Lab 2

Sofia Hein Machado

2024-08-07

Q1 Load the data.

```
Heart <- read.csv("~/Desktop/UH courses/INDE 4364/Lab 2/Heart.csv", header = T,row.names
= 1)
summary (Heart)
```

```
##       Age             Sex          ChestPain             RestBP
##  Min.   :29.00   Min.   :0.0000   Length:303         Min.   : 94.0
##  1st Qu.:48.00   1st Qu.:0.0000   Class :character   1st Qu.:120.0
##  Median :56.00   Median :1.0000   Mode  :character   Median :130.0
##  Mean   :54.44   Mean   :0.6799                      Mean   :131.7
##  3rd Qu.:61.00   3rd Qu.:1.0000                      3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000                      Max.   :200.0
##
##       Chol            Fbs            RestECG          MaxHR
##  Min.   :126.0   Min.   :0.0000   Min.   :0.0000   Min.   : 71.0
##  1st Qu.:211.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:133.5
##  Median :241.0   Median :0.0000   Median :1.0000   Median :153.0
##  Mean   :246.7   Mean   :0.1485   Mean   :0.9901   Mean   :149.6
##  3rd Qu.:275.0   3rd Qu.:0.0000   3rd Qu.:2.0000   3rd Qu.:166.0
##  Max.   :564.0   Max.   :1.0000   Max.   :2.0000   Max.   :202.0
##
##      ExAng           Oldpeak         Slope             Ca
##  Min.   :0.0000   Min.   :0.00   Min.   :1.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000
##  Median :0.0000   Median :0.80   Median :2.000   Median :0.0000
##  Mean   :0.3267   Mean   :1.04   Mean   :1.601   Mean   :0.6722
##  3rd Qu.:1.0000   3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :6.20   Max.   :3.000   Max.   :3.0000
##                                                  NA's   :4
##      Thal               AHD
##  Length:303         Length:303
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
##
```

How many variables in the dataset? What are they? Are they quantitative or qualitative variables? Is there any missing value?

There are 14 features, they are both quantitative and qualitative variables. There are 4 missing values in the variable Ca.

Q2 Impute the missing value using mean value for quantitative variable and majority class for qualitative variable.

```
MeanCA <- mean(Heart$Ca,na.rm = T)
Index <- is.na(Heart$Ca)
MeanCA
```

```
## [1] 0.6722408
```

```
Heart$Ca[Index == TRUE] <- MeanCA
Heart$Ca[Index == TRUE]
```

```
## [1] 0.6722408 0.6722408 0.6722408 0.6722408
```

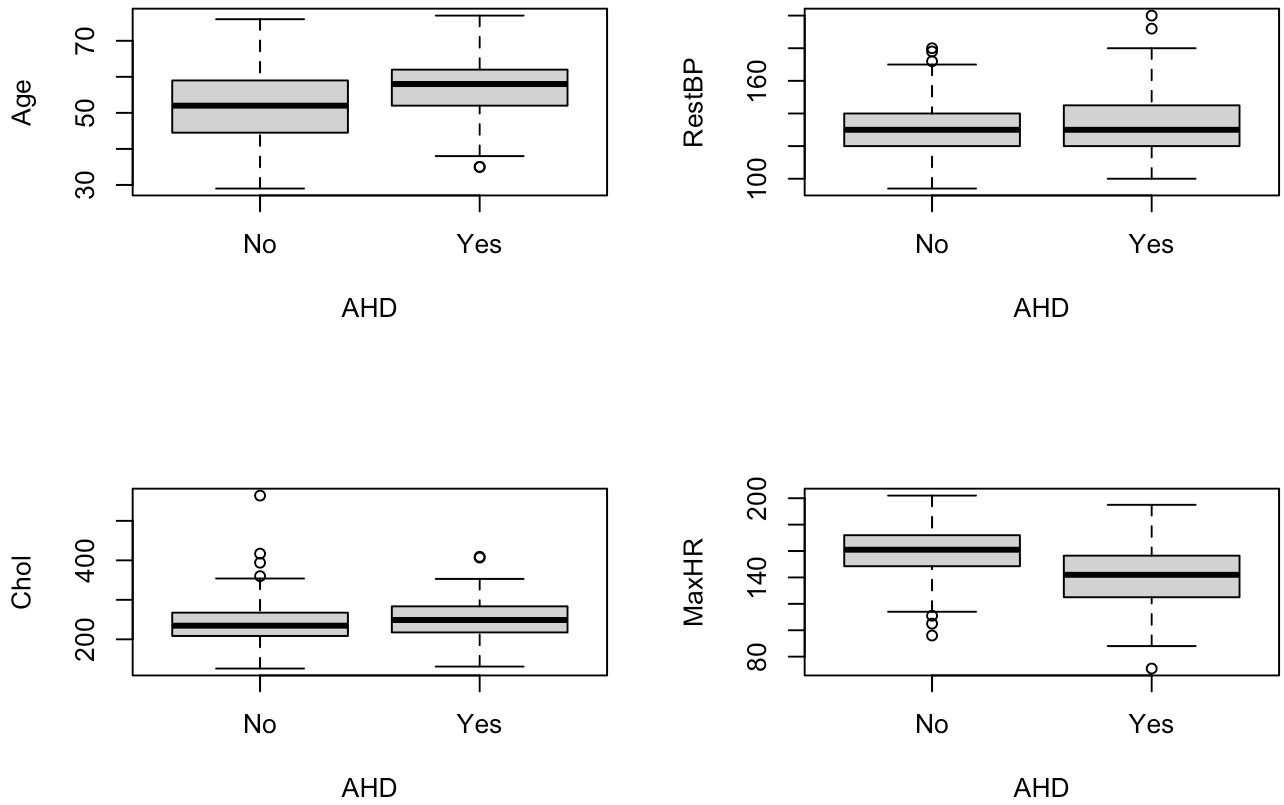
```
summary(Heart)
```

```
##       Age             Sex          ChestPain             RestBP
##  Min.   :29.00   Min.   :0.0000   Length:303         Min.   : 94.0
##  1st Qu.:48.00   1st Qu.:0.0000   Class :character   1st Qu.:120.0
##  Median :56.00   Median :1.0000   Mode  :character   Median :130.0
##  Mean   :54.44   Mean   :0.6799                      Mean   :131.7
##  3rd Qu.:61.00   3rd Qu.:1.0000                      3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000                      Max.   :200.0
##       Chol            Fbs            RestECG           MaxHR
##  Min.   :126.0   Min.   :0.0000   Min.   :0.0000   Min.   : 71.0
##  1st Qu.:211.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:133.5
##  Median :241.0   Median :0.0000   Median :1.0000   Median :153.0
##  Mean   :246.7   Mean   :0.1485   Mean   :0.9901   Mean   :149.6
##  3rd Qu.:275.0   3rd Qu.:0.0000   3rd Qu.:2.0000   3rd Qu.:166.0
##  Max.   :564.0   Max.   :1.0000   Max.   :2.0000   Max.   :202.0
##      ExAng           Oldpeak         Slope             Ca
##  Min.   :0.0000   Min.   :0.00   Min.   :1.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000
##  Median :0.0000   Median :0.80   Median :2.000   Median :0.0000
##  Mean   :0.3267   Mean   :1.04   Mean   :1.601   Mean   :0.6722
##  3rd Qu.:1.0000   3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :6.20   Max.   :3.000   Max.   :3.0000
##      Thal               AHD
##  Length:303         Length:303
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

Q3 Data Visualization

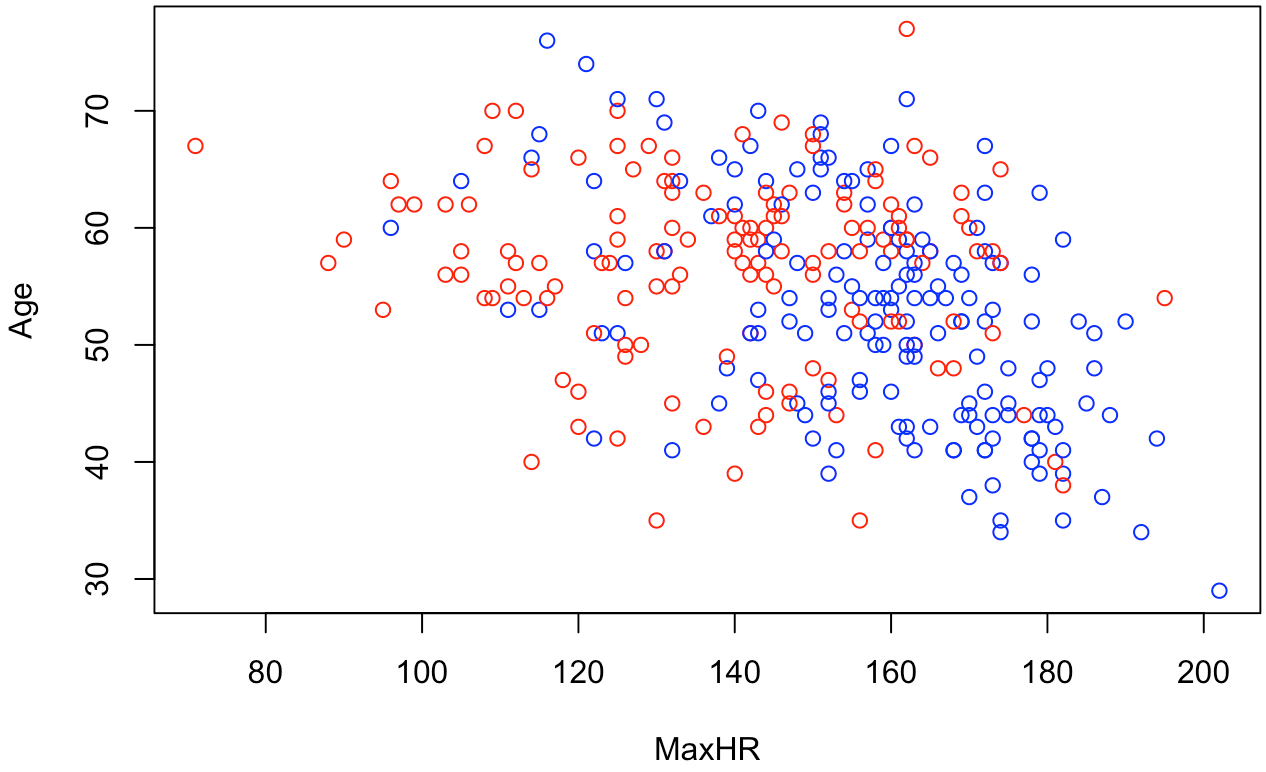
a) Please use the boxplot to visualize the relationship of quantitative predictor with outcome (AHD).

