FE590. Assignment #3.

Liting Hu

2017-04-14

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:

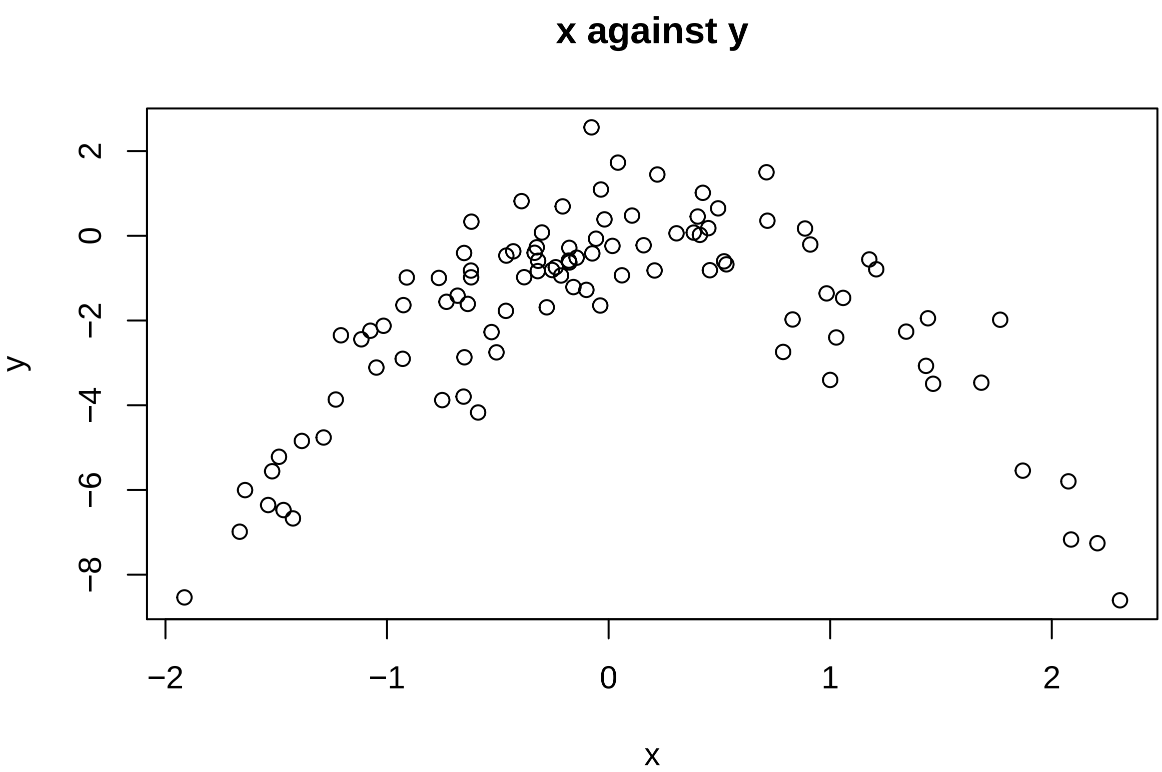
```
set.seed(1)
y <- rnorm(100)
x <- rnorm(100)
y <- x - 2*x^2 + rnorm(100)
```

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

** n is 100. 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")
```



This plot 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)
df <- data.frame(y, x)
cv.error <- seq(1:4)
for (i in cv.error) {
    glm.fit <- glm(y~poly(x, i), data = df)
    cv.error[i] <- cv.glm(df, glm.fit)$delta[1]
}
names(cv.error) <- c("poly=1","poly=2","poly=3","poly=4")
print(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.

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

```
## poly=2
```

```
## 2
```

The second model has the smallest LOOCV error, which is what I want. Because the original setting is $Y = X - 2X^2 + \epsilon$.

(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?

```
coefs <- as.data.frame(matrix(nrow = 4, ncol = 5))
for (i in 1:4) {
    glm.fit <- glm(y ~ poly(x, i), data = df)
    coefs[i, 1:(i+1)] <- glm.fit$coefficients
}
rownames(coefs) <- c("poly=1","poly=2","poly=3","poly=4")
colnames(coefs) <- c("Intercept","poly(x, i)1","poly(x, i)2","poly(x, i)3","poly(x, i)4")
```

