Project

Fatemeh Nosrat

5/10/2021

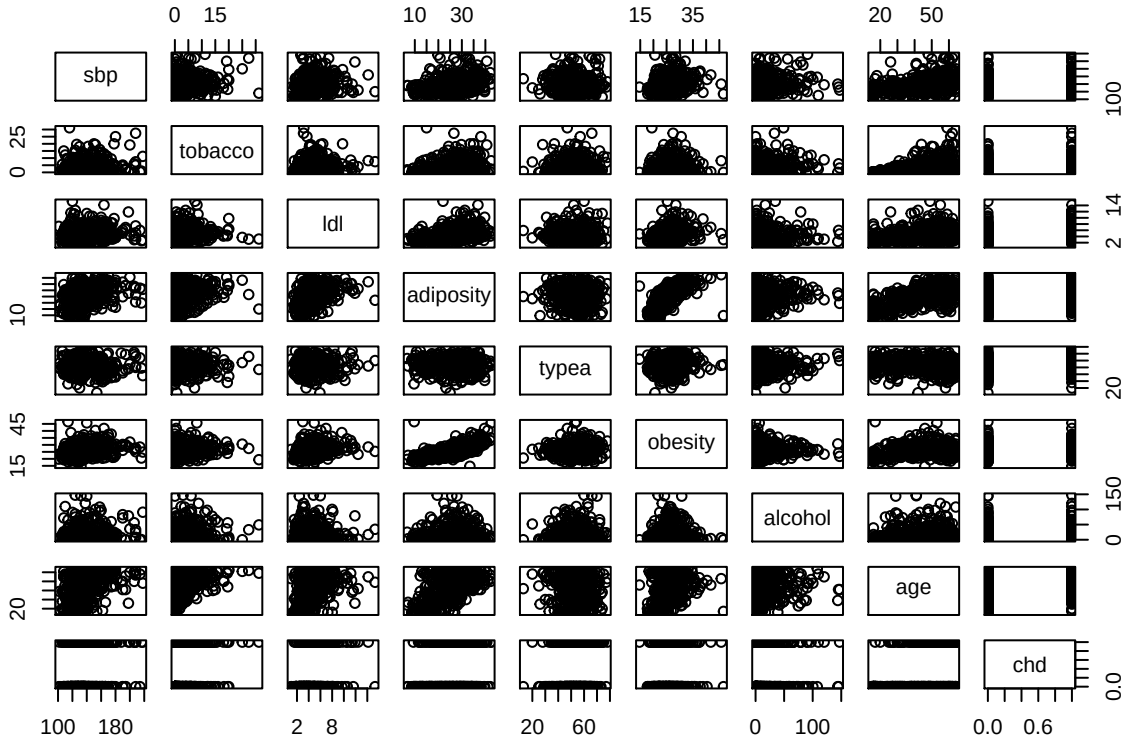
```
library(ISLR)
SA.Heart = read.table("http://www-stat.stanford.edu/~tibs/ElemStatLearn/datasets/SAheart.data",
                      sep=",", head=T, row.names=1)
names(SA.Heart)
```

```
##  [1] "sbp"       "tobacco"   "ldl"       "adiposity" "famhist"   "typea"
##  [7] "obesity"   "alcohol"   "age"       "chd"
```

```
dim(SA.Heart)
```

```
## [1] 462  10
```

```
pairs(~ sbp + tobacco + ldl + adiposity  + typea + obesity + alcohol +
        age + chd , data = SA.Heart)
```




```
summary(SA.Heart)
```

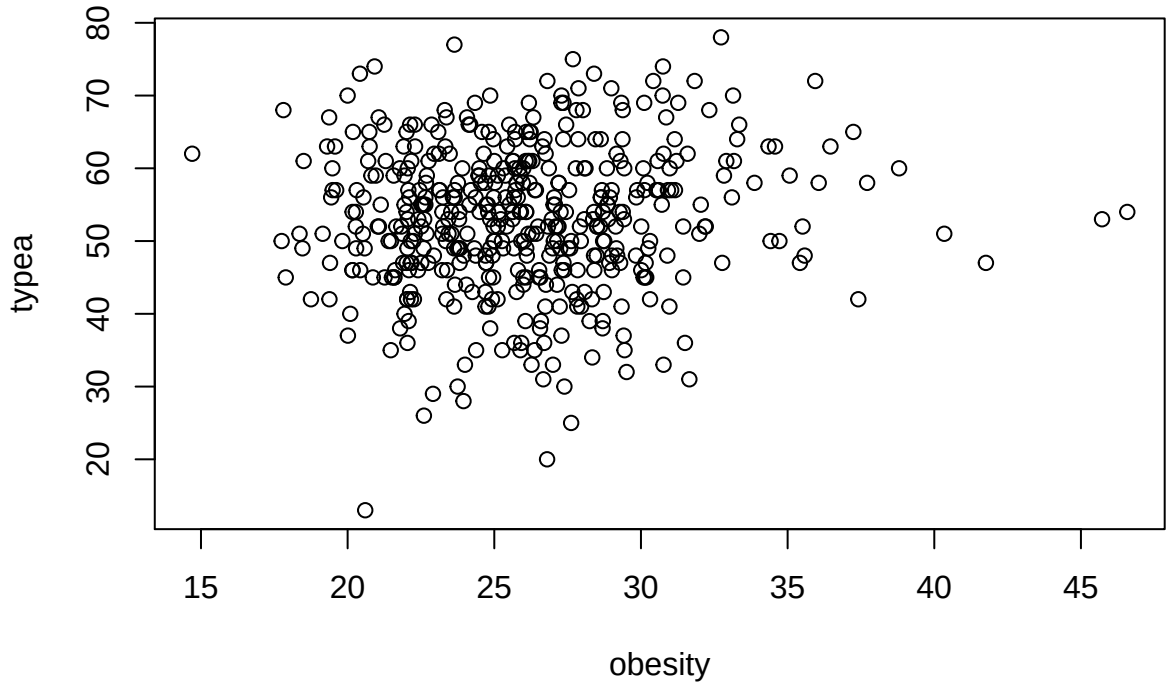
```
##       sbp             tobacco            ldl              adiposity
##  Min.   :101.0   Min.   : 0.0000   Min.   : 0.980   Min.   : 6.74
##  1st Qu.:124.0   1st Qu.: 0.0525   1st Qu.: 3.283   1st Qu.:19.77
##  Median :134.0   Median : 2.0000   Median : 4.340   Median :26.11
```

```
##  Mean   :138.3   Mean   : 3.6356   Mean   : 4.740   Mean   :25.41
##  3rd Qu.:148.0   3rd Qu.: 5.5000   3rd Qu.: 5.790   3rd Qu.:31.23
##  Max.   :218.0   Max.   :31.2000   Max.   :15.330   Max.   :42.49
##    famhist              typea          obesity          alcohol
##  Length:462         Min.   :13.0   Min.   :14.70   Min.   :  0.00
##  Class :character   1st Qu.:47.0   1st Qu.:22.98   1st Qu.:  0.51
##  Mode  :character   Median :53.0   Median :25.80   Median :  7.51
##                     Mean   :53.1   Mean   :26.04   Mean   : 17.04
##                     3rd Qu.:60.0   3rd Qu.:28.50   3rd Qu.: 23.89
##                     Max.   :78.0   Max.   :46.58   Max.   :147.19
##       age              chd
##  Min.   :15.00   Min.   :0.0000
##  1st Qu.:31.00   1st Qu.:0.0000
##  Median :45.00   Median :0.0000
##  Mean   :42.82   Mean   :0.3463
##  3rd Qu.:55.00   3rd Qu.:1.0000
##  Max.   :64.00   Max.   :1.0000
```

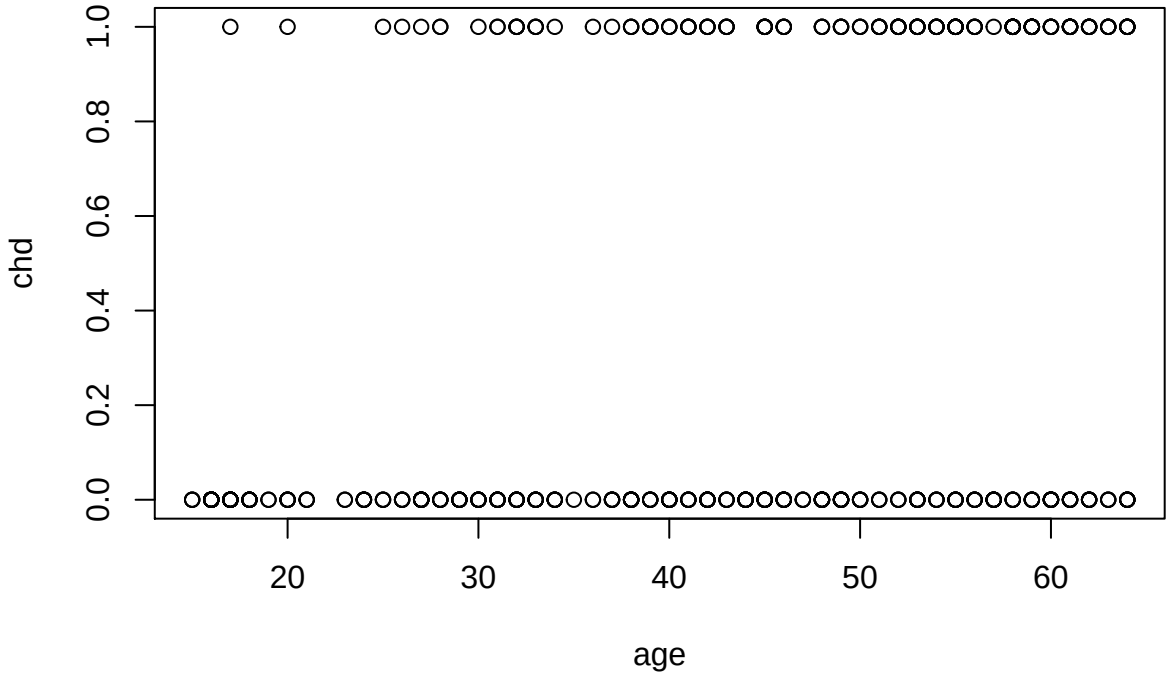
```r
# As I explained in the note the possible range for typea is between 12 and 84,
# and the summary confirms this range, since the min of typea is f13 and max is 78.
# convert famhist to a qualitative variable
SA.Heart$famhist = as.factor(SA.Heart$famhist)

# correlation
cor(SA.Heart[,-5])
```

```
##                    sbp     tobacco         ldl   adiposity       typea
## sbp         1.00000000  0.21224652  0.15829633  0.35650008 -0.05745431
## tobacco     0.21224652  1.00000000  0.15890546  0.28664037 -0.01460788
## ldl         0.15829633  0.15890546  1.00000000  0.44043175  0.04404758
## adiposity   0.35650008  0.28664037  0.44043175  1.00000000 -0.04314364
## typea      -0.05745431 -0.01460788  0.04404758 -0.04314364  1.00000000
## obesity     0.23806661  0.12452941  0.33050586  0.71655625  0.07400610
## alcohol     0.14009559  0.20081339 -0.03340340  0.10033013  0.03949794
## age         0.38877060  0.45033016  0.31179923  0.62595442 -0.10260632
## chd         0.19235411  0.29971754  0.26305268  0.25412139  0.10315583
##               obesity     alcohol        age        chd
## sbp        0.23806661  0.14009559  0.3887706 0.19235411
## tobacco    0.12452941  0.20081339  0.4503302 0.29971754
## ldl        0.33050586 -0.03340340  0.3117992 0.26305268
## adiposity  0.71655625  0.10033013  0.6259544 0.25412139
## typea      0.07400610  0.03949794 -0.1026063 0.10315583
## obesity    1.00000000  0.05161957  0.2917771 0.10009508
## alcohol    0.05161957  1.00000000  0.1011246 0.06253068
## age        0.29177713  0.10112465  1.0000000 0.37297334
## chd        0.10009508  0.06253068  0.3729733 1.00000000
```

