FE590. Assignment #3.

Kejia

2016-05-08

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.

Generate a simulated data set as follows:

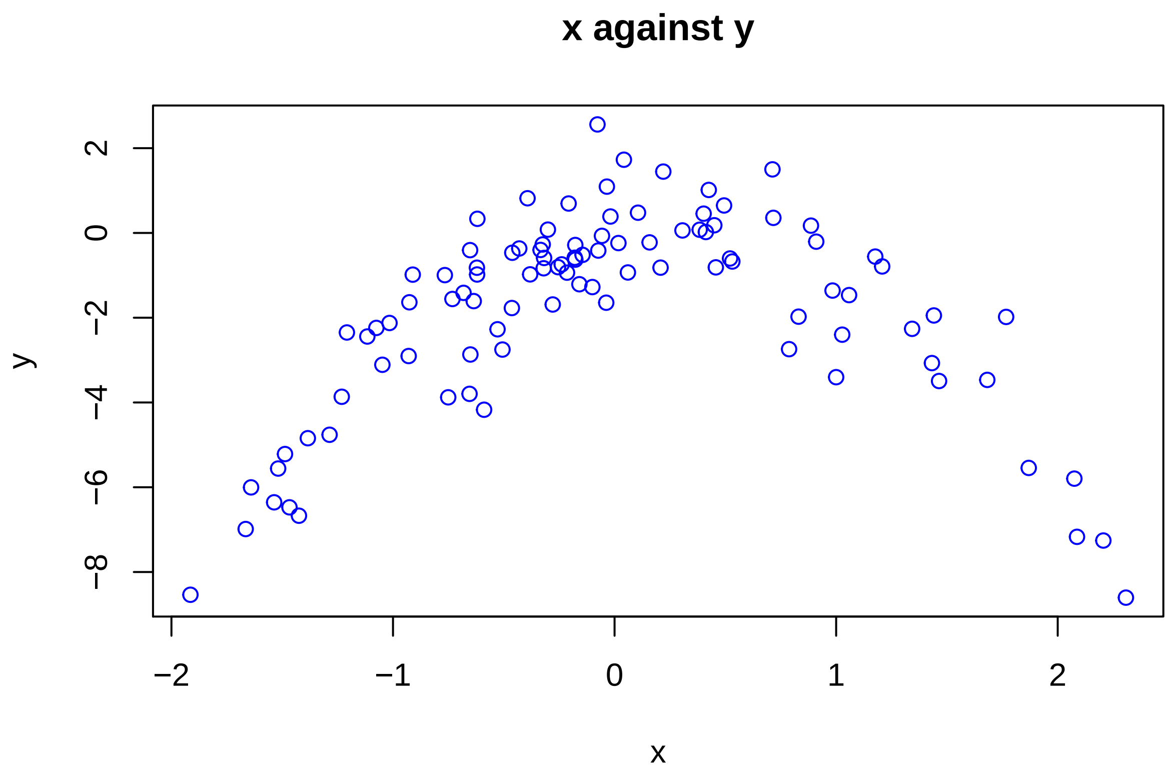
```
set.seed(1)
y <- rnorm(100)
x <- rnorm(100)
y <- x - 2*x^2 + rnorm(100)
```

(a) In this data set, what is n and what is p?

n is 100, and p is 1

(b) Create a scatterplot of x against y. Comment on what you find.

```
plot(x,y,type="p",main="x against y",col="blue",xlab="x",ylab="y")
```



It looks like a parabola

(c) Set a random seed of 2, and then 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. $Y = \beta_0 + \beta_1 X + +\beta_2 X^2 + \beta_3 X^3 + \beta_4 X^4 + \epsilon$

```
library(boot)
set.seed(2)
y <- rnorm(100)
x <- rnorm(100)
y <- x - 2*x^2 + rnorm(100)
data_1 <- data.frame(y = y,
                     x = x)
cv.error <- rep(0,4)
for(i in 1:4){
    glm.fit <- glm(y~poly(x,i),data=data_1)
    cv.error[i] <- cv.glm(data_1,glm.fit)$delta[1]
}
names(cv.error) <- c("poly=1","poly=2","poly=3","poly=4")
cv.error
```

```
##    poly=1    poly=2    poly=3    poly=4
```

```
## 6.140266 1.169795 1.191309 1.180095
```

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

```r
which.min(cv.error)
```

```
## poly=2
##      2
```

The second model has the smallest LOOCV error. This is what I expected, because I set $Y = X - 2X^2 + \epsilon$

(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?

Intercept coef.poly(x,1) coef.poly(x,2) coef.poly(x,3) poly=1 -1.751957 8.765237 NA NA poly=2 -1.751957 8.765237 -21.48335 NA poly=3 -1.751957 8.765237 -21.48335 0.2519422 poly=4 -1.751957 8.765237 -21.48335 0.2519422 coef.poly(x,4) poly=1 NA poly=2 NA poly=3 NA poly=4 1.758921 ##### When poly=3 or poly=4, the coefficient of x^3 and x^4 are very small. Results agree with the conclusions drawnbased on the cross-validation results.

Question 2 (based on JWHT Chapter 6, Problem 8)

In this exercise, we will generate simulated data, and will then use this data to perform best subset selection.