```
knitr::kable(coefs)
```

	Intercept	poly(x, i)1	poly(x, i)2	poly(x, i)3	poly(x, i)4
poly=1	-1.751957	8.765237	NA	NA	NA
poly=2	-1.751957	8.765237	-21.48335	NA	NA
poly=3	-1.751957	8.765237	-21.48335	0.2519422	NA
poly=4	-1.751957	8.765237	-21.48335	0.2519422	1.758921

In cubic and quartic polynomial, the coefficients of x^3 and x^4 are relatively small, which agree with the conclusions drawnbased on the cross-validation

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`.

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

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

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

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

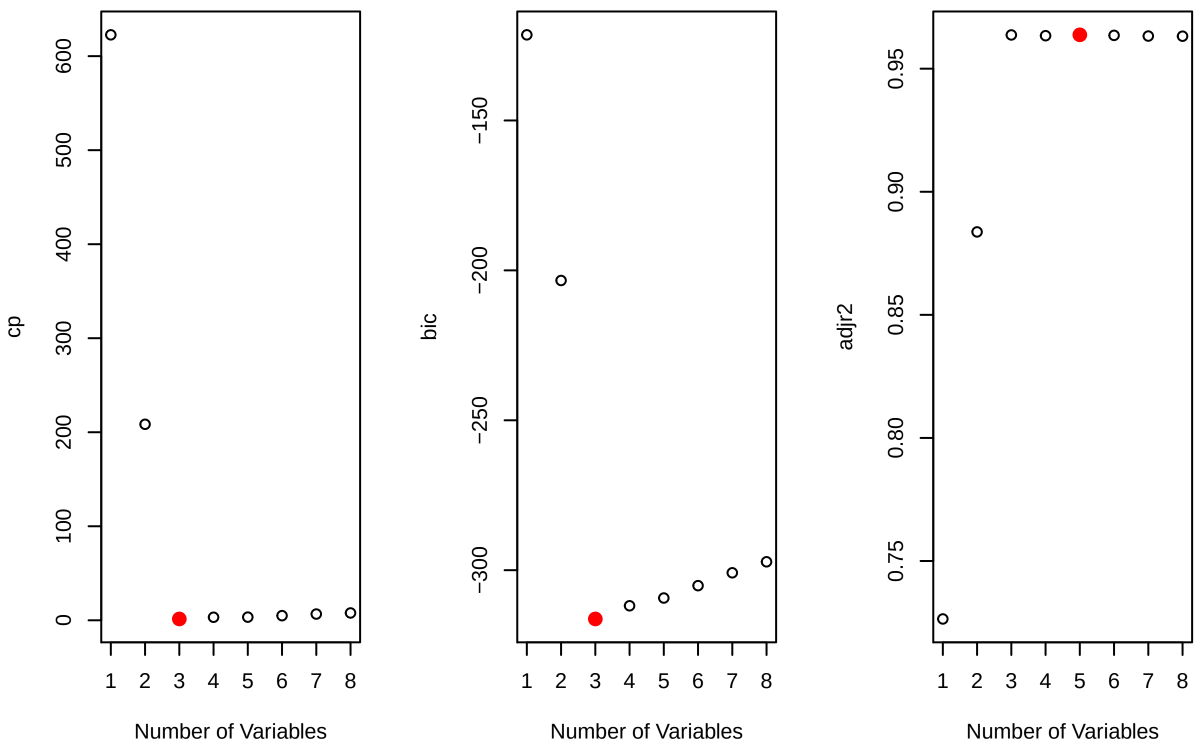
(c) Use the `regsubsets()` function to perform best subset selection in order to choose the best model containing the predictors $X, X^2, \ldots, X^{10}$. What is the best model obtained according to C_p, BIC, and adjusted R^2? Show some plots to provide evidence for your answer, and report the coefficients 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 .

```
num <- matrix(seq(1,10),nrow=1,ncol=10)
x_poly10 <- apply(num,2,function(n){return(x^n)})
df2 <- as.data.frame(cbind(y, x_poly10))
```

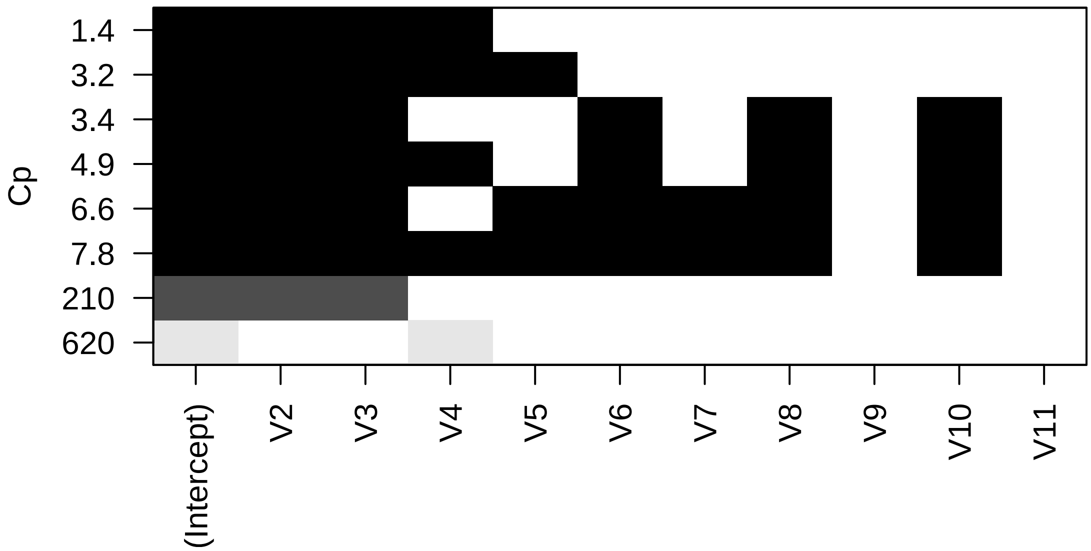
```r
regfit.full <- regsubsets(y ~ ., df2)
reg.summary <- summary(regfit.full)
names(reg.summary) # Elements obtained by summary.regsubsets
```

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

