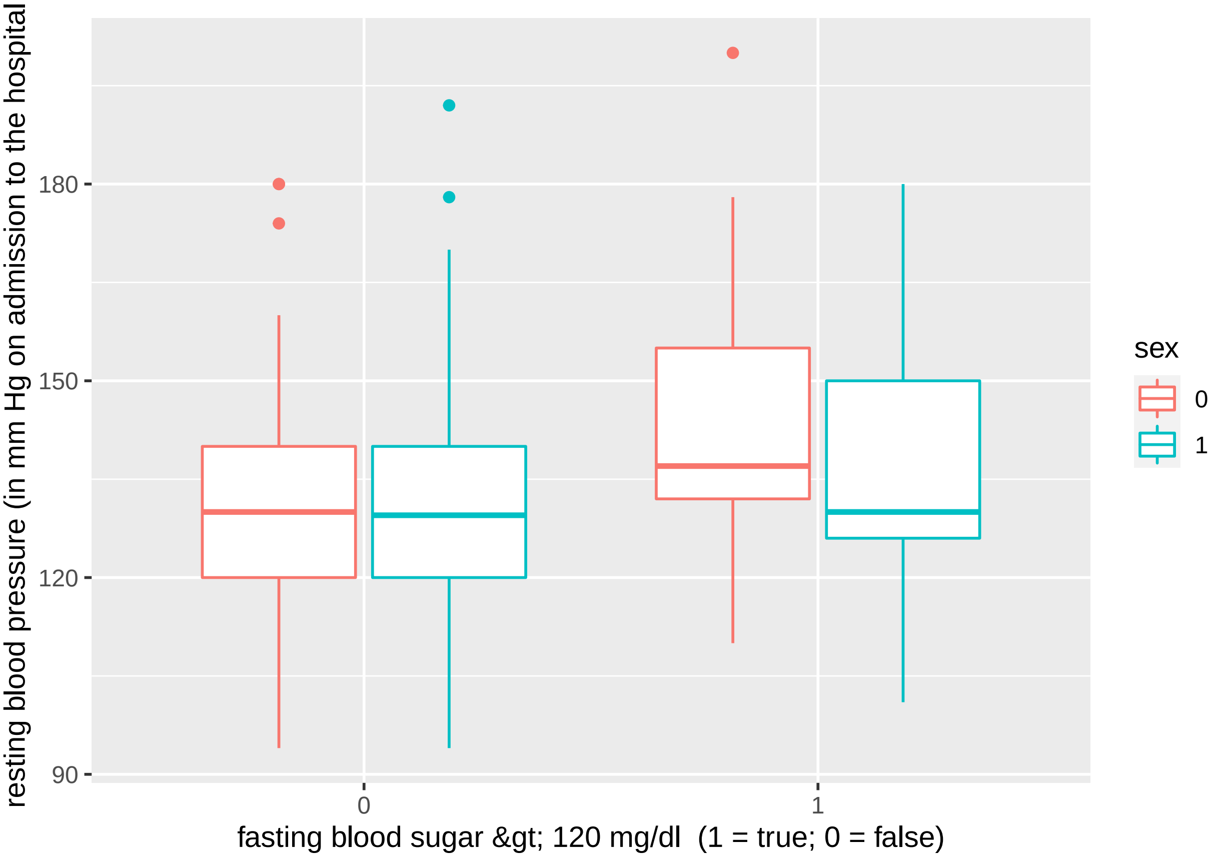
```
ggplot(data,aes(ca,trestbps))+geom_boxplot(aes(color=sex))+labs(x='number of major vessels (0-3) colore
```