(a) Set the random seed to be 10. Use the `rnorm()` function to generate a predictor `X` of length `n = 100`, as well as a noise vector ϵ of length `n = 100`.

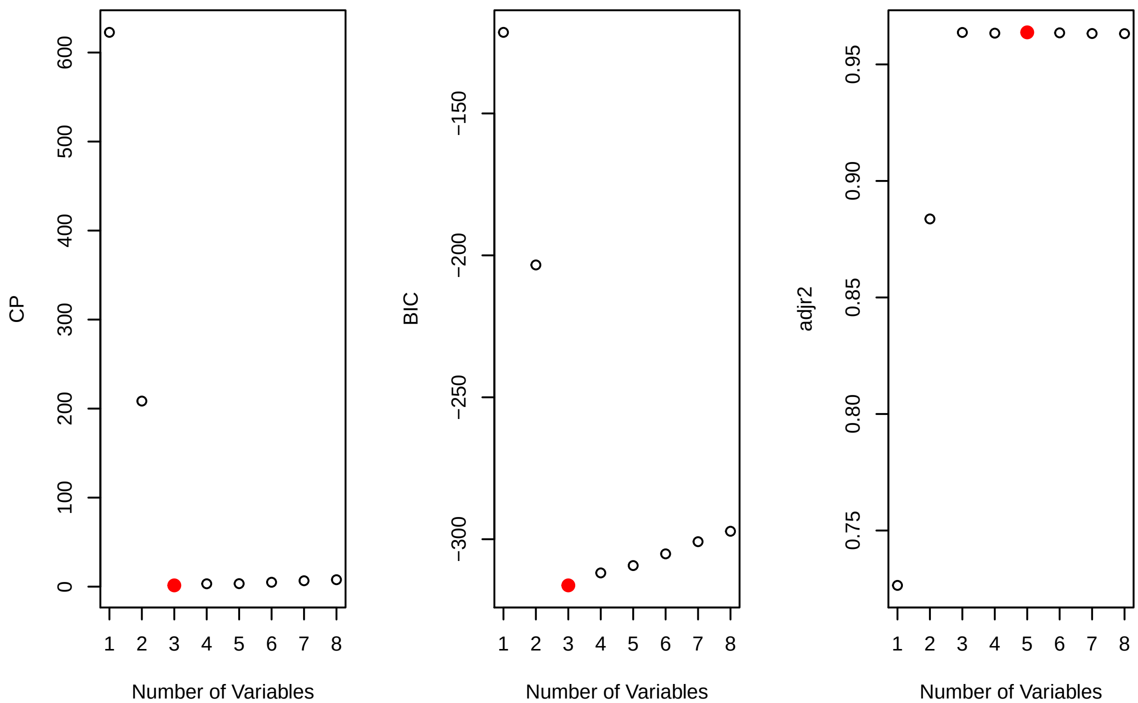
```r
set.seed(10)
x <- rnorm(100)
epsilon <- rnorm(100)
```

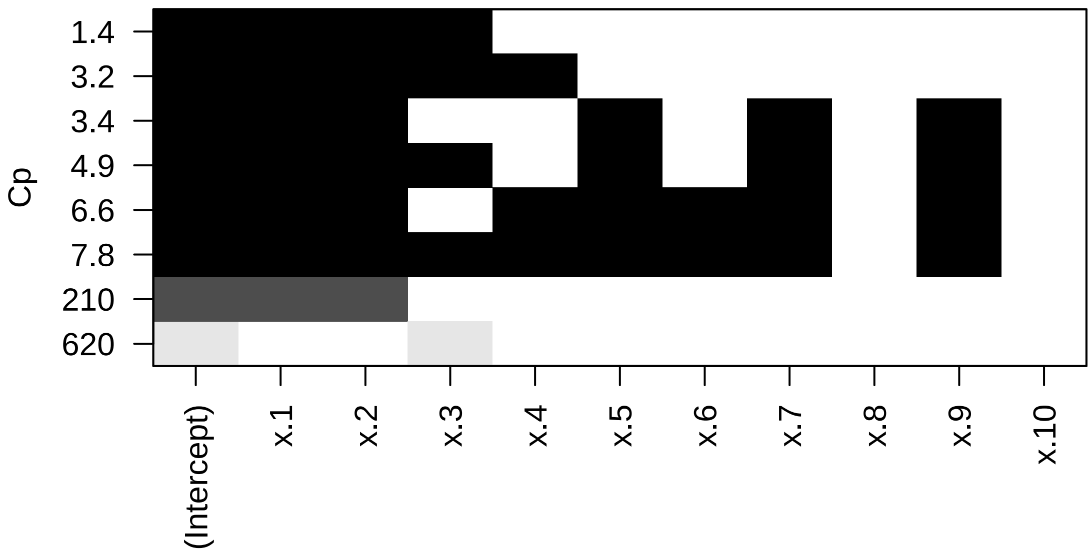
(b) Generate a response vector `Y` of length `n = 100` according to the model