```
par(mfrow=c(2,2))
boxplot(Age ~ AHD, xlab = "AHD", ylab = "Age", data = Heart)
boxplot(RestBP ~ AHD, xlab = "AHD", ylab = "RestBP", data = Heart)
boxplot(Chol ~ AHD, xlab = "AHD", ylab = "Chol", data = Heart)
boxplot(MaxHR ~ AHD, xlab = "AHD", ylab = "MaxHR", data = Heart)
```



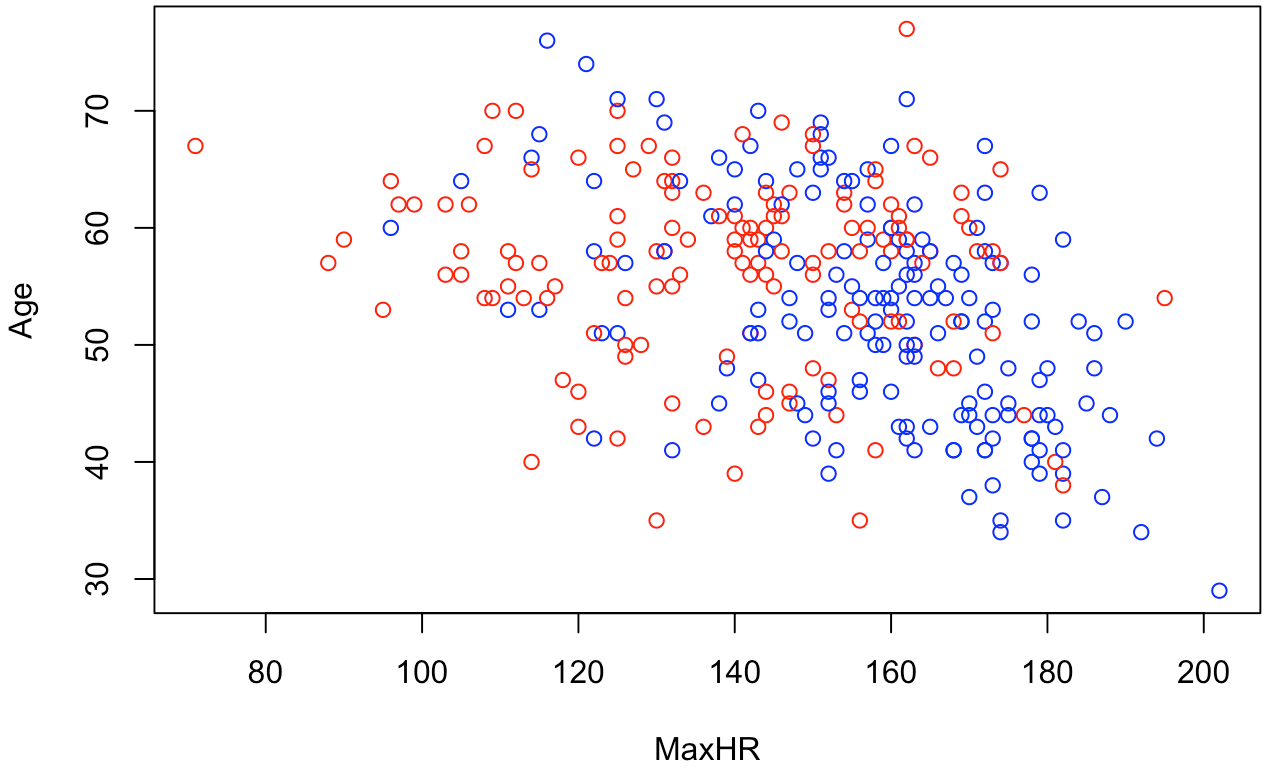
b) Please use the corrplots to visualize the correlations between quantitative variables.

```
col = rep(0,nrow(Heart)) #indicate the color of each data point
col[Heart$AHD == "No"] <- 'blue'
col[Heart$AHD == 'Yes'] <- 'red'
plot(Age~MaxHR, col = col, data= Heart)
```



c) Please use the scatterplots to find the patterns in predictors that distinguish AHD=Yes and AHD=No patients.

```
col = rep(0,nrow(Heart)) #indicate the color of each data point
col[Heart$AHD == "No"] <- 'blue'
col[Heart$AHD == 'Yes'] <- 'red'
plot(Age~MaxHR, col = col, data = Heart)
```



Q4 Please separate the dataset into training and testing data with ratio of 3:1. Use the sample function.

```
set.seed(123)
y <- rep(0, nrow(Heart))
y[Heart$AHD == 'Yes'] <- 1 #assigned those who have heart disease to number 1
y
```

```
##   [1] 0 1 1 0 0 0 1 0 1 1 0 0 1 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0 1 1 0 0 0 1
##  [38] 1 1 0 1 0 0 0 1 1 0 1 0 0 0 0 1 0 1 1 1 1 0 0 1 0 1 0 1 1 1 0 1 1 0 1 1 1
##  [75] 1 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 1 0 1 1 1 1 1
## [112] 1 0 1 1 0 0 0 1 1 1 1 0 1 1 0 1 1 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 1 0 1 1 0
## [149] 0 0 0 0 0 0 1 1 1 1 1 1 0 0 1 0 0 0 0 0 0 1 0 1 0 1 0 1 1 0 1 0 0 1 1 0 0 1
## [186] 0 0 1 1 1 0 1 1 1 0 1 0 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 0 1 1 0 0 0 0 0 0 0
## [223] 0 1 1 0 0 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 0 1 0 1 1 1 1 0 0 1 0 0 0 0 0 0 0
## [260] 1 0 1 0 0 1 1 1 1 1 0 1 0 1 0 1 0 0 0 1 0 1 0 1 0 1 1 1 0 0 0 1 0 1 1 1 0
## [297] 1 1 1 1 1 1 0
```

```
Heart <- data.frame(Heart[,-14], y)
N= nrow(Heart)
testid = sample(c(1:N), round(0.25*N), replace = FALSE)
train = Heart[-testid,]
test = Heart[testid,]
nrow(train)
```

```
## [1] 227
```

```
nrow(test)
```

```
## [1] 76
```

Q5 Logistic regression.

Please fit a logistic regression model on training data to predict AHD (whether the patient will have a heart disease or not) using the other variables.