```r
par(mfrow=c(1,3))
plot(reg.summary$cp, xlab="Number of Variables", ylab="cp")
points(which.min(reg.summary$cp),
       reg.summary$cp[which.min(reg.summary$cp)],
       col="red", cex=2, pch=20)
plot(reg.summary$bic, xlab="Number of Variables", ylab="bic")
points(which.min(reg.summary$bic),
       reg.summary$bic[which.min(reg.summary$bic)],
       col="red", cex=2, pch=20)
plot(reg.summary$adjr2, xlab="Number of Variables", ylab="adjr2")
points(which.max(reg.summary$adjr2),
       reg.summary$adjr2[which.max(reg.summary$adjr2)],
       col="red", cex=2, pch=20)
```



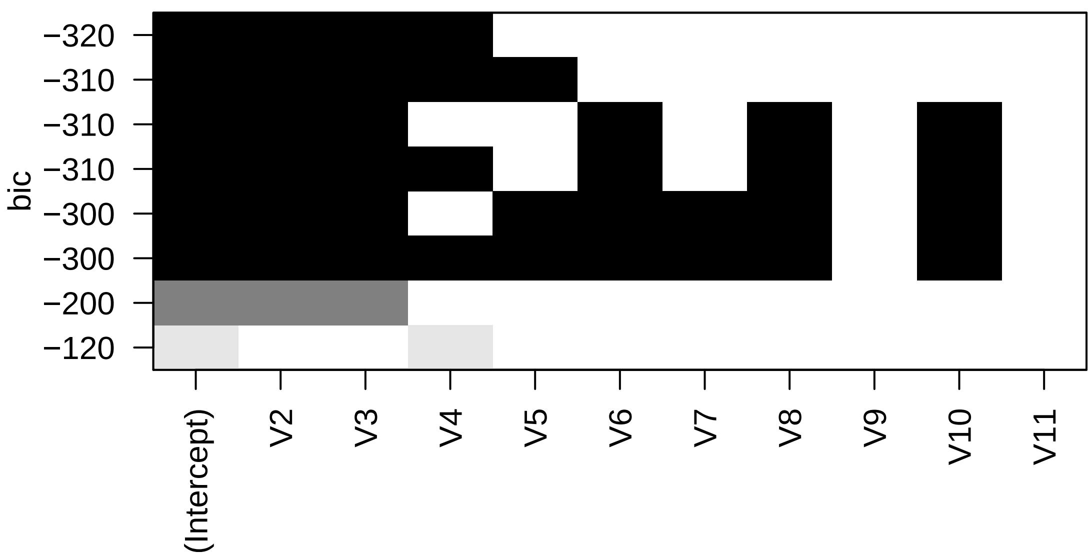

```r
par(mfrow=c(1,1))
plot(regfit.full, scale = "Cp")
```




```r
# The coefficient estimates associated with best cp
coef(regfit.full,which.min(reg.summary$cp))
```