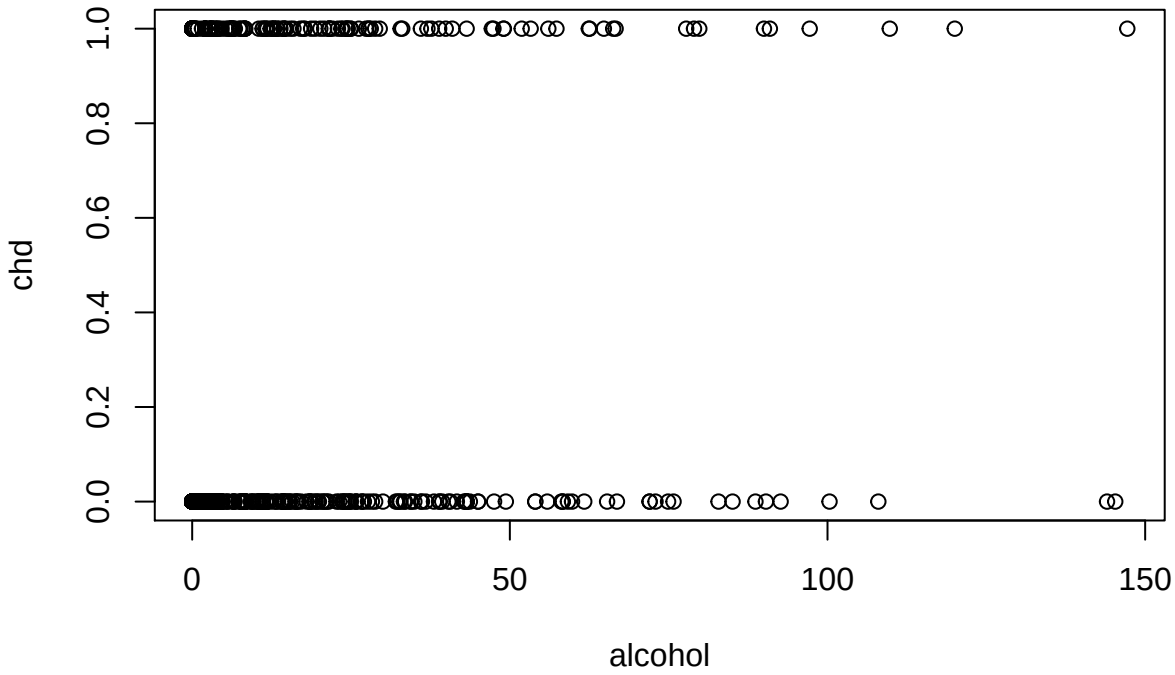
```r
# age has the highest correlation with chd and alcohol has the lowest correlation with chd
# obesity and typea has the highest correlation
attach(SA.Heart)
plot(obesity,typea)
```




```
plot(age,chd)
```




```
plot(alcohol,chd)
```




```
# Logistic Regression
# We will fit a logistic regression model in order to predict "chd" using "sbp", "tobacco",
# "ldl", "adiposity", "famhist", "typea", "obesity", "alcohol", and "age".

glm.fit = glm(chd ~ sbp + tobacco + ldl + adiposity + famhist + typea + obesity + alcohol +
                age, data = SA.Heart, family = binomial)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = chd ~ sbp + tobacco + ldl + adiposity + famhist +
##     typea + obesity + alcohol + age, family = binomial, data = SA.Heart)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7781  -0.8213  -0.4387   0.8889   2.5435
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -6.1507209  1.3082600  -4.701 2.58e-06 ***
## sbp             0.0065040  0.0057304   1.135 0.256374
## tobacco         0.0793764  0.0266028   2.984 0.002847 **
## ldl             0.1739239  0.0596617   2.915 0.003555 **
## adiposity       0.0185866  0.0292894   0.635 0.525700
## famhistPresent  0.9253704  0.2278940   4.061 4.90e-05 ***
## typea           0.0395950  0.0123202   3.214 0.001310 **
## obesity        -0.0629099  0.0442477  -1.422 0.155095
## alcohol         0.0001217  0.0044832   0.027 0.978350
## age             0.0452253  0.0121298   3.728 0.000193 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 596.11  on 461  degrees of freedom
## Residual deviance: 472.14  on 452  degrees of freedom
## AIC: 492.14
##
## Number of Fisher Scoring iterations: 5
```

```
# variables with statistically significant p-values are tobacco,ldl, famhistPresent, typea,
# and age.
# Also, famhistPresent has the smallest p-value.
coef(glm.fit)
```

```
##    (Intercept)            sbp        tobacco            ldl      adiposity
##  -6.1507208650   0.0065040171   0.0793764457   0.1739238981   0.0185865682
## famhistPresent          typea        obesity        alcohol            age
##   0.9253704194   0.0395950250  -0.0629098693   0.0001216624   0.0452253496
```

```
# the coefficient corresponding to famhistPresent is the biggest among other coefficients,
# which means that the present of family history of coronary heart disease have a high impact
# on having diagnosed with that disease.
# the coefficients corresponding to "sbp" and "alcohol" are positive but close to zero,
# which means they do not have a high effect on indicating whether an individual has the
# disease or not. The coefficient associated with obesity is negative, which means that
# if an individual put on more weight and got more obese, then t is less likely for that
# individual to diagnosed with coronary disease.

# consider the first 115 data as the training data and the rest as the test data
train = SA.Heart[1:347,]
dim(train)
```

```
## [1] 347  10
```

```
test = SA.Heart[348:462,]
dim(test)
```

```
## [1] 115  10
```

```
# know we fit the model with the training data
glm.fit = glm(chd ~ sbp + tobacco + ldl + adiposity + famhist + typea + obesity + alcohol +
                age, data = train, family = binomial)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = chd ~ sbp + tobacco + ldl + adiposity + famhist +
##     typea + obesity + alcohol + age, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8575  -0.8476  -0.4446   0.9479   2.4746
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -5.3828517  1.4899493  -3.613 0.000303 ***
## sbp            -0.0006045  0.0069083  -0.088 0.930271
## tobacco         0.0788708  0.0304316   2.592 0.009549 **
## ldl             0.1667246  0.0694720   2.400 0.016400 *
```

```
## adiposity       0.0178851  0.0340952   0.525 0.599886
## famhistPresent  0.8554214  0.2607152   3.281 0.001034 **
## typea           0.0409574  0.0140507   2.915 0.003557 **
## obesity        -0.0514798  0.0500117  -1.029 0.303313
## alcohol         0.0022175  0.0054141   0.410 0.682107
## age             0.0451540  0.0135781   3.326 0.000883 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 454.70  on 346  degrees of freedom
## Residual deviance: 365.23  on 337  degrees of freedom
## AIC: 385.23
##
## Number of Fisher Scoring iterations: 4
```

```
# using the first 347 observations as the training data caused many changes in the values
# of p-values. The only variables with statistically significant p-values are tobacco and
# typea. Interestingly, famhistPresent, which used to be a variable with the smallest p-value
# in the main  model, now does not even have a significant p-value.

prob = predict(glm.fit, test, type= "response")
pred = rep(0, 462-347)
pred[prob > 0.5] = 1
table(pred,test$chd)
```

```
##
## pred  0  1
##    0 66 11
##    1 15 23
```

```
mean(pred == test$chd)
```

```
## [1] 0.773913
```

```
# logistic regression correctly predicted the number of cases with coronary heart disease
# 77.39% of the time. That is surprisingly good and it shows that the predictors in the model
# are good indicators of having someone diagnosed with the disease or not.

mean(pred!=test$chd)
```

```
## [1] 0.226087
```

```
# the test error in this case is 28.24%

# Let refit the model with only those variable with significant p-values in the main model,
# that is a linear model with only tobacco,ldl, famhistPresent, typea, and age as the predictors.
glm.fit = glm(chd ~ tobacco + ldl + famhist + typea + age,
              data = train, family = binomial)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = chd ~ tobacco + ldl + famhist + typea + age, family = binomial,
##     data = train)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9314  -0.8449  -0.4502   0.9651   2.5334
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -6.29310    1.03496  -6.081 1.20e-09 ***
## tobacco         0.08181    0.03005   2.722  0.00649 **
## ldl             0.15459    0.06470   2.389  0.01688 *
## famhistPresent  0.84041    0.25868   3.249  0.00116 **
## typea           0.03990    0.01398   2.853  0.00433 **
## age             0.04711    0.01132   4.163 3.14e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 454.7  on 346  degrees of freedom
## Residual deviance: 366.7  on 341  degrees of freedom
## AIC: 378.7
##
## Number of Fisher Scoring iterations: 4
```

```
prob = predict(glm.fit, test, type = "response")
pred = rep(0, 462-347)
pred[prob > 0.5] = 1
table(pred, test$chd)
```

```
##
## pred  0  1
##    0 68 14
##    1 13 20
```

```
mean(pred == test$chd)
```

```
## [1] 0.7652174
```

```
# logistic regression correctly predicted the number of cases with coronary heart disease 76.52\%
# of the time.

# If we refit the main model with the variables which were significant in the model with all
# variables and  training data, then we have
glm.fit = glm(chd ~ tobacco + typea,
              data = train, family = binomial)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = chd ~ tobacco + typea, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9745  -0.8868  -0.7122   1.2214   1.9320
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -2.76181    0.70925  -3.894 9.86e-05 ***
## tobacco      0.14735    0.02910   5.064 4.10e-07 ***
## typea        0.03039    0.01260   2.411   0.0159 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 454.70  on 346  degrees of freedom
## Residual deviance: 417.62  on 344  degrees of freedom
## AIC: 423.62
##
## Number of Fisher Scoring iterations: 4
```

```
prob = predict(glm.fit, test, type = "response")
pred = rep(0, 462-347)
pred[prob > 0.5] = 1
table(pred, test$chd)
```

```
##
## pred  0  1
##    0 74 26
##    1  7  8
```

```
mean(pred == test$chd)
```

```
## [1] 0.7130435
```

```
# logistic regression correctly predicted the number of cases with coronary heart disease
# 71.3% of the time, which is worse than the other two model. So, we can conclude that
# the model with tobacco , ldl , famhist , typea , and age as the predictors and with
# that training data predicts the result in this case better.

# now we perform LDA on our data. we fit the model using the training data we have already
# obtained. Since we could obtain the least test error when we had tobacco,ldl, famhistPresent,
# typea, and age as the predictors, we again use only these variables as the predictors
library(MASS)
set.seed(1, sample.kind = "Rounding")
```

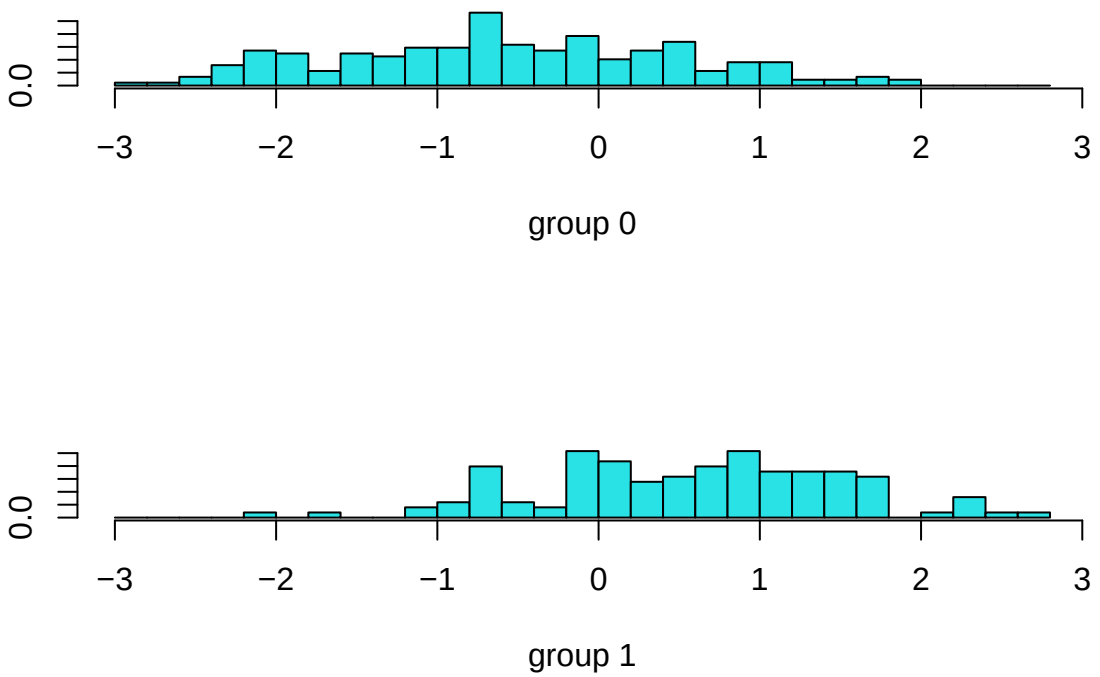
```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
lda.fit = lda(chd ~ sbp + tobacco + ldl + adiposity + famhist + typea + obesity + alcohol +
                age, data = train)
lda.fit
```

```
## Call:
## lda(chd ~ sbp + tobacco + ldl + adiposity + famhist + typea +
##     obesity + alcohol + age, data = train)
##
## Prior probabilities of groups:
##         0         1
## 0.6368876 0.3631124
##
## Group means:
##        sbp  tobacco      ldl adiposity famhistPresent    typea  obesity
## 0 134.9548 2.628959 4.378597  23.56597      0.3031674 52.63801 25.70670
```

```
## 1 140.2381 5.424365 5.478413  27.62667      0.5714286 55.37302 26.61968
##    alcohol      age
## 0 14.21950 38.66968
## 1 18.31159 49.84127
##
## Coefficients of linear discriminants:
##                           LD1
## sbp            -0.0002365733
## tobacco         0.0837262014
## ldl             0.1576033911
## adiposity       0.0076164496
## famhistPresent  0.8137435589
## typea           0.0323872936
## obesity        -0.0417462089
## alcohol         0.0010380326
## age             0.0369194340
```

```
# LDA output indicates that pi_{1}^{bar} = 0.6368876 and pi_{1}^{bar} = 0.3631124; in other word,
# 38% of the training observations correspond to those having diagnoised with the disease.
# and 62% of the training observations correspond to those not having diagnosed with the disease
# Based on the group mean provided by LDA, they suggest that the consumption of tobacco is higher
# when the observation is diagnosed with the disease and the mean of consumption of tobacco is
# lower when the observation is not diagnosed with the disease.
# this is true about all other predictors, that is, ldl, famhistPresent, typea, and age are all
# higher when the observation is diagnosed with the disease and lower otherwise.
plot(lda.fit)
```




```
lda.fit$coefs
```

```
## NULL
```

```
# Based on the coefficients of linear discriminants, we hace the following linear combination of variab
# 0.08774733 * tobacco + 0.10201265 * ldl + 0.30663925  * famhistPresent + 0.04760699 * typea +  0.0454
# If the value of this linear combination is high, then LDA classifier will predict that the individual
```

```
# that disease and if it is small, then the prediction is that the person is not diagnoised with the di
# prediction
lda.pred = predict(lda.fit, test)
 lda.pred$class
```

```
##   [1] 0 0 0 1 0 1 1 1 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 1 1 0 1 0 0 0 0 0 0 0 0 1 0
##  [38] 1 1 1 1 0 0 0 1 0 0 0 0 1 0 1 0 0 0 1 1 0 1 1 1 0 0 1 0 1 1 1 0 1 0 0 0 0
##  [75] 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0
## [112] 0 0 1 1
## Levels: 0 1
```

```
table( lda.pred$class, test$chd)
```

```
##
##      0  1
##   0 68 13
##   1 13 21
```

```
mean(lda.pred$class == test$chd)
```

```
## [1] 0.773913
```

```
# the LDA and logistic regression predictions are almost identical.

# We now use Quadratic Discriminant Analysis for obtaining the prediction on the data.
# Again, we use the model containing only tobacco,ldl, famhistPresent, typea, and age.
qda.fit = qda(chd ~ tobacco + ldl + famhist + typea + age,
              data = train)
qda.fit
```

```
## Call:
## qda(chd ~ tobacco + ldl + famhist + typea + age, data = train)
##
## Prior probabilities of groups:
##         0         1
## 0.6368876 0.3631124
##
## Group means:
##    tobacco      ldl famhistPresent    typea      age
## 0 2.628959 4.378597      0.3031674 52.63801 38.66968
## 1 5.424365 5.478413      0.5714286 55.37302 49.84127
```

```
#prediction
qda.pred = predict(qda.fit, test)
table(qda.pred$class, test$chd)
```

```
##
##      0  1
##   0 69 18
##   1 12 16
```

```
mean(qda.pred$class == test$chd)
```

```
## [1] 0.7391304
```

```
# QDA method has the almost same level of accuracy as the logistic regression and LDA methods.
```

```r
# KNN method
library(class)
# we now use KNN method for prediction
# we need to standardize the data so that all variables are given a mean of zero and a standard deviati
# we remove the column corresponding to famhist, because it is a qualitative variable.
# standardized variable
#stand.X = scale(SA.Heart[,-5])
# We fit a KNN model on the training data using K = 1
train.X = cbind(sbp, tobacco, ldl, adiposity, famhist, typea, obesity, alcohol, age)[1:347,-10]
test.X = cbind(sbp, tobacco, ldl, adiposity, famhist, typea, obesity, alcohol, age)[348:462,-10]
train.Y = SA.Heart$chd[1:347]
test.Y = SA.Heart$chd[348:462]
set.seed(1, sample.kind = "Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```r
knn.pred = knn(train.X, test.X, train.Y, k=1)
table(knn.pred, test.Y)
```

```
##         test.Y
## knn.pred  0  1
##        0 57 20
##        1 24 14
```

```r
mean(test.Y == knn.pred)
```

```
## [1] 0.6173913
```

```r
mean(test.Y != knn.pred)
```

```
## [1] 0.3826087
```

```r
mean(test.Y != "absent")
```