```
fit.logit <- glm(y~., data = train, family = binomial())
summary(fit.logit)
```

```
##
## Call:
## glm(formula = y ~ ., family = binomial(), data = train)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -5.114743   3.364504  -1.520 0.128459
## Age                 -0.002056   0.029852  -0.069 0.945089
## Sex                  1.566666   0.594090   2.637 0.008362 **
## ChestPainnonanginal -1.553989   0.563088  -2.760 0.005784 **
## ChestPainnontypical -1.181997   0.664911  -1.778 0.075457 .
## ChestPaintypical    -2.259252   0.762420  -2.963 0.003044 **
## RestBP               0.022941   0.012076   1.900 0.057466 .
## Chol                 0.008867   0.005029   1.763 0.077879 .
## Fbs                 -0.633965   0.677226  -0.936 0.349211
## RestECG              0.300914   0.216033   1.393 0.163648
## MaxHR               -0.019875   0.011757  -1.690 0.090948 .
## ExAng                0.503591   0.500469   1.006 0.314301
## Oldpeak              0.527682   0.272642   1.935 0.052936 .
## Slope                0.157328   0.436946   0.360 0.718801
## Ca                   1.066934   0.288201   3.702 0.000214 ***
## Thalnormal           0.140674   0.952862   0.148 0.882632
## Thalreversable       1.753720   0.924119   1.898 0.057733 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 311.16  on 224  degrees of freedom
## Residual deviance: 152.00  on 208  degrees of freedom
##   (2 observations deleted due to missingness)
## AIC: 186
##
## Number of Fisher Scoring iterations: 6
```

a) Is there any significant predictor?

Yes: Sex, ChestPainnonanginal, CheatPaintypical and Ca.

b) Interpret the effects of significant predictors.

Sex: Males are approximately 4.79 times more likely to have heart disease than females, controlling for other factors in the model. ChestPainnonanginal: Patients with non-anginal chest pain are about 0.21 times as likely to have a heart disease. ChestPaintypical: patients with typical chest pain are about 0.10 as likely to have heart disease. Ca: the odds of having a heart disease increase by 2.91 times when Ca is increased.

c) How good this model fits the data: confusion matrix, accuracy, ROC and AUC

This matrix tells us:

True negatives: 109

False positives: 10

False negatives: 18

True positives: 88

Accuracy: 86.7%

```
fit.p = predict(fit.logit, newdata = train, type = 'response')
fit.p
```

```
##           1           2           3           5           6           8
## 0.133293364 0.997341302 0.994586266 0.033253886 0.056397039 0.188234411
##           9          10          11          12          15          17
## 0.957352449 0.931312761 0.237424930 0.131132996 0.235513633 0.297270348
##          18          19          20          21          24          27
## 0.359637455 0.023991772 0.093308754 0.130573472 0.936436844 0.015661208
##          28          29          31          33          36          37
## 0.087405707 0.422066270 0.141323100 0.130820808 0.159962651 0.942344465
##          38          40          42          44          45          46
## 0.942717988 0.182142344 0.214558548 0.074849582 0.142978171 0.762140108
##          47          48          49          51          53          54
## 0.049185970 0.969025296 0.221227358 0.014910944 0.605435414 0.061053052
##          55          56          57          58          59          60
## 0.948464040 0.991669389 0.570812896 0.447414534 0.401187112 0.393848600
##          61          62          64          65          66          67
## 0.700587054 0.065411439 0.008996572 0.915984303 0.997802897 0.307097254
##          68          70          71          73          74          75
## 0.589762489 0.431958253 0.043828254 0.994135127 0.752163638 0.285090049
##          77          79          80          82          83          84
## 0.987868335 0.110186977 0.967716784 0.186275242 0.136858515 0.848118551
##          85          87          88          92          93          95
## 0.098211426 0.124041618          NA 0.998168170 0.957641973 0.018245505
##          96          97          98          99         100         101
## 0.861073436 0.953692301 0.971979335 0.208530790 0.157570272 0.186333244
##         102         103         104         105         106         107
## 0.016066353 0.309564827 0.039452676 0.941531228 0.307246690 0.797965740
##         108         111         113         114         115         117
## 0.671435240 0.833157516 0.011917968 0.906241142 0.588747932 0.041006254
##         119         120         122         123         124         125
## 0.997850702 0.987358517 0.998802986 0.100259964 0.992986651 0.181830497
##         126         128         129         130         131         132
## 0.031176422 0.973293471 0.039254827 0.029714308 0.446989841 0.350500348
##         133         134         136         138         139         142
## 0.042556635 0.396458075 0.075857627 0.943521670 0.848263926 0.535100812
##         144         145         146         147         148         149
## 0.784492593 0.436646933 0.035936851 0.997484515 0.025090639 0.163278088
##         150         151         152         154         155         156
## 0.030844356 0.221840395 0.173310257 0.987294766 0.964168717 0.995944120
##         157         158         161         162         163         164
## 0.883971387 0.957242519 0.070160007 0.962358169 0.014783567 0.170683451
##         165         167         169         170         171         172
## 0.144895321 0.082456687 0.843827035 0.009115889 0.977089092 0.917952673
##         173         174         175         176         177         178
## 0.285029362 0.508278865 0.952969725 0.991627797 0.923978519 0.924476368
##         180         181         182         183         184         185
## 0.495215446 0.779415642 0.991836006 0.397286984 0.937664566 0.229869732
##         186         187         188         189         190         191
## 0.068419113 0.104397442 0.932144573 0.855338881 0.985861834 0.030793344
##         192         193         194         196         198         199
## 0.999576042 0.885904862 0.920455111 0.950010349 0.213573422 0.021365560
##         200         201         202         203         204         205
## 0.194357649 0.062848534 0.378871214 0.161148958 0.191544667 0.370313398
```

```
##         206         207         208         212         213         214
## 0.997011209 0.997192441 0.928936945 0.468950539 0.177557750 0.897972441
##         216         218         219         220         221         222
## 0.267925257 0.206027047 0.837876137 0.306189155 0.021407677 0.012692867
##         225         226         227         228         230         231
## 0.575008793 0.007060419 0.147064577 0.051325947 0.660391145 0.015266678
##         232         233         234         236         237         238
## 0.926193283 0.699291890 0.242369459 0.991451021 0.965202580 0.800777996
##         239         242         245         246         247         248
## 0.024486013 0.023688374 0.008970345 0.677093961 0.652919629 0.893531874
##         249         250         251         252         253         255
## 0.911009960 0.069563851 0.436844107 0.976312951 0.962542190 0.305678723
##         258         259         260         261         262         264
## 0.055447801 0.256337571 0.396199669 0.039651851 0.251860608 0.029410561
##         266         267         268         269         270         272
## 0.826689653          NA 0.239662253 0.536862128 0.035802182 0.775306859
##         273         274         275         276         277         278
## 0.996306134 0.072097367 0.196010793 0.470045397 0.131594254 0.014803177
##         279         280         281         282         283         284
## 0.365874303 0.088140037 0.985349279 0.037955396 0.874107961 0.030441743
##         286         287         288         290         291         292
## 0.994936662 0.896300369 0.525906480 0.062360354 0.559246095 0.061970114
##         296         297         298         301         303
## 0.018188850 0.943151542 0.586525321 0.921660128 0.052862006
```

```
fit.y = fit.p > 0.5
confusion.table = table(fit.y,train$y)
confusion.table
```

```
##
## fit.y     0   1
##   FALSE 109  18
##   TRUE   10  88
```

```
accuracy = sum(diag(confusion.table))/nrow(train)
accuracy
```

```
## [1] 0.8678414
```

ROC and AUC

The logistic regression had a good classification performance on training data.

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
fit.ROC = roc(train$y,fit.p)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
#plot(fit.ROC)
auc(fit.ROC)
```

```
## Area under the curve: 0.9301
```

d) Evaluate the model on testing data.

The logistic regression had a good classification performance on testing data.

