FE590. Assignment #3.

This content is protected and may not be shared, uploaded, or distributed

Enter Your Name Here, or "Anonymous" if you want to remain anonymous..

2022-04-11

Instructions

In this assignment, you should use R markdown to answer the questions below. Simply type your R code into embedded chunks as shown above.

When you have completed the assignment, knit the document into a PDF file, and upload *both* the .pdf and .Rmd files to Canvas.

Note that you must have LaTeX installed in order to knit the equations below. If you do not have it installed, simply delete the questions below.

Question 1 (based on JWHT Chapter 5, Problem 8)

In this problem, you will perform cross-validation on a simulated data set.

You will use this personalized simulated data set for this problem:

```
library(leaps)
library(boot)
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.1.2
```

```
library(ISLR)

CWID = 10474181 #Place here your Campus wide ID number, this will personalize
#your results, but still maintain the reproduceable nature of using seeds.
#If you ever need to reset the seed in this assignment, use this as your seed
#Papers that use -1 as this CWID variable will earn 0's so make sure you change
#this value before you submit your work.
personal = CWID %% 10000
set.seed(personal)
y <- rnorm(100)
x <- rnorm(100)
y <- x - 2*x^2 + rnorm(100)
```

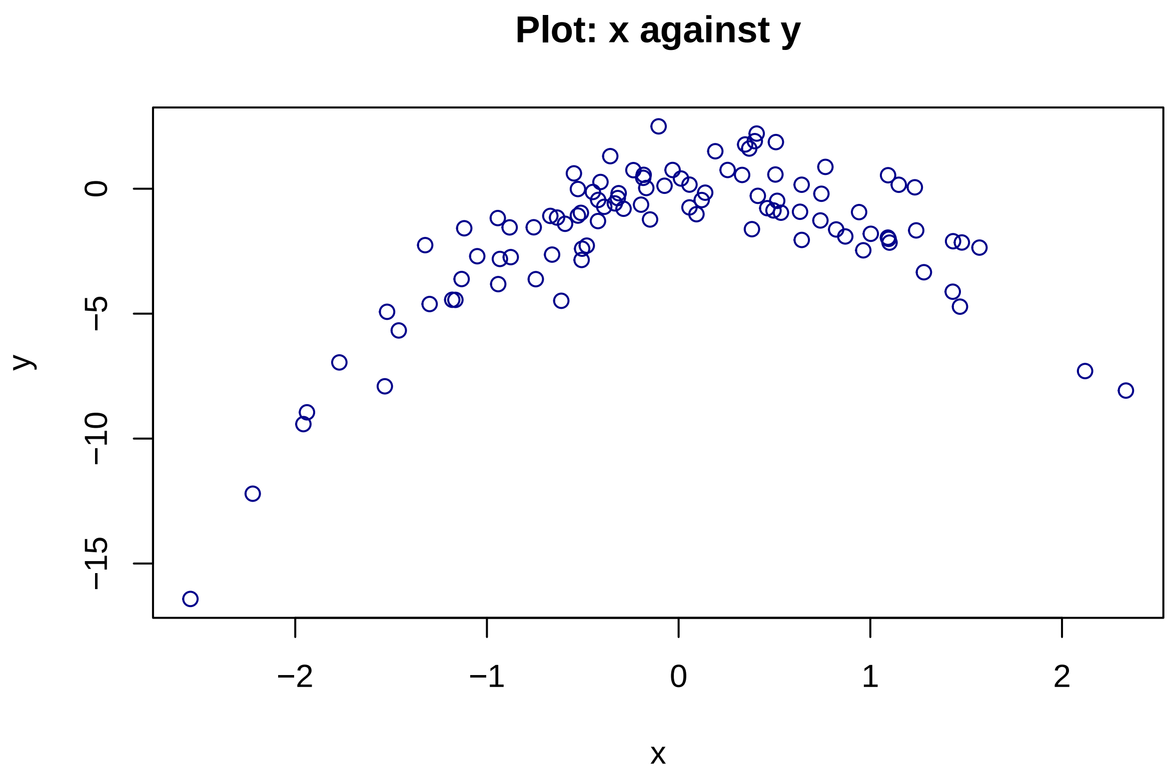
(a) In this data set, what is n and what is p?
(b) Create a scatterplot of x against y. Comment on what you find.

(c) Compute the LOOCV errors that result from fitting the following four models using least squares: 1. $Y = \beta_0 + \beta_1 X + \epsilon$ 2. $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon$ 3. $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$ 4. $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \beta_4 X^4 + \epsilon$

(d) Which of the models in (c) had the smallest LOOCV error? Is this what you expected? Explain your answer.

(e) Comment on the statistical significance of the coefficient estimates that results from fitting each of the models in (c) using least squares. Do these results agree with the conclusions drawnbased on the cross-validation results?

```
# Enter your R code here!

#(a)
# n showcases the number of observations. Here n = 100
# p is the number of  predictors. Here p = 2 i.e. x and x^2.
# #(b)
plot(x=x, y=y, main="Plot: x against y", xlab = "x", ylab = "y", col = "darkblue")
```




```
#The relation between x and y in not linear. It quadratic relationship.
# as we define y as x - 2*x^2 + epsilon which is quadratic.
df_1 <- data.frame(x,y)
#Model 1
modelA <- glm(y ~ x)
modelA1 <- cv.glm(df_1, modelA)
modelA1$delta
```

```
## [1] 8.174068 8.170247
```

```r
#Model 2
modelB <- glm(y ~ poly(x, 2))
modelB2 <- cv.glm(df_1, modelB)
modelB2$delta
```

```
## [1] 1.320872 1.320515
```

```r
#Model 3
modelC <- glm(y ~ poly(x, 3))
modelC3 <- cv.glm(df_1, modelC)
modelC3$delta
```

```
## [1] 1.328940 1.328498
```

```r
#Model 4
modelD <- glm(y ~ poly(x, 4))
modelD4 <- cv.glm(df_1, modelD)
modelD4$delta
```

```
## [1] 1.373301 1.372551
```

```r
# (d)
#  modelB has the smallest LOOCV error. It was
#expected that modelB should have least LOOCV error as we defined y as x - 2*x^2
# + epsilon.
#When we see the delta vector we get two values. The values are
# identical upto two decimal place.
#(e)
summary(modelA)
```

```
##
## Call:
## glm(formula = y ~ x)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -11.465   -1.235    0.610    1.761    4.371
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.7447     0.2757  -6.327 7.49e-09 ***
## x             1.2597     0.2890   4.359 3.22e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 7.585236)
##
##     Null deviance: 887.49  on 99  degrees of freedom
## Residual deviance: 743.35  on 98  degrees of freedom
## AIC: 490.39
##
## Number of Fisher Scoring iterations: 2
```

```r
summary(modelB)
```

```
##
## Call:
## glm(formula = y ~ poly(x, 2))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2490  -0.7895  -0.0546   0.7124   2.5203
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.8027     0.1136  -15.87   <2e-16 ***
## poly(x, 2)1  12.0056     1.1360   10.57   <2e-16 ***
## poly(x, 2)2 -24.8629     1.1360  -21.89   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.290605)
##
##     Null deviance: 887.49  on 99  degrees of freedom
## Residual deviance: 125.19  on 97  degrees of freedom
## AIC: 314.25
##
## Number of Fisher Scoring iterations: 2
```

```r
summary(modelC)
```

```
##
## Call:
## glm(formula = y ~ poly(x, 3))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3022  -0.7270  -0.0264   0.6871   2.5199
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.8027     0.1139 -15.824   <2e-16 ***
## poly(x, 3)1  12.0056     1.1392  10.538   <2e-16 ***
## poly(x, 3)2 -24.8629     1.1392 -21.824   <2e-16 ***
## poly(x, 3)3   0.7721     1.1392   0.678      0.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.297839)
##
##     Null deviance: 887.49  on 99  degrees of freedom
## Residual deviance: 124.59  on 96  degrees of freedom
## AIC: 315.78
##
## Number of Fisher Scoring iterations: 2
```

```
summary(modelD)
```

```
##
## Call:
## glm(formula = y ~ poly(x, 4))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2994  -0.7364  -0.0264   0.6852   2.5316
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.8027     0.1145 -15.743   <2e-16 ***
## poly(x, 4)1  12.0056     1.1451  10.484   <2e-16 ***
## poly(x, 4)2 -24.8629     1.1451 -21.712   <2e-16 ***
## poly(x, 4)3   0.7721     1.1451   0.674    0.502
## poly(x, 4)4  -0.1477     1.1451  -0.129    0.898
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.311271)
##
##     Null deviance: 887.49  on 99  degrees of freedom
## Residual deviance: 124.57  on 95  degrees of freedom
## AIC: 317.76
##
## Number of Fisher Scoring iterations: 2
```

```
# modelB i.e. x and x^2 (linear and quadratic
# part) are significant.
#  modelC the variable x^3 is not significant.
# modelD the variable x^3 and x^4 are not significant.
# modelB has least LOOCV error.
#Thus the statistical significance of the coefficient
# estimates agree with the conclusions drawnbased
#on the cross-validation results.
```