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

```r
# KKN with K=1 could predict the result correctly 66.28% of time, which is not a weak prediction compar
# methods
# Let consider K = 3
knn.pred = knn(train.X, test.X, train.Y, k=3)
table(knn.pred,test.Y)
```

```
##         test.Y
## knn.pred  0  1
##        0 61 18
##        1 20 16
```

```r
mean(test.Y == knn.pred)
```

```
## [1] 0.6695652
```

```r
mean(test.Y != knn.pred)
```

```
## [1] 0.3304348
```

```r
# this time the mean test error is less than when K = 1, which mean we have a better prediction with K

# Let consider K = 10
```

```
knn.pred = knn(train.X, test.X, train.Y, k=10)
table(knn.pred,test.Y)
```

```
##         test.Y
## knn.pred  0  1
##        0 64 15
##        1 17 19
```

```
mean(test.Y == knn.pred)
```

```
## [1] 0.7217391
```

```
mean(test.Y != knn.pred)
```

```
## [1] 0.2782609
```

```
# when K =10, we have a better prediction compared to KKN with lower K, but the test error for KNN with
# is still higher than those earlier method we use to fit our method.



# LOOCV
error = rep(0,462)
for (i in 1:462){
  l.fit = glm(chd ~ .,
              data = SA.Heart[-i,], family = "binomial")
  chd.pred = predict.glm(l.fit, SA.Heart[i,], type="response") > 0.5
  if(chd.pred != SA.Heart[i,]$chd)
    error[i] = 1

}

mean(error)
```

```
## [1] 0.2813853
```

```
# K-fold CV, K=10
set.seed(1, sample.kind = "Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
library(boot)
cv.error.10 = rep(0,10)
for (i in 1:10){
  glm.fit = glm(chd ~ .,
              data = SA.Heart, family = "binomial")
  cv.error.10[i] = cv.glm(SA.Heart, glm.fit, K=10)$delta[1]
}
cv.error.10
```

```
##  [1] 0.1795869 0.1802834 0.1783256 0.1800535 0.1804971 0.1809238 0.1809298
##  [8] 0.1813956 0.1803258 0.1840430
```

```
# bootstrap
# we have already considered the logistic regression when we have tobacco, ldl, famhist, typea,
# and age as the predictors and chd as the response
#we now compute the standard error of the income and balance logistic regression coefficients in two di
```

```r
# using bootstrap and using the standard formula for computing the standard error in the glm()
glm.fit = glm(chd ~ tobacco + ldl + famhist + typea + age,
              data = SA.Heart, family = "binomial")
summary(glm.fit)
```

```
##
## Call:
## glm(formula = chd ~ tobacco + ldl + famhist + typea + age, family = "binomial",
##     data = SA.Heart)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9165  -0.8054  -0.4430   0.9329   2.6139
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -6.44644    0.92087  -7.000 2.55e-12 ***
## tobacco         0.08038    0.02588   3.106  0.00190 **
## ldl             0.16199    0.05497   2.947  0.00321 **
## famhistPresent  0.90818    0.22576   4.023 5.75e-05 ***
## typea           0.03712    0.01217   3.051  0.00228 **
## age             0.05046    0.01021   4.944 7.65e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 596.11  on 461  degrees of freedom
## Residual deviance: 475.69  on 456  degrees of freedom
## AIC: 487.69
##
## Number of Fisher Scoring iterations: 5
```

```r
data("SA.Heart")
```

```
## Warning in data("SA.Heart"): data set 'SA.Heart' not found
```

```r
# boot.fn function which returns coefficients estimates for all considered predictors in the logistic r
boot.fn = function(data,index){
  l.fit =  glm(chd ~ .,
              data = SA.Heart, family = "binomial", subset = index)
  return(coef(l.fit))
}

boot(SA.Heart, boot.fn,1000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = SA.Heart, statistic = boot.fn, R = 1000)
##
##
## Bootstrap Statistics :
```

```
##           original        bias    std. error
## t1*  -6.1507208650 -0.1707868160 1.401744462
## t2*   0.0065040171  0.0003754193 0.006272245
## t3*   0.0793764457  0.0040679689 0.027143726
## t4*   0.1739238981  0.0071858451 0.064855486
## t5*   0.0185865682  0.0027323337 0.031264157
## t6*   0.9253704194  0.0191337765 0.226487594
## t7*   0.0395950250  0.0019499504 0.012470782
## t8*  -0.0629098693 -0.0064039140 0.051579661
## t9*   0.0001216624 -0.0001893718 0.004912838
## t10*  0.0452253496  0.0009075547 0.012449730
```

```
# model selection: so far based on the methods we used, we figured that the logistic regression
# and LDA models with all variables as predictors provide us best predictions for the response
# compared to other linear methods. Having all 9 variables in the model is not necessary and
# makes the model very complicated. In the next chapter, we will find what subset of variables
# are best to use as predictors in the linear model  Now we still consider a linear formulation
# of variables and try to obtain the best linear model (find a subset of predictors which gives
# us the best accuracy)

# best subset selection
library(leaps)
set.seed(1, sample.kind = "Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
regfit.f = regsubsets(chd ~ ., SA.Heart)
summary(regfit.f)
```

```
## Subset selection object
## Call: regsubsets.formula(chd ~ ., SA.Heart)
## 9 Variables  (and intercept)
##                Forced in Forced out
## sbp                FALSE      FALSE
## tobacco            FALSE      FALSE
## ldl                FALSE      FALSE
## adiposity          FALSE      FALSE
## famhistPresent     FALSE      FALSE
## typea              FALSE      FALSE
## obesity            FALSE      FALSE
## alcohol            FALSE      FALSE
## age                FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          sbp tobacco ldl adiposity famhistPresent typea obesity alcohol age
## 1  ( 1 ) " " " "     " " " "       " "            " "    " "     " "     "*"
## 2  ( 1 ) " " " "     " " " "       "*"            " "    " "     " "     "*"
## 3  ( 1 ) " " "*"     " " " "       "*"            " "    " "     " "     "*"
## 4  ( 1 ) " " "*"     "*" " "       "*"            " "    " "     " "     "*"
## 5  ( 1 ) " " "*"     "*" " "       "*"            "*"    " "     " "     "*"
## 6  ( 1 ) " " "*"     "*" " "       "*"            "*"    "*"     " "     "*"
## 7  ( 1 ) "*" "*"     "*" " "       "*"            "*"    "*"     " "     "*"
## 8  ( 1 ) "*" "*"     "*" "*"       "*"            "*"    "*"     " "     "*"
```

```
# based on the summary, the best subset selection method identifies:
# the best model with only 1 variable is the one with "age"
# the best model with only 2 variable is the one with "age" and "famhistPresent"
# the best model with only 3 variable is the one with "age", "famhistPresent", and "tobacco"
# the best model with only 4 variable is the one with "age", "famhistPresent", "tobacco", and "ldl"
# the best model with only 5 variable is the one with "age", "famhistPresent", "tobacco", "ldl", and "t
# the best model with only 6 variable is the one with "age", "famhistPresent", "tobacco", "ldl", "typea
# the best model with only 7 variable is the one with "age", "famhistPresent", "tobacco", "ldl", "typea


regfit.f = regsubsets(chd ~ ., SA.Heart, nvmax = 9)
regfit.f.sum = summary(regfit.f)
regfit.f.sum
```

```
## Subset selection object
## Call: regsubsets.formula(chd ~ ., SA.Heart, nvmax = 9)
## 9 Variables  (and intercept)
##                 Forced in Forced out
## sbp                 FALSE      FALSE
## tobacco             FALSE      FALSE
## ldl                 FALSE      FALSE
## adiposity           FALSE      FALSE
## famhistPresent      FALSE      FALSE
## typea               FALSE      FALSE
## obesity             FALSE      FALSE
## alcohol             FALSE      FALSE
## age                 FALSE      FALSE
## 1 subsets of each size up to 9
## Selection Algorithm: exhaustive
##          sbp tobacco ldl adiposity famhistPresent typea obesity alcohol age
## 1  ( 1 ) " " " "     " " " "       " "             " "   " "     " "     "*"
## 2  ( 1 ) " " " "     " " " "       "*"             " "   " "     " "     "*"
## 3  ( 1 ) " " "*"     " " " "       "*"             " "   " "     " "     "*"
## 4  ( 1 ) " " "*"     "*" " "       "*"             " "   " "     " "     "*"
## 5  ( 1 ) " " "*"     "*" " "       "*"             "*"   " "     " "     "*"
## 6  ( 1 ) " " "*"     "*" " "       "*"             "*"   "*"     " "     "*"
## 7  ( 1 ) "*" "*"     "*" " "       "*"             "*"   "*"     " "     "*"
## 8  ( 1 ) "*" "*"     "*" "*"       "*"             "*"   "*"     " "     "*"
## 9  ( 1 ) "*" "*"     "*" "*"       "*"             "*"   "*"     "*"     "*"
```

```
names(regfit.f.sum)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```
regfit.f.sum$rsq
```

```
## [1] 0.1391091 0.1746324 0.1976777 0.2155700 0.2285384 0.2327238 0.2355085
## [8] 0.2358771 0.2360147
```

```
regfit.f.sum$rss
```

```
## [1] 90.03950 86.32416 83.91389 82.04255 80.68620 80.24846 79.95720 79.91866
## [9] 79.90426
```

```
regfit.f.sum$adjr2
```

```
## [1] 0.1372376 0.1710361 0.1924223 0.2087041 0.2200794 0.2226058 0.2237212
```

```
## [8] 0.2223826 0.2208026
```

```
regfit.f.sum$cp
```

```
## [1] 51.332678 32.315862 20.681509 12.095770  6.423213  5.947026  6.299469
## [8]  8.081428 10.000000
```

```
regfit.f.sum$bic
```

```
## [1] -56.93070 -70.26332 -77.21085 -81.49484 -83.06102 -79.43872 -74.98299
## [8] -69.07020 -63.01786
```

```
par(mfrow = c(2,2))
plot(regfit.f.sum$cp, xlab = "Number of Variables", ylab = "CP", type ="l")
points(which.min(regfit.f.sum$cp), regfit.f.sum$cp[which.min(regfit.f.sum$cp)], col = "red", pch = 20)
plot(regfit.f.sum$adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq", type ="l")
points(which.max(regfit.f.sum$adjr2), regfit.f.sum$adjr2[which.max(regfit.f.sum$adjr2)], col = "red", p
plot(regfit.f.sum$bic, xlab = "Number of Variables", ylab = "BIC", type ="l")
points(which.min(regfit.f.sum$bic), regfit.f.sum$bic[which.min(regfit.f.sum$bic)], col = "red", pch = 2
# the model with 6 variables has the lowest CP and it has the following coefficients
coef(regfit.f,6)
```

```
##    (Intercept)        tobacco            ldl famhistPresent          typea
##   -0.395225652    0.016663878    0.034654688    0.171923918    0.005876749
##        obesity            age
##   -0.007962060    0.007953219
```

```
# the model with 7 variables has the highest adjusted R^2
coef(regfit.f,7)
```

```
##    (Intercept)            sbp        tobacco            ldl famhistPresent
##   -0.535432209    0.001347086    0.016371096    0.034559620    0.172590484
##          typea        obesity            age
##    0.005971345   -0.008873812    0.007341973
```

```
# the model with 5 variables has the lowest BIC
coef(regfit.f,5)
```

```
##    (Intercept)        tobacco            ldl famhistPresent          typea
##   -0.545452392    0.016791566    0.030494337    0.170682179    0.005587670
##            age
##    0.007439075
```

```
# Forward and Backward Stepwise Selection
regfit.fwd = regsubsets(chd ~ ., SA.Heart, nvmax = 9, method ="forward")
summary(regfit.fwd)
```

```
## Subset selection object
## Call: regsubsets.formula(chd ~ ., SA.Heart, nvmax = 9, method = "forward")
## 9 Variables  (and intercept)
##                Forced in Forced out
## sbp                FALSE      FALSE
## tobacco            FALSE      FALSE
## ldl                FALSE      FALSE
## adiposity          FALSE      FALSE
## famhistPresent     FALSE      FALSE
## typea              FALSE      FALSE
## obesity            FALSE      FALSE
## alcohol            FALSE      FALSE
```

```
## age                 FALSE      FALSE
## 1 subsets of each size up to 9
## Selection Algorithm: forward
##          sbp tobacco ldl adiposity famhistPresent typea obesity alcohol age
## 1  ( 1 ) " " " "     " " " "       " "             " "    " "      " "     "*"
## 2  ( 1 ) " " " "     " " " "       "*"             " "    " "      " "     "*"
## 3  ( 1 ) " " "*"     " " " "       "*"             " "    " "      " "     "*"
## 4  ( 1 ) " " "*"     "*" " "       "*"             " "    " "      " "     "*"
## 5  ( 1 ) " " "*"     "*" " "       "*"             "*"    " "      " "     "*"
## 6  ( 1 ) " " "*"     "*" " "       "*"             "*"    "*"      " "     "*"
## 7  ( 1 ) "*" "*"     "*" " "       "*"             "*"    "*"      " "     "*"
## 8  ( 1 ) "*" "*"     "*" "*"       "*"             "*"    "*"      " "     "*"
## 9  ( 1 ) "*" "*"     "*" "*"       "*"             "*"    "*"      "*"     "*"
```

```
regfit.bwd = regsubsets(chd ~ ., SA.Heart, nvmax = 9, method ="backward")
summary(regfit.bwd)
```

```
## Subset selection object
## Call: regsubsets.formula(chd ~ ., SA.Heart, nvmax = 9, method = "backward")
## 9 Variables  (and intercept)
##                Forced in Forced out
## sbp                FALSE      FALSE
## tobacco            FALSE      FALSE
## ldl                FALSE      FALSE
## adiposity          FALSE      FALSE
## famhistPresent     FALSE      FALSE
## typea              FALSE      FALSE
## obesity            FALSE      FALSE
## alcohol            FALSE      FALSE
## age                FALSE      FALSE
## 1 subsets of each size up to 9
## Selection Algorithm: backward
##          sbp tobacco ldl adiposity famhistPresent typea obesity alcohol age
## 1  ( 1 ) " " " "     " " " "       " "             " "    " "      " "     "*"
## 2  ( 1 ) " " " "     " " " "       "*"             " "    " "      " "     "*"
## 3  ( 1 ) " " "*"     " " " "       "*"             " "    " "      " "     "*"
## 4  ( 1 ) " " "*"     "*" " "       "*"             " "    " "      " "     "*"
## 5  ( 1 ) " " "*"     "*" " "       "*"             "*"    " "      " "     "*"
## 6  ( 1 ) " " "*"     "*" " "       "*"             "*"    "*"      " "     "*"
## 7  ( 1 ) "*" "*"     "*" " "       "*"             "*"    "*"      " "     "*"
## 8  ( 1 ) "*" "*"     "*" "*"       "*"             "*"    "*"      " "     "*"
## 9  ( 1 ) "*" "*"     "*" "*"       "*"             "*"    "*"      "*"     "*"
```

```
# the best models with different # number o variables are the same in all three methods of best subset
# forward and backward stepwise selection

# Instead of using CP, BIC, or adjusted R^2 to choose the best model, we try to use the validation set
# and cross-validation to select the best one
# we need to divide the data into a training set and a test set
set.seed(1, sample.kind = "Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
train = sample(c(TRUE,FALSE), nrow(SA.Heart), rep = TRUE)
test = (!train)
```

```
# best subset selection
regfit.b = regsubsets(chd ~ ., data = SA.Heart[train,], nvmax = 9)
# validation test error
# a model matrix from the test data
test.mat = model.matrix(chd ~ ., data = SA.Heart[test,])
val.error = rep(NA,9)
for(i in 1:9){
  coefi = coef(regfit.b,id = 9)
  pred = test.mat[, names(coefi)]%*%coefi
  val.error[i] = mean((SA.Heart$chd[test]-pred)^2)
}
val.error
```

```
## [1] 0.1886678 0.1886678 0.1886678 0.1886678 0.1886678 0.1886678 0.1886678
## [8] 0.1886678 0.1886678
```

```
which.min(val.error)
```

