Heart Attack Prediction & Analysis

Carlos Kelaidis

4/7/2021

Link to data: https://www.kaggle.com/fedesoriano/heart-failure-prediction?select=heart.csv

Read in data set:

```
heart_data <- read.csv("~/Documents/My Working Directory/Personal Projects/Heart Attack Prediciton & Ana
#View(heart_data)
```

Load libraries:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.3
## v tibble  3.1.0     v dplyr   0.8.4
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(knitr)
```

The main goal is to create a model to predict the heart attack probability (output), based on the variables given.

- Classification
  - Some more **linear approaches** to begin:
    - **Logistic Regression** (We have two classes; high risk (1) or low risk (0))
    - QDA (less linear than Logistic Regression); to compare to logistic regression, see if a linear or more non-lonear method is preferred. If more linear is preferred we can also try KNN. (Use ROC curve)
    - KNN
  - Try **more flexible, less linear** methods:
    - GAMs (fit either using splines, or polynomial logistic regression)
    - **SVM**
    - **Classification Tree**

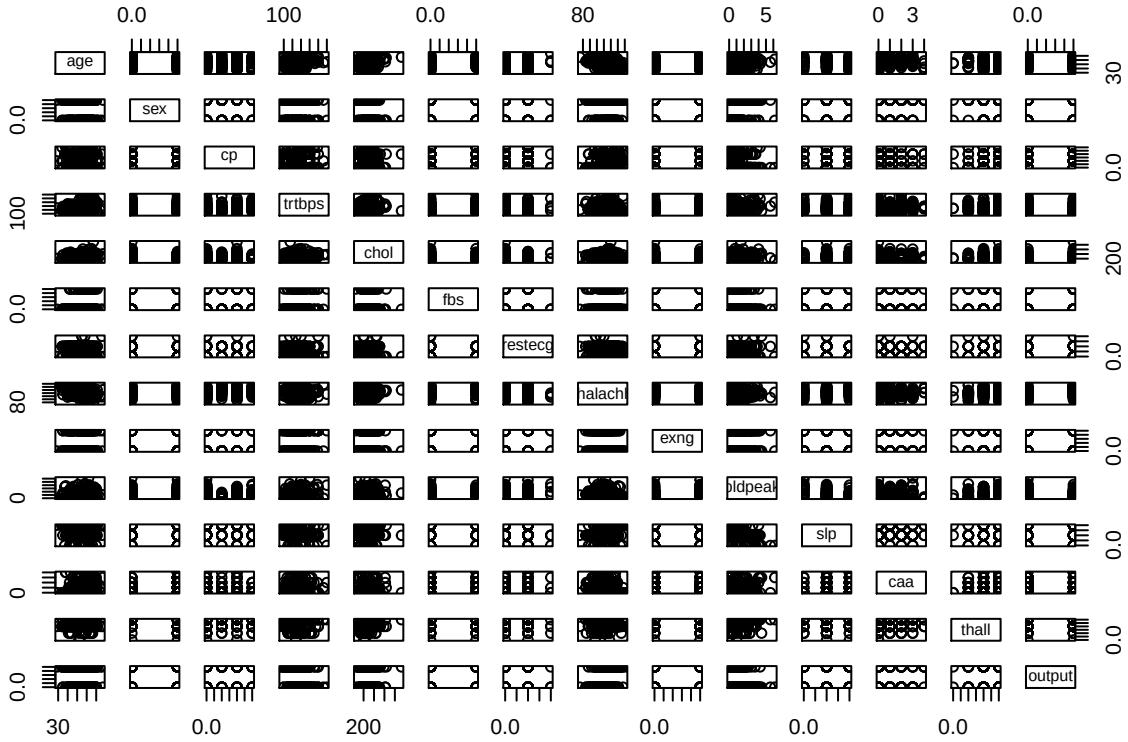
We begin by analysing the data to observe possible correlations:

```
#Observe the structure of our data
str(heart_data)
```

```
## 'data.frame':    303 obs. of  14 variables:
##  $ age     : int  63 37 41 56 57 57 56 44 52 57 ...
```

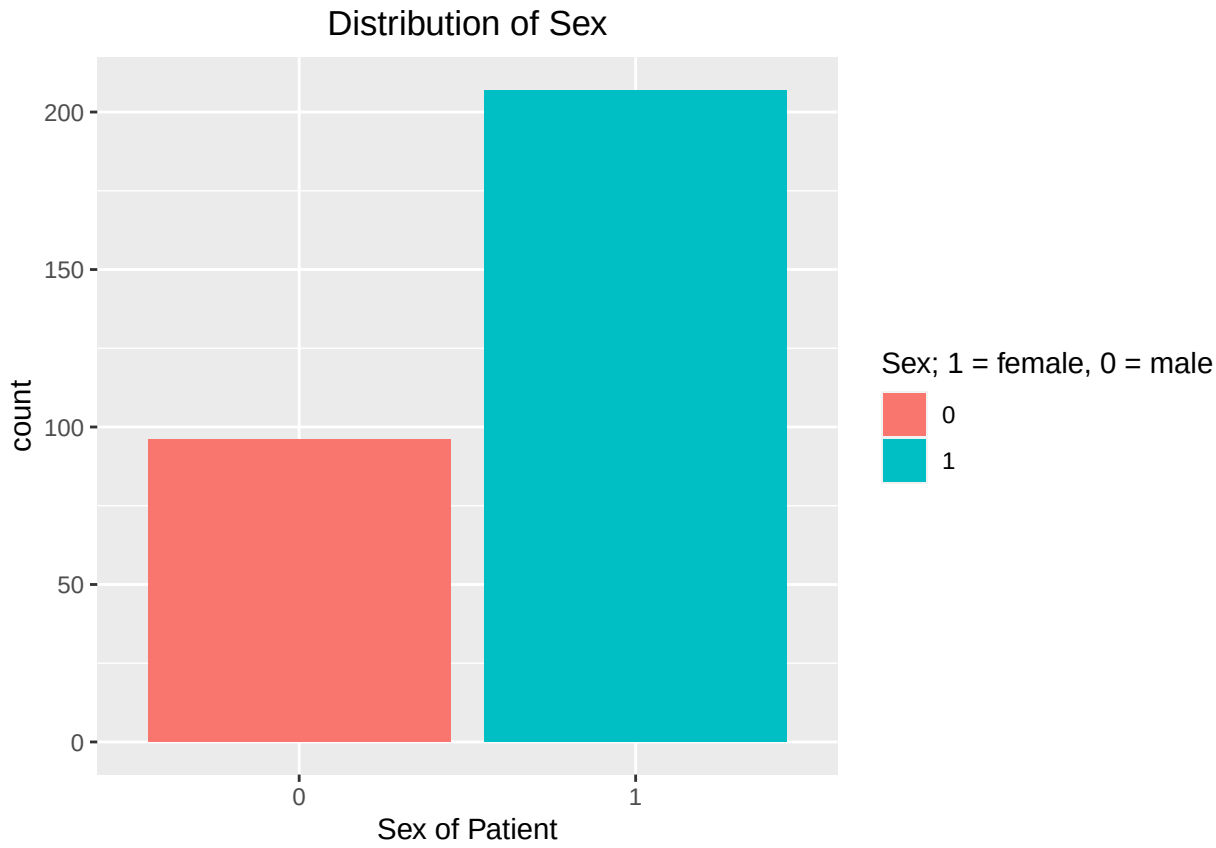
```
##  $ sex     : int  1 1 0 1 0 1 0 1 1 1 ...
##  $ cp      : int  3 2 1 1 0 0 1 1 2 2 ...
##  $ trtbps  : int  145 130 130 120 120 140 140 120 172 150 ...
##  $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
##  $ fbs     : int  1 0 0 0 0 0 0 0 1 0 ...
##  $ restecg : int  0 1 0 1 1 1 0 1 1 1 ...
##  $ thalachh: int  150 187 172 178 163 148 153 173 162 174 ...
##  $ exng    : 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 ...
##  $ slp     : int  0 0 2 2 2 1 1 2 2 2 ...
##  $ caa     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ thall   : int  1 2 2 2 2 1 2 3 3 2 ...
##  $ output  : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
#cor(heart_data)
pairs(heart_data)
```

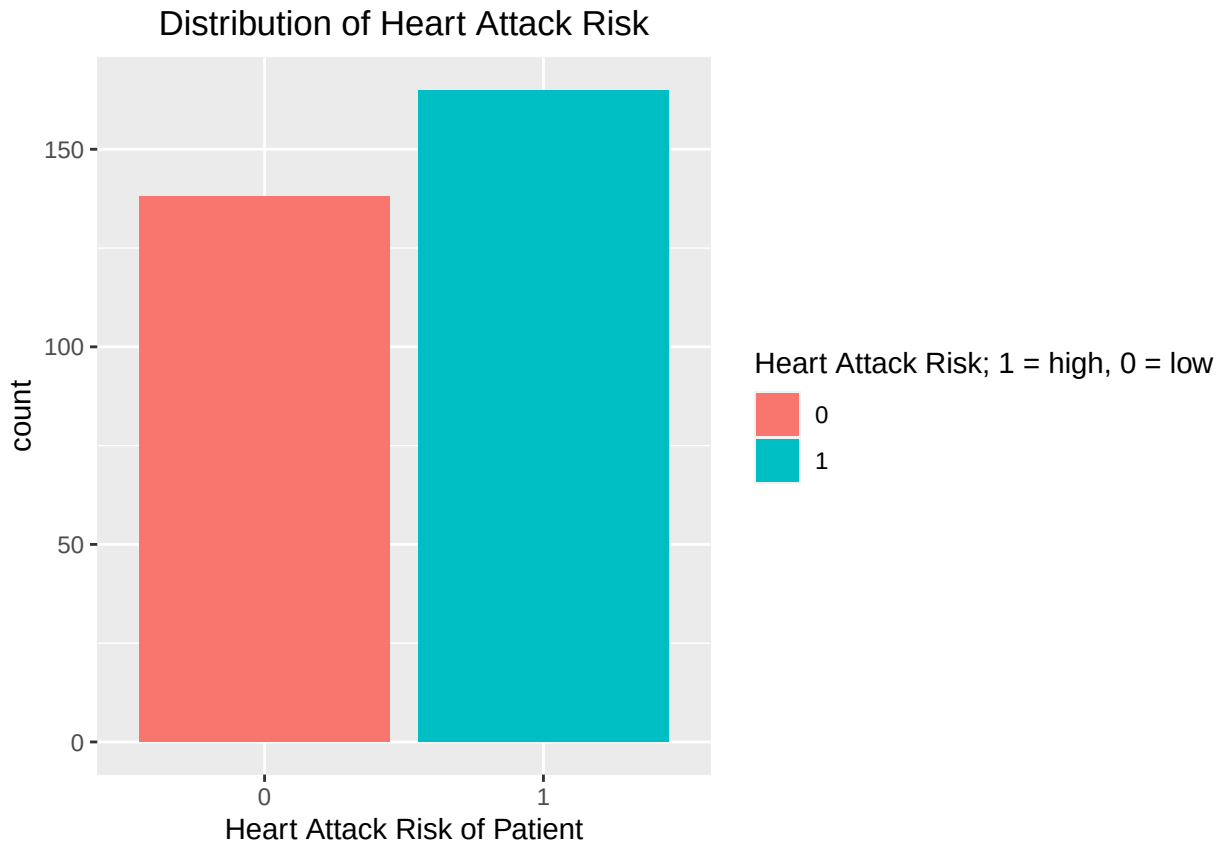


Let's look at our data:

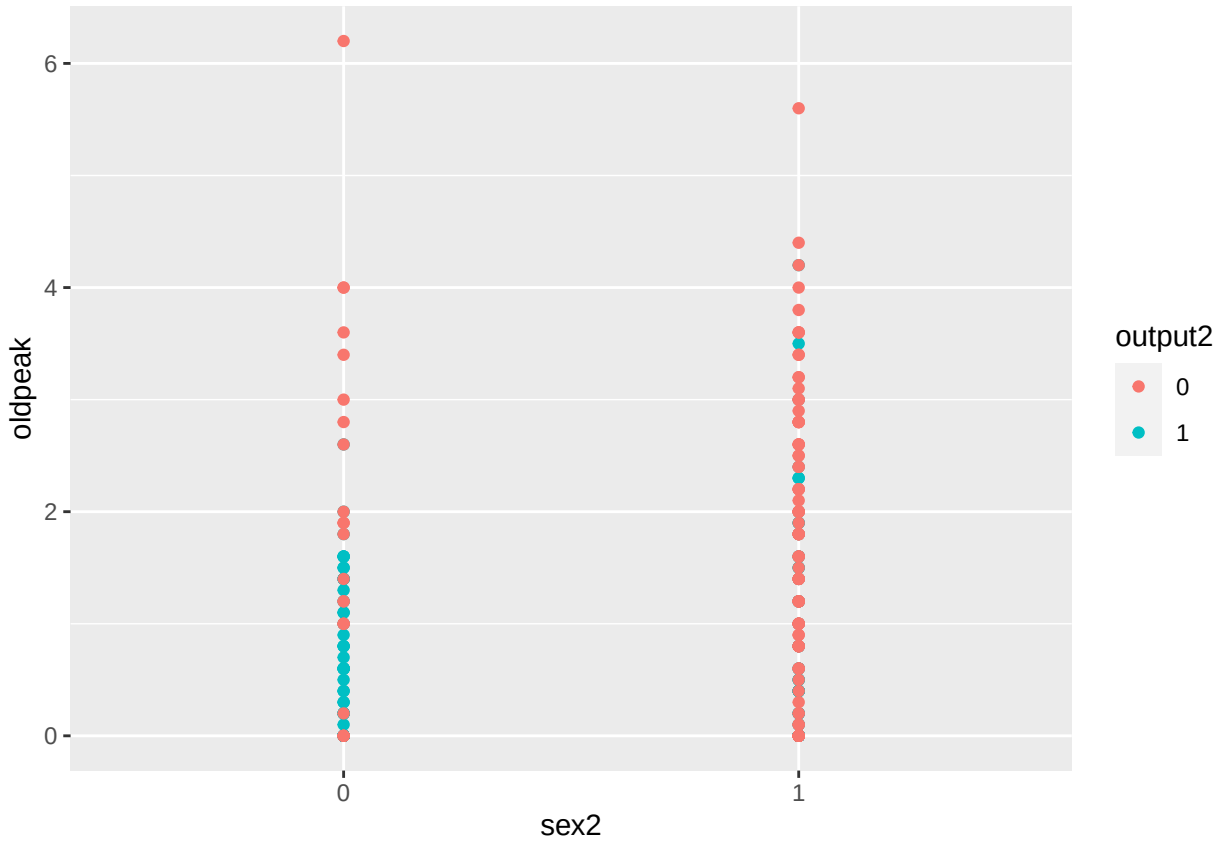
```
heart_data%>%
  #select(sex, output, oldpeak, chol, age)%>%
  mutate(sex2=as.factor(sex))%>%
  ggplot(aes(x=sex2))+
  geom_bar(aes(fill=sex2))+
  labs(fill="Sex; 1 = female, 0 = male", x="Sex of Patient")+
  ggtitle("Distribution of Sex")+
  theme(plot.title = element_text(hjust = .5)) #Center title
```