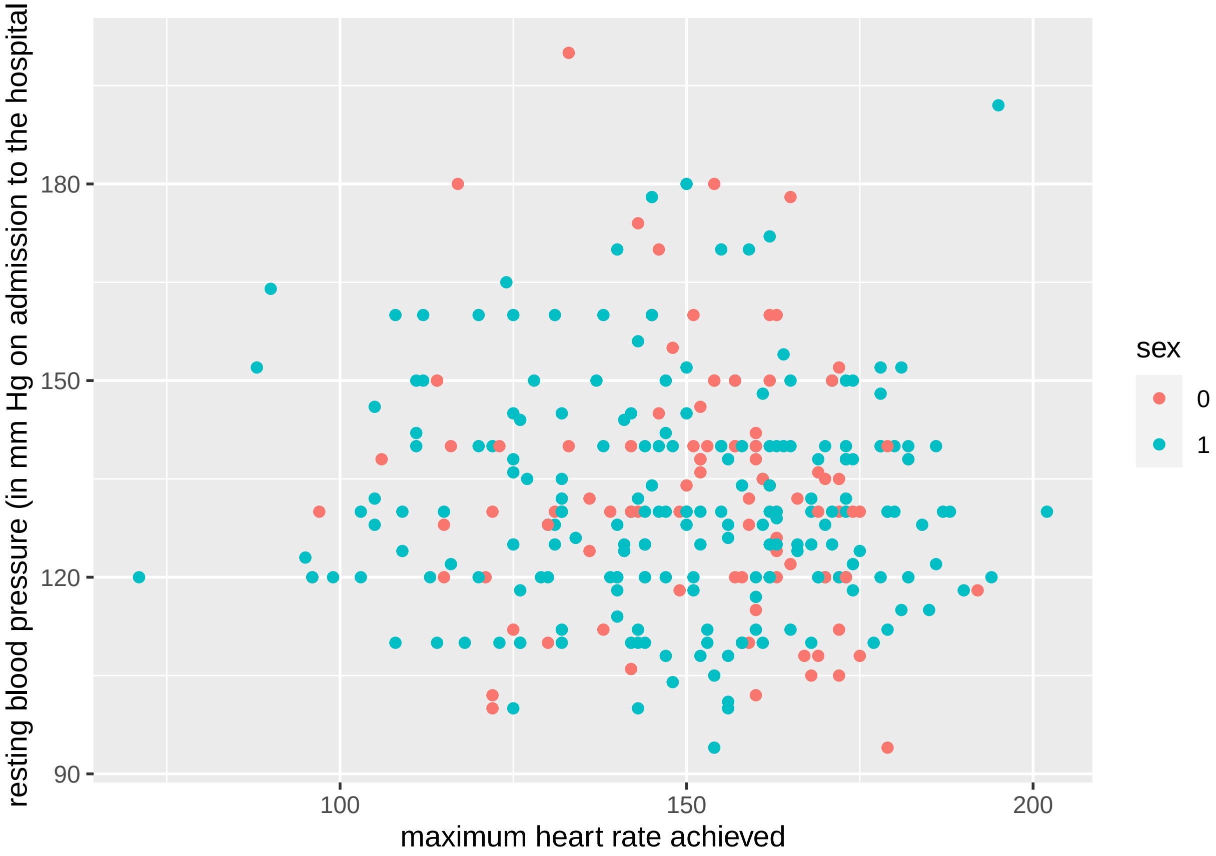
```
ggplot(data,aes(exang,trestbps))+geom_boxplot(aes(color=sex))+labs(x='exercise induced angina',y='restin
```