True negatives: 38

False positives: 6

False negatives: 7

True positives: 25

AUC: 0.92 - the model is very good.

```
fit.p = predict(fit.logit, newdata = test, type = 'response')
fit.y = fit.p > 0.5
confusion.table = table(fit.y,test$y)
confusion.table
```

```
##
## fit.y    0  1
##   FALSE 38  7
##   TRUE   6 25
```

```
fit.ROC = roc(test$y,fit.p)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
auc(fit.ROC)
```

```
## Area under the curve: 0.9212
```

e) Use the feature selection algorithm to find optimal feature set. How the model perform on training and testing data?

Overall, the model demonstrates good performance on both training and testing data.

train data

```
train_clean <- na.omit(train)
fit.logit <- glm(y~., data = train_clean, family = binomial())
fit.stepwise <- step(fit.logit, direction = "both")
```

```
## Start:  AIC=186
## y ~ Age + Sex + ChestPain + RestBP + Chol + Fbs + RestECG + MaxHR +
##     ExAng + Oldpeak + Slope + Ca + Thal
##
##             Df Deviance    AIC
## - Age        1   152.01 184.01
## - Slope      1   152.13 184.13
## - Fbs        1   152.91 184.91
## - ExAng      1   153.01 185.01
## - RestECG    1   153.97 185.97
## <none>           152.00 186.00
## - MaxHR      1   154.95 186.95
## - Chol       1   155.18 187.18
## - RestBP     1   155.79 187.79
## - Oldpeak    1   155.94 187.94
## - Sex        1   159.55 191.55
## - ChestPain  3   165.28 193.28
## - Thal       2   165.84 195.84
## - Ca         1   168.64 200.64
##
## Step:  AIC=184.01
## y ~ Sex + ChestPain + RestBP + Chol + Fbs + RestECG + MaxHR +
##     ExAng + Oldpeak + Slope + Ca + Thal
##
##             Df Deviance    AIC
## - Slope      1   152.14 182.14
## - Fbs        1   152.91 182.91
## - ExAng      1   153.02 183.02
## - RestECG    1   153.97 183.97
## <none>           152.01 184.01
## - Chol       1   155.21 185.21
## - MaxHR      1   155.42 185.42
## - RestBP     1   155.97 185.97
## - Oldpeak    1   155.99 185.99
## + Age        1   152.00 186.00
## - Sex        1   159.88 189.88
## - ChestPain  3   165.97 191.97
## - Thal       2   165.90 193.90
## - Ca         1   169.49 199.49
##
## Step:  AIC=182.14
## y ~ Sex + ChestPain + RestBP + Chol + Fbs + RestECG + MaxHR +
##     ExAng + Oldpeak + Ca + Thal
##
##             Df Deviance    AIC
## - Fbs        1   153.01 181.01
## - ExAng      1   153.17 181.17
## <none>           152.14 182.14
## - RestECG    1   154.24 182.24
## - Chol       1   155.34 183.34
## + Slope      1   152.01 184.01
## - RestBP     1   156.02 184.02
```

```
## - MaxHR      1   156.02 184.02
## + Age        1   152.13 184.13
## - Oldpeak    1   157.66 185.66
## - Sex        1   159.89 187.89
## - ChestPain  3   165.97 189.97
## - Thal       2   166.81 192.81
## - Ca         1   169.59 197.59
##
## Step:  AIC=181.01
## y ~ Sex + ChestPain + RestBP + Chol + RestECG + MaxHR + ExAng +
##     Oldpeak + Ca + Thal
##
##             Df Deviance    AIC
## - ExAng      1   154.04 180.04
## <none>           153.01 181.01
## - RestECG    1   155.22 181.22
## - Chol       1   156.07 182.07
## + Fbs        1   152.14 182.14
## - RestBP     1   156.42 182.42
## + Slope      1   152.91 182.91
## + Age        1   153.00 183.00
## - MaxHR      1   157.20 183.20
## - Oldpeak    1   158.55 184.55
## - Sex        1   160.78 186.78
## - ChestPain  3   168.71 190.71
## - Thal       2   168.11 192.11
## - Ca         1   169.61 195.61
##
## Step:  AIC=180.04
## y ~ Sex + ChestPain + RestBP + Chol + RestECG + MaxHR + Oldpeak +
##     Ca + Thal
##
##             Df Deviance    AIC
## <none>           154.04 180.04
## - RestECG    1   156.07 180.07
## + ExAng      1   153.01 181.01
## + Fbs        1   153.17 181.17
## - Chol       1   157.38 181.38
## - RestBP     1   157.47 181.47
## + Slope      1   153.92 181.92
## + Age        1   154.03 182.03
## - MaxHR      1   159.25 183.25
## - Oldpeak    1   160.20 184.20
## - Sex        1   161.71 185.71
## - Thal       2   171.08 193.08
## - ChestPain  3   173.51 193.51
## - Ca         1   170.76 194.76
```

```
fit.p_train <- predict(fit.stepwise, newdata = train, type = 'response')
fit.y_train <- fit.p_train > 0.5
confusion.table_train <- table(fit.y_train, train$y)
accuracy_train <- sum(diag(confusion.table_train)) / sum(confusion.table_train)
fit.ROC_train <- roc(train$y, fit.p_train)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
auc_train <- auc(fit.ROC_train)
auc_train
```

```
## Area under the curve: 0.9287
```

```
accuracy_train
```

```
## [1] 0.8711111
```

test data

```
fit.logit <- glm(y~., data = test, family = binomial())
fit.stepwise <- step(fit.logit, direction = "both")
```

```
## Start:  AIC=61.11
## y ~ Age + Sex + ChestPain + RestBP + Chol + Fbs + RestECG + MaxHR +
##     ExAng + Oldpeak + Slope + Ca + Thal
##
##             Df Deviance    AIC
## - ChestPain  3   29.445 57.445
## - RestECG    1   27.106 59.106
## - RestBP     1   27.114 59.114
## - MaxHR      1   27.277 59.277
## - Chol       1   27.492 59.492
## - Age        1   27.571 59.571
## - Thal       2   29.675 59.675
## - Sex        1   28.117 60.117
## - Oldpeak    1   28.363 60.363
## - Fbs        1   28.723 60.723
## <none>           27.106 61.106
## - ExAng      1   32.121 64.121
## - Slope      1   37.083 69.083
## - Ca         1   40.022 72.022
##
## Step:  AIC=57.44
## y ~ Age + Sex + RestBP + Chol + Fbs + RestECG + MaxHR + ExAng +
##     Oldpeak + Slope + Ca + Thal
##
##             Df Deviance    AIC
## - RestBP     1   29.488 55.488
## - Chol       1   29.506 55.506
## - RestECG    1   29.547 55.547
## - Age        1   30.132 56.132
## - MaxHR      1   30.902 56.902
## - Sex        1   31.146 57.146
## <none>           29.445 57.445
## - Thal       2   34.342 58.342
## - Oldpeak    1   32.554 58.554
## - Fbs        1   34.967 60.967
## + ChestPain  3   27.106 61.106
## - ExAng      1   41.040 67.040
## - Slope      1   43.712 69.712
## - Ca         1   49.791 75.791
##
## Step:  AIC=55.49
## y ~ Age + Sex + Chol + Fbs + RestECG + MaxHR + ExAng + Oldpeak +
##     Slope + Ca + Thal
##
##             Df Deviance    AIC
## - Chol       1   29.538 53.538
## - RestECG    1   29.567 53.567
## - Age        1   30.162 54.162
## - MaxHR      1   30.969 54.969
## - Sex        1   31.191 55.191
## <none>           29.488 55.488
## - Thal       2   34.397 56.397
```

```
## - Oldpeak    1   32.594 56.594
## + RestBP     1   29.445 57.445
## - Fbs        1   35.084 59.084
## + ChestPain  3   27.114 59.114
## - ExAng      1   41.098 65.098
## - Slope      1   43.722 67.722
## - Ca         1   49.929 73.929
##
## Step:  AIC=53.54
## y ~ Age + Sex + Fbs + RestECG + MaxHR + ExAng + Oldpeak + Slope +
##     Ca + Thal
##
##             Df Deviance    AIC
## - RestECG    1   29.674 51.674
## - Age        1   30.162 52.162
## - MaxHR      1   30.977 52.977
## - Sex        1   31.305 53.305
## <none>           29.538 53.538
## - Thal       2   34.397 54.397
## - Oldpeak    1   32.866 54.866
## + Chol       1   29.488 55.488
## + RestBP     1   29.506 55.506
## - Fbs        1   35.121 57.121
## + ChestPain  3   27.548 57.548
## - ExAng      1   41.994 63.994
## - Slope      1   44.329 66.329
## - Ca         1   50.530 72.530
##
## Step:  AIC=51.67
## y ~ Age + Sex + Fbs + MaxHR + ExAng + Oldpeak + Slope + Ca +
##     Thal
##
##             Df Deviance    AIC
## - Age        1   30.215 50.215
## - MaxHR      1   31.246 51.246
## - Sex        1   31.523 51.523
## <none>           29.674 51.674
## - Thal       2   34.458 52.458
## - Oldpeak    1   32.922 52.922
## + RestECG    1   29.538 53.538
## + Chol       1   29.567 53.567
## + RestBP     1   29.667 53.667
## - Fbs        1   35.524 55.524
## + ChestPain  3   27.630 55.630
## - ExAng      1   42.213 62.213
## - Slope      1   44.376 64.376
## - Ca         1   50.601 70.601
##
## Step:  AIC=50.21
## y ~ Sex + Fbs + MaxHR + ExAng + Oldpeak + Slope + Ca + Thal
##
##             Df Deviance    AIC
```

```
## - MaxHR      1   31.507 49.507
## <none>           30.215 50.215
## - Thal       2   34.511 50.511
## - Sex        1   32.748 50.748
## - Oldpeak    1   33.030 51.030
## + Age        1   29.674 51.674
## + RestECG    1   30.162 52.162
## + RestBP     1   30.169 52.169
## + Chol       1   30.211 52.211
## + ChestPain  3   28.108 54.108
## - Fbs        1   36.275 54.275
## - ExAng      1   42.565 60.565
## - Slope      1   44.527 62.527
## - Ca         1   51.578 69.578
##
## Step:  AIC=49.51
## y ~ Sex + Fbs + ExAng + Oldpeak + Slope + Ca + Thal
##
##             Df Deviance    AIC
## <none>           31.507 49.507
## - Sex        1   33.918 49.918
## - Thal       2   36.178 50.178
## + MaxHR      1   30.215 50.215
## - Oldpeak    1   34.293 50.293
## + RestBP     1   31.231 51.231
## + Age        1   31.246 51.246
## + RestECG    1   31.352 51.352
## + Chol       1   31.496 51.496
## + ChestPain  3   28.252 52.252
## - Fbs        1   36.917 52.917
## - ExAng      1   45.389 61.389
## - Slope      1   47.080 63.080
## - Ca         1   54.432 70.432
```

```
fit.p_test <- predict(fit.stepwise, newdata = test, type = 'response')
fit.y_test <- fit.p_test > 0.5
confusion.table_test <- table(fit.y_test, test$y)

accuracy_test <- sum(diag(confusion.table_test)) / sum(confusion.table_test)
fit.ROC_test <- roc(test$y, fit.p_test)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
auc_test <- auc(fit.ROC_test)
accuracy_test
```

```
## [1] 0.9210526
```

```
auc_test
```

```
## Area under the curve: 0.9716
```