```
heart_data%>%
  #select(sex, output, oldpeak, chol, age)%>%
  mutate(sex2=as.factor(sex))%>%
  mutate(output2=as.factor(output))%>%
  ggplot(aes(x=output2))+
  geom_bar(aes(fill=output2))+
  labs(fill="Heart Attack Risk; 1 = high, 0 = low", x="Heart Attack Risk of Patient")+
  ggtitle("Distribution of Heart Attack Risk")+
  theme(plot.title = element_text(hjust = .5))
```

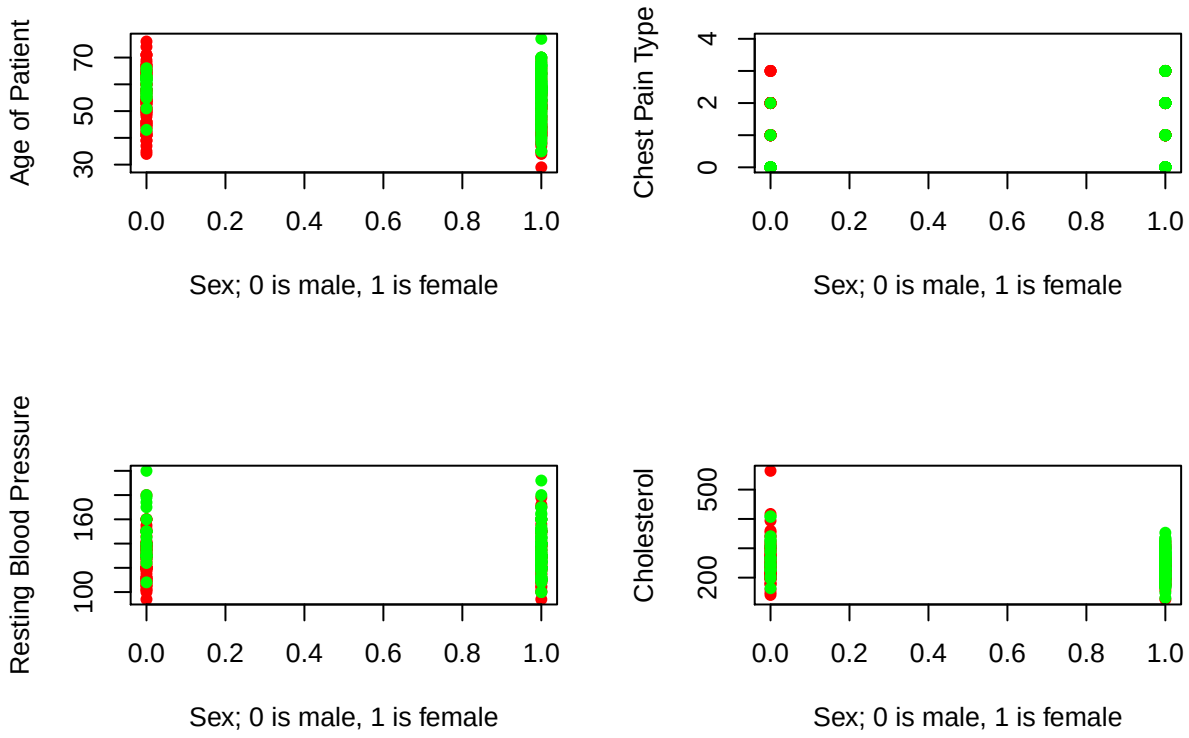