$$Y = 4 + 3X + 2X^2 + X^3 + \epsilon.$$

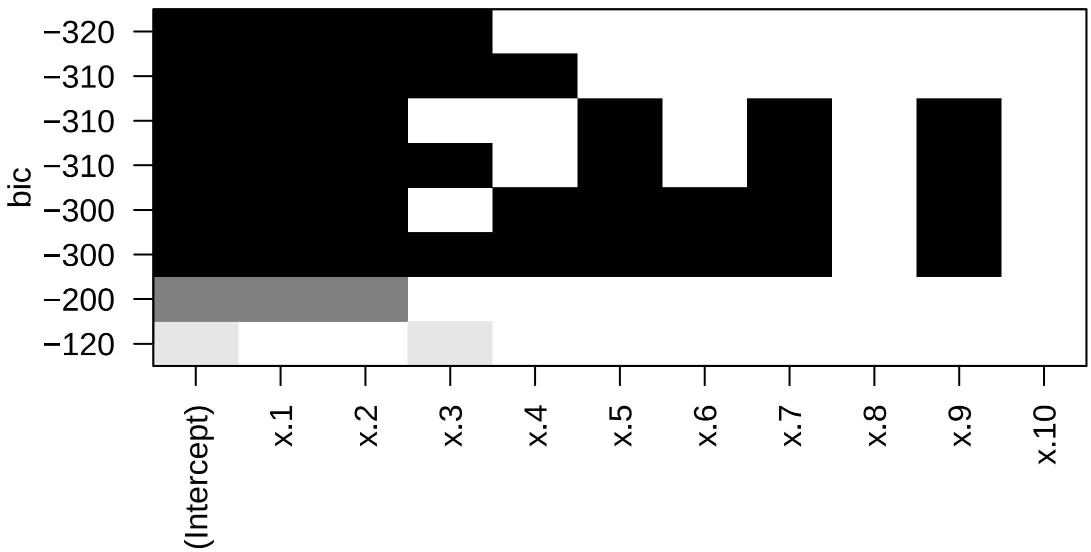
```r
y <- 4 + 3*x + 2*x^2 + x^3 + epsilon
```

(c) Use the `regsubsets()` function to perform best subset selection in order to choose the best model containing the predictors $X, X^2, \ldots, X^10$. What is the best model obtained according to C_p, BIC, and adjusted R^2? Show some plots to provide evidence for your answer, and report the coefficients of the best model obtained. Note you will need to use the data.frame() function to create a single data set containing both X and Y . [1] “which” “rsq” “rss” “adjr2” “cp” “bic” “outmat” “obj”