f) Compare the models with and without feature selection, what you observe?

The AUC of the model with feature selection (0.9716) is higher than that of the model without feature selection (0.9287), and a higher AUC indicates better overall performance of the model.

Q6 CART model

a) Please fit a CART model on training data to predict AHD. What is the tree size? Which predictors are used?

Tree size: number of terminal nodes is 21.

Predictors used: Ca, ExAng, MaxHR, Sex, Chol, ResECG, Age, RestBP and OldPeak.

```
library(tree)
train$y <- as.factor(train$y)
summary(train)
```

```
##       Age             Sex         ChestPain            RestBP
##  Min.   :29.00   Min.   :0.000   Length:227         Min.   : 94
##  1st Qu.:48.00   1st Qu.:0.000   Class :character   1st Qu.:120
##  Median :55.00   Median :1.000   Mode  :character   Median :130
##  Mean   :54.44   Mean   :0.696                      Mean   :132
##  3rd Qu.:60.50   3rd Qu.:1.000                      3rd Qu.:140
##  Max.   :77.00   Max.   :1.000                      Max.   :192
##       Chol            Fbs           RestECG          MaxHR
##  Min.   :126.0   Min.   :0.000   Min.   :0.000   Min.   : 71.0
##  1st Qu.:212.5   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:131.0
##  Median :244.0   Median :0.000   Median :2.000   Median :152.0
##  Mean   :248.5   Mean   :0.141   Mean   :1.031   Mean   :148.7
##  3rd Qu.:276.5   3rd Qu.:0.000   3rd Qu.:2.000   3rd Qu.:167.5
##  Max.   :417.0   Max.   :1.000   Max.   :2.000   Max.   :202.0
##      ExAng          Oldpeak         Slope             Ca
##  Min.   :0.000   Min.   :0.00   Min.   :1.000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000
##  Median :0.000   Median :0.80   Median :2.000   Median :0.0000
##  Mean   :0.326   Mean   :1.01   Mean   :1.595   Mean   :0.6859
##  3rd Qu.:1.000   3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :1.000   Max.   :6.20   Max.   :3.000   Max.   :3.0000
##      Thal           y
##  Length:227         0:120
##  Class :character   1:107
##  Mode  :character
##
##
##
```

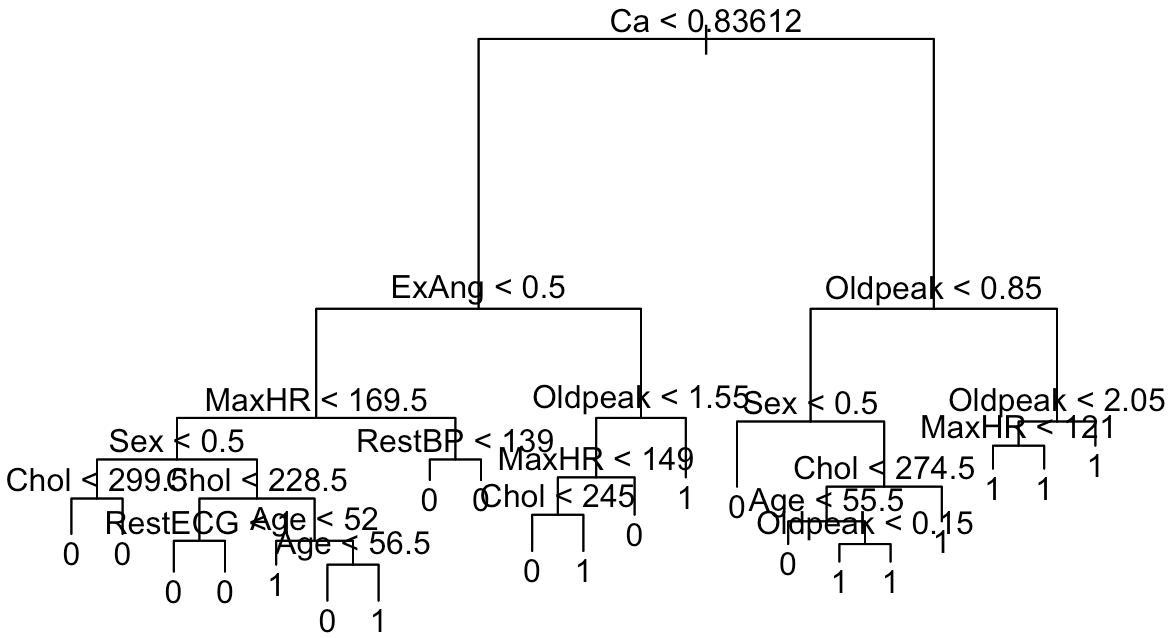
```
fit.CART <- tree(y~.,train)
```

```
## Warning in tree(y ~ ., train): NAs introduced by coercion
```

```
summary(fit.CART)
```

```
##
## Classification tree:
## tree(formula = y ~ ., data = train)
## Variables actually used in tree construction:
## [1] "Ca"      "ExAng"   "MaxHR"   "Sex"     "Chol"    "RestECG" "Age"
## [8] "RestBP"  "Oldpeak"
## Number of terminal nodes:  21
## Residual mean deviance:  0.5344 = 109 / 204
## Misclassification error rate: 0.1111 = 25 / 225
```

```
plot(fit.CART)
text(fit.CART)
```