```
heart_data%>%
  select(sex, output, oldpeak, chol, age)%>%
  mutate(sex2=as.factor(sex))%>%
  mutate(output2=as.factor(output))%>%
  ggplot(aes(x=sex2, y=oldpeak))+
  geom_point(aes(col=output2))
```

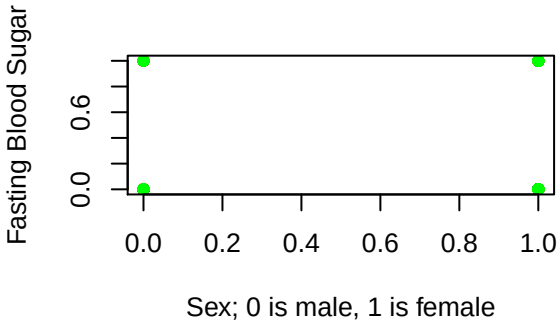


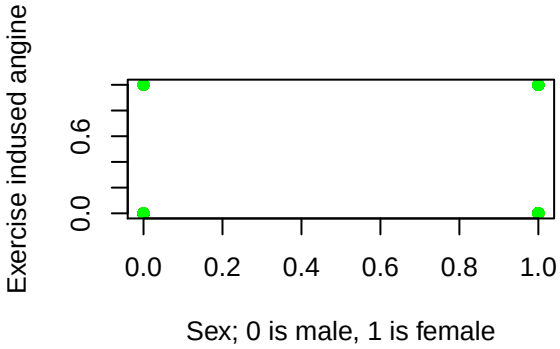
Some graphs comparing males and females:

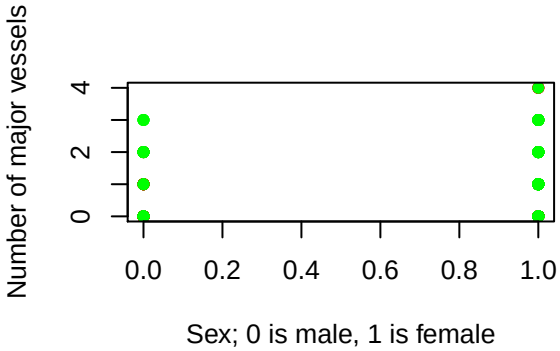
```
par(mfrow=c(2,2))
plot(heart_data$sex, heart_data$age,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Sex; 0 is male, 1 is female", ylab="Age of Patient")
plot(heart_data$sex, heart_data$cp,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Sex; 0 is male, 1 is female", ylab="Chest Pain Type", ylim=c(0,4))
plot(heart_data$sex, heart_data$trtbps,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Sex; 0 is male, 1 is female", ylab="Resting Blood Pressure")
plot(heart_data$sex, heart_data$chol,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Sex; 0 is male, 1 is female", ylab="Cholesterol")
```

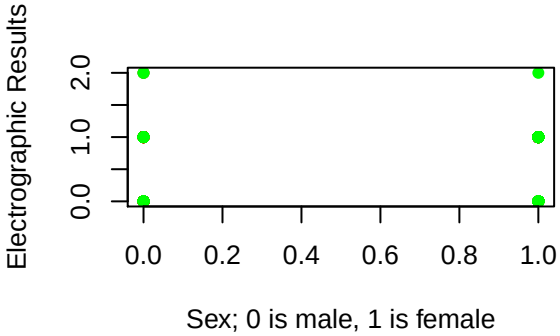



```
par(mfrow=c(2,2))
plot(heart_data$sex, heart_data$fbs,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Sex; 0 is male, 1 is female", ylab="Fasting Blood Sugar")
plot(heart_data$sex, heart_data$exng,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Sex; 0 is male, 1 is female", ylab="Exercise indused angine")
plot(heart_data$sex, heart_data$caa,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Sex; 0 is male, 1 is female", ylab="Number of major vessels")
plot(heart_data$sex, heart_data$restecg,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Sex; 0 is male, 1 is female", ylab="Electrographic Results")
```

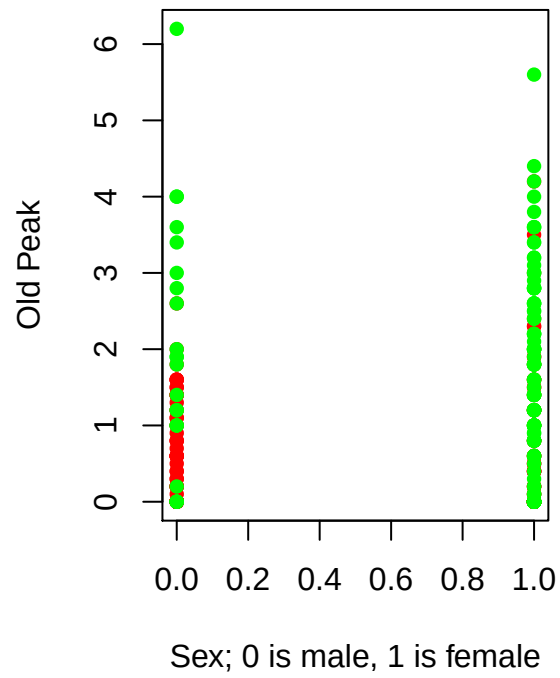


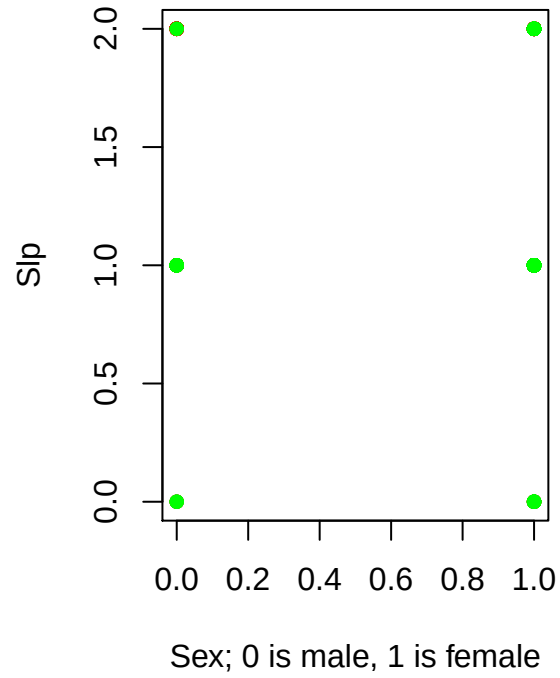




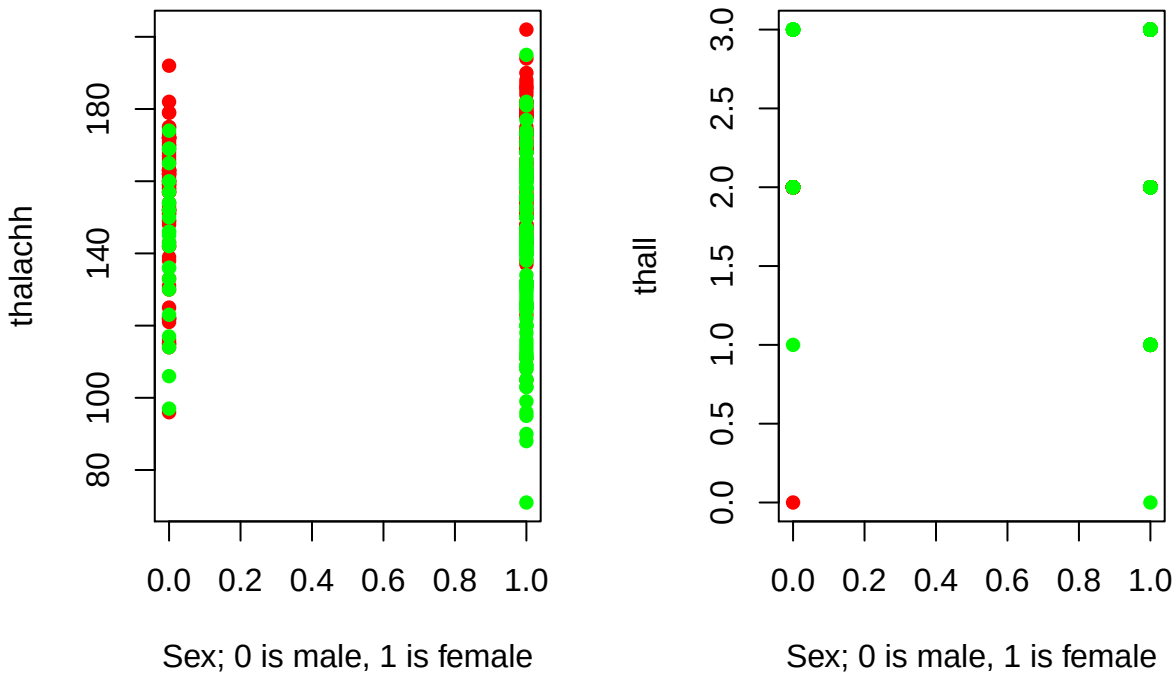



```
par(mfrow=c(1,2))
plot(heart_data$sex, heart_data$oldpeak,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Sex; 0 is male, 1 is female", ylab="Old Peak")
plot(heart_data$sex, heart_data$slp,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Sex; 0 is male, 1 is female", ylab="Slp")
```