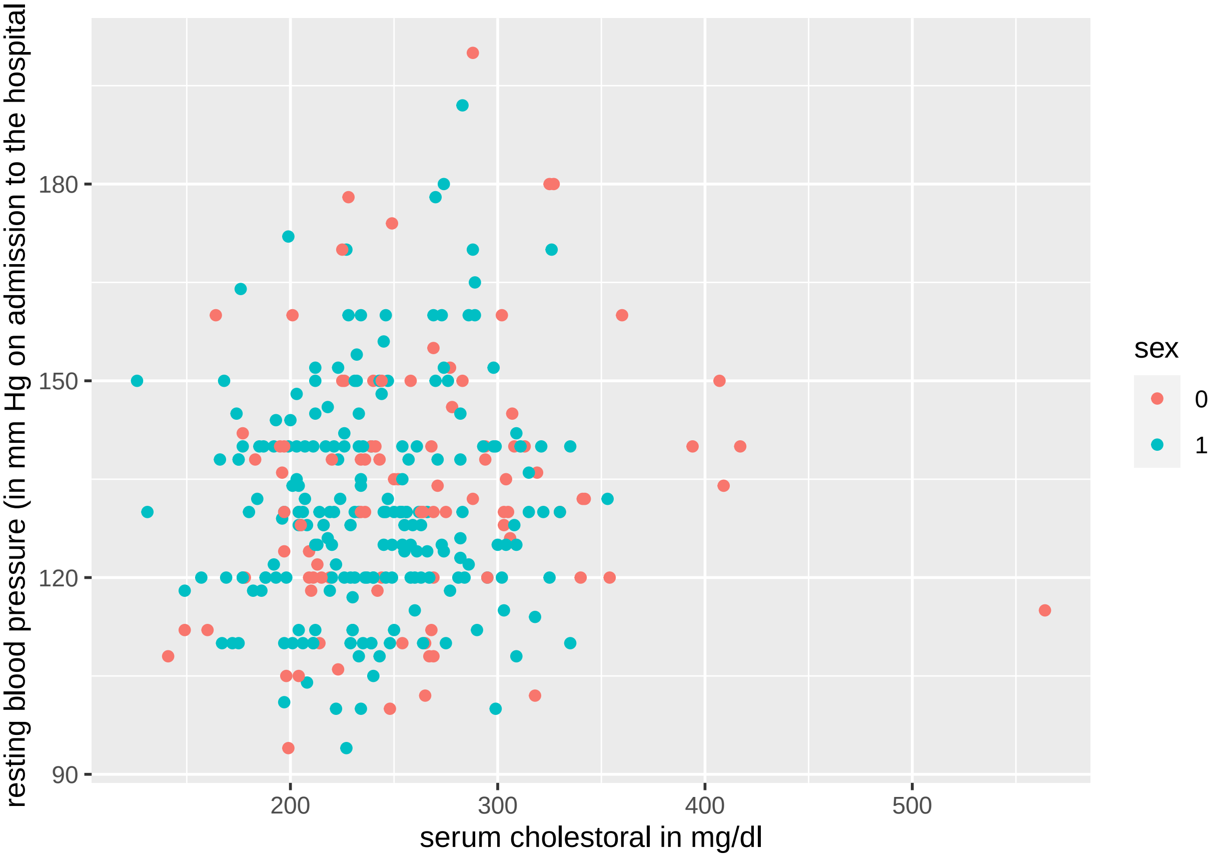
```
ggplot(data,aes(fbs,trestbps))+geom_boxplot(aes(color=sex))+labs(x='fasting blood sugar &gt; 120 mg/dl
```



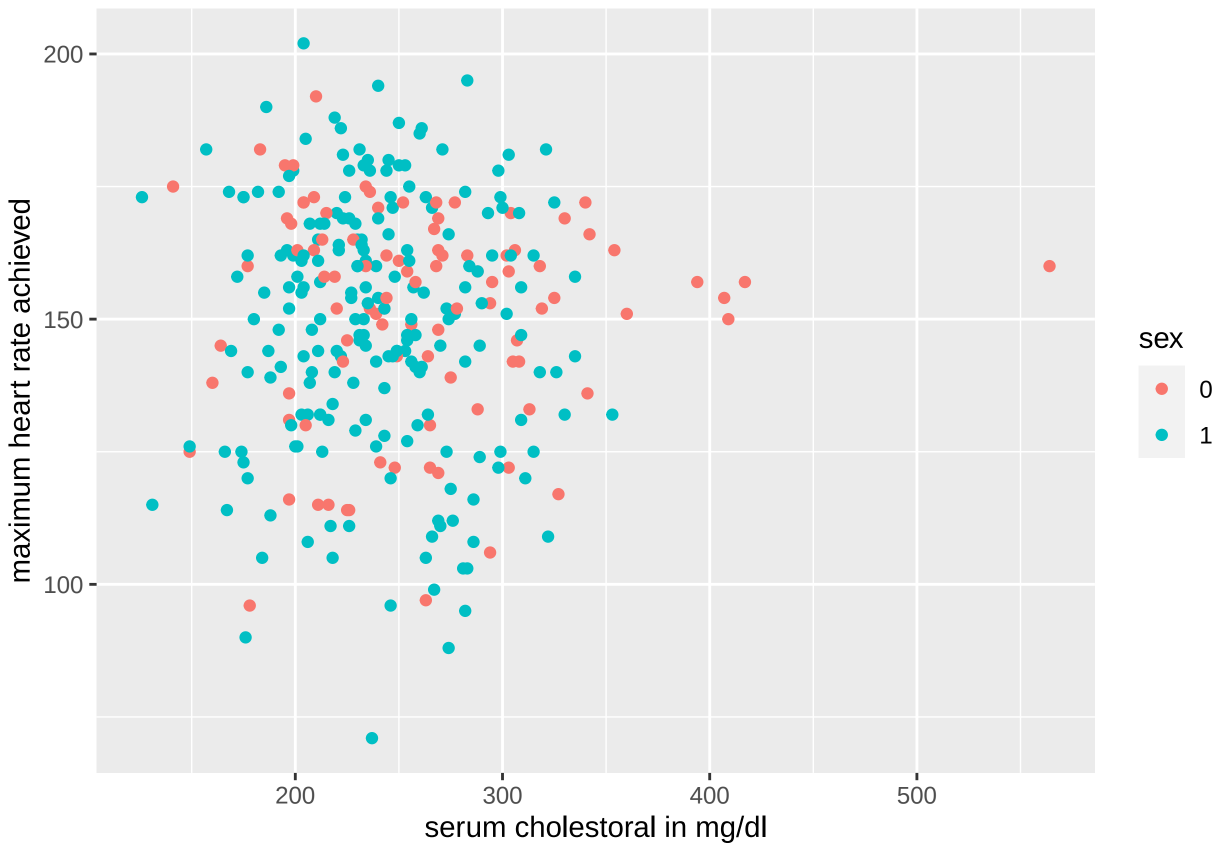
```
ggplot(data,aes(thalach,trestbps))+geom_point(aes(color=sex))+labs(x='maximum heart rate achieved',y='r
```