b) Plot the tree.

```
library(tree)
train$y <- as.factor(train$y)
summary(train)
```

```
##       Age             Sex         ChestPain            RestBP
##  Min.   :29.00   Min.   :0.000   Length:227         Min.   : 94
##  1st Qu.:48.00   1st Qu.:0.000   Class :character   1st Qu.:120
##  Median :55.00   Median :1.000   Mode  :character   Median :130
##  Mean   :54.44   Mean   :0.696                      Mean   :132
##  3rd Qu.:60.50   3rd Qu.:1.000                      3rd Qu.:140
##  Max.   :77.00   Max.   :1.000                      Max.   :192
##       Chol            Fbs           RestECG          MaxHR
##  Min.   :126.0   Min.   :0.000   Min.   :0.000   Min.   : 71.0
##  1st Qu.:212.5   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:131.0
##  Median :244.0   Median :0.000   Median :2.000   Median :152.0
##  Mean   :248.5   Mean   :0.141   Mean   :1.031   Mean   :148.7
##  3rd Qu.:276.5   3rd Qu.:0.000   3rd Qu.:2.000   3rd Qu.:167.5
##  Max.   :417.0   Max.   :1.000   Max.   :2.000   Max.   :202.0
##      ExAng          Oldpeak         Slope             Ca
##  Min.   :0.000   Min.   :0.00   Min.   :1.000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000
##  Median :0.000   Median :0.80   Median :2.000   Median :0.0000
##  Mean   :0.326   Mean   :1.01   Mean   :1.595   Mean   :0.6859
##  3rd Qu.:1.000   3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :1.000   Max.   :6.20   Max.   :3.000   Max.   :3.0000
##      Thal           y
##  Length:227         0:120
##  Class :character   1:107
##  Mode  :character
##
##
##
```

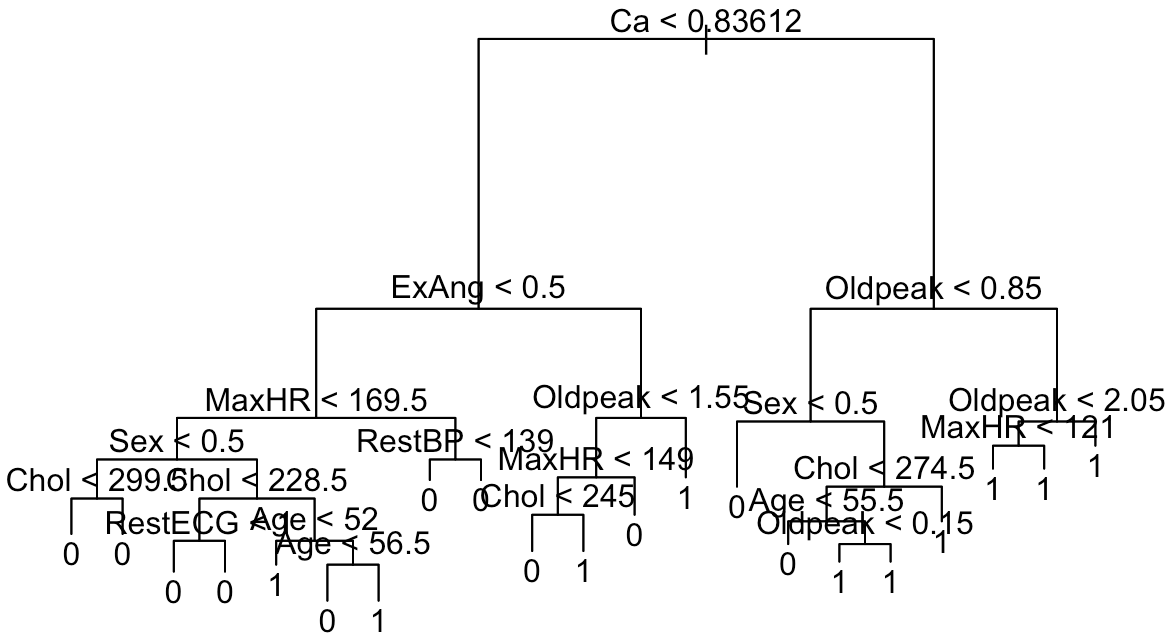
```
fit.CART <- tree (y~., train)
```

```
## Warning in tree(y ~ ., train): NAs introduced by coercion
```

```
summary(fit.CART)
```

```
##
## Classification tree:
## tree(formula = y ~ ., data = train)
## Variables actually used in tree construction:
## [1] "Ca"      "ExAng"   "MaxHR"   "Sex"     "Chol"    "RestECG" "Age"
## [8] "RestBP"  "Oldpeak"
## Number of terminal nodes:  21
## Residual mean deviance:  0.5344 = 109 / 204
## Misclassification error rate: 0.1111 = 25 / 225
```

```
plot(fit.CART)
text(fit.CART)
```



c) Evaluate the prediction accuracy on training and testing data.

The model shows good performance on both the training and testing datasets, though there's a slight decrease in accuracy and AUC from training to testing data. ##### train data

```
fit.p.CART <- predict(fit.CART,newdata = train)
```

```
## Warning in pred1.tree(object, tree.matrix(newdata)): NAs introduced by coercion
```

```
head(fit.p.CART)
```

```
##       0     1
## 1 0.375 0.625
## 2 0.000 1.000
## 3 0.000 1.000
## 5 1.000 0.000
## 6 1.000 0.000
## 8 0.900 0.100
```

```
fit.y.CART <- fit.p.CART[,2] > 0.5
head(fit.y.CART)
```

```
##     1     2     3     5     6     8
##  TRUE  TRUE  TRUE FALSE FALSE FALSE
```

```
confusion.table.CART <- table(fit.y.CART,train$y)
confusion.table.CART
```

```
##
## fit.y.CART   0   1
##      FALSE 109  15
##      TRUE   11  92
```

```
sum(diag(confusion.table.CART))/nrow(train) ##accuracy
```

```
## [1] 0.8854626
```

```
fit.ROC.CART <- roc(train$y,fit.p.CART[,2])
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
auc(fit.ROC.CART)
```

```
## Area under the curve: 0.9579
```

test data

```
fit.p.CART <- predict(fit.CART,newdata = test)
```

```
## Warning in pred1.tree(object, tree.matrix(newdata)): NAs introduced by coercion
```

```
head(fit.p.CART)
```

```
##             0         1
## 179 0.2105263 0.7894737
## 14  1.0000000 0.0000000
## 195 1.0000000 0.0000000
## 118 1.0000000 0.0000000
## 299 0.1428571 0.8571429
## 229 0.6363636 0.3636364
```

```
fit.y.CART <- fit.p.CART[,2] > 0.5
head(fit.y.CART)
```

```
##   179    14   195   118   299   229
##  TRUE FALSE FALSE FALSE  TRUE FALSE
```

```
confusion.table.CART <- table(fit.y.CART,test$y)
confusion.table.CART
```

```
##
## fit.y.CART  0  1
##      FALSE 34  7
##      TRUE  10 25
```

```
sum(diag(confusion.table.CART))/nrow(test) ##accuracy
```

```
## [1] 0.7763158
```

```
fit.ROC.CART <- roc(test$y,fit.p.CART[,2])
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
auc(fit.ROC.CART)
```

```
## Area under the curve: 0.8626
```