Question 2 (based on JWTH Chapter 7, Problem 10)

The question refers to the 'College' data set

(a) Split the data into a training set and a test set. Using out-of-state tuition as the response and the other variables as the predictors, perform subset selection (your choice on how) in order to identify a satisfactory model that uses just a subset of the predictors (if your approach suggests using all of the predictors, then follow your results and use them all).
(b) Fit a GAM on the training data, using out-of-state tuition as the response and the features selected in the previous step as the predictors, using splines of each feature with 5 df.
(c) Evaluate the model obtained on the test set, and explain the results obtained
(d) For which variables, if any, is there evidence of a non-linear relationship with the response? Which are probably linear? Justify your answers.

```r
# Enter your R code here!

#(a)
set.seed(personal)
attach(College)
length(College)
```

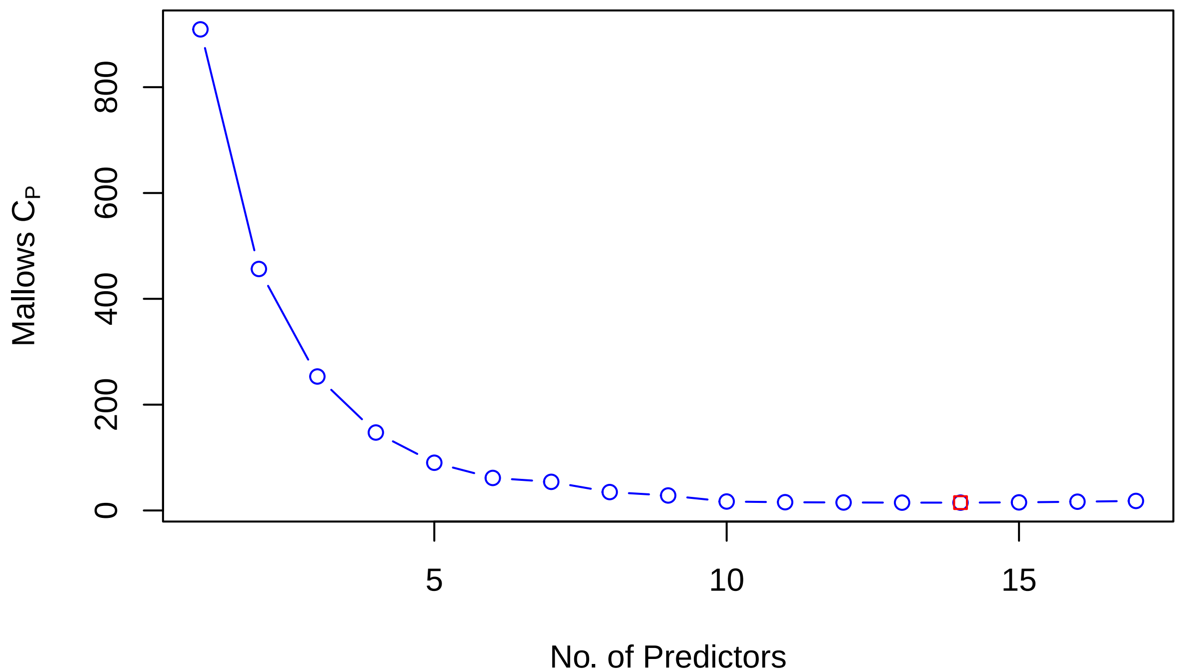
```
## [1] 18
```

```r
xvar <- sample(length(Apps), as.integer(length(Apps)*0.85))
college_Train_set <- College[xvar,]
college_Test_set <- College[-xvar,]
s_college <- regsubsets(Outstate ~ ., data = college_Train_set, method = "exhaustive", nvmax
= 18)

summary(s_college)[7]
```

```
## $outmat
##           PrivateYes Apps Accept Enroll Top10perc Top25perc F.Undergrad
## 1  ( 1 )  " "        " "  " "    " "    " "       " "       " "
## 2  ( 1 )  "*"        " "  " "    " "    " "       " "       " "
## 3  ( 1 )  "*"        " "  " "    " "    " "       " "       " "
## 4  ( 1 )  "*"        " "  " "    " "    " "       " "       " "
## 5  ( 1 )  "*"        " "  " "    " "    " "       " "       " "
## 6  ( 1 )  "*"        " "  " "    " "    " "       " "       " "
## 7  ( 1 )  "*"        " "  "*"    " "    " "       " "       "*"
## 8  ( 1 )  "*"        " "  "*"    "*"    " "       " "       " "
## 9  ( 1 )  "*"        "*"  "*"    "*"    " "       " "       " "
## 10  ( 1 ) "*"        "*"  "*"    "*"    "*"       " "       " "
## 11  ( 1 ) "*"        "*"  "*"    "*"    "*"       " "       " "
## 12  ( 1 ) "*"        "*"  "*"    "*"    "*"       " "       "*"
## 13  ( 1 ) "*"        "*"  "*"    "*"    "*"       " "       "*"
## 14  ( 1 ) "*"        "*"  "*"    "*"    "*"       " "       "*"
## 15  ( 1 ) "*"        "*"  "*"    "*"    "*"       " "       "*"
## 16  ( 1 ) "*"        "*"  "*"    "*"    "*"       " "       "*"
## 17  ( 1 ) "*"        "*"  "*"    "*"    "*"       "*"       "*"
##           P.Undergrad Room.Board Books Personal PhD Terminal S.F.Ratio
## 1  ( 1 )  " "         " "        " "   " "      " " " "      " "
## 2  ( 1 )  " "         " "        " "   " "      " " " "      " "
## 3  ( 1 )  " "         "*"        " "   " "      " " " "      " "
## 4  ( 1 )  " "         "*"        " "   " "      " " " "      " "
## 5  ( 1 )  " "         "*"        " "   " "      "*" " "      " "
## 6  ( 1 )  " "         "*"        " "   " "      "*" " "      " "
## 7  ( 1 )  " "         "*"        " "   " "      "*" " "      " "
## 8  ( 1 )  " "         "*"        " "   " "      "*" " "      " "
## 9  ( 1 )  " "         "*"        " "   " "      "*" " "      " "
## 10  ( 1 ) " "         "*"        " "   " "      " " "*"      " "
## 11  ( 1 ) " "         "*"        "*"   " "      " " "*"      " "
## 12  ( 1 ) " "         "*"        "*"   " "      " " "*"      " "
## 13  ( 1 ) " "         "*"        "*"   " "      "*" "*"      " "
## 14  ( 1 ) " "         "*"        "*"   " "      "*" "*"      "*"
## 15  ( 1 ) " "         "*"        "*"   "*"      "*" "*"      "*"
```

```
## 16  ( 1 ) "*"         "*"         "*"   "*"       "*" "*"       "*"
## 17  ( 1 ) "*"         "*"         "*"   "*"       "*" "*"       "*"
##           perc.alumni Expend Grad.Rate
## 1  ( 1 )  " "         "*"    " "
## 2  ( 1 )  " "         "*"    " "
## 3  ( 1 )  " "         "*"    " "
## 4  ( 1 )  "*"         "*"    " "
## 5  ( 1 )  "*"         "*"    " "
## 6  ( 1 )  "*"         "*"    "*"
## 7  ( 1 )  "*"         "*"    " "
## 8  ( 1 )  "*"         "*"    "*"
## 9  ( 1 )  "*"         "*"    "*"
## 10  ( 1 ) "*"         "*"    "*"
## 11  ( 1 ) "*"         "*"    "*"
## 12  ( 1 ) "*"         "*"    "*"
## 13  ( 1 ) "*"         "*"    "*"
## 14  ( 1 ) "*"         "*"    "*"
## 15  ( 1 ) "*"         "*"    "*"
## 16  ( 1 ) "*"         "*"    "*"
## 17  ( 1 ) "*"         "*"    "*"
```

```r
c_p <- summary(s_college)$cp
plot(c_p , type='b', xlab="No. of Predictors",
 ylab=expression("Mallows C"[P]), col="blue")
points(which.min(c_p), c_p[which.min(c_p)], pch=22,
 col="red")
```



```r
which.min(c_p)
```

```
## [1] 14
```

```r
# Model with 14 variables is the best: has the minimum Mallows Cp.
# The 14 coefficients are as follow:

coef(s_college,14)
```

```
##   (Intercept)    PrivateYes          Apps        Accept        Enroll
## -2.026256e+03  2.359814e+03 -2.713083e-01  9.092887e-01 -9.128134e-01
##     Top10perc   F.Undergrad    Room.Board         Books           PhD
##  2.535235e+01 -9.693813e-02  8.591396e-01 -7.457145e-01  1.542012e+01
##      Terminal     S.F.Ratio   perc.alumni        Expend     Grad.Rate
##  2.279360e+01 -4.008436e+01  4.525095e+01  1.871600e-01  2.434782e+01
```

```r
#(b)

library(gam)
```

```
## Warning: package 'gam' was built under R version 4.1.2

## Loading required package: splines

## Loading required package: foreach

## Warning: package 'foreach' was built under R version 4.1.2

## Loaded gam 1.20.1
```

```r
colnames(College)
```

```
##  [1] "Private"     "Apps"        "Accept"      "Enroll"      "Top10perc"
##  [6] "Top25perc"   "F.Undergrad" "P.Undergrad" "Outstate"    "Room.Board"
## [11] "Books"       "Personal"    "PhD"         "Terminal"    "S.F.Ratio"
## [16] "perc.alumni" "Expend"      "Grad.Rate"
```