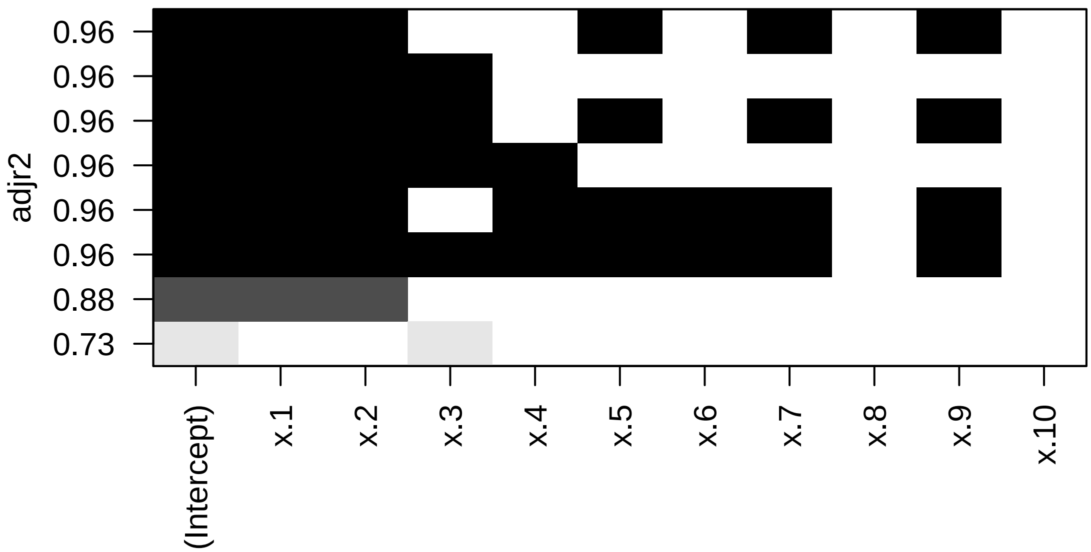




(Intercept) x.1 x.2 x.3 3.928974 2.884212 1.963622 1.021113



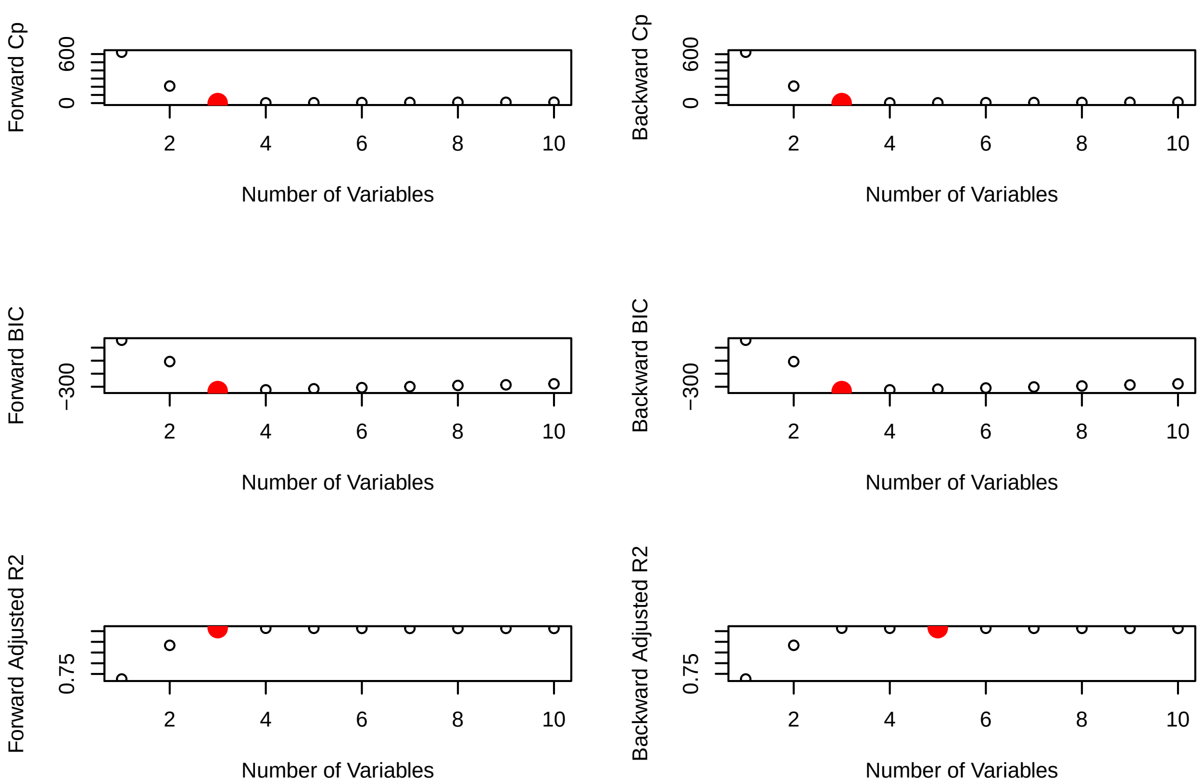
(Intercept) x.1 x.2 x.3 3.928974 2.884212 1.963622 1.021113



(Intercept) x.1 x.2 x.5 x.7 x.9 3.95409012 3.12434155 1.93068856 1.04218301 -0.36785066 0.04036764 (d) Repeat (c), using forward stepwise selection and also using backwards stepwise selection. How does your answer compare to the results in (c)?

```
forward  <- regsubsets(y ~ poly(x, 10, raw = T), data = data_2, nvmax = 10, method = "forward" )
backward <- regsubsets(y ~ poly(x, 10, raw = T), data = data_2, nvmax = 10, method = "backward")
forwardsummary  <- summary(forward)
backwardsummary <- summary(backward)

par(mfrow = c(3, 2))
plot(forwardsummary$cp,ylab = "Forward Cp",xlab="Number of Variables")
points(which.min(forwardsummary$cp),
       forwardsummary$cp[which.min(forwardsummary$cp)],
       pch = 20, col = "red", lwd = 7)

plot(backwardsummary$cp, ylab = "Backward Cp",xlab="Number of Variables")
points(which.min(backwardsummary$cp),
       backwardsummary$cp[which.min(backwardsummary$cp)],
       pch = 20, col = "red", lwd = 7)

plot(forwardsummary$bic, ylab = "Forward BIC",xlab="Number of Variables")
points(which.min(forwardsummary$bic),
       forwardsummary$bic[which.min(forwardsummary$bic)],
       pch = 20, col = "red", lwd = 7)

plot(backwardsummary$bic, ylab = "Backward BIC",xlab="Number of Variables")
points(which.min(backwardsummary$bic),
```

```
       backwardsummary$bic[which.min(backwardsummary$bic)],
       pch = 20, col = "red", lwd = 7)

plot(forwardsummary$adjr2, ylab = "Forward Adjusted R2",xlab="Number of Variables")
points(which.max(forwardsummary$adjr2),
       forwardsummary$adjr2[which.max(forwardsummary$adjr2)],
       pch = 20, col = "red", lwd = 7)

plot(backwardsummary$adjr2, ylab = "Backward Adjusted R2",xlab="Number of Variables")
points(which.max(backwardsummary$adjr2),
       backwardsummary$adjr2[which.max(backwardsummary$adjr2)],
       pch = 20, col = "red", lwd = 7)
```