d) Please prune the tree using cv.tree function.

```
set.seed(1)
cv.Heart <- cv.tree(fit.CART, FUN = prune.misclass)
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

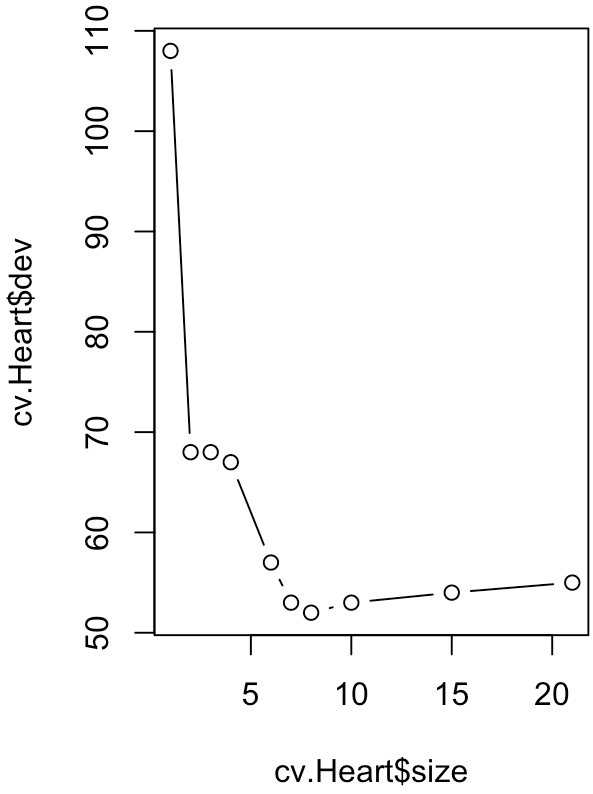
```
names(cv.Heart)
```

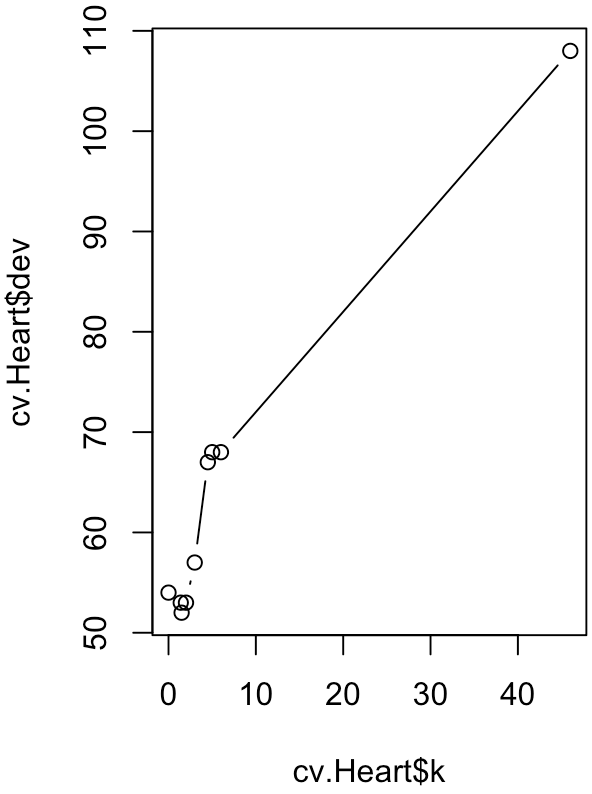
```
## [1] "size"   "dev"    "k"      "method"
```

```
cv.Heart
```

```
## $size
##  [1] 21 15 10  8  7  6  4  3  2  1
##
## $dev
##  [1]  55  54  53  52  53  57  67  68  68 108
##
## $k
##  [1] -Inf  0.0  1.4  1.5  2.0  3.0  4.5  5.0  6.0 46.0
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

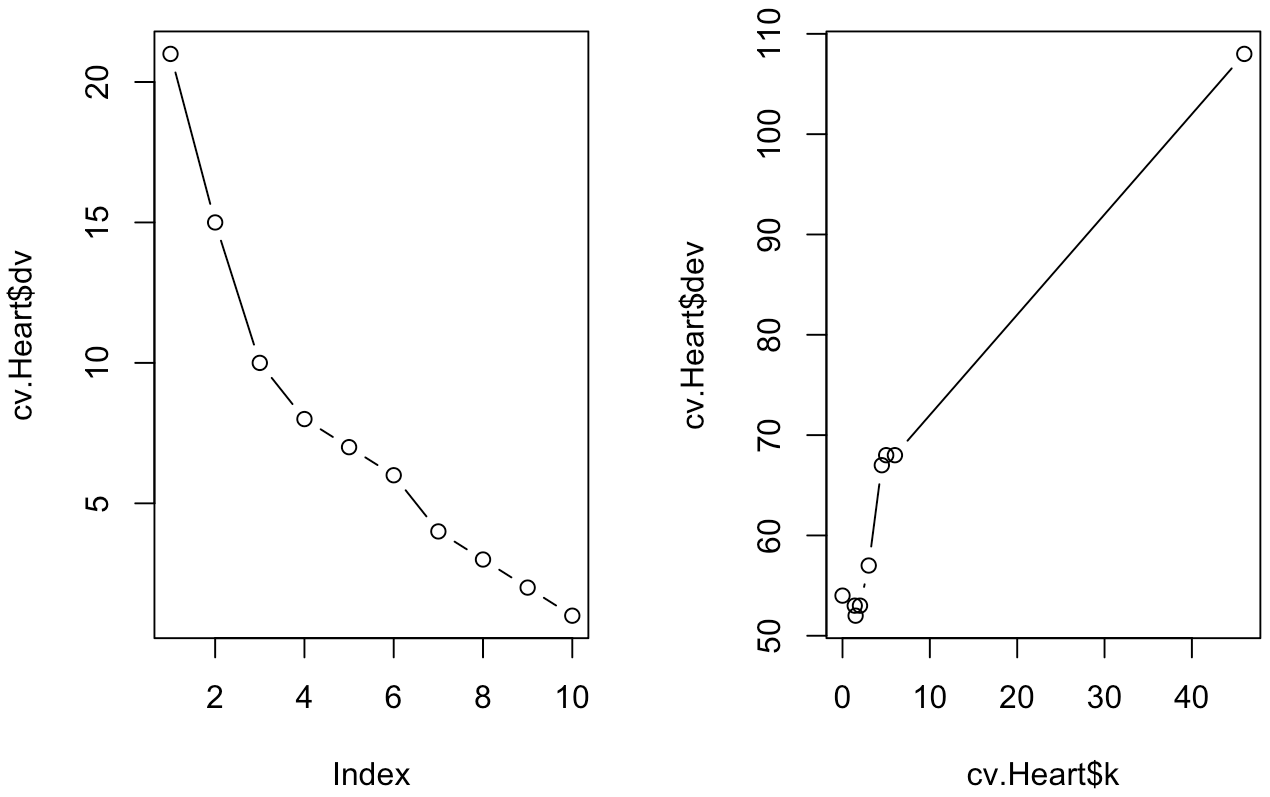
```
par(mfrow =c(1,2))
plot(cv.Heart$size ,cv.Heart$dev ,type="b")
plot(cv.Heart$k ,cv.Heart$dev ,type="b")
```





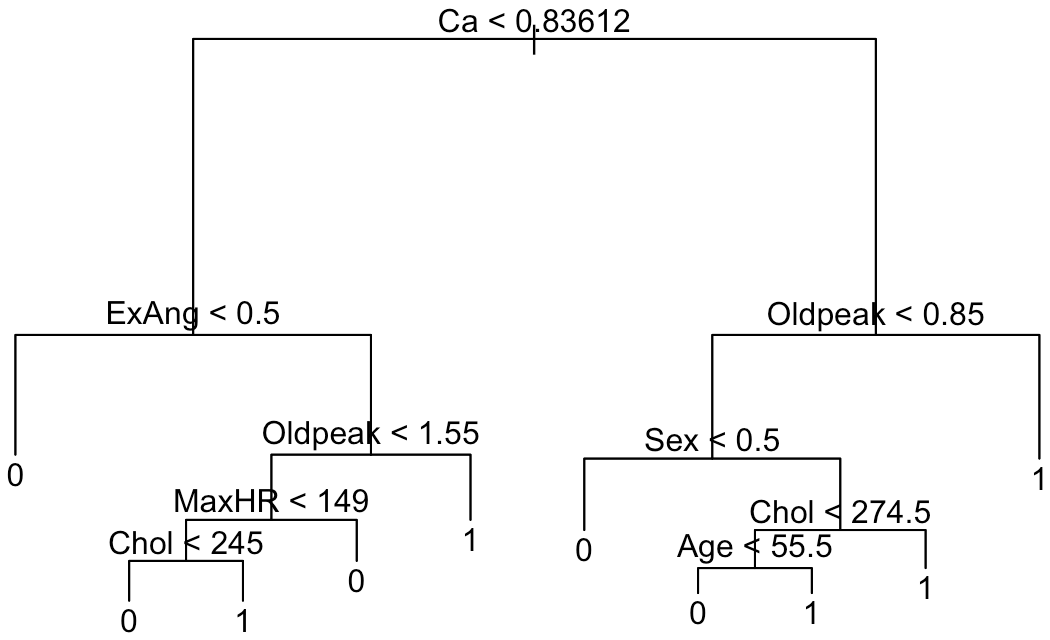
e) Please plot the CV prediction error as a function of both tree size and k.

```
par(mfrow =c(1,2))
plot(cv.Heart$size, cv.Heart$dv, type="b")
plot(cv.Heart$k, cv.Heart$dev, type = "b")
```



f) Please find the best tree size and use prune.misclass function to obtain the best tree.

```
prune.Heart=prune.misclass(fit.CART,best =9)
plot(prune.Heart)
text(prune.Heart,pretty =0)
```



g) Evaluate the prediction accuracy of pruned tree on training and testing data.