```r
gamModel <- gam(Outstate ~ Private + s(Apps,5) + s(Accept,5) +
 s(Enroll,5) + s(Top10perc,5) + s(Room.Board,5) +s(S.F.Ratio ,5)
 +s(Personal,5) + s(PhD,5) + s(Terminal,5) +s(perc.alumni,5)+
 s(perc.alumni,5) + s(Expend,5) + s(Grad.Rate,5),data =college_Train_set)

#(c)

mean((college_Test_set$Outstate - predict(gamModel, college_Test_set))^2)
```

```
## [1] 4017552
```

```r
#The MSE is very large. Thus, we can say that the model does not fit well

summary(gamModel)
```

```
##
## Call: gam(formula = Outstate ~ Private + s(Apps, 5) + s(Accept, 5) +
##     s(Enroll, 5) + s(Top10perc, 5) + s(Room.Board, 5) + s(S.F.Ratio,
##     5) + s(Personal, 5) + s(PhD, 5) + s(Terminal, 5) + s(perc.alumni,
##     5) + s(perc.alumni, 5) + s(Expend, 5) + s(Grad.Rate, 5),
##     data = college_Train_set)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -6565.28 -1002.49    67.75  1010.05  6823.67
##
## (Dispersion Parameter for gaussian family taken to be 2971188)
##
##     Null Deviance: 10717168763 on 659 degrees of freedom
## Residual Deviance: 1776769454 on 597.9997 degrees of freedom
## AIC: 11770.84
##
## Number of Local Scoring Iterations: NA
##
## Anova for Parametric Effects
##                    Df     Sum Sq    Mean Sq   F value    Pr(>F)
## Private             1 3079550659 3079550659 1036.4712 < 2.2e-16 ***
## s(Apps, 5)          1 1245904769 1245904769  419.3288 < 2.2e-16 ***
## s(Accept, 5)        1  157987146  157987146   53.1731 9.728e-13 ***
## s(Enroll, 5)        1  319124384  319124384  107.4063 < 2.2e-16 ***
## s(Top10perc, 5)     1 1082354468 1082354468  364.2834 < 2.2e-16 ***
## s(Room.Board, 5)    1  652594046  652594046  219.6408 < 2.2e-16 ***
## s(S.F.Ratio, 5)     1  143817296  143817296   48.4040 9.134e-12 ***
## s(Personal, 5)      1   37656838   37656838   12.6740 0.0004002 ***
## s(PhD, 5)           1  114187432  114187432   38.4316 1.057e-09 ***
## s(Terminal, 5)      1   22819469   22819469    7.6803 0.0057565 **
## s(perc.alumni, 5)   1  152708726  152708726   51.3965 2.235e-12 ***
## s(Expend, 5)        1  470348201  470348201  158.3031 < 2.2e-16 ***
## s(Grad.Rate, 5)     1   62752316   62752316   21.1203 5.264e-06 ***
## Residuals         598 1776769454    2971188
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                   Npar Df  Npar F     Pr(F)
## (Intercept)
## Private
## s(Apps, 5)              4  2.0041  0.092446 .
## s(Accept, 5)            4 12.1481 1.681e-09 ***
## s(Enroll, 5)            4  4.6803  0.001001 **
## s(Top10perc, 5)         4  1.4248  0.224224
## s(Room.Board, 5)        4  0.9934  0.410508
## s(S.F.Ratio, 5)         4  3.8589  0.004183 **
## s(Personal, 5)          4  3.4789  0.008036 **
## s(PhD, 5)               4  2.1198  0.076926 .
```

```
## s(Terminal, 5)          4  1.8710  0.113925
## s(perc.alumni, 5)       4  2.1700  0.070984 .
## s(Expend, 5)            4 22.6423 < 2.2e-16 ***
## s(Grad.Rate, 5)         4  2.2664  0.060790 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Variables Expend and Accept have lower p-value:non-linear relationship
#with the response variable.
# Variables Top10perc and perc.alumni have higher p-values:
# linear relationship with the response variable.
```

Question 3 (based on JWHT Chapter 7, Problem 6)

In this exercise, you will further analyze the `Wage` data set.

(a) Perform polynomial regression to predict `wage` using `age.` Use cross-validation to select the optimal degree d for the polynomial. What degree was chosen? Make a plot of the resulting polynomial fit to the data.
(b) Fit a step function to predict `wage` using `age`, and perform cross-validation to choose the optimal number of cuts. Make a plot of the fit obtained.

```r
# Enter your R code here!
#(a)
attach(Wage)
summary(Wage)
```

```
##       year           age                     maritl           race     
##  Min.   :2003   Min.   :18.00   1. Never Married: 648   1. White:2480  
##  1st Qu.:2004   1st Qu.:33.75   2. Married      :2074   2. Black: 293  
##  Median :2006   Median :42.00   3. Widowed      :  19   3. Asian: 190  
##  Mean   :2006   Mean   :42.41   4. Divorced     : 204   4. Other:  37  
##  3rd Qu.:2008   3rd Qu.:51.00   5. Separated    :  55                  
##  Max.   :2009   Max.   :80.00                                          
##                                                                        
##               education                     region               jobclass   
##  1. < HS Grad      :268   2. Middle Atlantic   :3000   1. Industrial :1544  
##  2. HS Grad        :971   1. New England       :   0   2. Information:1456  
##  3. Some College   :650   3. East North Central:   0                        
##  4. College Grad   :685   4. West North Central:   0                        
##  5. Advanced Degree:426   5. South Atlantic    :   0                        
##                           6. East South Central:   0                        
##                           (Other)              :   0                        
##             health      health_ins      logwage           wage       
##  1. <=Good     : 858   1. Yes:2083   Min.   :3.000   Min.   : 20.09  
##  2. >=Very Good:2142   2. No : 917   1st Qu.:4.447   1st Qu.: 85.38  
##                                      Median :4.653   Median :104.92  
##                                      Mean   :4.654   Mean   :111.70  
##                                      3rd Qu.:4.857   3rd Qu.:128.68  
##                                      Max.   :5.763   Max.   :318.34  
## 
```

```r
summary(Wage$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   33.75   42.00   42.41   51.00   80.00
```