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

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

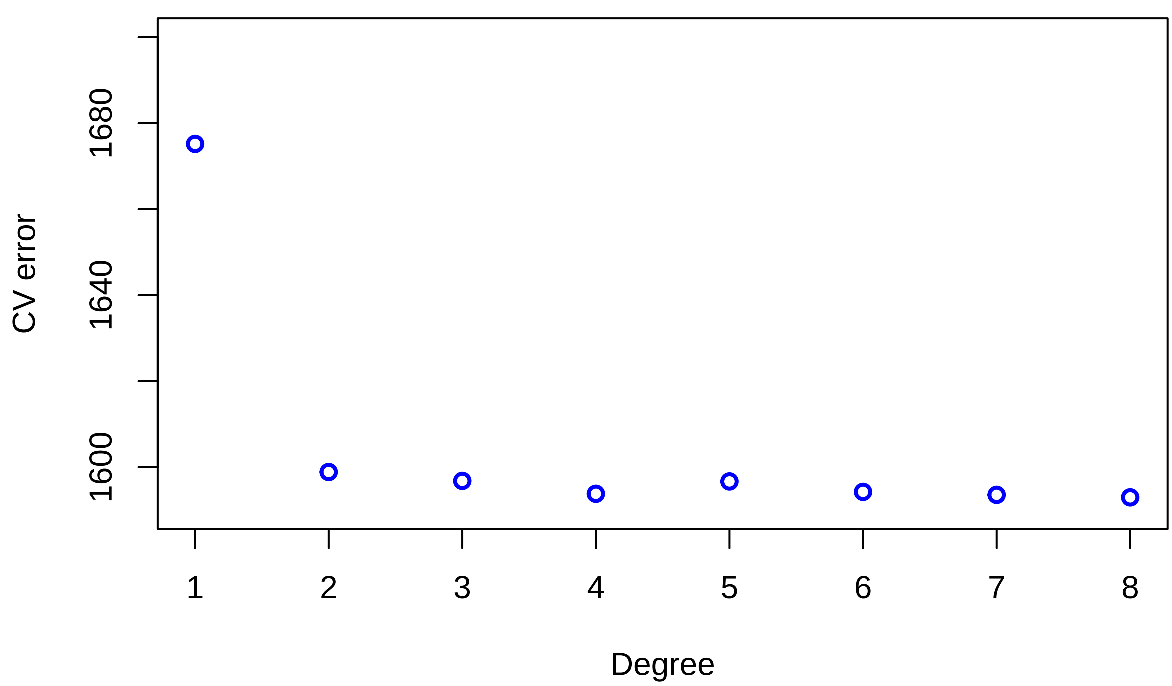
(a) Perform polynomial regression to predict `wage` using `age`. Use cross-validation to select the optimal degree d for the polynomial. What degree was chosen? Make a plot of the resulting polynomial fit to the data.

```
library(ISLR)
attach(Wage)
```

```r
fit <- lm(wage~poly(age,8),data=Wage)
coef(summary(fit))
```

```
##                  Estimate Std. Error     t value     Pr(>|t|)
## (Intercept)     111.70361  0.7286244 153.3075323 0.000000e+00
## poly(age, 8)1  447.06785 39.9084018  11.2023492 1.458528e-28
## poly(age, 8)2 -478.31581 39.9084018 -11.9853410 2.309469e-32
## poly(age, 8)3  125.52169 39.9084018   3.1452446 1.675770e-03
## poly(age, 8)4  -77.91118 39.9084018  -1.9522501 5.100170e-02
## poly(age, 8)5  -35.81289 39.9084018  -0.8973772 3.695899e-01
## poly(age, 8)6   62.70772 39.9084018   1.5712911 1.162208e-01
## poly(age, 8)7   50.54979 39.9084018   1.2666453 2.053808e-01
## poly(age, 8)8  -11.25473 39.9084018  -0.2820141 7.779522e-01
```

```r
# cross-validation
library(boot)
cv.error <- rep(NA, 8)
for (i in 1:8) {
  glm.fit <- glm(wage~poly(age, i), data=Wage)
  cv.error[i] = cv.glm(Wage, glm.fit, K=8)$delta[1]
}
par(mfrow = c(1, 1))
plot(1:8, cv.error, xlab="Degree", ylab="CV error", lwd=2, ylim=c(1590, 1700),col="blue")
```

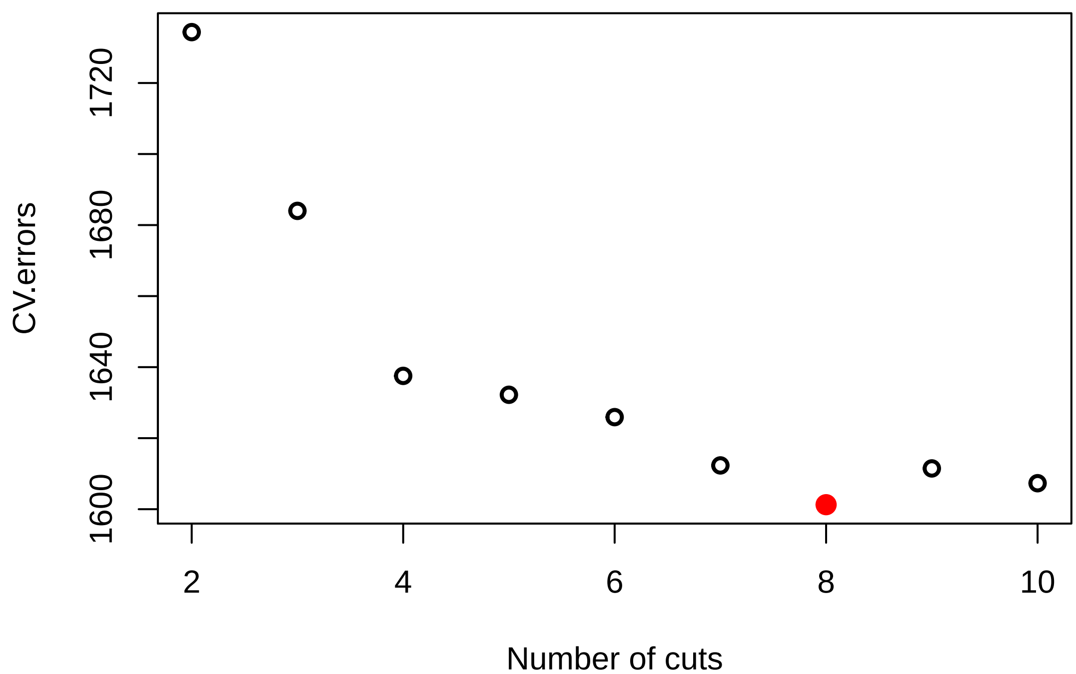


We can see that p-values from the table that there is a sharp drop in the estimated test MSE