```
plot(heart_data$sex, heart_data$thalachh,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Sex; 0 is male, 1 is female", ylab="thalachh")
plot(heart_data$sex, heart_data$thall,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Sex; 0 is male, 1 is female", ylab="thall")
```



Red dots show **high risk** and green dots **low risk** of heart attack.

Overall, we see that in most cases women are categorized at lower risk of heart attach, even with matching cholesterol levels or resting blood pressure levels as males. So **Sex** is an important factor.

Furthermore, we see that in males the high risk patients are concentrated in the ages above 65. We will look at age against risk in more detail below.

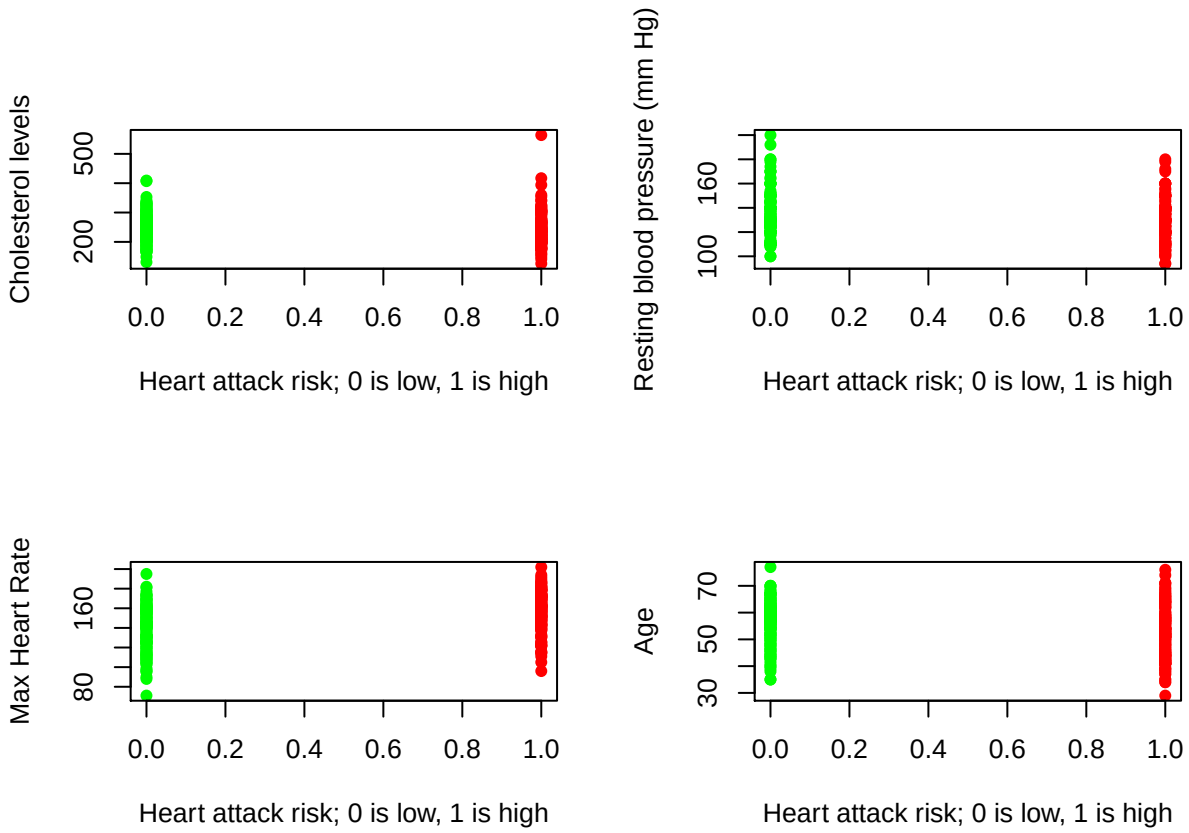
Looking at **Chest Pain Type**, we see that in males, non-anginal chest pain is strictly related to high risk of heart attack, while we see patients in the low & high risks experiencing atypical or typical angina.

Looking at the second set of plots, it appears to be no distinction between high risk and low risk patients.

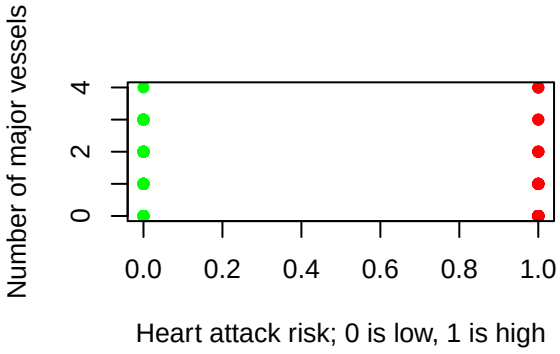
Now some more plots to see how some variables behave for high or low risk:

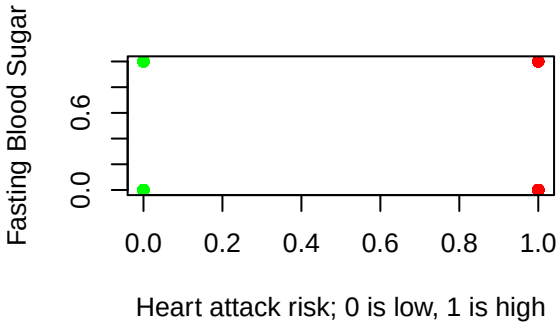
```
par(mfrow=c(2,2))
plot(heart_data$output, heart_data$chol,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Heart attack risk; 0 is low, 1 is high", ylab="Cholesterol levels")

plot(heart_data$output, heart_data$trtbps,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Heart attack risk; 0 is low, 1 is high", ylab="Resting blood pressure (mm Hg)")

plot(heart_data$output, heart_data$thalachh,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Heart attack risk; 0 is low, 1 is high", ylab="Max Heart Rate")
```

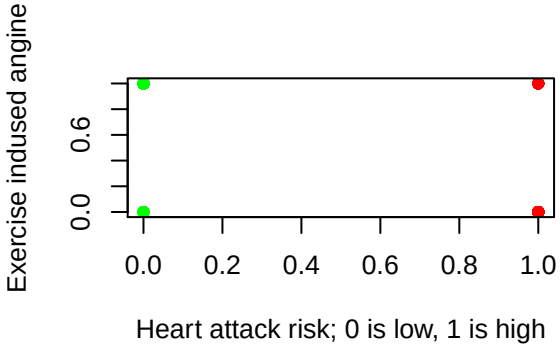
```r
plot(heart_data$output, heart_data$age,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Heart attack risk; 0 is low, 1 is high", ylab="Age")
```

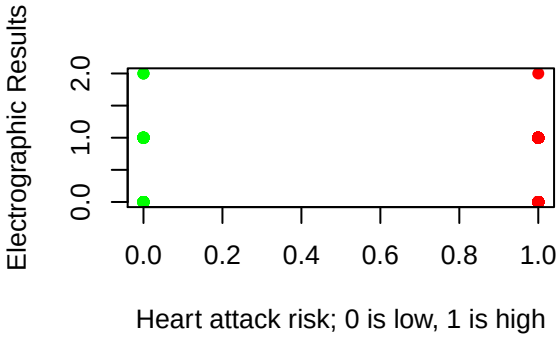



```r
par(mfrow=c(2,2))
plot(heart_data$output, heart_data$caa,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Heart attack risk; 0 is low, 1 is high", ylab="Number of major vessels")
plot(heart_data$output, heart_data$fbs,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Heart attack risk; 0 is low, 1 is high", ylab="Fasting Blood Sugar")
plot(heart_data$output, heart_data$exng,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Heart attack risk; 0 is low, 1 is high", ylab="Exercise indused angine")
plot(heart_data$output, heart_data$restecg,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Heart attack risk; 0 is low, 1 is high", ylab="Electrographic Results")
```

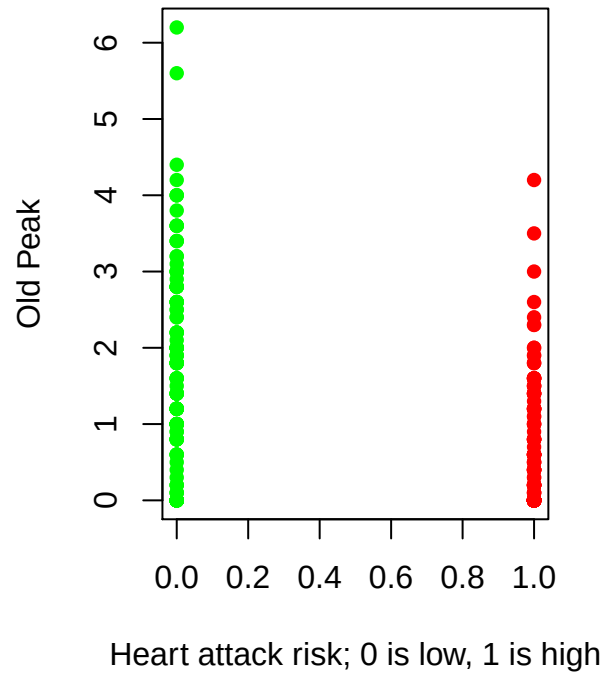


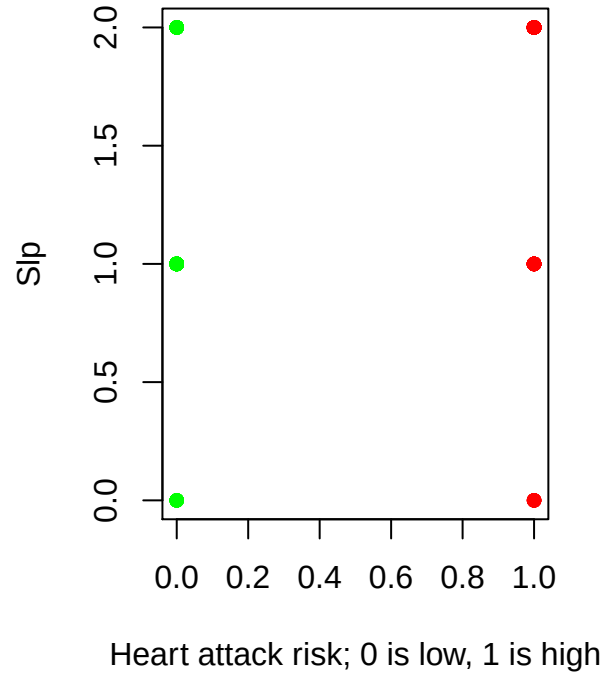







```
par(mfrow=c(1,2))
plot(heart_data$output, heart_data$oldpeak,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Heart attack risk; 0 is low, 1 is high", ylab="Old Peak")
plot(heart_data$output, heart_data$slp,
     col=ifelse(heart_data$output==1,"red","green"), pch=16,
     xlab="Heart attack risk; 0 is low, 1 is high", ylab="Slp")
```





For **cholesterol levels**, we see that higher levels are correlated with high risk patients.

For **Resting blood pressure** we see that values 180 and above marked patients as low risk.

For **Max Heart Rate** we see that while for values in the 100 to 180 range patients were evaluated at both low and high risk, we see that values in the 180 - 200 range primarily marked patients as **high risk**.

Looking at **Age**, high risk and low risk patients of ages ranging from 35 to 70, however with a higher number of high risks below or around 35 and above or around 70. This inequality at the extremes is probably due to the higher amount of high risk patients recorded (165 patients), as opposed the 138 low risk patients recorded.

From the second set of plots, we confirm what we said before that we see no distinction between these variables and heart attack risk.

Finally, let's look at the correlation between the variables:

```
cor(heart_data)
```

```
##                  age         sex          cp      trtbps         chol
## age       1.00000000 -0.09844660 -0.06865302  0.27935091  0.213677957
## sex      -0.09844660  1.00000000 -0.04935288 -0.05676882 -0.197912174
## cp       -0.06865302 -0.04935288  1.00000000  0.04760776 -0.076904391
## trtbps    0.27935091 -0.05676882  0.04760776  1.00000000  0.123174207
## chol      0.21367796 -0.19791217 -0.07690439  0.12317421  1.000000000
## fbs       0.12130765  0.04503179  0.09444403  0.17753054  0.013293602
## restecg  -0.11621090 -0.05819627  0.04442059 -0.11410279 -0.151040078
## thalachh -0.39852194 -0.04401991  0.29576212 -0.04669773 -0.009939839
## exng      0.09680083  0.14166381 -0.39428027  0.06761612  0.067022783
## oldpeak   0.21001257  0.09609288 -0.14923016  0.19321647  0.053951920
## slp      -0.16881424 -0.03071057  0.11971659 -0.12147458 -0.004037770
## caa       0.27632624  0.11826141 -0.18105303  0.10138899  0.070510925
## thall     0.06800138  0.21004110 -0.16173557  0.06220989  0.098802993
## output   -0.22543872 -0.28093658  0.43379826 -0.14493113 -0.085239105
##                  fbs     restecg     thalachh        exng      oldpeak
## age       0.121307648 -0.11621090 -0.398521938  0.09680083  0.210012567
## sex       0.045031789 -0.05819627 -0.044019908  0.14166381  0.096092877
## cp        0.094444035  0.04442059  0.295762125 -0.39428027 -0.149230158
## trtbps    0.177530542 -0.11410279 -0.046697728  0.06761612  0.193216472
## chol      0.013293602 -0.15104008 -0.009939839  0.06702278  0.053951920
## fbs       1.000000000 -0.08418905 -0.008567107  0.02566515  0.005747223
## restecg  -0.084189054  1.00000000  0.044123444 -0.07073286 -0.058770226
## thalachh -0.008567107  0.04412344  1.000000000 -0.37881209 -0.344186948
## exng      0.025665147 -0.07073286 -0.378812094  1.00000000  0.288222808
## oldpeak   0.005747223 -0.05877023 -0.344186948  0.28822281  1.000000000
## slp      -0.059894178  0.09304482  0.386784410 -0.25774837 -0.577536817
## caa       0.137979327 -0.07204243 -0.213176928  0.11573938  0.222682322
## thall    -0.032019339 -0.01198140 -0.096439132  0.20675379  0.210244126
## output   -0.028045760  0.13722950  0.421740934 -0.43675708 -0.430696002
##                  slp         caa       thall      output
## age      -0.16881424  0.27632624  0.06800138 -0.22543872
## sex      -0.03071057  0.11826141  0.21004110 -0.28093658
## cp        0.11971659 -0.18105303 -0.16173557  0.43379826
## trtbps   -0.12147458  0.10138899  0.06220989 -0.14493113
## chol     -0.00403777  0.07051093  0.09880299 -0.08523911
## fbs      -0.05989418  0.13797933 -0.03201934 -0.02804576
## restecg   0.09304482 -0.07204243 -0.01198140  0.13722950
## thalachh  0.38678441 -0.21317693 -0.09643913  0.42174093
```

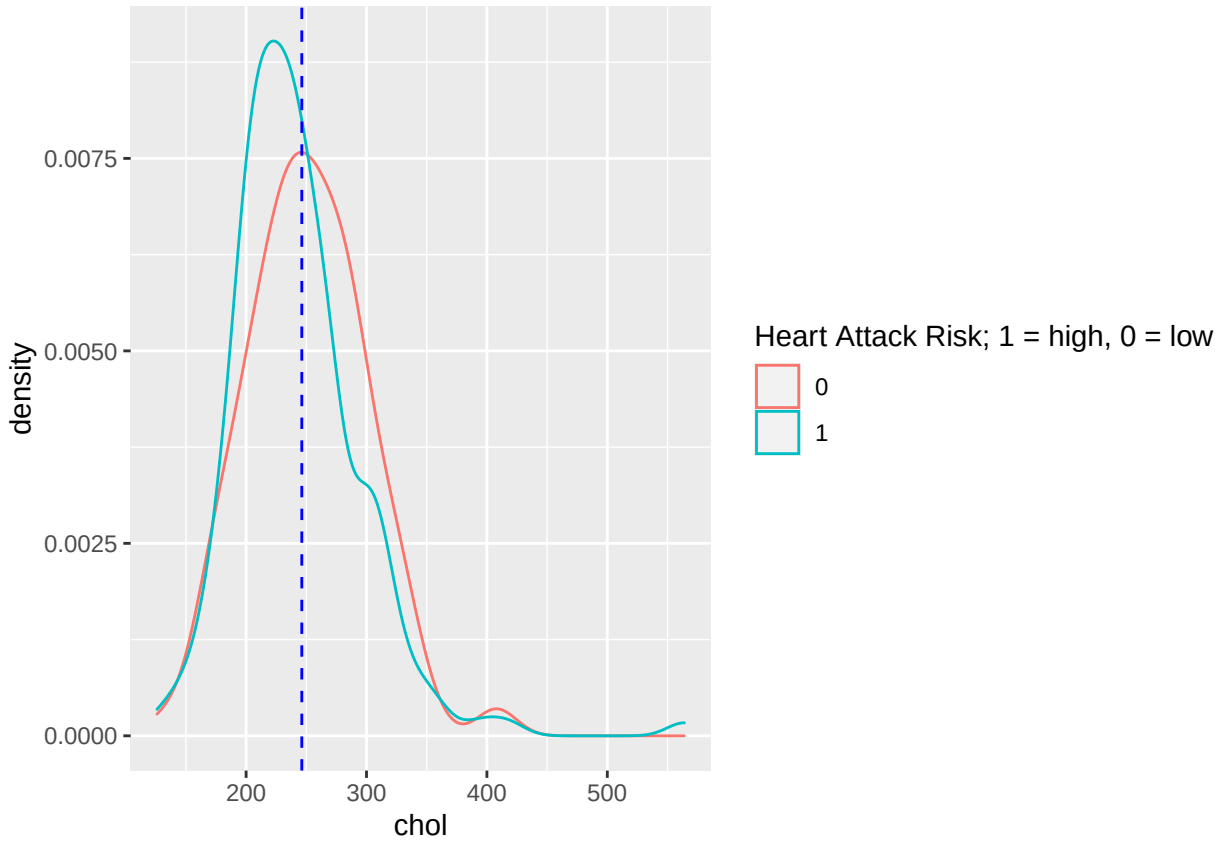
```
## exng     -0.25774837  0.11573938  0.20675379 -0.43675708
## oldpeak  -0.57753682  0.22268232  0.21024413 -0.43069600
## slp       1.00000000 -0.08015521 -0.10476379  0.34587708
## caa      -0.08015521  1.00000000  0.15183213 -0.39172399
## thall    -0.10476379  0.15183213  1.00000000 -0.34402927
## output    0.34587708 -0.39172399 -0.34402927  1.00000000
```

Based on the plots above and the correlation matrix, variables that seem to play a substantial role in determing the risk of heart disease in the patients are **age**, **oldpeak**, **cp**, **trtbps**, **chol**, **thalachh**,**sex**, **thall** and **slp**.

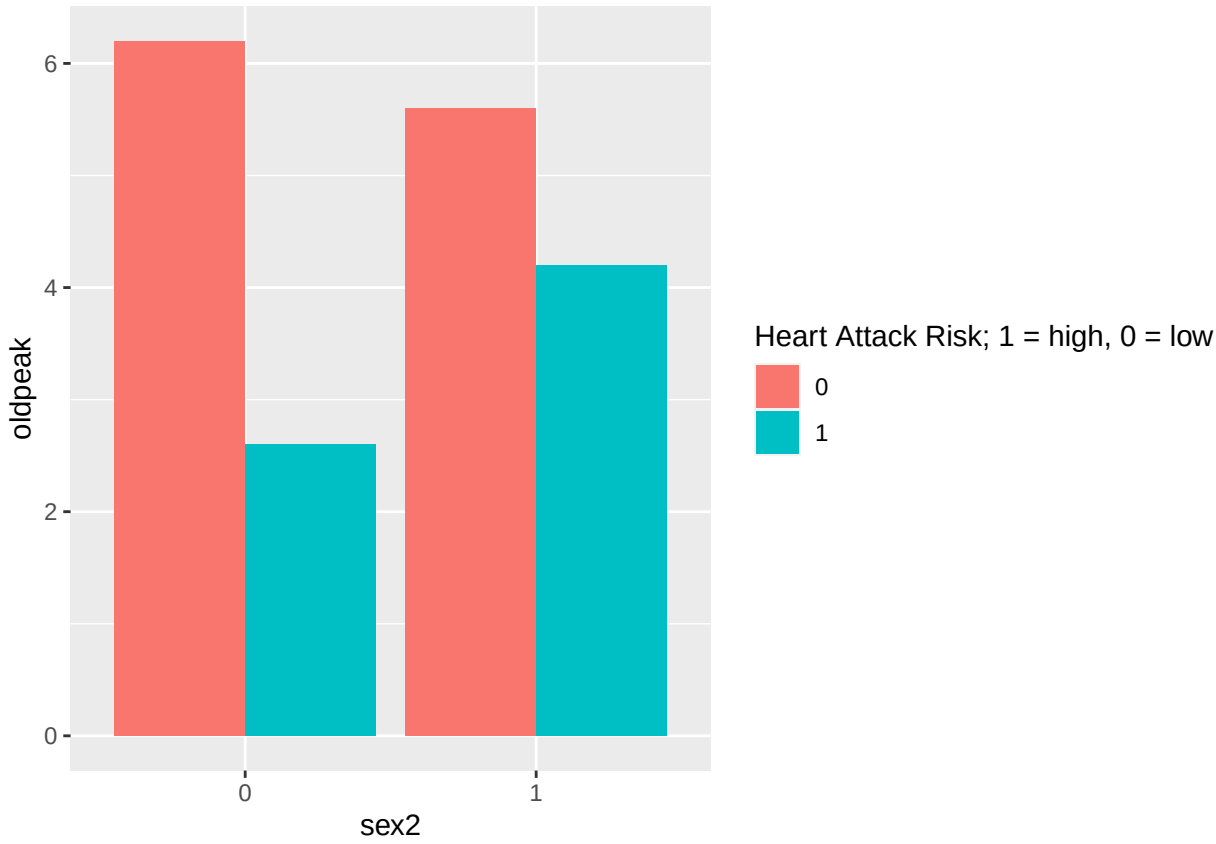
NOTE: Doubt on using, **thall** because there is no documentation on these two variables adn thus am not sure what they represent, although doing that would make more sense the inference domain but we are in the prediction one.

Some more EDA

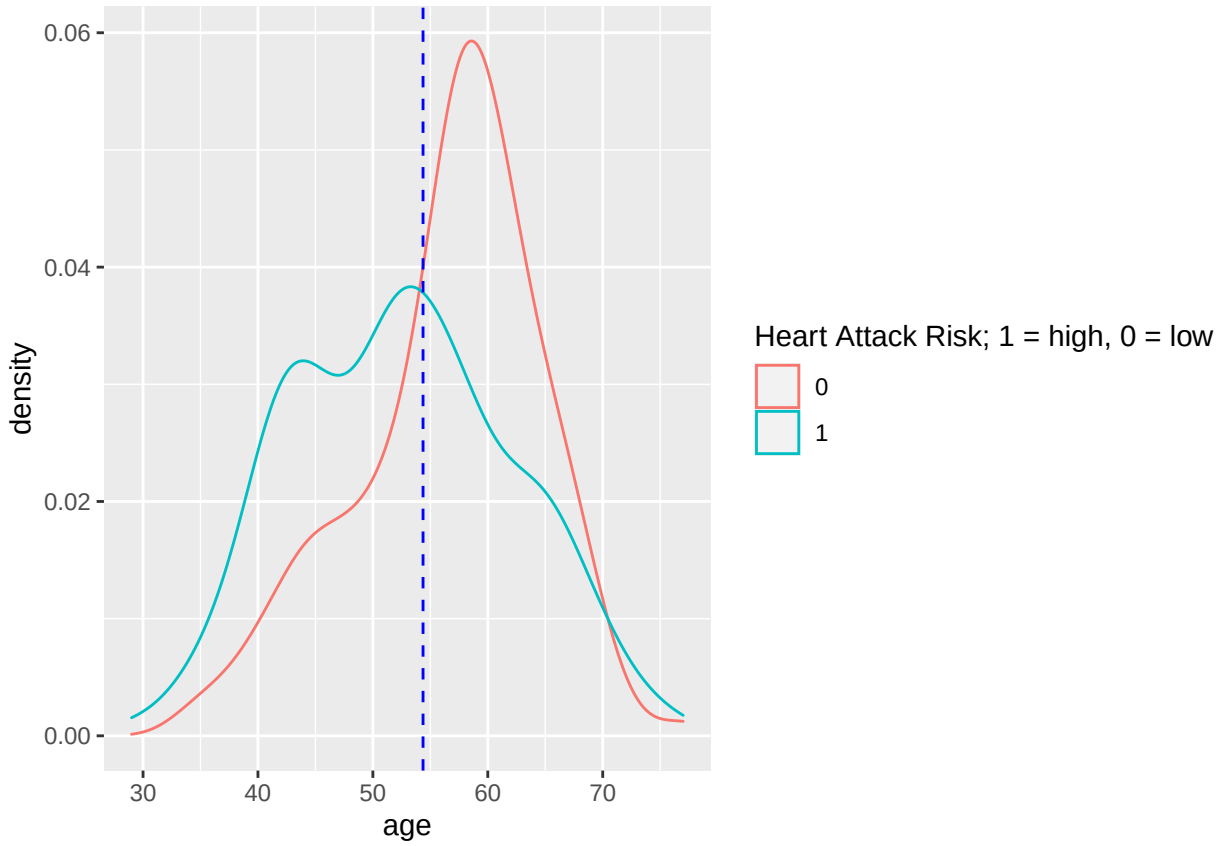
```
library(tidyverse)
library(ggthemes)