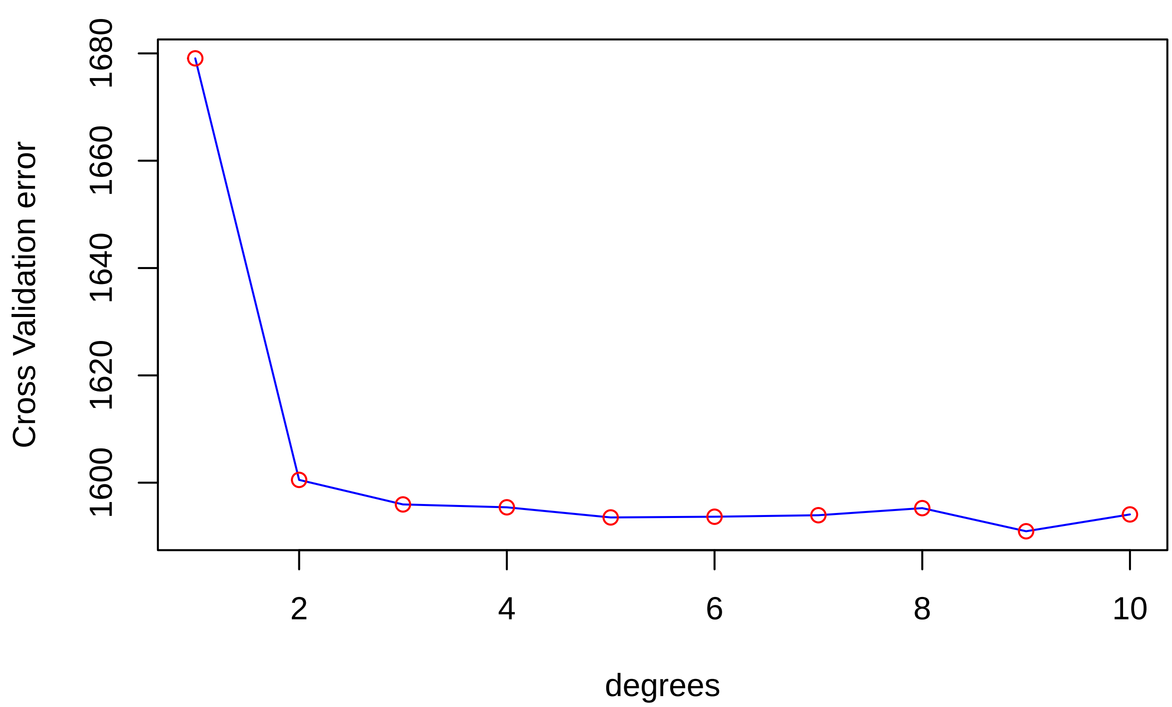
```r
summary(Wage$wage)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   20.09   85.38  104.92  111.70  128.68  318.34
```

```r
d <- NULL
for(i in 1:10)
{
 Wage.Model <- glm(formula = wage ~ poly(age, i), data = Wage)
 d[i] <- cv.glm(Wage, Wage.Model, K=10)$delta[2]
}
d
```

```
##  [1] 1679.074 1600.526 1595.950 1595.417 1593.529 1593.663 1593.941 1595.261
##  [9] 1590.955 1594.088
```

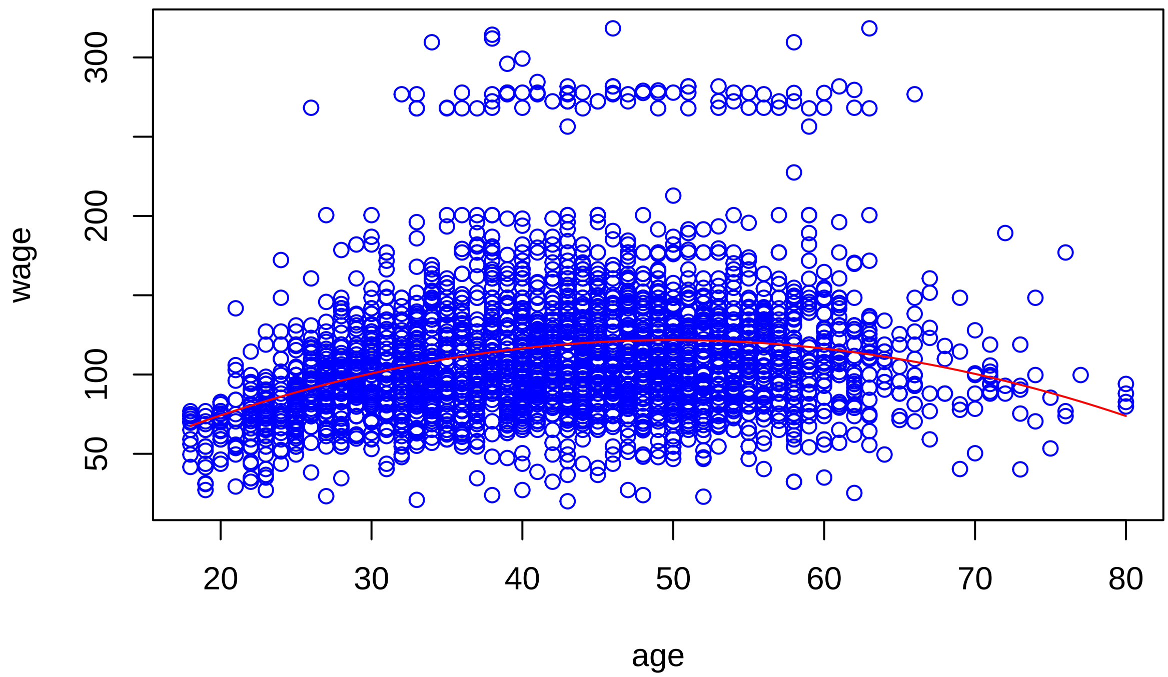
```r
plot(x=c(1:10), y=d, type = "l",xlab = "degrees",
 ylab = "Cross Validation error", col="blue")
points(d, col="red")
```