Both values are similar, with the test accuracy being lower than the train accuracy, which suggests that the pruned tree is a good fit for the data.

```
train_pred <- predict(prune.Heart, newdata = train, type = "class")
```

```
## Warning in pred1.tree(object, tree.matrix(newdata)): NAs introduced by coercion
```

```
train_accuracy <- mean(train_pred == train$y)
train_accuracy
```

```
## [1] 0.8546256
```

```
test_pred <- predict(prune.Heart, newdata = test, type = "class")
```

```
## Warning in pred1.tree(object, tree.matrix(newdata)): NAs introduced by coercion
```

```
test_accuracy <- mean(test_pred == test$y)
test_accuracy
```

```
## [1] 0.8157895
```

h) Compare the models with and without tree pruning, what you observe?

Train: in the original CART model (without tree pruning), the accuracy on the training data was 88.5% with an AUC of 0.95. With the tree pruning, the accuracy was slighter lower (85.4%), which reduces the complexity of the model.

Test: in the original CART model (without tree pruning), the accuracy on the testing data was 77.63% with an AUC of 0.86. With the tree pruning, the accuracy was slightly higher (81.5%), which helps with reducing overfitting.

Q7 Ensemble tree.

a) Please fit a bagging model using the 'randomForest' package. How many trees are fitted?

500 trees are fitted.

```
library(randomForest)
```

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

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
num_predictors <- ncol(train) - 1
bagging_model <- randomForest(y ~ ., data = train, mtry = num_predictors, importance = T
RUE, na.action = na.exclude)
print(bagging_model)
```

```
##
## Call:
##  randomForest(formula = y ~ ., data = train, mtry = num_predictors,      importance =
TRUE, na.action = na.exclude)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 13
##
##         OOB estimate of  error rate: 21.33%
## Confusion matrix:
##    0  1 class.error
## 0 99 20   0.1680672
## 1 28 78   0.2641509
```

b) Please fit a bagging model with 50 trees.

```
library(randomForest)
num_predictors <- ncol(train) - 1
bagging_model <- randomForest(y ~ ., data = train, mtry = num_predictors, ntree = 50, im
portance = TRUE, na.action = na.exclude)
print(bagging_model)
```

```
##
## Call:
##  randomForest(formula = y ~ ., data = train, mtry = num_predictors,      ntree = 50,
importance = TRUE, na.action = na.exclude)
##                Type of random forest: classification
##                      Number of trees: 50
## No. of variables tried at each split: 13
##
##         OOB estimate of  error rate: 22.67%
## Confusion matrix:
##    0  1 class.error
## 0 97 22   0.1848739
## 1 29 77   0.2735849
```

c) Please fit a bagging model with maximum tree size of 4.

```
library(randomForest)
num_predictors <- ncol(train) - 1
bagging_model <- randomForest(y ~ ., data = train, mtry = num_predictors, maxnodes = 16,
importance = TRUE, na.action = na.exclude)
print(bagging_model)
```

```
##
## Call:
##  randomForest(formula = y ~ ., data = train, mtry = num_predictors,      maxnodes = 1
6, importance = TRUE, na.action = na.exclude)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 13
##
##         OOB estimate of  error rate: 19.11%
## Confusion matrix:
##     0  1 class.error
## 0 103 16   0.1344538
## 1  27 79   0.2547170
```

d) Which model give the best performance?

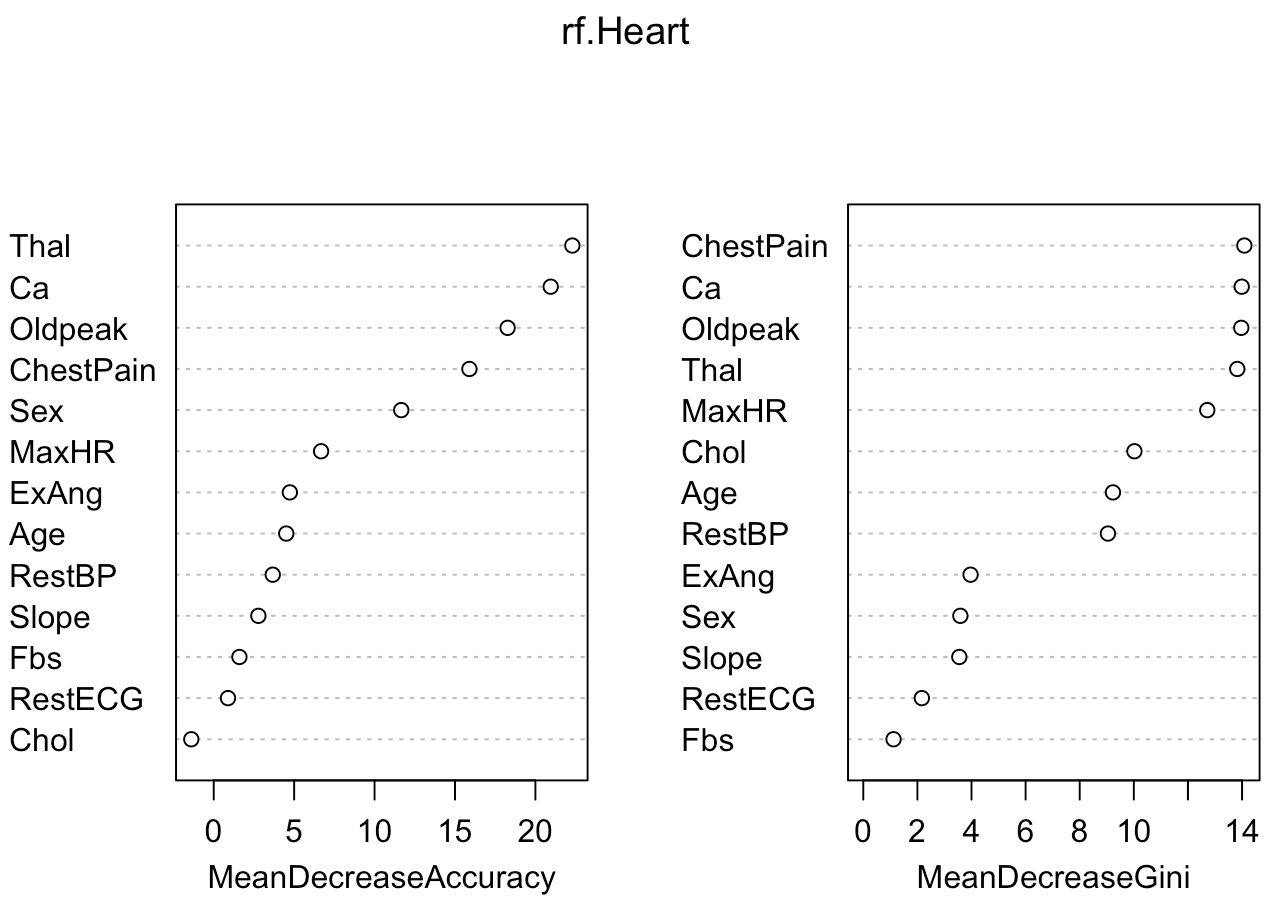
The model with a maximum of 4 trees has the lowest OOB error rate and lowest class error rates for both classes.

e) Please fit a random forest model with 4 features are considered in each split. Compare the performance of random forest and bagging tree. What you observe?

```
library(randomForest)
rf.Heart =randomForest(y~.,data=train,mtry=4,importance =TRUE,na.action = na.exclude)
rf.Heart
```

```
##
## Call:
##  randomForest(formula = y ~ ., data = train, mtry = 4, importance = TRUE,      na.act
ion = na.exclude)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 21.33%
## Confusion matrix:
##    0  1 class.error
## 0 99 20   0.1680672
## 1 28 78   0.2641509
```

```
varImpPlot(rf.Heart)
```



f) Please visualize the importance of variables in random forest.

The variables at the top of the top of the plot are the most important, because they have higher values in mean decrease accuracy and mean decrease in gini.

g) Please fit a boosting tree using the ‘gbm’ package with 500 trees. Compare the performance of boosting tree and random forest, which one is better?

Could not figure it out how to do it.

h) Please fit a rulefit model using the ‘pre’ package.

Could not figure it out how to do it.