par(mfrow=c(1,3))
heart_data%>%
  select(sex, output,oldpeak, chol)%>%
  mutate(sex2=as.factor(sex))%>%
  mutate(output2=as.factor(output))%>%
  ggplot(aes(x=chol))+
  geom_density(aes(col=output2))+
  geom_vline(aes(xintercept=mean(chol)), col="blue", lty=2)+
  labs(col="Heart Attack Risk; 1 = high, 0 = low")
```




```
heart_data%>%
  select(sex, output,oldpeak, chol)%>%
  mutate(sex2=as.factor(sex))%>%
  mutate(output2=as.factor(output))%>%
  ggplot(aes(x=sex2, y=oldpeak))+
  geom_col(aes(fill=output2), position="dodge")+
  labs(fill="Heart Attack Risk; 1 = high, 0 = low")
```




```
heart_data%>%
  select(sex, output, oldpeak, chol, age)%>%
  mutate(sex2=as.factor(sex))%>%
  mutate(output2=as.factor(output))%>%
  ggplot(aes(x=age))+
  geom_density(aes(col=output2))+
  geom_vline(aes(xintercept=mean(age)), col="blue", lty=2)+
  labs(col="Heart Attack Risk; 1 = high, 0 = low")
```



We have a set of candidates for our predictors, however, before settling down on them we will perform BSS and backward stepwise selection in order to see which variables these methods propose.

We first need to split the data into a train set and a test set:

```
#Split data into train and test set
set.seed(1)
test.sample<-sample(nrow(heart_data), nrow(heart_data)/3)#take a third of the data for a the test sampl
heart.train<-heart_data[-test.sample,]
heart.test<-heart_data[test.sample,]
```

BSS:

```
library(leaps)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-1
```

```
#We fit BSS on the whole data set, because we evaluate Cp, BIC and Adjusted R^2
bss.fit<-regsubsets(output~., data=heart_data, nvmax=13)
```

```
sum.bss.fit<-summary(bss.fit)
sum.bss.fit
```

```
## Subset selection object
## Call: regsubsets.formula(output ~ ., data = heart_data, nvmax = 13)
## 13 Variables  (and intercept)
##          Forced in Forced out
## age          FALSE      FALSE
## sex          FALSE      FALSE
## cp           FALSE      FALSE
## trtbps       FALSE      FALSE
## chol         FALSE      FALSE
## fbs          FALSE      FALSE
## restecg      FALSE      FALSE
## thalachh     FALSE      FALSE
## exng         FALSE      FALSE
## oldpeak      FALSE      FALSE
## slp          FALSE      FALSE
## caa          FALSE      FALSE
## thall        FALSE      FALSE
## 1 subsets of each size up to 13
## Selection Algorithm: exhaustive
##           age sex cp  trtbps chol fbs restecg thalachh exng oldpeak slp caa
## 1  ( 1 )  " " " " " " " " " "    " "  " " " " " "      " "      "*"  " "     " " " "
## 2  ( 1 )  " " " " " "*" " " "    " "  " " " " " "      " "      " "  "*"     " " " "
## 3  ( 1 )  " " " " " "*" " " "    " "  " " " " " "      " "      " "  "*"     " " "*"
## 4  ( 1 )  " " "*" "*" " " "    " "  " " " " " "      " "      " "  "*"     " " "*"
## 5  ( 1 )  " " "*" "*" " " "    " "  " " " " " "      " "      "*"  "*"     " " "*"
## 6  ( 1 )  " " "*" "*" " " "    " "  " " " " " "      "*"      " "  "*"     " " "*"
## 7  ( 1 )  " " "*" "*" " " "    " "  " " " " " "      "*"      "*"  "*"     " " "*"
## 8  ( 1 )  " " "*" "*" "*" "    " "  " " " " " "      "*"      "*"  "*"     " " "*"
## 9  ( 1 )  " " "*" "*" "*" "    " "  " " " " " "      "*"      "*"  "*"     "*" "*"
## 10  ( 1 ) " " "*" "*" "*" "    " "  " " " "*"      "*"      "*"  "*"     "*" "*"
## 11  ( 1 ) " " "*" "*" "*" "    "*"  " " " "*"      "*"      "*"  "*"     "*" "*"
## 12  ( 1 ) "*" "*" "*" "*" "    "*"  " " " "*"      "*"      "*"  "*"     "*" "*"
## 13  ( 1 ) "*" "*" "*" "*" "    "*"  "*" "*"      "*"      "*"  "*"     "*" "*"
##           thall
## 1  ( 1 )  " "
## 2  ( 1 )  " "
## 3  ( 1 )  " "
## 4  ( 1 )  " "
## 5  ( 1 )  " "
## 6  ( 1 )  "*"
## 7  ( 1 )  "*"
## 8  ( 1 )  "*"
## 9  ( 1 )  "*"
## 10  ( 1 ) "*"
## 11  ( 1 ) "*"
## 12  ( 1 ) "*"
## 13  ( 1 ) "*"
```

Let's observe Cp, BIC and AdjR2