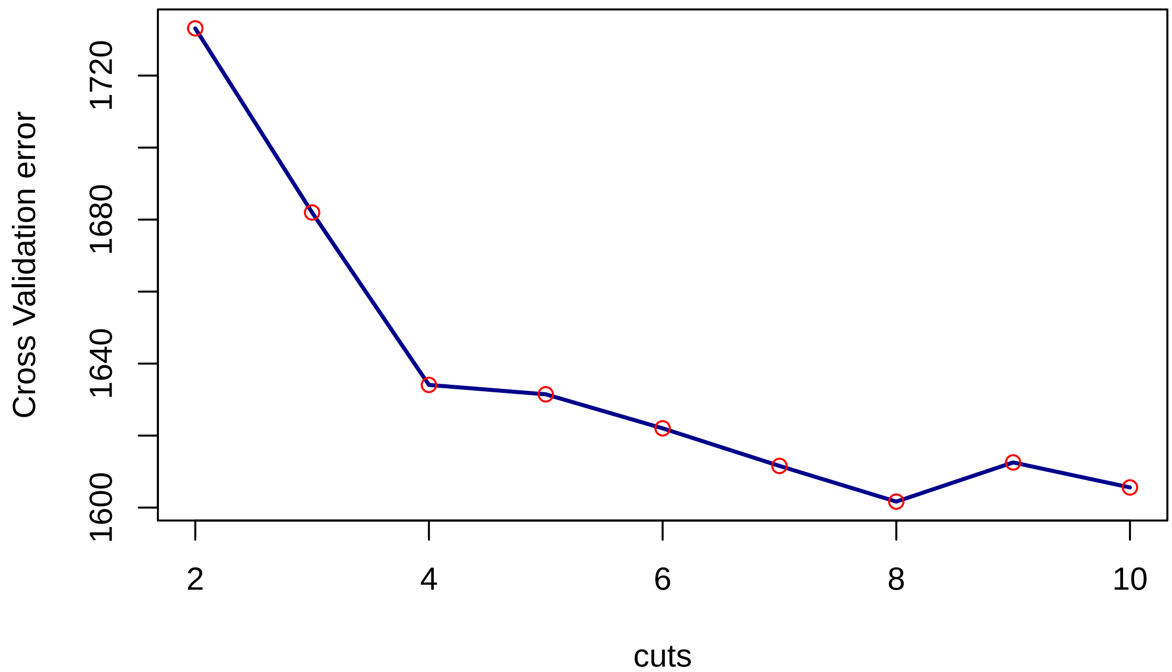
```r
# The optimal degree can be d = 3.

fit1 = lm(wage~poly(age, 1), data=Wage)
fit2 = lm(wage~poly(age, 2), data=Wage)
fit3 = lm(wage~poly(age, 3), data=Wage)
fit4 = lm(wage~poly(age, 4), data=Wage)
fit5 = lm(wage~poly(age, 5), data=Wage)
fit6 = lm(wage~poly(age, 6), data=Wage)
fit7 = lm(wage~poly(age, 7), data=Wage)
fit8 = lm(wage~poly(age, 8), data=Wage)
fit9 = lm(wage~poly(age, 9), data=Wage)
fit10 = lm(wage~poly(age, 10), data=Wage)
anova(fit1, fit2, fit3, fit4, fit5, fit6, fit7, fit8, fit9, fit10)
```

```
## Analysis of Variance Table
##
## Model  1: wage ~ poly(age, 1)
## Model  2: wage ~ poly(age, 2)
## Model  3: wage ~ poly(age, 3)
## Model  4: wage ~ poly(age, 4)
## Model  5: wage ~ poly(age, 5)
## Model  6: wage ~ poly(age, 6)
## Model  7: wage ~ poly(age, 7)
## Model  8: wage ~ poly(age, 8)
## Model  9: wage ~ poly(age, 9)
## Model 10: wage ~ poly(age, 10)
##    Res.Df     RSS Df Sum of Sq        F    Pr(>F)
## 1    2998 5022216
## 2    2997 4793430  1    228786 143.7638 < 2.2e-16 ***
## 3    2996 4777674  1     15756   9.9005  0.001669 **
## 4    2995 4771604  1      6070   3.8143  0.050909 .
## 5    2994 4770322  1      1283   0.8059  0.369398
## 6    2993 4766389  1      3932   2.4709  0.116074
## 7    2992 4763834  1      2555   1.6057  0.205199
## 8    2991 4763707  1       127   0.0796  0.777865
## 9    2990 4756703  1      7004   4.4014  0.035994 *
## 10   2989 4756701  1         3   0.0017  0.967529
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# According to Anova polynomials above degree 2 are insignificant.
# Hence, we take degree 2 as the optimal degree.

plot(wage~age, data=Wage, col="blue")
age.grid <- seq(18,80)
lm.fit <- fit2
lm.pred <- predict(lm.fit, data.frame(age=age.grid))
lines(age.grid, lm.pred, col="red")
```



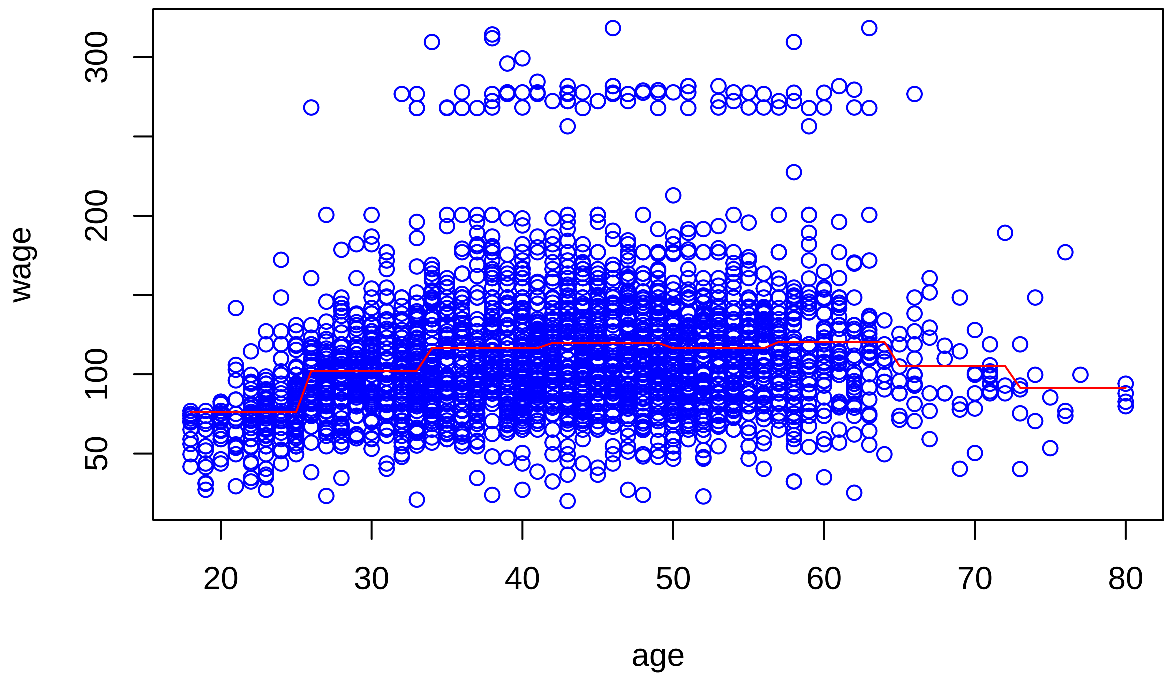

```
#(b)

errors <- NULL
for(i in 2:10){
 Wage$ageCuts <- cut(age,i)
 Wage.Model_1 <- glm(wage ~ ageCuts,data = Wage)
 errors[i] <- cv.glm(Wage,Wage.Model_1,K=10)$delta[2]
}
error_1 <- errors[-1]
plot(2:10, error_1, type = "l", lwd = 2, col = "dark blue",
 xlab = "cuts", ylab = "Cross Validation error")
points(errors, col="red")
```




```
#Minimum error gives optimal number of cuts.

min <- which.min(errors)
min
```

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

```
#Therefore, optimal number of cuts is equal to 8.

model <- glm(wage ~ cut(age, min),data = Wage)
prediction <- predict(model,
 newdata = data.frame(age = range(age)[1]:range(age)[2]))
plot(wage ~ age, data=Wage, col= "blue")
lines(18:80, prediction, col="red")
```



Question 4 (based on JWHT Chapter 8, Problem 8)

In the lab, a classification tree was applied to the `Carseats` data set after converting Sales into a qualitative response variable. Now we will seek to predict Sales using regression trees and related approaches, treating the response as a quantitative variable.