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

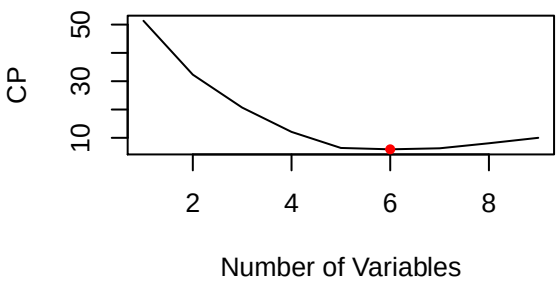
```
# all of the models have the same validation test errors.

# Now let perform the cross-validation approach to obtain the best model
# we consider K=10
k =10
set.seed(1, sample.kind = "Rounding")
```

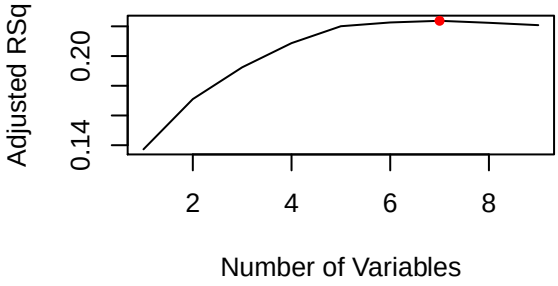
```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

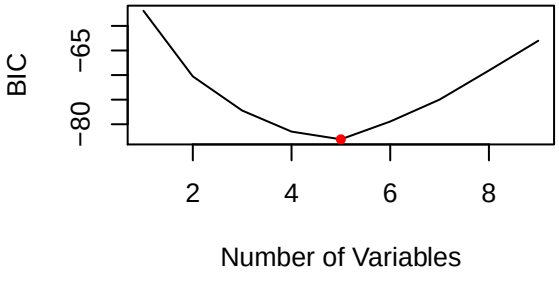
```
folds = sample(1:k, nrow(SA.Heart), replace = TRUE)
cv.errors=matrix(NA,k,9, dimnames=list(NULL, paste(1:9)))
# prediction
for(j in 1:k){
  best.f = regsubsets(chd ~ ., data = SA.Heart[folds != j,],nvmax = 9)
  test.mat = model.matrix(chd ~ ., data = SA.Heart[folds == j,])
  for(i in 1:9){
    coefi = coef(best.f, id = i)
    pred = test.mat[,names(coefi)]%*%coefi
    cv.errors [j,i] = mean((SA.Heart$chd[folds == j] - pred)^2)
  }
}

mean.cv.errors = apply(cv.errors, 2, mean)
mean.cv.errors
```

```
##         1         2         3         4         5         6         7         8
## 0.1987293 0.1990457 0.1955844 0.1906058 0.1839181 0.1852709 0.1850963 0.1855882
##         9
## 0.1862998
```

```
par(mfrow = c(1,1))
```







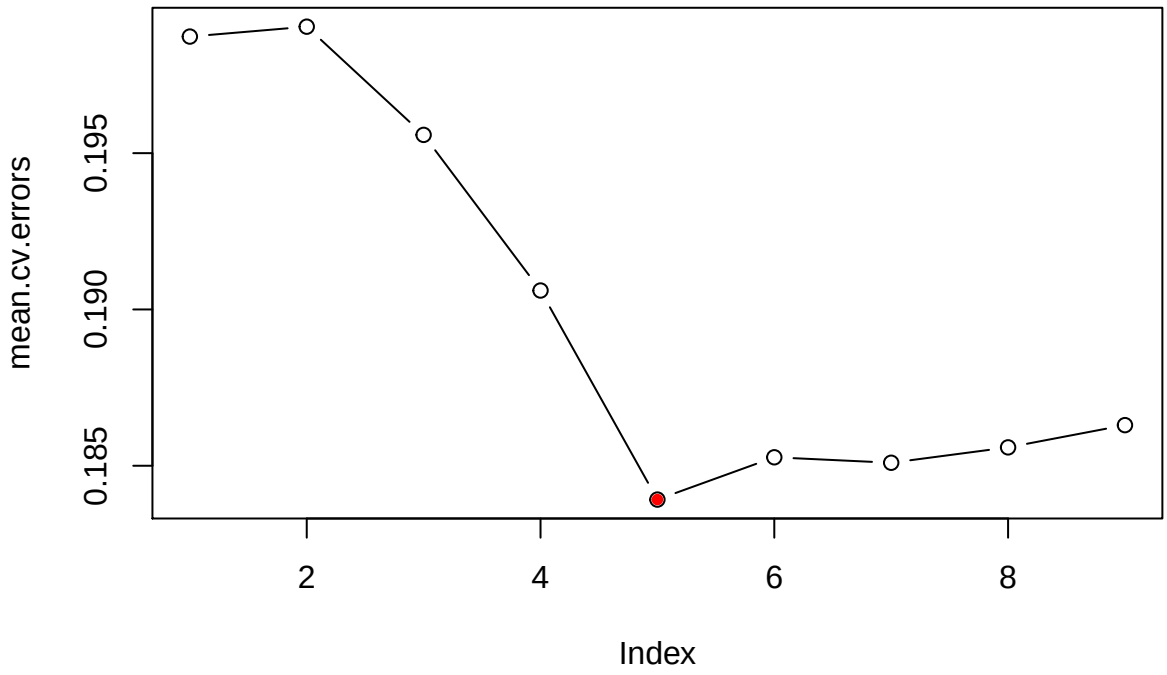

```
plot(mean.cv.errors, type = 'b')
points(which.min(mean.cv.errors), mean.cv.errors[which.min(mean.cv.errors)],col = "red", pch = 20)

# the best model is the one with 5 variables
# we now perform the best subset selection method with whole data and obtain the coefficients
# correspond to the model with 5 variables
reg.best = regsubsets(chd ~ ., data = SA.Heart, nvmax = 9)
coef(reg.best,5)
```

```
##    (Intercept)        tobacco            ldl famhistPresent          typea
##   -0.545452392    0.016791566    0.030494337    0.170682179    0.005587670
##            age
##    0.007439075
```

```
# the variables in this model are "tobacco", "ldl", "famhistPresent", "typea", and "age"