```
par(mfrow=c(1,3))
plot(sum.bss.fit$cp, xlab="Number of Variables", ylab="Cp", type="l")+
```

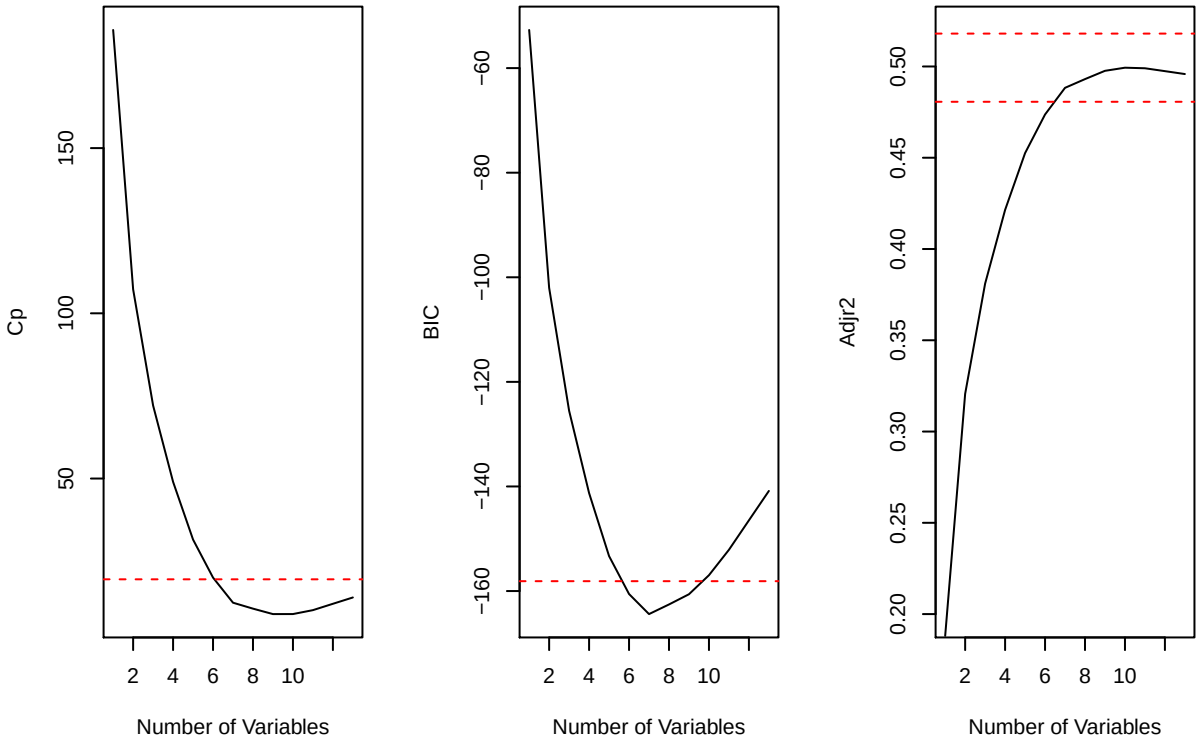
```
  abline(h=min(sum.bss.fit$cp)+.2*sd(sum.bss.fit$cp), col=2, lty=2)+
  abline(h=min(sum.bss.fit$cp)-.2*sd(sum.bss.fit$cp), col=2, lty=2)
```

```
## integer(0)
```

```
plot(sum.bss.fit$bic, xlab="Number of Variables", ylab="BIC", type="l")+
  abline(h=min(sum.bss.fit$bic)+.2*sd(sum.bss.fit$bic), col=2, lty=2)+
  abline(h=min(sum.bss.fit$bic)-.2*sd(sum.bss.fit$bic), col=2, lty=2)
```

```
## integer(0)
```

```
plot(sum.bss.fit$adjr2, xlab="Number of Variables", ylab="Adjr2", type="l", ylim=c(.2, .52))+
  abline(h=max(sum.bss.fit$adjr2)+.2*sd(sum.bss.fit$adjr2), col=2, lty=2)+
  abline(h=max(sum.bss.fit$adjr2)-.2*sd(sum.bss.fit$adjr2), col=2, lty=2)
```




```
## integer(0)
```

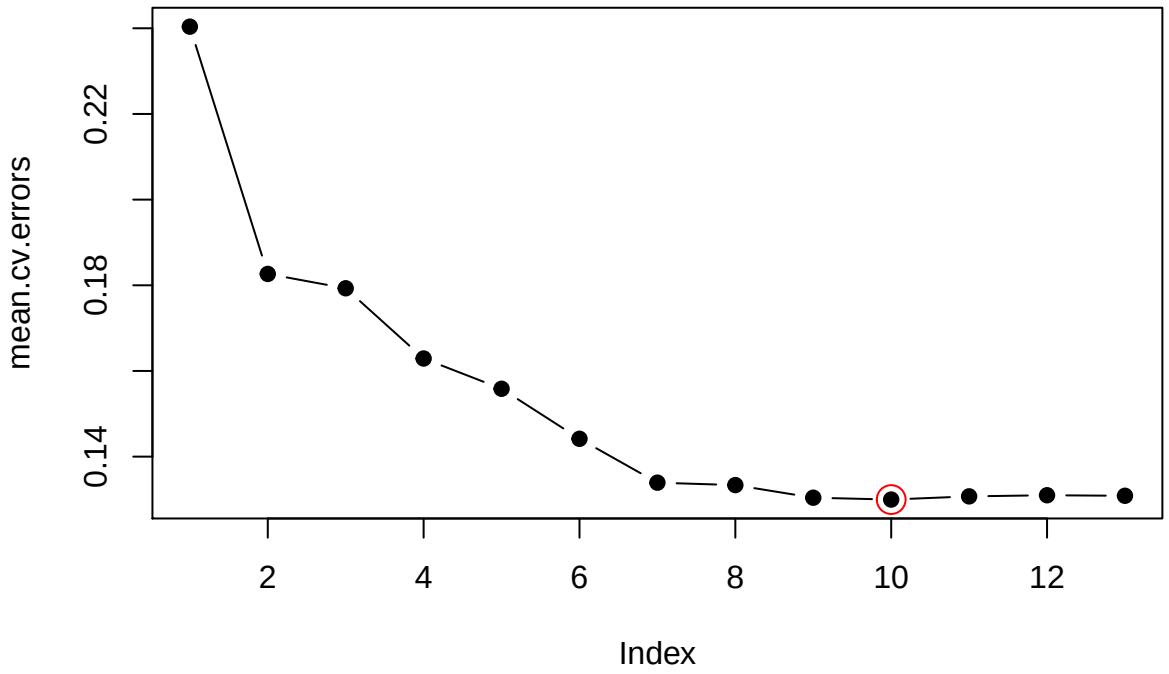
Cp, BIC and Adjr2 all agree on a model of size 7; note that at that size there is a noticeable bend in the curves, also to fact that at 7 variables the curves are within the 0.2 standard deviations from the optimum.

- Let's see what size model 10-fold CV picks:

Need to create a predict function and then perform 10-fold CV:

```
predict.regsubsets<-function(object, newdata, id,...){
  form<-as.formula(object$call[[2]])
  mat<-model.matrix(form, newdata)
  coefs<-coef(object, id=id)
  xvars<-names(coefs)
  mat[,xvars]%*%coefs
}

#10-fold CV:
```

```r
k<-10
set.seed(1)
folds<-sample(rep(1:k, length=nrow(heart_data)))
cv.errors<-matrix(NA, k, 13, dimnames = list(NULL, paste(1:13)))

for(i in 1:k){
  bss.fit<-regsubsets(output~., data=heart_data[folds!=i,], nvmax=13)
  for(j in 1:13){
    preds<-predict(bss.fit, heart_data[folds==i,], id=j)
    cv.errors[i,j]<-mean((heart_data$output[folds==i]-preds)^2)
  }
}
mean.cv.errors<-apply(cv.errors, 2, mean)
plot(mean.cv.errors, pch=19, type="b")+
  points(which.min(mean.cv.errors), mean.cv.errors[which.min(mean.cv.errors)],
         col=2, cex=2)
```




```
## integer(0)
```

We see that the 10 variable model achieves the lowest test MSE, although the 7 variable model's test MSE is not too far off and it is a simpler model.

```r
mean.cv.errors[7]
```

```
##         7
## 0.1339199
```

```r
mean.cv.errors[10]
```

```
##        10
## 0.1299776
```

Backward Stepwise Selection:

```r
library(leaps)
bkwd.fit<-regsubsets(output~., data=heart.train, nvmax=13, method="backward")
sum.bkwd.fit<-summary(bkwd.fit)
sum.bkwd.fit
```