(a) Split the data set into a training set and a test set.
(b) Fit a regression tree to the training set. Plot the tree, and interpret the results. What test MSE do you obtain?
(c) Use cross-validation in order to determine the optimal level of tree complexity. Does pruning the tree improve the test MSE?
(d) Use the bagging approach in order to analyze this data. What test MSE do you obtain? Use the `importance()` function to determine which variables are most important.

```r
# Enter your R code here!
#(a)
attach(Carseats)
summary(Carseats)
```

```
##      Sales          CompPrice       Income        Advertising
##  Min.   : 0.000   Min.   : 77   Min.   : 21.00   Min.   : 0.000
##  1st Qu.: 5.390   1st Qu.:115   1st Qu.: 42.75   1st Qu.: 0.000
##  Median : 7.490   Median :125   Median : 69.00   Median : 5.000
```

```
##  Mean   : 7.496   Mean   :125   Mean   : 68.66   Mean   : 6.635
##  3rd Qu.: 9.320   3rd Qu.:135   3rd Qu.: 91.00   3rd Qu.:12.000
##  Max.   :16.270   Max.   :175   Max.   :120.00   Max.   :29.000
##    Population        Price        ShelveLoc        Age          Education
##  Min.   : 10.0   Min.   : 24.0   Bad   : 96   Min.   :25.00   Min.   :10.0
##  1st Qu.:139.0   1st Qu.:100.0   Good  : 85   1st Qu.:39.75   1st Qu.:12.0
##  Median :272.0   Median :117.0   Medium:219   Median :54.50   Median :14.0
##  Mean   :264.8   Mean   :115.8                Mean   :53.32   Mean   :13.9
##  3rd Qu.:398.5   3rd Qu.:131.0                3rd Qu.:66.00   3rd Qu.:16.0
##  Max.   :509.0   Max.   :191.0                Max.   :80.00   Max.   :18.0
##  Urban       US
##  No :118   No :142
##  Yes:282   Yes:258
##
##
##
##
```

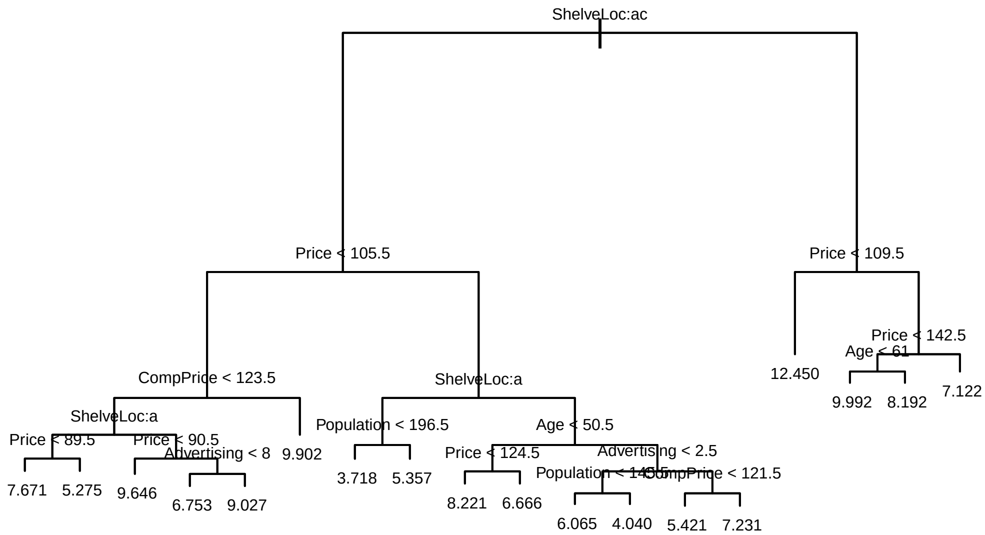
```
set.seed(personal)
samples <- sample(1:400, 320)
training_set <- Carseats[samples,]
testing_set <- Carseats[-samples,]
#(b)
library(tree)
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 4.1.2
```

```
treeCarseats_model <- tree(Sales ~ ., data = training_set,
 method = "recursive.partition",
 split = c("deviance", "gini"),
 model = TRUE)
summary(treeCarseats_model)
```

```
##
## Regression tree:
## tree(formula = Sales ~ ., data = training_set, method = "recursive.partition",
##     split = c("deviance", "gini"), model = TRUE)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "CompPrice"   "Advertising" "Population"
## [6] "Age"
## Number of terminal nodes:  18
## Residual mean deviance:  2.578 = 778.6 / 302
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -4.04000 -1.09200 -0.04444  0.00000  0.96080  4.78300
```

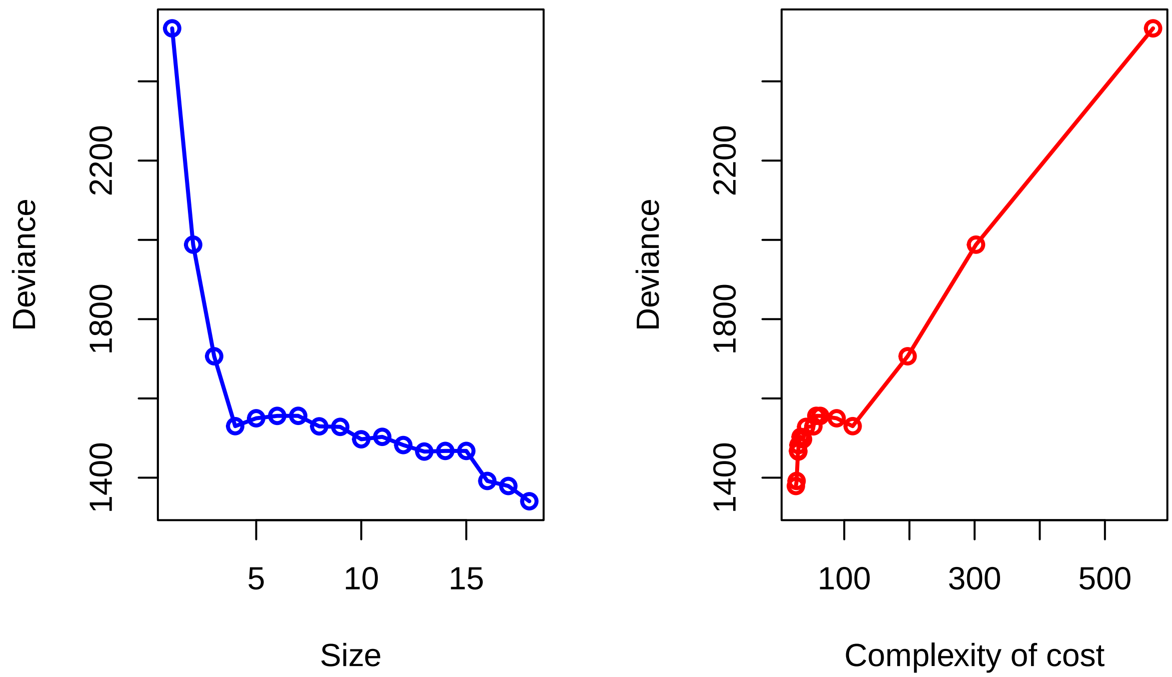
```
plot(treeCarseats_model)
text(treeCarseats_model, cex=0.51)
```




```
predCarseats <- predict(treeCarseats_model, testing_set)

#Mean Square Error(MSE):
mean((testing_set$Sales - predCarseats)^2)
```

```
## [1] 4.503883
```

```
#(c)
treeCV <- cv.tree(treeCarseats_model, FUN = prune.tree)
par(mfrow = c(1, 2))
plot(treeCV$size, treeCV$dev, type = "o", col = "blue",
 lwd = 2, xlab = "Size", ylab = "Deviance")
plot(treeCV$k, treeCV$dev, type = "o", col = "red",
 lwd = 2,xlab = "Complexity of cost",ylab = "Deviance")
```




```r
names(treeCV)
```

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

```r
#We check the deviance in treeCV

treeCV$dev
```

```
##  [1] 1340.622 1379.076 1391.768 1467.524 1467.524 1465.955 1482.345 1503.073
##  [9] 1496.949 1528.218 1529.609 1556.078 1555.966 1549.928 1530.061 1706.383
## [17] 1987.944 2533.644
```