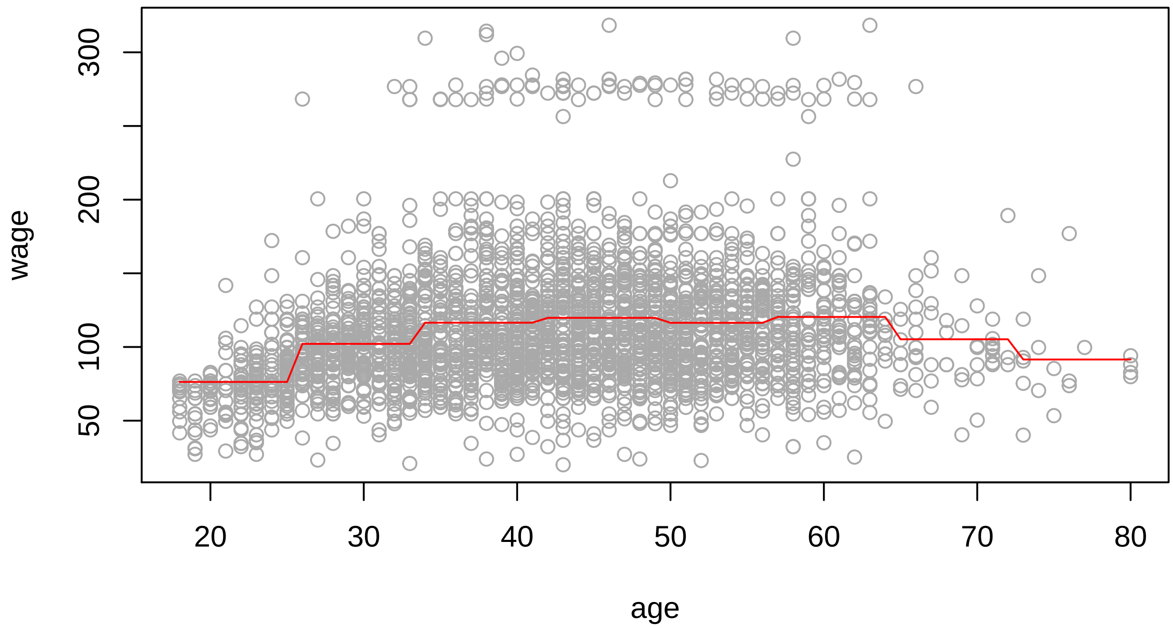
between the lnear and quadratic fits, but then no clear improvement from using higher-order polynomials. We choose degree two.

(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.

```
cv.errors <- rep(0, 9)
for (i in 2:10) {
  Wage$age.cut <- cut(Wage$age, i)
  glm.fit <- glm(wage~age.cut, data=Wage)
  cv.errors[i-1] <- cv.glm(Wage, glm.fit, K=10)$delta[1]
}
par(mfrow = c(1, 1))
plot(2:10, cv.errors, xlab="Number of cuts", ylab="CV.errors", lwd=2)
n <- which.min(cv.errors)
points(n+1,cv.errors[n],col="red", cex=2, pch=20)
```