# we now perform ridge regression and the lasso in order to predict chd on the SA.Heart
x = model.matrix(chd ~ ., data = SA.Heart)[,-1]
y = SA.Heart$chd
library(glmnet)
```

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

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



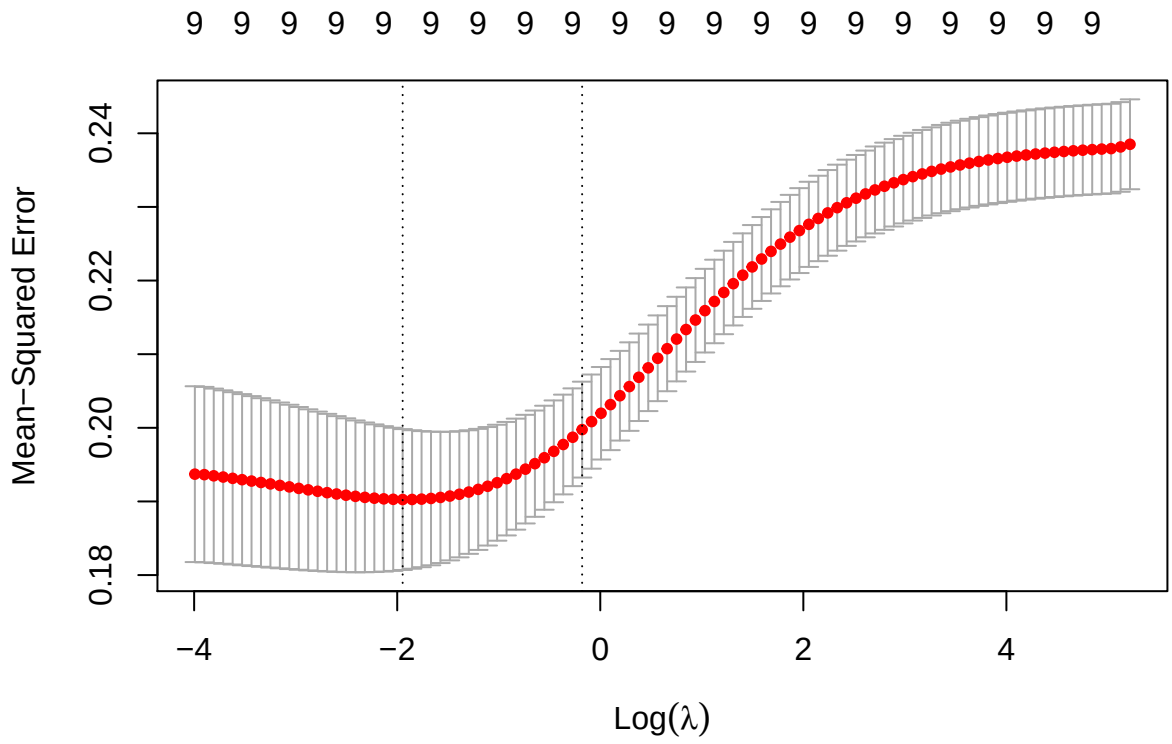

```
grid = 10^seq(10,-2,length =100)
# split the data into two groups: training data, test data
set.seed(1, sample.kind = "Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
train = sample(462, 231)
test = (-train)
y.test = y[test]
ridge.mob = glmnet(x[train,], y[train], alpha= 0, lambda = grid, thresh = 1e-12)
# Let consider that lambda = 5
ridge.pred = predict(ridge.mob, s = 5, newx=x[test,])
prob = rep(0,length(y.test))
prob[ridge.pred > 0.5] = 1
mean(prob==y.test)
```

```
## [1] 0.6926407
```

```
cv.out = cv.glmnet(x[train,],y[train], alpha = 0)
plot(cv.out)
```




```
bestlam = cv.out$lambda.min
bestlam
```

```
## [1] 0.1428105
```

```
# cross-validation tells us that the best lambda to use in the ridge regression is lambda = 0.1428105;
# by using it, we will get the lowest cross-validation error
ridge.pred = predict(ridge.mob, s = bestlam,newx=x[test,])
prob = rep(0,length(y.test))
prob[ridge.pred > 0.5] = 1
mean(prob==y.test)
```

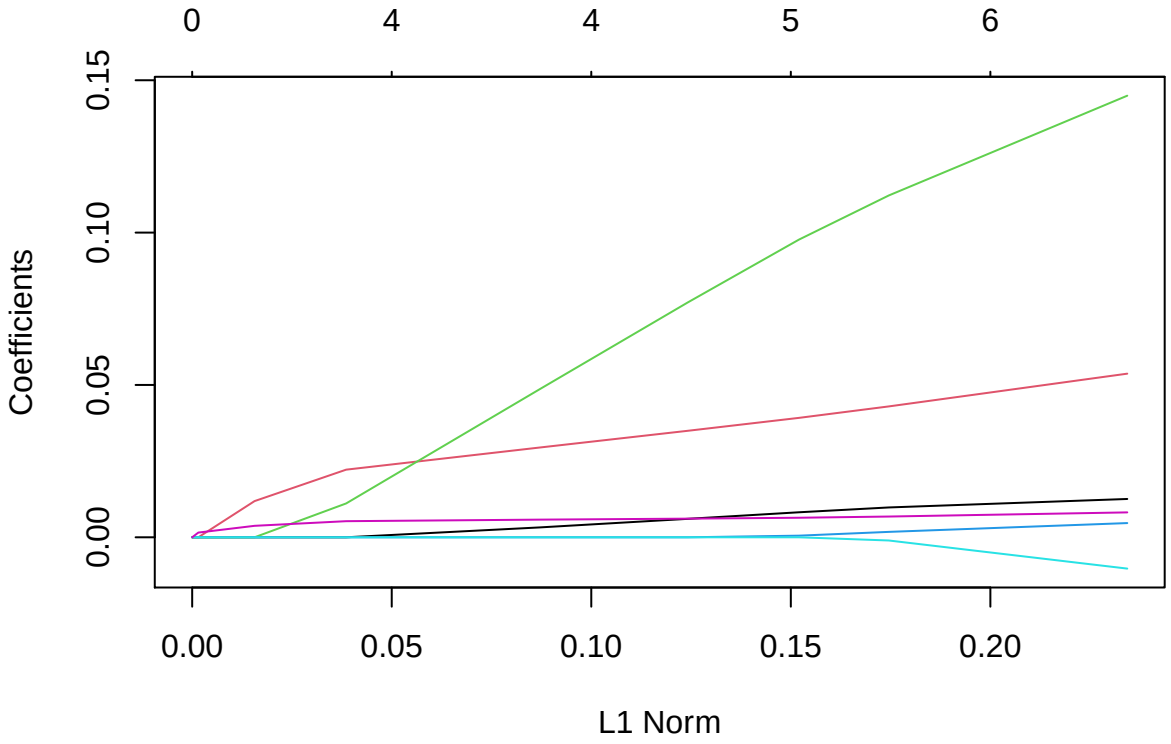
```
## [1] 0.7359307
```

```
# the performance when using lambda = 0.1428105 is lower than when we use lambda = 5
out=glmnet(x,y,alpha=0,lambda=grid)
ridge.coef=predict(out,type="coefficients",s=bestlam)
ridge.coef
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                               1
## (Intercept)    -4.403512e-01
## sbp             1.260818e-03
## tobacco         1.416250e-02
## ldl             2.690249e-02
## adiposity       2.671563e-03
## famhistPresent  1.420947e-01
## typea           4.474348e-03
## obesity        -6.300249e-03
## alcohol        -3.794395e-05
## age             5.452350e-03
```

```r
# the Lasso
lasso.mob  = glmnet(x[train,], y[train], alpha = 1, lambda = grid)
plot(lasso.mob)
```

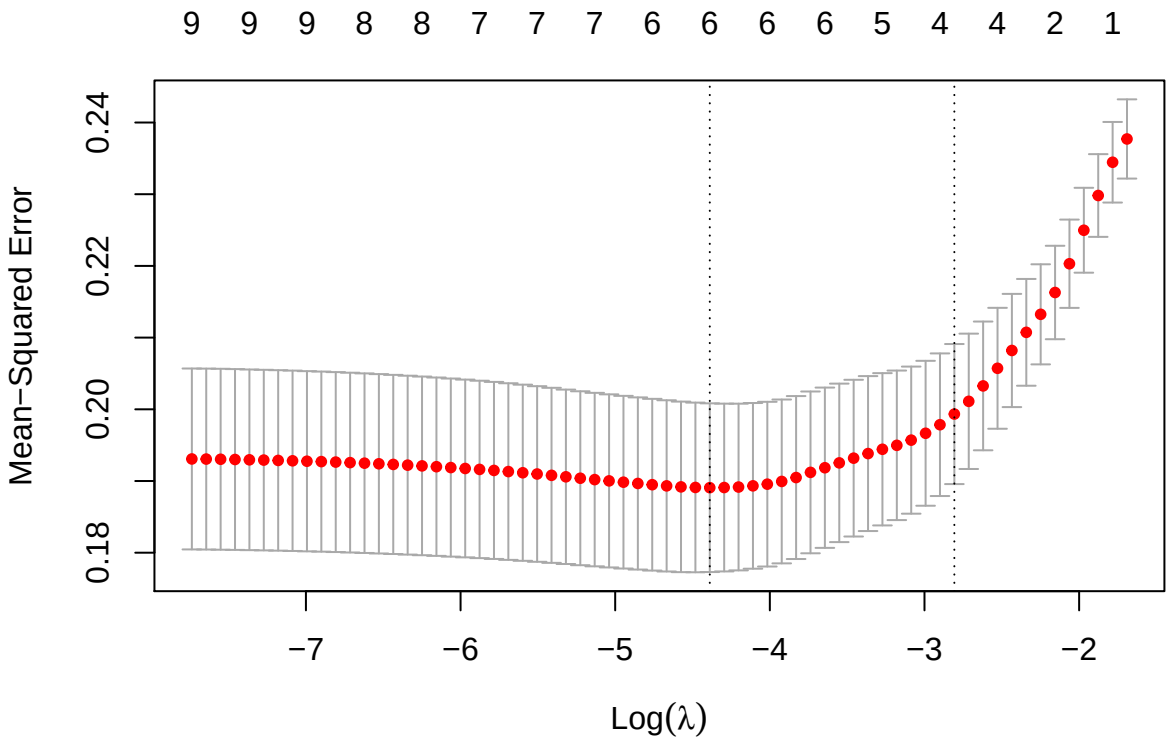
```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```




```r
set.seed(1, sample.kind = "Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```r
cv.out = cv.glmnet(x[train,], y[train], alpha = 1)
plot(cv.out)
```




```
bestlam = cv.out$lambda.min
bestlam
```

```
## [1] 0.01242093
```

```
lasso.pred = predict(lasso.mob, s = bestlam, newx = x[test,])
prob = rep(0,length(y.test))
prob[lasso.pred > 0.5] = 1
mean(prob==y.test)
```

```
## [1] 0.7272727
```

```
out=glmnet(x,y,alpha=1,lambda=grid)
lasso.coef=predict(out,type="coefficients",s=bestlam)
lasso.coef
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                               1
## (Intercept)    -0.4629296467
## sbp             0.0007576448
## tobacco         0.0148204649
## ldl             0.0281474098
## adiposity       .
## famhistPresent  0.1540212427
## typea           0.0044657783
## obesity        -0.0029968018
## alcohol         .
## age             0.0068672314
```

```
# based on the coefficients provided by the ridge regression and the lasso,
# it is clear that none of coefficients in the ridge regression are not zero, but
# there are several variables with coefficients equal zero in the Lasso.
```

```r
# Principal Components regression
library(pls)
```

```
##
## Attaching package: 'pls'
```

```
## The following object is masked from 'package:stats':
##
##     loadings
```

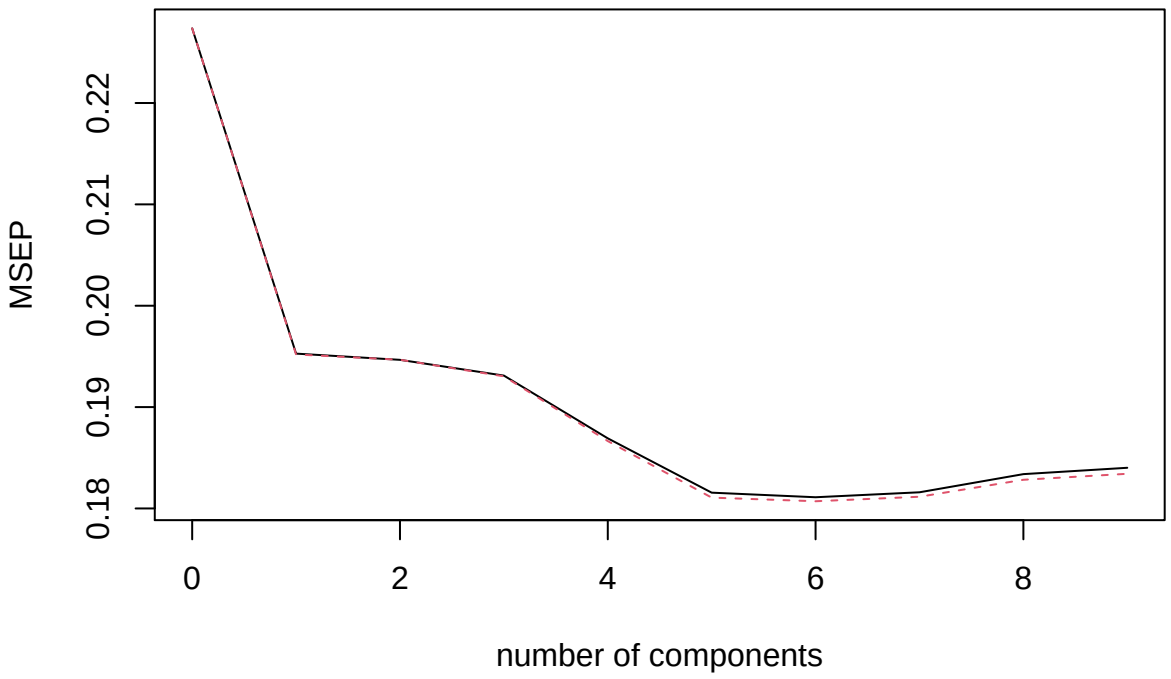
```r
set.seed(1, sample.kind = "Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```r
pcr.fit = pcr(chd ~ ., data = SA.Heart, scale = TRUE, validation = "CV")
summary(pcr.fit)
```

```
## Data:    X dimension: 462 9
##  Y dimension: 462 1
## Fit method: svdpc
## Number of components considered: 9
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV          0.4768   0.4419   0.4412   0.4394   0.4323   0.4261   0.4256
## adjCV       0.4768   0.4418   0.4412   0.4394   0.4320   0.4255   0.4251
##        7 comps  8 comps  9 comps
## CV      0.4261   0.4282   0.4290
## adjCV   0.4256   0.4276   0.4283
##
## TRAINING: % variance explained
##      1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X      32.05    45.35    57.31    67.80    77.09    85.54    92.94    98.06
## chd    14.31    14.81    16.45    20.02    22.67    23.16    23.38    23.60
##      9 comps
## X      100.0
## chd     23.6
```

```r
validationplot(pcr.fit, val.type="MSEP")
```




```
# the lowest cross-validation error obtain with 6 components, and with this number of components we are
# when we use all 9 components, we can capture 100% of the information about the predictors and the res

# we now compute the test MSE
pcr.pred = predict(pcr.fit, x[test,], ncomp = 6)
prob = rep(0,length(y.test))
prob[pcr.pred > 0.5] = 1
mean(prob == y.test)
```

```
## [1] 0.7705628
```

```
# the test MSE when we used PCR is lower than that the ridge regression and the Lasso

# fit the PCR on the full data set, using 6 components
pcr.fit = pcr(y~x, data = SA.Heart, scale = TRUE, ncomp = 6)
summary(pcr.fit)
```

```
## Data:    X dimension: 462 9
##  Y dimension: 462 1
## Fit method: svdpc
## Number of components considered: 6
## TRAINING: % variance explained
##    1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## X    32.05    45.35    57.31    67.80    77.09    85.54
## y    14.31    14.81    16.45    20.02    22.67    23.16
```

```
# Partial Least Square
library(pls)
set.seed(1, sample.kind = "Rounding")
```