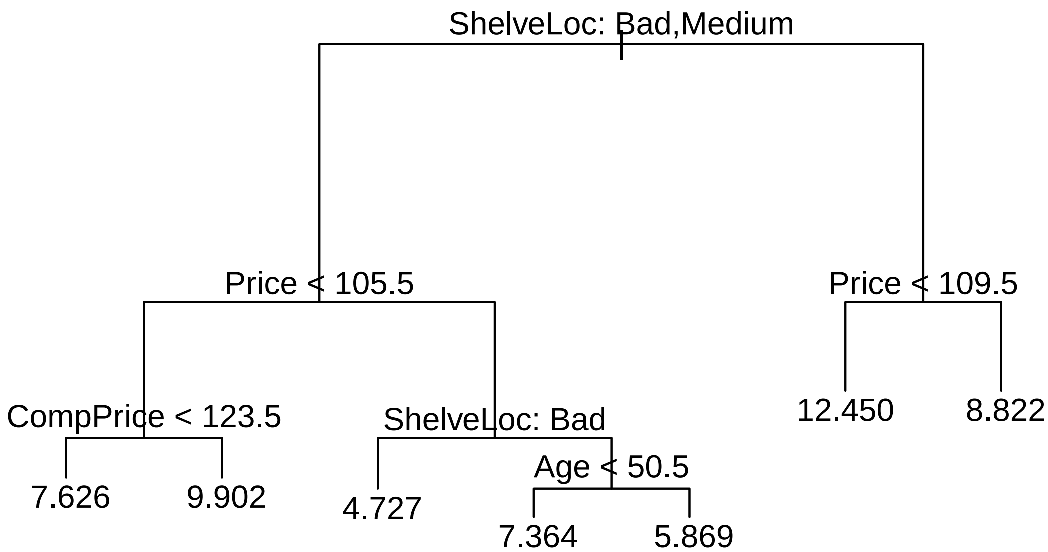
```r
m <- which.min(treeCV$dev)
m
```

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

```r
# we choose 1 due to minimum deviance.
# cheacking size at 1st position

treeCV$size[1]
```

```
## [1] 18
```

```r
prunedCarseats = prune.tree(treeCarseats_model, best = 7)
par(mfrow = c(1, 1))
plot(prunedCarseats)
text(prunedCarseats, pretty = 0)
```




```r
prediction <- predict(prunedCarseats, testing_set)

#Pruning Tree Mean Square Error(MSE).

mean((testing_set$Sales - prediction)^2)
```

```
## [1] 5.170744
```

```r
#no improvement

mean((testing_set$Sales - prediction)^2) #MSE with pruning
```

```
## [1] 5.170744
```

```r
mean((testing_set$Sales - predCarseats)^2) #Test MSE
```

```
## [1] 4.503883
```

```r
#(d)
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.1.2

## randomForest 4.7-1

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

```r
set.seed(personal)
bagCarseats <- randomForest(Sales ~ ., data = training_set, mtry = 10, ntree = 500,
importance = T)
bagPred <- predict(bagCarseats, testing_set)
mean((testing_set$Sales - bagPred)^2)
```

```
## [1] 2.772075
```

```r
#Bagging improves the test MSE to 2.772075.

importance(bagCarseats)
```

```
##                %IncMSE IncNodePurity
## CompPrice   32.6511473     260.36968
## Income       5.0551090     121.63439
## Advertising 17.9361347     151.68827
## Population   0.7769228      84.12906
## Price       82.5727763     791.06098
## ShelveLoc   80.3810590     728.36347
## Age         21.5321298     231.89760
## Education    2.3834524      65.76321
## Urban       -1.9221052      10.07288
## US           4.5113501      15.70894
```

```r
#Price, ShelveLoc and CompPrice are three most important predictors of Sale.
```

Question 5 (based on JWTH Chapter 8, Problem 10)

Use boosting (and bagging) to predict Salary in the Hitters data set

(a) Remove the observations for which salary is unknown, and then log-transform the salaries
(b) Split the data into training and testing sets for cross validation purposes.
(c) Perform boosting on the training set with 1000 trees for a range of values of the shrinkage parameter λ. Produce a plot with different shrinkage parameters on the x-axis and the corresponding training set MSE on the y-axis
(d) Produce a plot similar to the last one, but this time using the test set MSE
(e) Fit the model using two other regression techniques (from previous classes) and compare the MSE of those techniques to the results of these boosted trees.
(f) Reproduce (c) and (d), but this time use bagging instead of boosting and compare to the boosted MSE's and the MSE's from (e)

```r
# Enter your R code here!
#for this code, I had to import the data set from the internet since the one that is Installed in R is 

#(a)
#library(ISLR2)
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.1.2
```

```r
Hitters <- read_csv("Hitters.csv")
```

```
## Rows: 322 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (17): AtBat, Hits, HmRun, Runs, RBI, Walks, Years, CAtBat, CHits, CHmRun...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
summary(Hitters)
```

```
##      AtBat            Hits          HmRun            Runs       
##  Min.   : 16.0   Min.   :  1   Min.   : 0.00   Min.   :  0.00  
##  1st Qu.:255.2   1st Qu.: 64   1st Qu.: 4.00   1st Qu.: 30.25  
##  Median :379.5   Median : 96   Median : 8.00   Median : 48.00  
##  Mean   :380.9   Mean   :101   Mean   :10.77   Mean   : 50.91  
##  3rd Qu.:512.0   3rd Qu.:137   3rd Qu.:16.00   3rd Qu.: 69.00  
##  Max.   :687.0   Max.   :238   Max.   :40.00   Max.   :130.00  
##                                                                
##       RBI             Walks            Years            CAtBat       
##  Min.   :  0.00   Min.   :  0.00   Min.   : 1.000   Min.   :   19.0  
##  1st Qu.: 28.00   1st Qu.: 22.00   1st Qu.: 4.000   1st Qu.:  816.8  
##  Median : 44.00   Median : 35.00   Median : 6.000   Median : 1928.0  
##  Mean   : 48.03   Mean   : 38.74   Mean   : 7.444   Mean   : 2648.7  
##  3rd Qu.: 64.75   3rd Qu.: 53.00   3rd Qu.:11.000   3rd Qu.: 3924.2  
##  Max.   :121.00   Max.   :105.00   Max.   :24.000   Max.   :14053.0  
##                                                                       
##      CHits            CHmRun           CRuns             CRBI        
##  Min.   :   4.0   Min.   :  0.00   Min.   :   1.0   Min.   :   0.00  
##  1st Qu.: 209.0   1st Qu.: 14.00   1st Qu.: 100.2   1st Qu.:  88.75  
##  Median : 508.0   Median : 37.50   Median : 247.0   Median : 220.50  
##  Mean   : 717.6   Mean   : 69.49   Mean   : 358.8   Mean   : 330.12  
##  3rd Qu.:1059.2   3rd Qu.: 90.00   3rd Qu.: 526.2   3rd Qu.: 426.25  
##  Max.   :4256.0   Max.   :548.00   Max.   :2165.0   Max.   :1659.00  
##                                                                       
##      CWalks            PutOuts          Assists          Errors     
##  Min.   :   0.00   Min.   :   0.0   Min.   :  0.0   Min.   : 0.00  
##  1st Qu.:  67.25   1st Qu.: 109.2   1st Qu.:  7.0   1st Qu.: 3.00  
##  Median : 170.50   Median : 212.0   Median : 39.5   Median : 6.00  
##  Mean   : 260.24   Mean   : 288.9   Mean   :106.9   Mean   : 8.04  
##  3rd Qu.: 339.25   3rd Qu.: 325.0   3rd Qu.:166.0   3rd Qu.:11.00  
##  Max.   :1566.00   Max.   :1378.0   Max.   :492.0   Max.   :32.00  
```

```
##
##       Salary
##   Min.   :  67.5
##   1st Qu.: 190.0
##   Median : 425.0
##   Mean   : 535.9
##   3rd Qu.: 750.0
##   Max.   :2460.0
##   NA's   :59
```