So the optimal number of cuts is 8



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

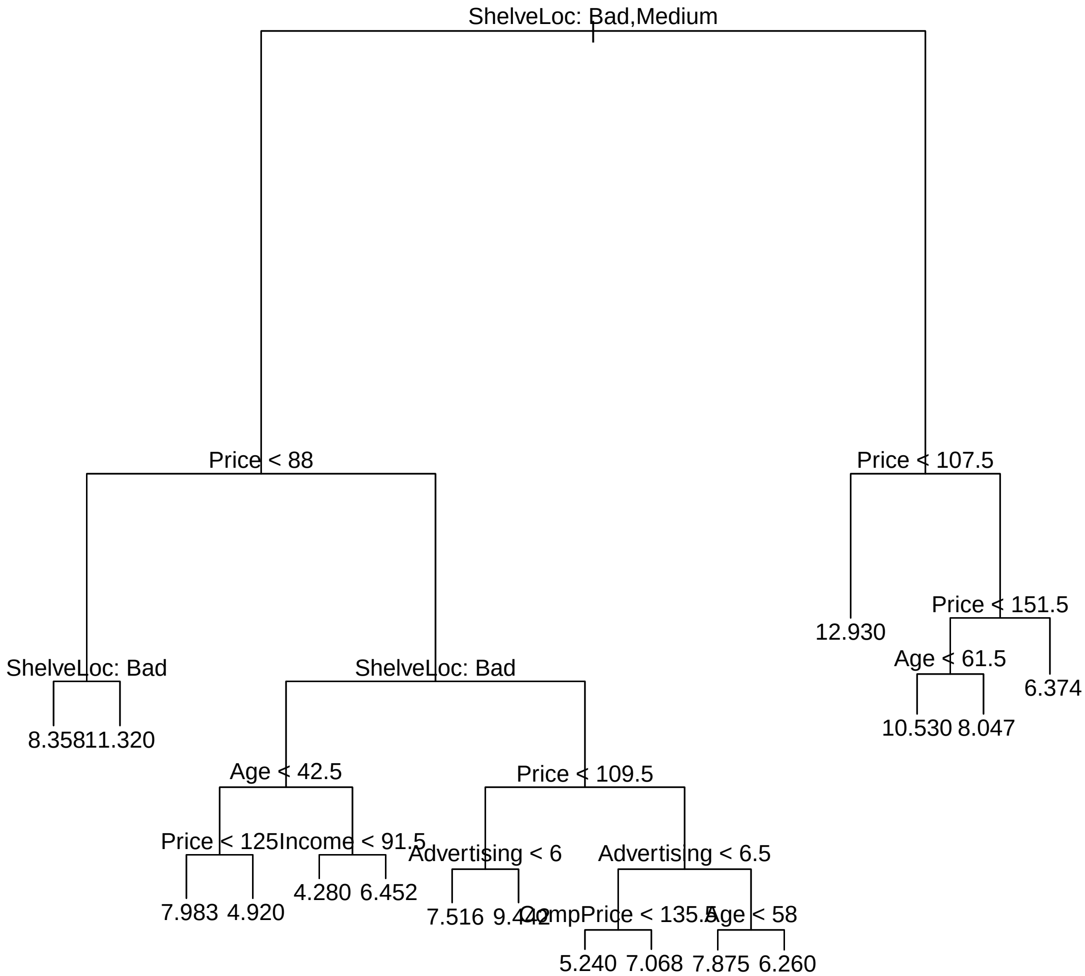
In the lab, a classification tree was applied to the `Carseats` data set after converting Sales into a qualitative response variable. Now we will seek to predict Sales using regression trees and related approaches, treating the response as a quantitative variable.

(a) Split the data set into a training set and a test set.

```
library(ISLR)
attach(Carseats)
set.seed(4)
Carseats.train <- sample(dim(Carseats)[1], dim(Carseats)[1]/2)
Carseats.test <- Carseats[-Carseats.train,]
```

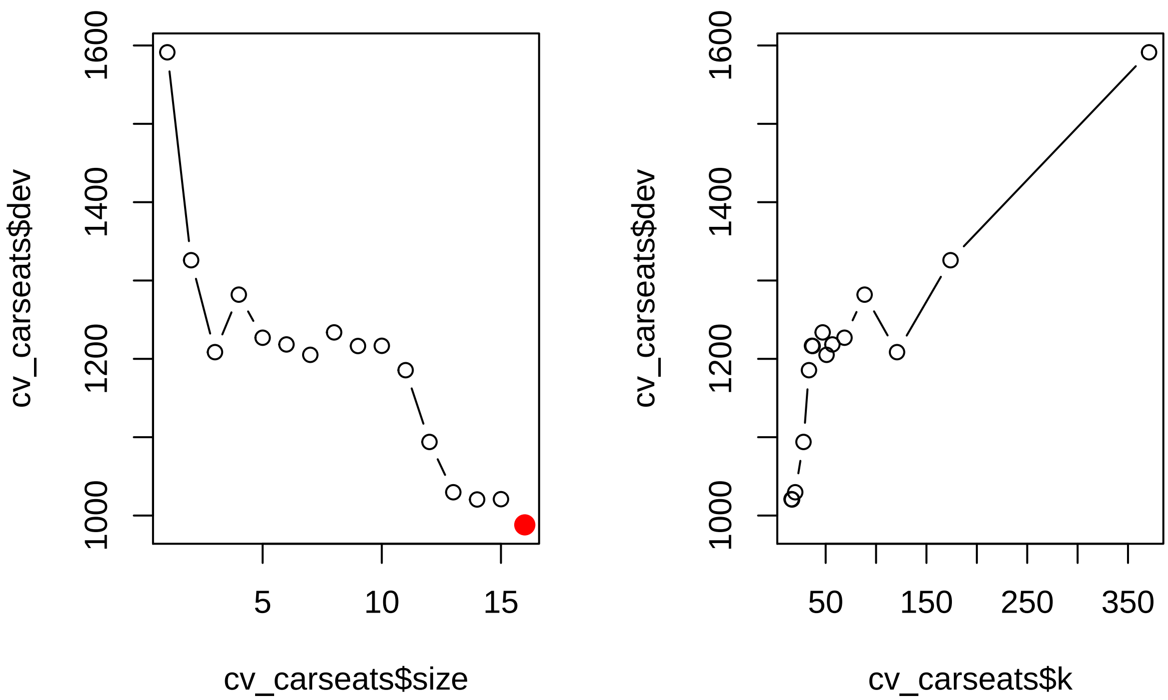
(b) Fit a regression tree to the training set. Plot the tree, and interpret the results. What test MSE do you obtain?