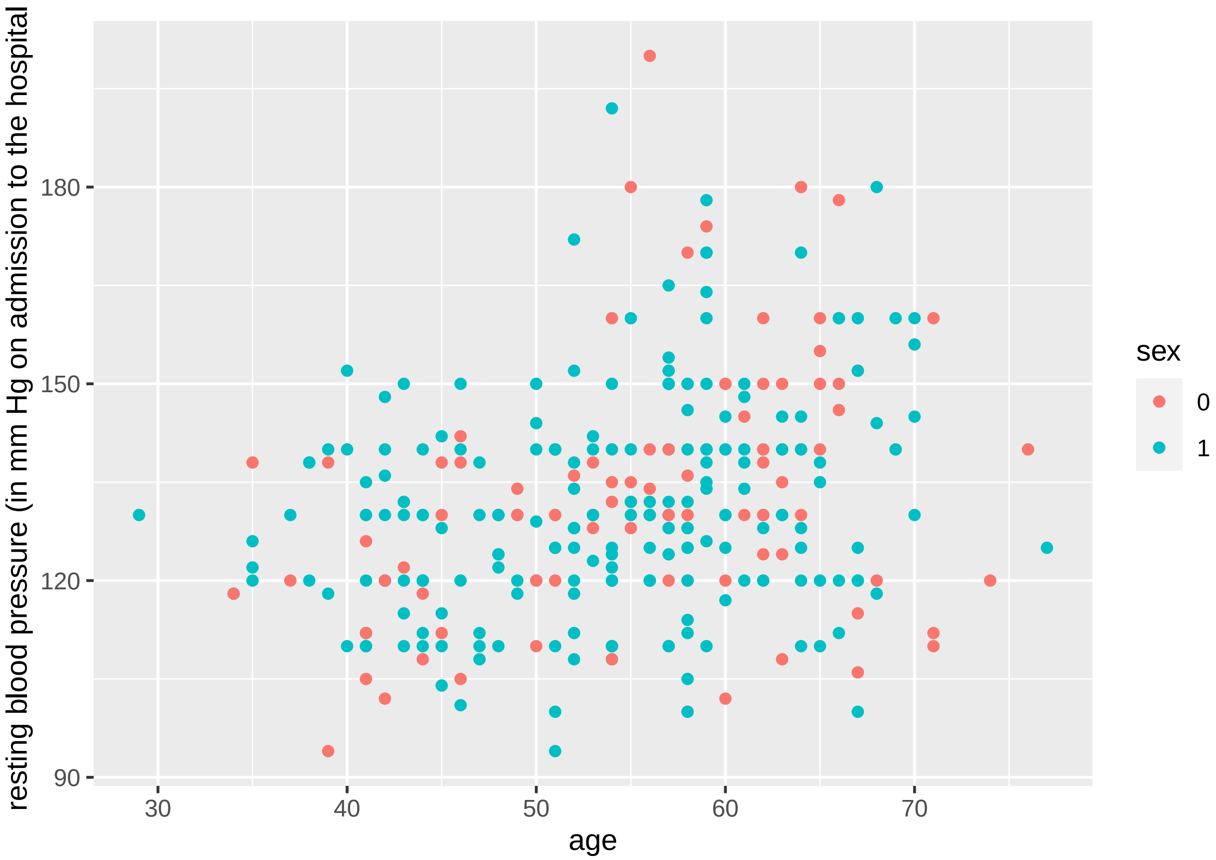
```
ggplot(data,aes(chol,trestbps))+geom_point(aes(color=sex))+labs(x='serum cholestoral in mg/dl',y='restin
```