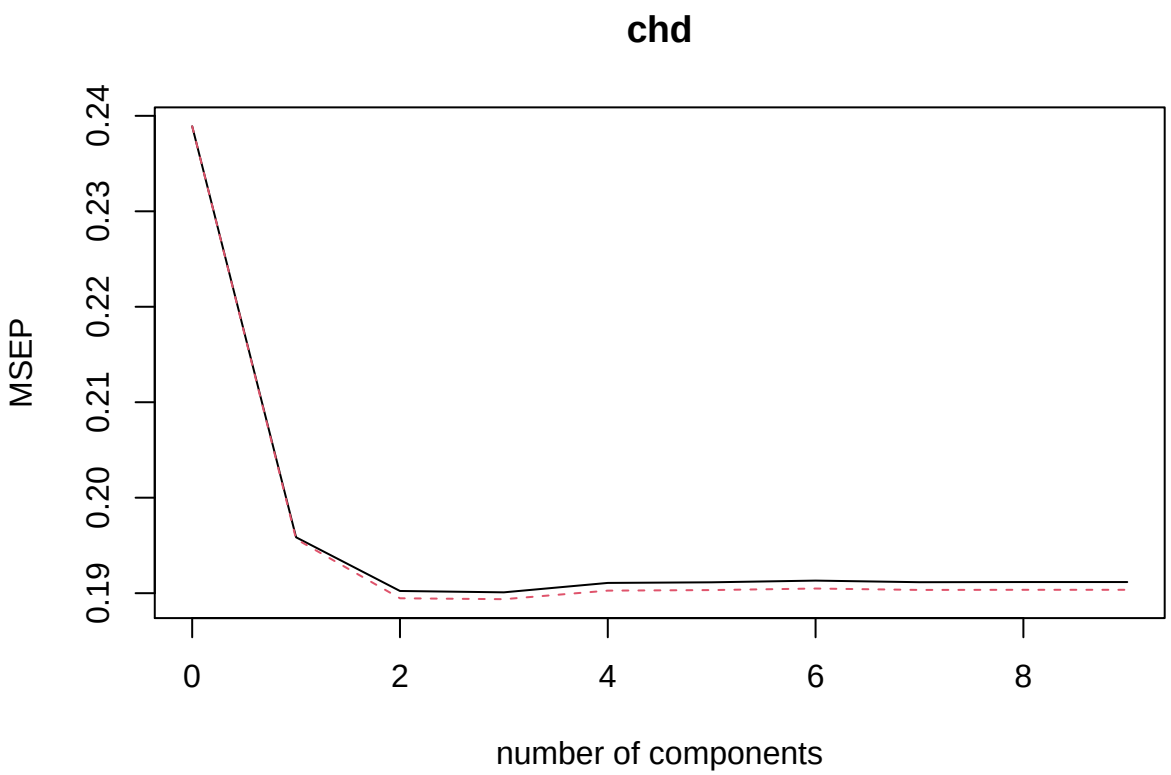
```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
```

```
## used
```

```
pls.fit = plsr(chd ~ ., data = SA.Heart, subset = train, scale = TRUE, validation = "CV")
summary(pls.fit)
```

```
## Data:    X dimension: 231 9
##  Y dimension: 231 1
## Fit method: kernelpls
## Number of components considered: 9
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV          0.4888   0.4426   0.4362   0.4360   0.4371   0.4372   0.4374
## adjCV       0.4888   0.4423   0.4353   0.4352   0.4362   0.4363   0.4364
##        7 comps  8 comps  9 comps
## CV      0.4372   0.4372   0.4372
## adjCV   0.4363   0.4363   0.4363
##
## TRAINING: % variance explained
##      1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X      30.00    41.25    50.55    59.97    67.00    74.71    81.76    89.89
## chd    19.97    25.46    25.64    25.66    25.67    25.67    25.68    25.68
##      9 comps
## X     100.00
## chd    25.68
```

```
validationplot(pls.fit, val.type= "MSEP")
```




```
# the lowest MSE obtain with 3 components, and with this number of components we can explain 52.77% of
# we now compute the test MSE
```

```r
pls.pred = predict(pls.fit, x[test,], ncomp = 3)
prob = rep(0,length(y.test))
prob[pls.pred > 0.5] = 1
mean(prob == y.test)
```

```
## [1] 0.7186147
```

```r
# fit the PCR on the full data set, using 3 components
pcl.fit = plsr(y~x, data = SA.Heart, scale = TRUE, ncomp = 3)
summary(pcl.fit)
```

```
## Data:    X dimension: 462 9
##  Y dimension: 462 1
## Fit method: kernelpls
## Number of components considered: 3
## TRAINING: % variance explained
##    1 comps  2 comps  3 comps
## X    30.69    42.54    52.77
## y    19.54    23.50    23.59
```

```r
# so far we were trying to fit a linear model to the data by using many different methods and different
# data, now we try nonlinear models
# recall that when we fit a linear model, only variables tobacco, ldl, famhist, typea, and age were
# statistically significant, so we only use these variables as the predictors and try to add more
# interactions and nonlinearlity to the model and then check which model is more accurate using
# anova() function

fit = glm(chd ~ ., data = SA.Heart, family = "binomial")
summary(fit)
```

```
##
## Call:
## glm(formula = chd ~ ., family = "binomial", data = SA.Heart)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7781  -0.8213  -0.4387   0.8889   2.5435
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -6.1507209  1.3082600  -4.701 2.58e-06 ***
## sbp             0.0065040  0.0057304   1.135 0.256374
## tobacco         0.0793764  0.0266028   2.984 0.002847 **
## ldl             0.1739239  0.0596617   2.915 0.003555 **
## adiposity       0.0185866  0.0292894   0.635 0.525700
## famhistPresent  0.9253704  0.2278940   4.061 4.90e-05 ***
## typea           0.0395950  0.0123202   3.214 0.001310 **
## obesity        -0.0629099  0.0442477  -1.422 0.155095
## alcohol         0.0001217  0.0044832   0.027 0.978350
## age             0.0452253  0.0121298   3.728 0.000193 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 596.11  on 461  degrees of freedom
## Residual deviance: 472.14  on 452  degrees of freedom
## AIC: 492.14
##
## Number of Fisher Scoring iterations: 5
```

```
# since age has the lowest p-value, we add higher exponents of this variable up to 5.
fit1 = glm(chd ~ . + I(age^2), data = SA.Heart, family = "binomial")
summary(fit1)
```

```
##
## Call:
## glm(formula = chd ~ . + I(age^2), family = "binomial", data = SA.Heart)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7801  -0.8313  -0.4275   0.9113   2.6566
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -7.2125042  1.7738731  -4.066 4.78e-05 ***
## sbp             0.0069995  0.0057301   1.222  0.22188
## tobacco         0.0775896  0.0264180   2.937  0.00331 **
## ldl             0.1709623  0.0598065   2.859  0.00426 **
## adiposity       0.0160774  0.0295141   0.545  0.58594
## famhistPresent  0.9127061  0.2281096   4.001 6.30e-05 ***
## typea           0.0387307  0.0123419   3.138  0.00170 **
## obesity        -0.0654878  0.0445509  -1.470  0.14157
## alcohol        -0.0004484  0.0045299  -0.099  0.92115
## age             0.1055191  0.0666886   1.582  0.11359
## I(age^2)       -0.0006874  0.0007420  -0.926  0.35424
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 596.11  on 461  degrees of freedom
## Residual deviance: 471.26  on 451  degrees of freedom
## AIC: 493.26
##
## Number of Fisher Scoring iterations: 5
```

```
fit2 = glm(chd ~ . + I(age^2) + I(age^3), data = SA.Heart, family = "binomial")
summary(fit2)
```

```
##
## Call:
## glm(formula = chd ~ . + I(age^2) + I(age^3), family = "binomial",
##     data = SA.Heart)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7773  -0.8439  -0.4176   0.8944   2.7924
##
## Coefficients:
```

```
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.062e+01  3.956e+00  -2.684  0.00727 **
## sbp             7.113e-03  5.749e-03   1.237  0.21599
## tobacco         7.981e-02  2.668e-02   2.992  0.00277 **
## ldl             1.713e-01  5.975e-02   2.867  0.00415 **
## adiposity       1.882e-02  2.967e-02   0.634  0.52600
## famhistPresent  9.237e-01  2.285e-01   4.042  5.3e-05 ***
## typea           3.799e-02  1.234e-02   3.078  0.00208 **
## obesity        -6.883e-02  4.488e-02  -1.534  0.12509
## alcohol        -6.729e-04  4.513e-03  -0.149  0.88147
## age             3.960e-01  3.002e-01   1.319  0.18712
## I(age^2)       -8.132e-03  7.451e-03  -1.091  0.27507
## I(age^3)        5.930e-05  5.869e-05   1.010  0.31234
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 596.11  on 461  degrees of freedom
## Residual deviance: 470.20  on 450  degrees of freedom
## AIC: 494.2
##
## Number of Fisher Scoring iterations: 6
```

```
fit3  = glm(chd ~ . + I(age^2) + I(age^3) + I(age^4), data = SA.Heart, family = "binomial")
summary(fit3)
```

```
##
## Call:
## glm(formula = chd ~ . + I(age^2) + I(age^3) + I(age^4), family = "binomial",
##     data = SA.Heart)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8021  -0.8205  -0.4442   0.8833   2.9841
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.589e+01  1.177e+01  -2.199  0.02786 *
## sbp             6.951e-03  5.776e-03   1.203  0.22885
## tobacco         7.955e-02  2.665e-02   2.985  0.00283 **
## ldl             1.685e-01  5.987e-02   2.815  0.00488 **
## adiposity       2.401e-02  2.998e-02   0.801  0.42313
## famhistPresent  9.321e-01  2.290e-01   4.071 4.68e-05 ***
## typea           3.883e-02  1.237e-02   3.140  0.00169 **
## obesity        -7.526e-02  4.526e-02  -1.663  0.09634 .
## alcohol        -1.059e-03  4.508e-03  -0.235  0.81431
## age             2.156e+00  1.274e+00   1.693  0.09042 .
## I(age^2)       -7.879e-02  4.941e-02  -1.595  0.11076
## I(age^3)        1.248e-03  8.157e-04   1.530  0.12594
## I(age^4)       -7.151e-06  4.862e-06  -1.471  0.14134
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 596.11  on 461  degrees of freedom
## Residual deviance: 467.93  on 449  degrees of freedom
## AIC: 493.93
##
## Number of Fisher Scoring iterations: 6
```

```
fit4  = glm(chd ~ . + I(age^2) + I(age^3) + I(age^4) + I(age^5), data = SA.Heart, family = "binomial")
summary(fit4)
```

```
##
## Call:
## glm(formula = chd ~ . + I(age^2) + I(age^3) + I(age^4) + I(age^5),
##     family = "binomial", data = SA.Heart)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8283  -0.8168  -0.4285   0.8809   2.9239
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.187e+00  3.297e+01  -0.066  0.94712
## sbp             7.095e-03  5.804e-03   1.222  0.22153
## tobacco         7.967e-02  2.675e-02   2.978  0.00290 **
## ldl             1.683e-01  5.983e-02   2.813  0.00491 **
## adiposity       2.588e-02  3.015e-02   0.858  0.39075
## famhistPresent  9.383e-01  2.295e-01   4.089 4.34e-05 ***
## typea           3.907e-02  1.242e-02   3.146  0.00165 **
## obesity        -7.597e-02  4.534e-02  -1.676  0.09382 .
## alcohol        -1.043e-03  4.516e-03  -0.231  0.81731
## age            -1.331e+00  4.750e+00  -0.280  0.77938
## I(age^2)        1.139e-01  2.590e-01   0.440  0.66010
## I(age^3)       -3.808e-03  6.743e-03  -0.565  0.57226
## I(age^4)        5.627e-05  8.427e-05   0.668  0.50431
## I(age^5)       -3.062e-07  4.068e-07  -0.753  0.45158
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 596.11  on 461  degrees of freedom
## Residual deviance: 467.37  on 448  degrees of freedom
## AIC: 495.37
##
## Number of Fisher Scoring iterations: 6
```

```
anova(fit,fit1,fit2,fit3,fit4)
```

```
## Analysis of Deviance Table
##
## Model 1: chd ~ sbp + tobacco + ldl + adiposity + famhist + typea + obesity +
##     alcohol + age
## Model 2: chd ~ sbp + tobacco + ldl + adiposity + famhist + typea + obesity +
##     alcohol + age + I(age^2)
## Model 3: chd ~ sbp + tobacco + ldl + adiposity + famhist + typea + obesity +
```

```
##     alcohol + age + I(age^2) + I(age^3)
## Model 4: chd ~ sbp + tobacco + ldl + adiposity + famhist + typea + obesity +
##     alcohol + age + I(age^2) + I(age^3) + I(age^4)
## Model 5: chd ~ sbp + tobacco + ldl + adiposity + famhist + typea + obesity +
##     alcohol + age + I(age^2) + I(age^3) + I(age^4) + I(age^5)
##   Resid. Df Resid. Dev Df Deviance
## 1       452     472.14
## 2       451     471.26  1  0.88352
## 3       450     470.20  1  1.05769
## 4       449     467.93  1  2.26401
## 5       448     467.37  1  0.56305
```

```
# interactions
fitt1 = glm(chd ~ . + age * tobacco,
              data = SA.Heart, family = "binomial")
summary(fitt1)
```

```
##
## Call:
## glm(formula = chd ~ . + age * tobacco, family = "binomial", data = SA.Heart)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7734  -0.8193  -0.4360   0.8907   2.5488
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -6.169e+00  1.325e+00  -4.656 3.23e-06 ***
## sbp             6.525e-03  5.732e-03   1.138 0.254943
## tobacco         9.264e-02  1.509e-01   0.614 0.539336
## ldl             1.738e-01  5.968e-02   2.912 0.003589 **
## adiposity       1.855e-02  2.929e-02   0.633 0.526465
## famhistPresent  9.237e-01  2.286e-01   4.040 5.34e-05 ***
## typea           3.950e-02  1.237e-02   3.194 0.001401 **
## obesity        -6.297e-02  4.426e-02  -1.423 0.154792
## alcohol         6.395e-05  4.531e-03   0.014 0.988738
## age             4.573e-02  1.338e-02   3.418 0.000631 ***
## tobacco:age    -2.482e-04  2.779e-03  -0.089 0.928827
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 596.11  on 461  degrees of freedom
## Residual deviance: 472.13  on 451  degrees of freedom
## AIC: 494.13
##
## Number of Fisher Scoring iterations: 5
```

```
fitt2 = glm(chd ~ . + age * tobacco + age * ldl,
              data = SA.Heart, family = "binomial")
summary(fitt2)
```

```
##
## Call:
```

```
## glm(formula = chd ~ . + age * tobacco + age * ldl, family = "binomial",
##     data = SA.Heart)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7734  -0.8191  -0.4360   0.8909   2.5481
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -6.163e+00  1.623e+00  -3.796 0.000147 ***
## sbp             6.525e-03  5.732e-03   1.138 0.254953
## tobacco         9.274e-02  1.516e-01   0.612 0.540834
## ldl             1.722e-01  2.379e-01   0.724 0.469079
## adiposity       1.856e-02  2.930e-02   0.633 0.526473
## famhistPresent  9.237e-01  2.286e-01   4.040 5.34e-05 ***
## typea           3.951e-02  1.240e-02   3.186 0.001440 **
## obesity        -6.297e-02  4.426e-02  -1.423 0.154800
## alcohol         6.734e-05  4.559e-03   0.015 0.988215
## age             4.558e-02  2.542e-02   1.793 0.072971 .
## tobacco:age    -2.501e-04  2.794e-03  -0.090 0.928658
## ldl:age         3.273e-05  4.901e-03   0.007 0.994672
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 596.11  on 461  degrees of freedom
## Residual deviance: 472.13  on 450  degrees of freedom
## AIC: 496.13
##
## Number of Fisher Scoring iterations: 5
```

```
fitt3 = glm(chd ~ . + age * tobacco + age * ldl + age * famhist,
              data = SA.Heart, family = "binomial")
summary(fitt3)
```

```
##
## Call:
## glm(formula = chd ~ . + age * tobacco + age * ldl + age * famhist,
##     family = "binomial", data = SA.Heart)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8665  -0.7949  -0.4550   0.8506   2.4604
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -5.9066620  1.6203200  -3.645 0.000267 ***
## sbp                0.0068874  0.0057789   1.192 0.233335
## tobacco            0.1104493  0.1523472   0.725 0.468462
## ldl                0.1919848  0.2364535   0.812 0.416829
## adiposity          0.0174504  0.0292991   0.596 0.551446
## famhistPresent    -0.2364191  0.9539538  -0.248 0.804265
## typea              0.0398622  0.0124394   3.205 0.001353 **
## obesity           -0.0612694  0.0442433  -1.385 0.166105
```

```
## alcohol             0.0002919  0.0045920   0.064 0.949319
## age                 0.0375242  0.0259833   1.444 0.148692
## tobacco:age        -0.0005244  0.0028029  -0.187 0.851588
## ldl:age            -0.0003379  0.0048862  -0.069 0.944870
## famhistPresent:age  0.0244457  0.0194745   1.255 0.209382
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 596.11  on 461  degrees of freedom
## Residual deviance: 470.52  on 449  degrees of freedom
## AIC: 496.52
##
## Number of Fisher Scoring iterations: 5
```

```
fitt4 = glm(chd ~ . + age * tobacco + age * ldl + age * famhist + age * typea,
              data = SA.Heart, family = "binomial")
summary(fitt4)
```

```
##
## Call:
## glm(formula = chd ~ . + age * tobacco + age * ldl + age * famhist +
##     age * typea, family = "binomial", data = SA.Heart)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8257  -0.7977  -0.4557   0.8427   2.5077
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -7.3021261  3.2050905  -2.278   0.0227 *
## sbp                 0.0068085  0.0057749   1.179   0.2384
## tobacco             0.1165124  0.1526645   0.763   0.4453
## ldl                 0.1916063  0.2363824   0.811   0.4176
## adiposity           0.0177863  0.0293376   0.606   0.5443
## famhistPresent     -0.2233923  0.9537144  -0.234   0.8148
## typea               0.0645814  0.0498988   1.294   0.1956
## obesity            -0.0607326  0.0442748  -1.372   0.1702
## alcohol             0.0003312  0.0045791   0.072   0.9423
## age                 0.0660549  0.0618440   1.068   0.2855
## tobacco:age        -0.0006444  0.0028069  -0.230   0.8184
## ldl:age            -0.0003618  0.0048786  -0.074   0.9409
## famhistPresent:age  0.0241538  0.0194575   1.241   0.2145
## typea:age          -0.0005124  0.0009987  -0.513   0.6079
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 596.11  on 461  degrees of freedom
## Residual deviance: 470.25  on 448  degrees of freedom
## AIC: 498.25
##
## Number of Fisher Scoring iterations: 5
```

```r
anova(fitt1,fitt2,fitt3,fitt4)
```

```
## Analysis of Deviance Table
##
## Model 1: chd ~ sbp + tobacco + ldl + adiposity + famhist + typea + obesity +
##     alcohol + age + age * tobacco
## Model 2: chd ~ sbp + tobacco + ldl + adiposity + famhist + typea + obesity +
##     alcohol + age + age * tobacco + age * ldl
## Model 3: chd ~ sbp + tobacco + ldl + adiposity + famhist + typea + obesity +
##     alcohol + age + age * tobacco + age * ldl + age * famhist
## Model 4: chd ~ sbp + tobacco + ldl + adiposity + famhist + typea + obesity +
##     alcohol + age + age * tobacco + age * ldl + age * famhist +
##     age * typea
##   Resid. Df Resid. Dev Df Deviance
## 1       451     472.13
## 2       450     472.13  1  0.00004
## 3       449     470.52  1  1.61421
## 4       448     470.25  1  0.26469
```

```r
# GAMs
# smothing splines
library(gam)
```

```
## Loading required package: splines
```

```
## Loading required package: foreach
```