Regression tree: tree(formula = Sales ~ ., data = Carseats, subset = Carseats.train) Variables actually used in tree construction: [1] "ShelveLoc" "Price" "Age" "Income" "Advertising" [6] "CompPrice"
Number of terminal nodes: 16 Residual mean deviance: 2.241 = 412.3 / 184 Distribution of residuals: Min. 1st Qu. Median Mean 3rd Qu. Max. -4.88700 -0.99420 -0.02147 0.00000 1.10000 3.34200

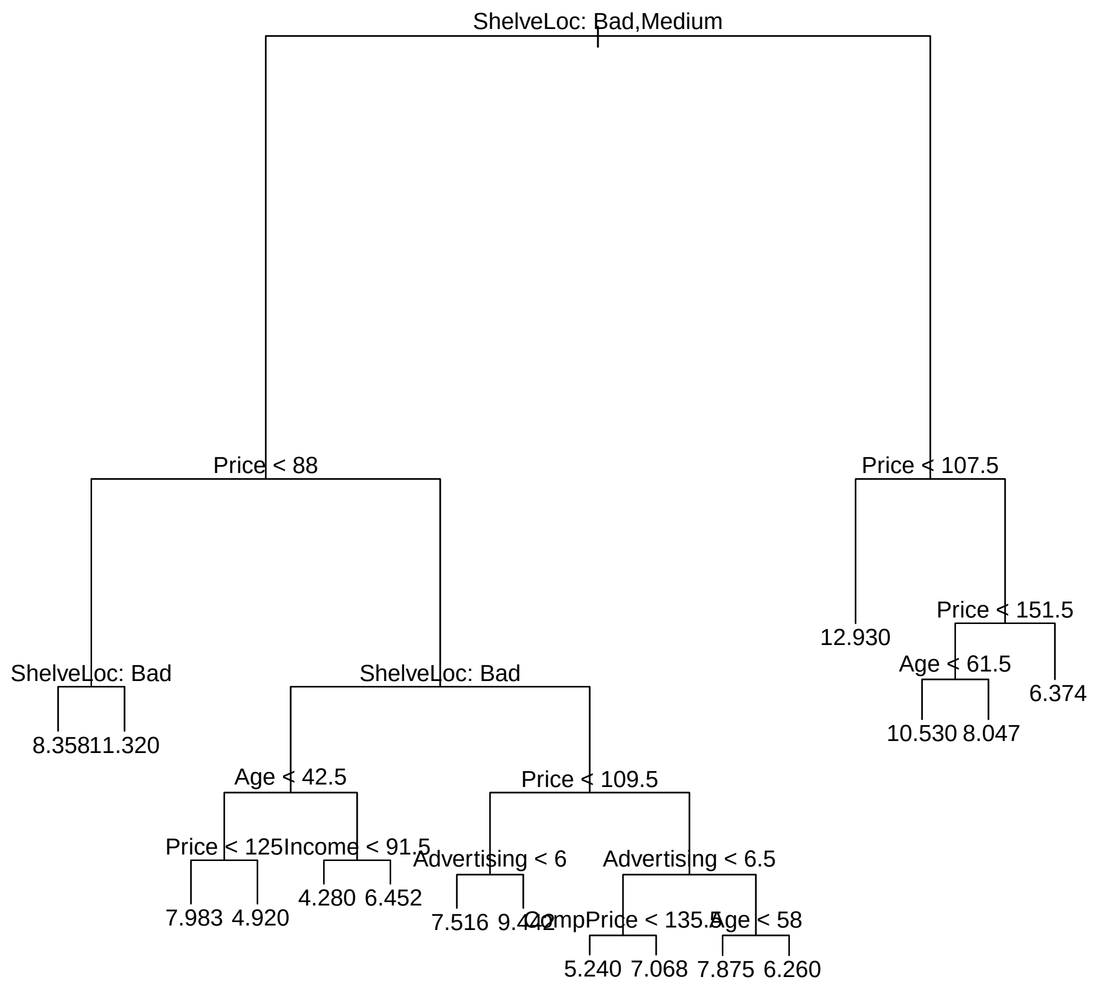


[1] 4.844354

(c) Use cross-validation in order to determine the optimal level of tree complexity. Does pruning the tree improve the test MSE?



cv_carseats$dev
1000
1200
1400
1600
5
10
15
cv_carseats$size
cv_carseats$dev
1000
1200
1400
1600
50
150
250
350
cv_carseats$k



[1] 4.844354 ##### The optimal level of tree complexity is 13

(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.

```
library(randomForest)
```

```
## randomForest 4.6-12
```

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

```
set.seed(6)
rf.carseats <- randomForest(Sales ~ .,Carseats, subset= Carseats.train, mtry = 10, ntree = 500, importa
```

```r
rf.pred <- predict(rf.carseats, Carseats.test)
mean((Carseats.test$Sales - rf.pred)^2)
```

```
## [1] 2.930925
```

```r
importance(rf.carseats)
```

```
##                %IncMSE IncNodePurity
## CompPrice   13.67358744    115.022728
## Income       9.31185846     97.133081
## Advertising 20.69244254    130.773750
## Population   0.08913758     50.639059
## Price       55.50946264    498.963448
## ShelveLoc   57.03057682    440.065309
## Age         15.83283290    158.948579
## Education    1.77320036     38.082431
## Urban        0.07844476      5.980857
## US           1.14084353      7.087098
```

We can see that `Price` and 'ShelveLOv' are the most important