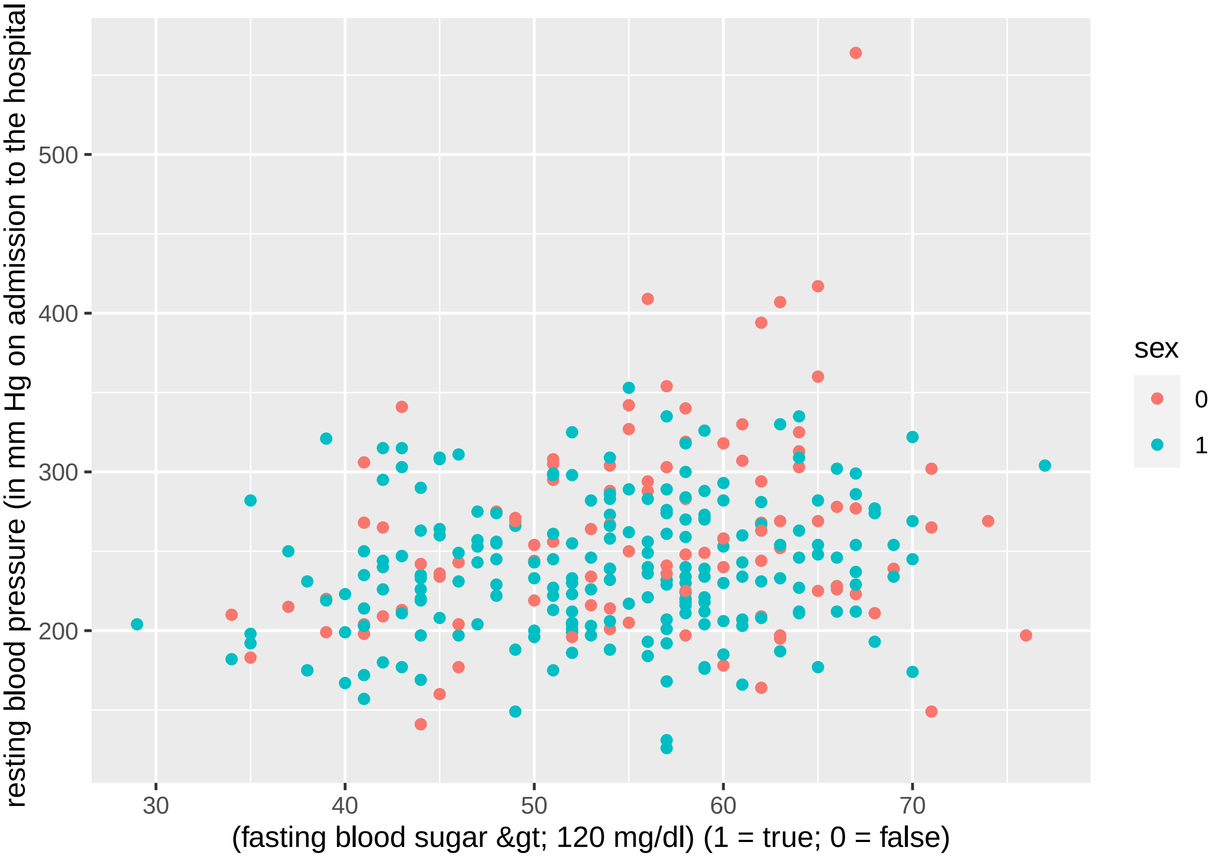
```
ggplot(data,aes(chol,thalach))+geom_point(aes(color=sex))+labs(x='serum cholestoral in mg/dl',y='maximu
```



```
ggplot(data,aes(ï..age,trestbps))+geom_point(aes(color=sex))+labs(x='age',y='resting blood pressure (in
```