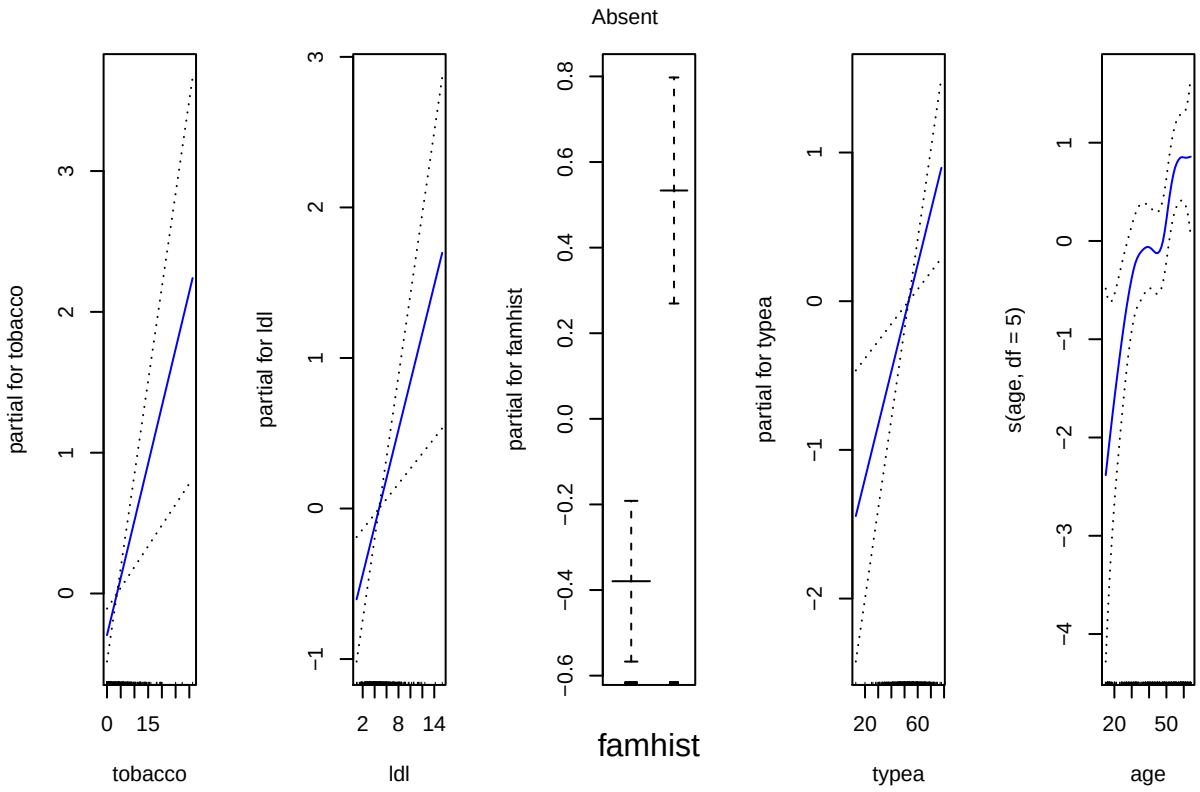
```
## Loaded gam 1.20
```

```r
gam = gam(chd ~ tobacco + ldl + famhist + typea + s(age, df = 5), family = "binomial", data = SA.Heart)
summary(gam)
```

```
##
## Call: gam(formula = chd ~ tobacco + ldl + famhist + typea + s(age,
##     df = 5), family = "binomial", data = SA.Heart)
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8323 -0.8278 -0.4313  0.9469  2.8769
##
## (Dispersion Parameter for binomial family taken to be 1)
##
##     Null Deviance: 596.1084 on 461 degrees of freedom
## Residual Deviance: 469.4874 on 451.9997 degrees of freedom
## AIC: 489.488
##
## Number of Local Scoring Iterations: NA
##
## Anova for Parametric Effects
##                  Df Sum Sq Mean Sq F value    Pr(>F)
## tobacco           1  19.47 19.4682 18.6002 1.981e-05 ***
## ldl               1  13.18 13.1751 12.5877 0.0004289 ***
## famhist           1  18.36 18.3627 17.5440 3.377e-05 ***
## typea             1   3.62  3.6201  3.4587 0.0635696 .
## s(age, df = 5)    1  20.92 20.9174 19.9848 9.882e-06 ***
## Residuals       452 473.09  1.0467
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                  Npar Df Npar Chisq P(Chi)
## (Intercept)
## tobacco
## ldl
## famhist
## typea
## s(age, df = 5)       4     5.9634 0.2019
```

```
par(mfrow = c(1,5))
plot(gam,se=TRUE, col ="blue")
```




```
# tree
library(tree)
set.seed(1, sample.kind = "Rounding")
```