```
sum(is.na(Hitters$Salary))
```

```
## [1] 59
```

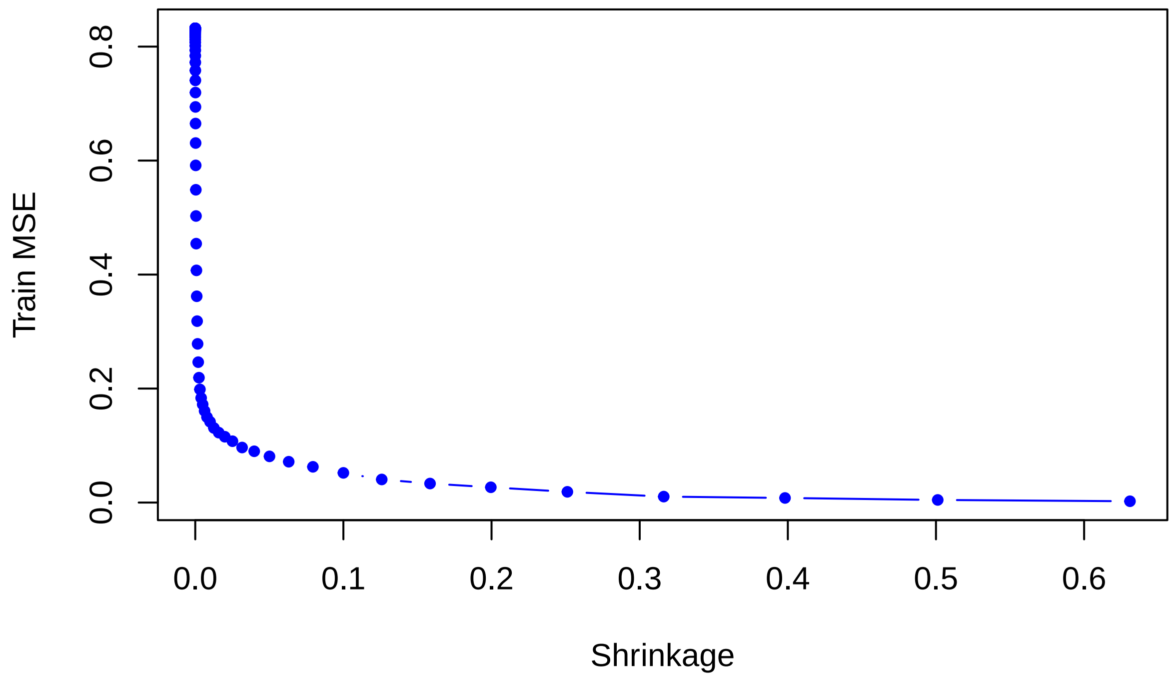
```
Hitters = Hitters[-which(is.na(Hitters$Salary)), ]
sum(is.na(Hitters$Salary))
```

```
## [1] 0
```

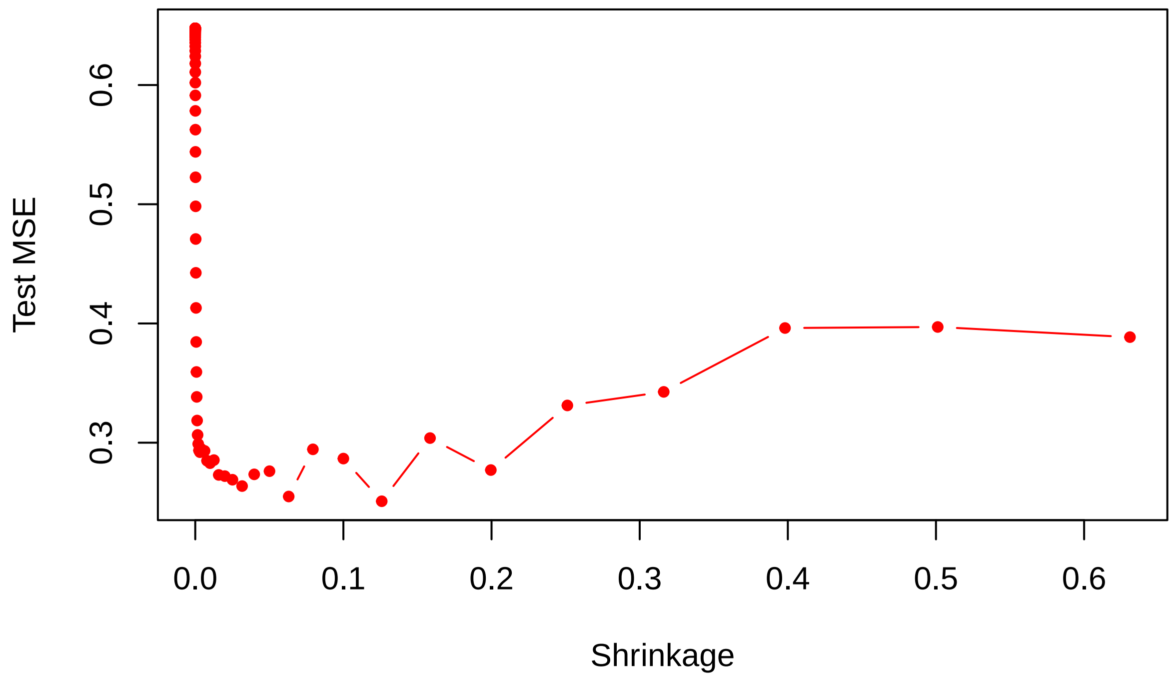
```
Hitters$Salary = log(Hitters$Salary)

#(b)
train_val = 1:200
Hitters.train_set = Hitters[train_val, ]
Hitters.test_set = Hitters[-train_val, ]

#(c)
library(gbm)
```

```
## Loaded gbm 2.1.8
```

```
pows = seq(-10, -0.2, by = 0.1)
lambdas = 10^pows
length.lambdas = length(lambdas)
train.e = rep(NA, length.lambdas)
test.e = rep(NA, length.lambdas)
for (i in 1:length.lambdas)
   {
 boost.hitters = gbm(Salary ~ ., data = Hitters.train_set, distribution ="gaussian", n.trees = 1000, sh
 train.p = predict(boost.hitters, Hitters.train_set, n.trees = 1000)
 test.p = predict(boost.hitters, Hitters.test_set, n.trees = 1000)
 train.e[i] = mean((Hitters.train_set$Salary - train.p)^2)
 test.e[i] = mean((Hitters.test_set$Salary - test.p)^2)
    }
plot(lambdas, train.e, type = "b", xlab = "Shrinkage", ylab = "Train MSE", col = "blue", pch = 20)
```




```
#(d)
plot(lambdas, test.e, type = "b", xlab = "Shrinkage", ylab = "Test MSE",
 col = "red", pch = 20)
```




```
min(test.e)
```

```
## [1] 0.2508828
```

```
lambdas[which.min(test.e)]
```

```
## [1] 0.1258925
```

```
#(e)
lm.fit_5 = lm(Salary ~ ., data = Hitters.train_set)
lm.pred_5 = predict(lm.fit_5, Hitters.test_set)
mean((Hitters.test_set$Salary - lm.pred_5)^2)
```

```
## [1] 0.5156972
```

```
library(glmnet)
```

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

```
## Warning: package 'Matrix' was built under R version 4.1.2
```

```
## Loaded glmnet 4.1-3
```

```r
x = model.matrix(Salary ~ ., data = Hitters.train_set)
y = Hitters.train_set$Salary
x.test = model.matrix(Salary ~ ., data = Hitters.test_set)
lasso.fit = glmnet(x, y, alpha = 1)
lasso.pred = predict(lasso.fit, s = 0.01, newx = x.test)
mean((Hitters.test_set$Salary - lasso.pred)^2)
```

```
## [1] 0.4838873
```

```r
#Both linear model and regularization like Lasso have higher test MSE than
#boosting.

#(f)

library(randomForest)
bag.hitters <- randomForest(Salary ~ ., data = Hitters.train_set)
mean((Hitters.test_set$Salary - predict(bag.hitters, Hitters.test_set))^2)
```

```
## [1] 0.2212287
```

```r
#The MSE of bagging is lower than boosting's
```