```
## Subset selection object
## Call: regsubsets.formula(output ~ ., data = heart.train, nvmax = 13,
##     method = "backward")
## 13 Variables  (and intercept)
##          Forced in Forced out
## age          FALSE      FALSE
## sex          FALSE      FALSE
## cp           FALSE      FALSE
## trtbps       FALSE      FALSE
## chol         FALSE      FALSE
## fbs          FALSE      FALSE
## restecg      FALSE      FALSE
## thalachh     FALSE      FALSE
## exng         FALSE      FALSE
## oldpeak      FALSE      FALSE
## slp          FALSE      FALSE
## caa          FALSE      FALSE
## thall        FALSE      FALSE
## 1 subsets of each size up to 13
## Selection Algorithm: backward
##           age sex cp  trtbps chol fbs restecg thalachh exng oldpeak slp caa
## 1  ( 1 )  " " " " " " " "    " "  " " " "     " "      " "  "*"     " " " "
## 2  ( 1 )  " " " " "*" " "    " "  " " " "     " "      " "  "*"     " " " "
## 3  ( 1 )  " " " " "*" " "    " "  " " " "     " "      " "  "*"     " " "*"
## 4  ( 1 )  " " " " "*" " "    " "  " " "*"     " "      " "  "*"     " " "*"
## 5  ( 1 )  " " " " "*" " "    " "  " " "*"     " "      "*"  "*"     " " "*"
## 6  ( 1 )  " " "*" "*" " "    " "  " " "*"     " "      "*"  "*"     " " "*"
## 7  ( 1 )  " " "*" "*" " "    " "  " " "*"     " "      "*"  "*"     " " "*"
## 8  ( 1 )  " " "*" "*" " "    " "  " " "*"     " "      "*"  "*"     "*" "*"
## 9  ( 1 )  "*" "*" "*" " "    " "  " " "*"     " "      "*"  "*"     "*" "*"
## 10  ( 1 ) "*" "*" "*" " "    " "  " " "*"     "*"      "*"  "*"     "*" "*"
## 11  ( 1 ) "*" "*" "*" "*"    " "  " " "*"     "*"      "*"  "*"     "*" "*"
## 12  ( 1 ) "*" "*" "*" "*"    "*"  " " "*"     "*"      "*"  "*"     "*" "*"
## 13  ( 1 ) "*" "*" "*" "*"    "*"  "*" "*"     "*"      "*"  "*"     "*" "*"
##           thall
## 1  ( 1 )  " "
## 2  ( 1 )  " "
## 3  ( 1 )  " "
## 4  ( 1 )  " "
## 5  ( 1 )  " "
## 6  ( 1 )  " "
## 7  ( 1 )  "*"
## 8  ( 1 )  "*"
## 9  ( 1 )  "*"
## 10  ( 1 ) "*"
## 11  ( 1 ) "*"
## 12  ( 1 ) "*"
## 13  ( 1 ) "*"
```

We observe Cp, BIC and Adjr2:

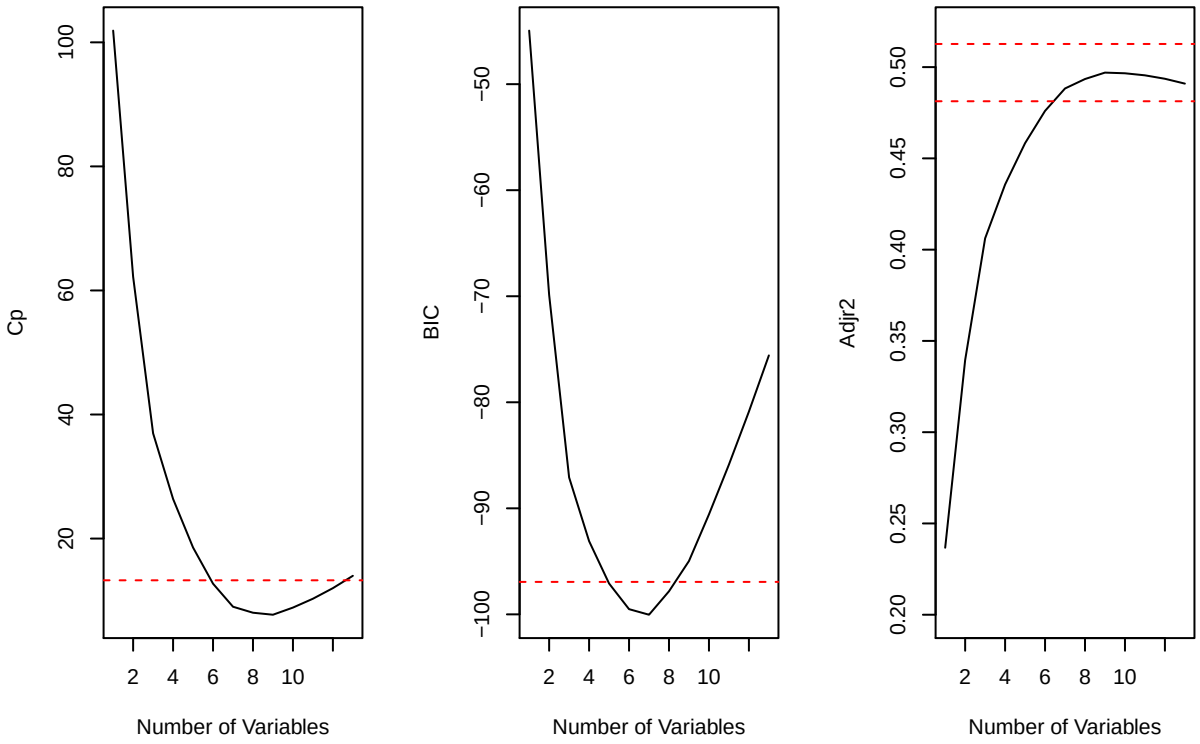
```
par(mfrow=c(1,3))
plot(sum.bkwd.fit$cp, xlab="Number of Variables", ylab="Cp", type="l")+
  abline(h=min(sum.bkwd.fit$cp)+.2*sd(sum.bkwd.fit$cp), col=2, lty=2)+
  abline(h=min(sum.bkwd.fit$cp)-.2*sd(sum.bkwd.fit$cp), col=2, lty=2)
```

```
## integer(0)
```

```
plot(sum.bkwd.fit$bic, xlab="Number of Variables", ylab="BIC", type="l")+
  abline(h=min(sum.bkwd.fit$bic)+.2*sd(sum.bkwd.fit$bic), col=2, lty=2)+
  abline(h=min(sum.bkwd.fit$bic)-.2*sd(sum.bkwd.fit$bic), col=2, lty=2)
```

```
## integer(0)
```

```
plot(sum.bkwd.fit$adjr2, xlab="Number of Variables", ylab="Adjr2", type="l", ylim=c(.2, .52))+
  abline(h=max(sum.bkwd.fit$adjr2)+.2*sd(sum.bkwd.fit$adjr2), col=2, lty=2)+
  abline(h=max(sum.bkwd.fit$adjr2)-.2*sd(sum.bkwd.fit$adjr2), col=2, lty=2)
```



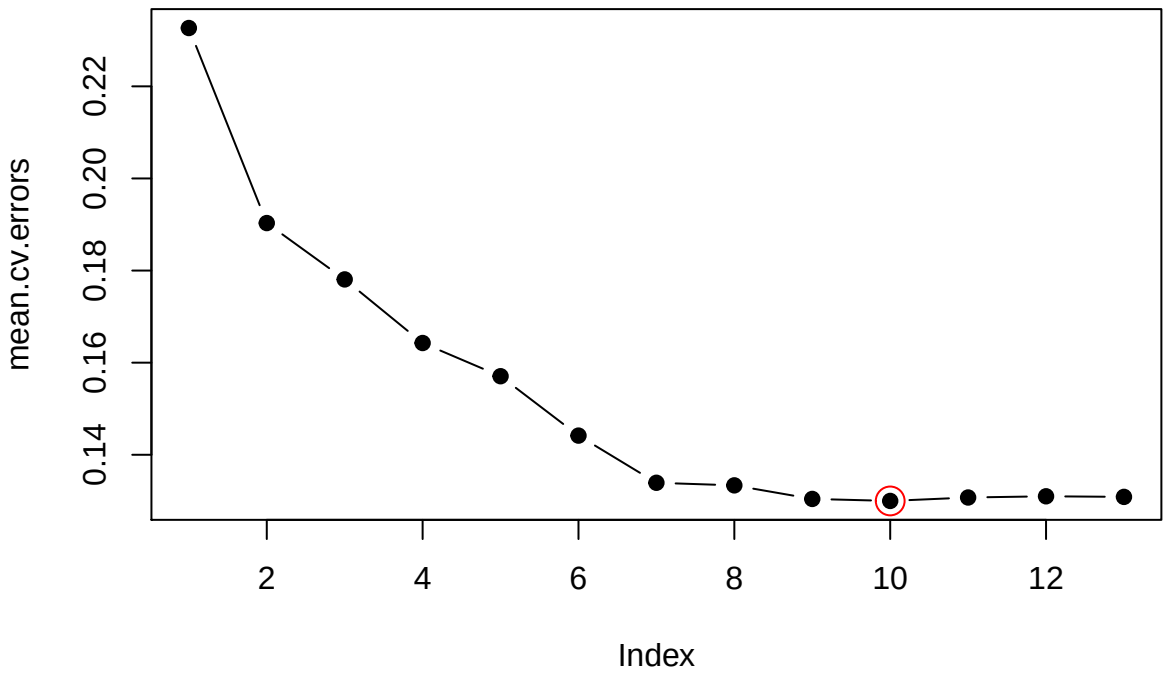

```
## integer(0)
```

Similarly to how BSS behaved, Backward Stepwise Selection agrees on a 7 variable model and perhaps even a 6 variable model.

Let's see what model size 10-fold CV picks:

```
#10-fold CV:
k<-10
set.seed(1)
folds<-sample(rep(1:k, length=nrow(heart_data)))
cv.errors<-matrix(NA, k, 13, dimnames = list(NULL, paste(1:13)))

for(i in 1:k){
  bkwd.fit<-regsubsets(output~., data=heart_data[folds!=i,], nvmax=13, method="backward")
```

```
  for(j in 1:13){
    preds<-predict(bkwd.fit, heart_data[folds==i,], id=j)
    cv.errors[i,j]<-mean((heart_data$output[folds==i]-preds)^2)
  }
}
mean.cv.errors<-apply(cv.errors, 2, mean)
plot(mean.cv.errors, pch=19, type="b")+
  points(which.min(mean.cv.errors), mean.cv.errors[which.min(mean.cv.errors)],
         col=2, cex=2)
```




```
## integer(0)
```

Just like BSS, Backward Stepwise Selection picks the 10 variable model as the one with the lowest CV error.

Let's compare the CV errors of the 7 & 10 variable models:

```
mean.cv.errors[7]
```

```
##         7
## 0.1339199
```

```
mean.cv.errors[10]
```

```
##        10
## 0.1299776
```

Again the errors of both models are almost the same but the 7 variable model is simpler. We proposed the option that a 6 variable model could also be a candidate but looking at the above plot we exclude this proposition.

Let's see the variables picked by the 7 variable BSS model and the 7 variable Backward SS model:

```
bss.fit<-bss.fit<-regsubsets(output~., data=heart.train, nvmax=13)
bkwd.fit<-regsubsets(output~., data=heart.train, nvmax=13, method="backward")
#The BSS fit
coef(bss.fit, 7)
```

```
## (Intercept)         sex          cp     restecg        exng     oldpeak
##   0.9679435  -0.1362930   0.1004686   0.1623166  -0.1600877  -0.1281846
##         caa       thall
##  -0.1183469  -0.1003198
```

```
#The Backward SS
coef(bkwd.fit, 7)
```

```
## (Intercept)         sex          cp     restecg        exng     oldpeak
##   0.9679435  -0.1362930   0.1004686   0.1623166  -0.1600877  -0.1281846
##         caa       thall
##  -0.1183469  -0.1003198
```

The models picked by BSS and Backward SS are identical and they use **sex**, **cp**, **restecg**, **exng**, **oldpeak**, **caa** and **thall** as variables.

From our earlier analysis on the variables that played an important role included **age**, **oldpeak**, **cp**, **trtbps**, **chol**, **thalachh**,**sex**, **thall** and **slp**.