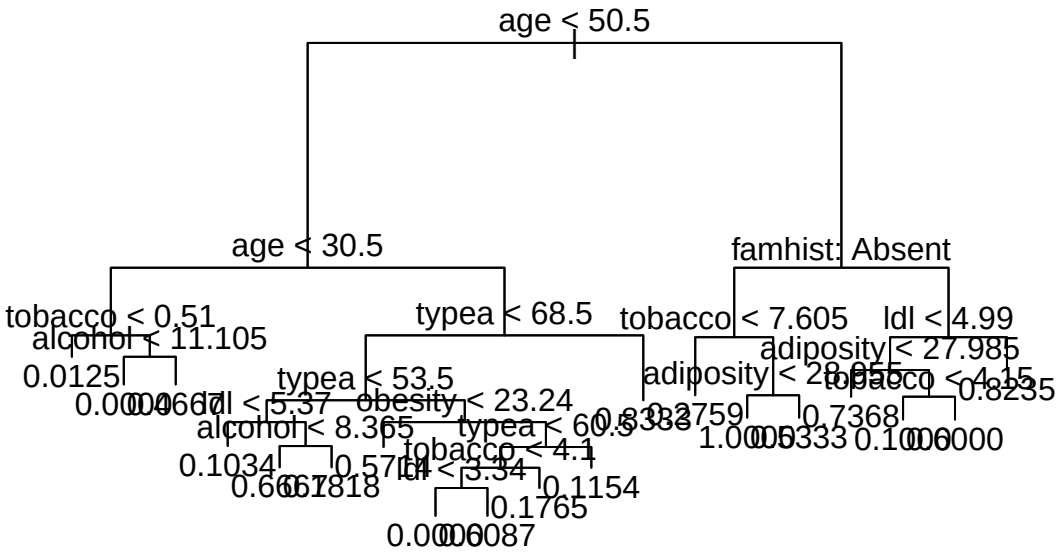
```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
tree.SA.Heart = tree(chd ~ ., SA.Heart)
summary(tree.SA.Heart)
```

```
##
## Regression tree:
## tree(formula = chd ~ ., data = SA.Heart)
## Variables actually used in tree construction:
## [1] "age"       "tobacco"   "alcohol"   "typea"     "ldl"       "obesity"
## [7] "famhist"   "adiposity"
## Number of terminal nodes:  19
```

```
## Residual mean deviance:  0.1374 = 60.86 / 443
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.8333 -0.1765 -0.0125  0.0000  0.1765  0.9875
```

```
plot(tree.SA.Heart)
text(tree.SA.Heart, pretty = 0)
```




```
train = sample(1:nrow(SA.Heart), 347)
SA.Heart.test = SA.Heart[-train,]
chd.test = chd[-train]
tree.SA.Heart = tree(chd ~ ., SA.Heart, subset = train)
summary(tree.SA.Heart)
```

```
##
## Regression tree:
## tree(formula = chd ~ ., data = SA.Heart, subset = train)
## Variables actually used in tree construction:
## [1] "age"       "tobacco"   "ldl"       "obesity"   "alcohol"   "famhist"
## [7] "typea"     "adiposity"
## Number of terminal nodes:  26
## Residual mean deviance:  0.1099 = 35.28 / 321
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.93330 -0.05000 -0.04762  0.00000  0.15790  0.95240
```

```
tree.pred = predict(tree.SA.Heart, SA.Heart.test)
mean((tree.pred - chd.test)^2)
```

```
## [1] 0.2281715
```

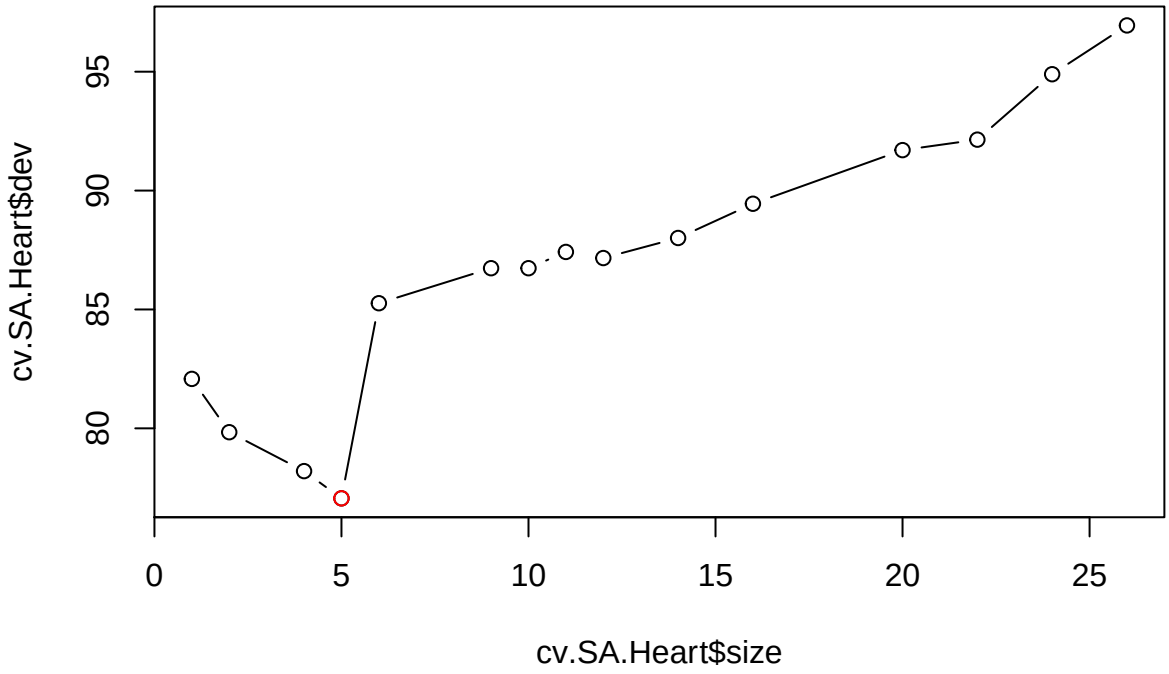
```
cv.SA.Heart = cv.tree(tree.SA.Heart)
plot(cv.SA.Heart$size, cv.SA.Heart$dev, type ="b")
cv.SA.Heart$dev
```

```
##  [1] 96.94509 94.89362 92.14153 91.70135 89.44603 88.00583 87.16295 87.41910
##  [9] 86.73491 86.73491 85.26208 77.05676 78.20120 79.83815 82.07954
```

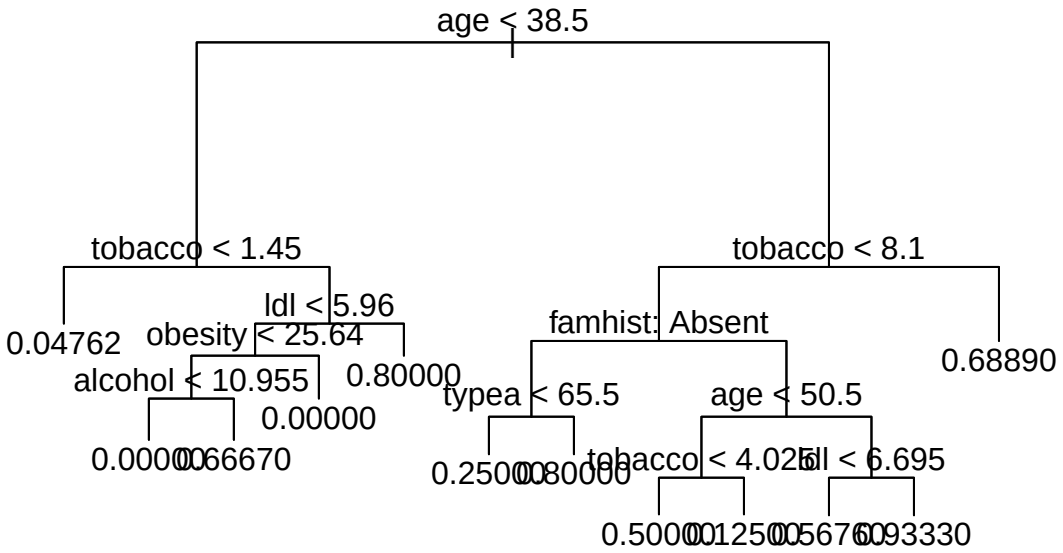
```
cv.SA.Heart$size
```

```
##  [1] 26 24 22 20 16 14 12 11 10  9  6  5  4  2  1
```

```
min.tree = which.min(cv.SA.Heart$dev)
points(cv.SA.Heart$size[min.tree], cv.SA.Heart$dev[min.tree], col = "red")
```




```
prune.SA.Heart = prune.tree(tree.SA.Heart, best = min.tree)
plot(prune.SA.Heart)
text(prune.SA.Heart, pretty = 0)
```




```
# prediction and test MSE
pred = predict(prune.SA.Heart, newdata = SA.Heart.test)
prob = rep(0,length(chd.test))
prob[pred>0.5] = 1
mean(prob == chd.test)
```

```
## [1] 0.7391304
```

```
# bagging
library(randomForest)
```

```
## randomForest 4.6-14
```

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

```
set.seed(1, sample.kind = "Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
# training data
SA.Heart.train = SA.Heart[train,]
#test data
SA.Heart.test = SA.Heart[-train,]
bag.SA.Heart = randomForest(chd ~ ., data = SA.Heart.train, mtry = 10, ntree= 500, importance = TRUE)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
## Warning in randomForest.default(m, y, ...): invalid mtry: reset to within valid
## range
```

```
pred = predict(bag.SA.Heart, newdata = SA.Heart.test)
prod = rep(0, length(chd.test))
prod[pred>0.5] = 1
mean(prod == chd.test)
```

```
## [1] 0.7130435
```

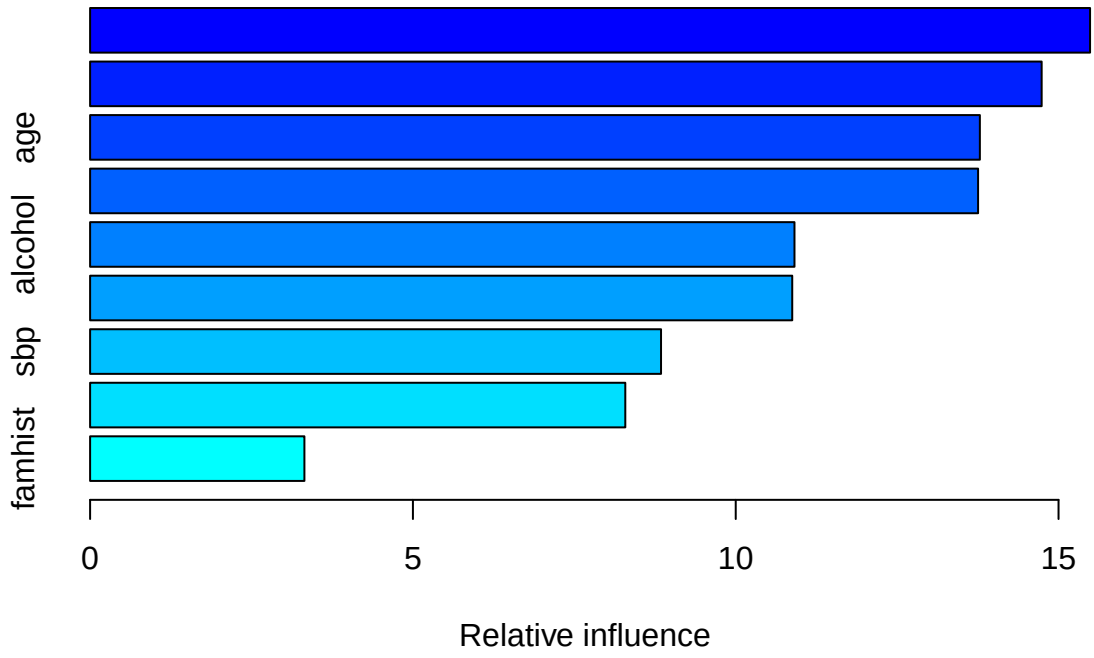
```
importance(bag.SA.Heart)
```

```
##               %IncMSE IncNodePurity
## sbp       -1.4032632      6.816259
## tobacco   12.0651588     11.475977
## ldl        5.8686620     10.289469
## adiposity  3.1491246      7.596888
## famhist    8.1885241      2.942255
## typea     -0.4583013      7.639277
## obesity   -2.3504579      7.902911
## alcohol   -0.1419028      5.590586
## age       14.2793093     13.644138
```

```
# tobacco and age the two important variables


# boosting
library(gbm)
```

```
## Loaded gbm 2.1.8
```

```
boost.SA.Heart = gbm(chd~., data = SA.Heart.train ,distribution="bernoulli",n.trees = 1000, interaction
summary(boost.SA.Heart)
```




```
##                 var   rel.inf
## tobacco     tobacco 15.488379
## obesity     obesity 14.738798
## age             age 13.781525
## ldl             ldl 13.753748
## alcohol     alcohol 10.910117
## adiposity adiposity 10.875480
## sbp             sbp  8.843083
## typea         typea  8.289749
## famhist     famhist  3.319120
```

```
yhat.boost=predict(boost.SA.Heart, newdata=SA.Heart.test, n.trees=1000, type = "response")
prob = rep(0,length(chd.test))
prob[yhat.boost > 0.2] = 1
table(prob, chd.test)
```

```
##     chd.test
## prob  0  1
##    0 52 15
##    1 25 23
```

```
mean(prob == chd.test)
```

```
## [1] 0.6521739
```