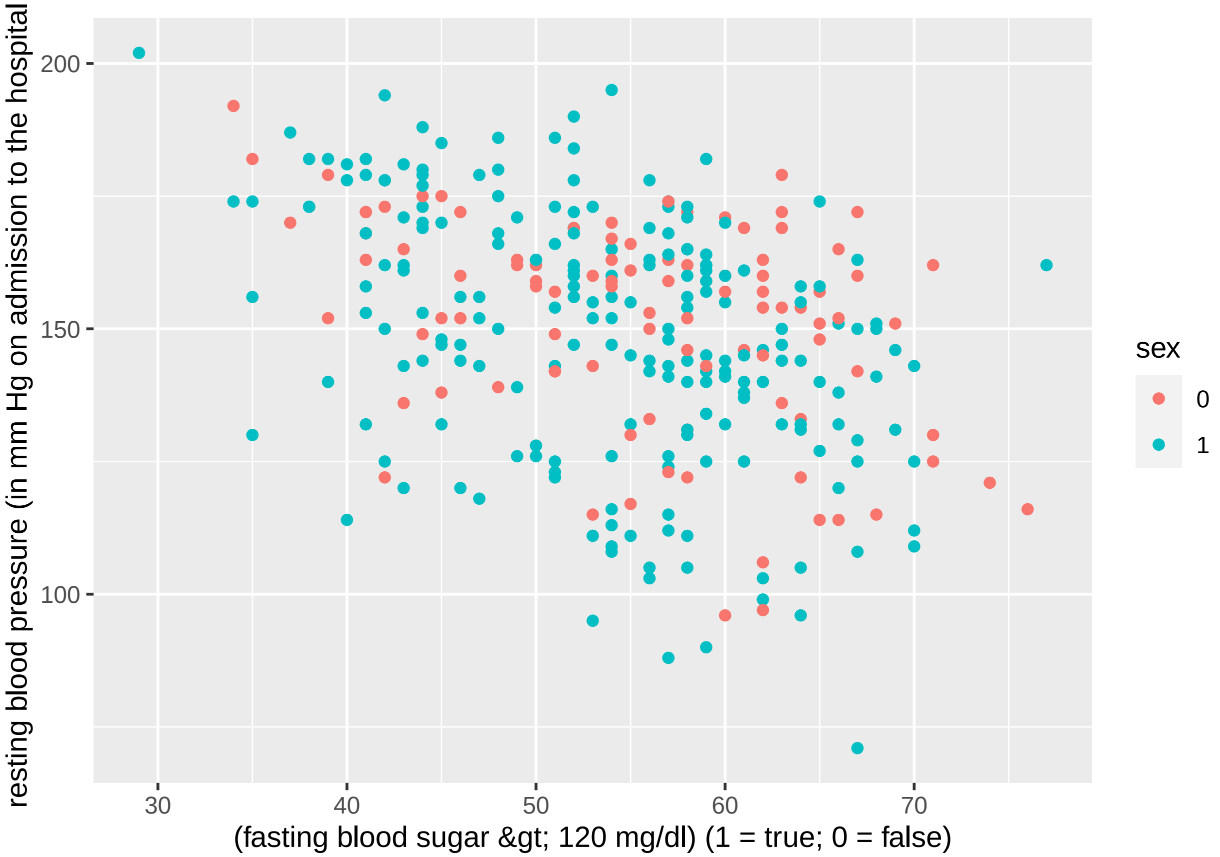


```
ggplot(data,aes(ï..age,chol))+geom_point(aes(color=sex))+labs(x='(fasting blood sugar &gt; 120 mg/dl) (
```



Till now, we see that most of the data is pretty much random.

```
ggplot(data,aes(ï..age,thalach))+geom_point(aes(color=sex))+labs(x='(fasting blood sugar &gt; 120 mg/dl
```



Here, we can see that there is somewhat negative relationship between the age and maximum heart rate achieved.

Train-Test Split

We split the data into training and test sets, with 70% of the data going to training and 30% going for testing.

```
sample<-sample.split(data,SplitRatio=0.7)
train<-subset(data,sample=T)
test<-subset(data,sample=F)
```

Model Building:

We build a random forest model and make predictions based on the model:

```
model<-randomForest(target~., train, importance = T, ntree=500)
predictions<-predict(model,test)
```

Why choose Random Forest?

From the exploratory data analysis, it was found that for most of the part, the data was randomly distributed and approximating the relationship to be linear would cause in a higher error rate and RMSE value. Also, as the data set is pretty small, even a Random Forest model will take very less time to train itself.

We check for the confusion matrix:

```
cm<-table(predictions,test$target)
cm
```

```
##
## predictions   0   1
##           0 138   0
##           1   0 165
```

Hence, we see that the random forest model works perfectly on the test set.