```
## (Intercept)          V2          V3          V4
##    3.928974    2.884212    1.963622    1.021113
```

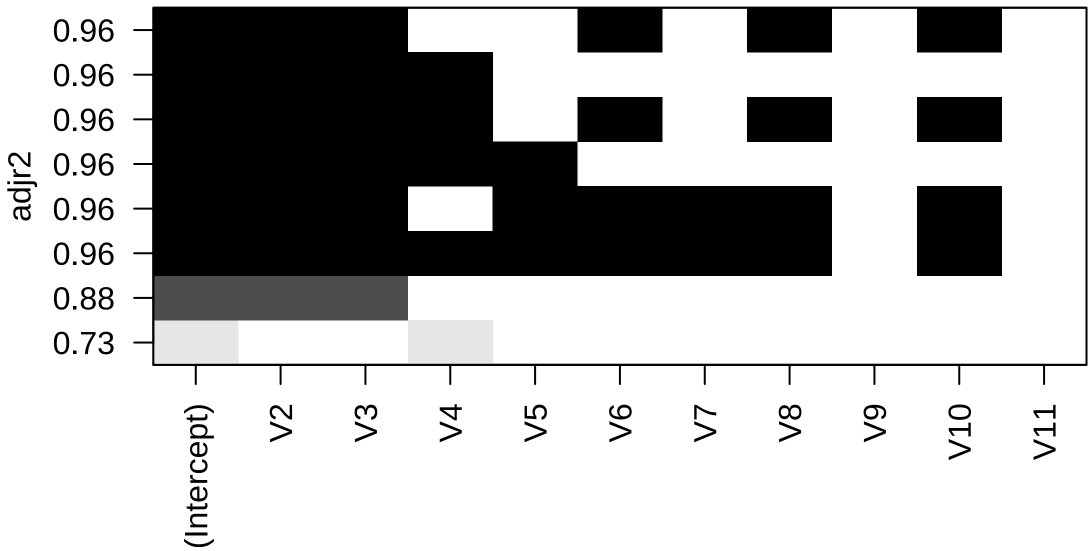
```r
plot(regfit.full,scale = "bic")
```




```r
# The coefficient estimates associated with best bic
coef(regfit.full,which.min(reg.summary$bic))
```

```
## (Intercept)          V2          V3          V4
##    3.928974    2.884212    1.963622    1.021113
```

```r
plot(regfit.full,scale = "adjr2")
```



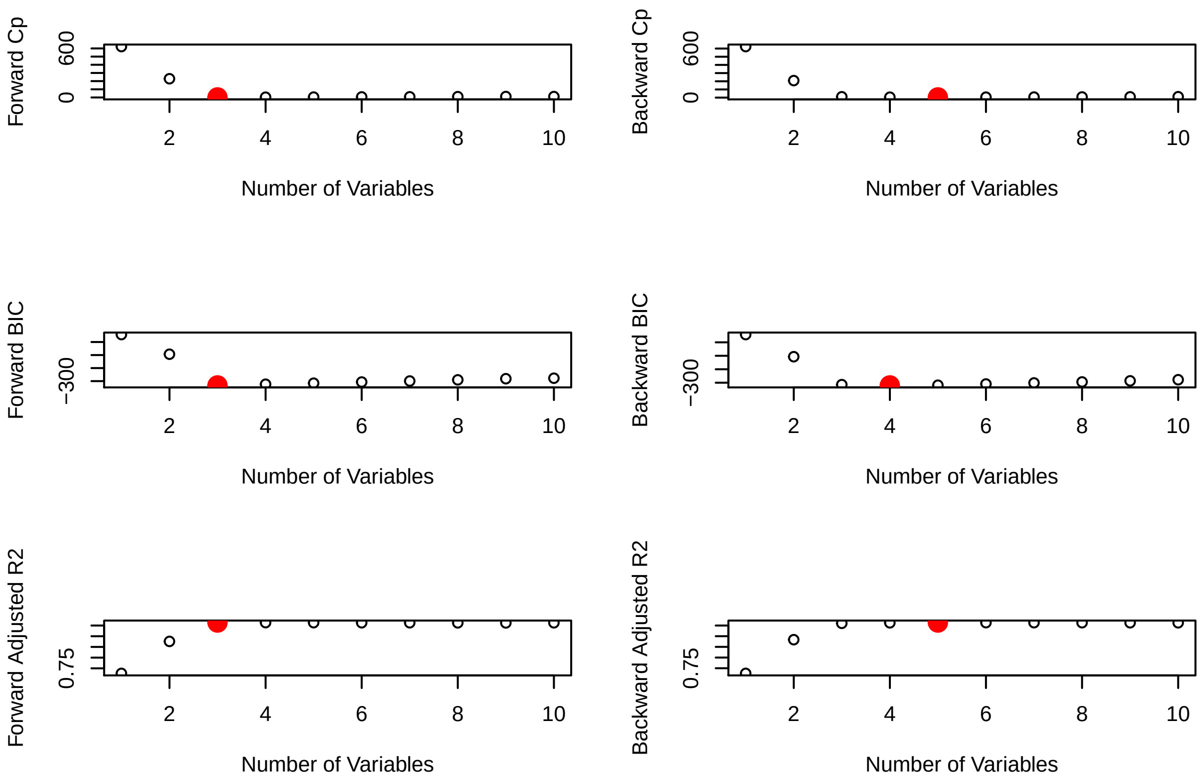

```
# The coefficient estimates associated with best adjr2
coef(regfit.full,which.max(reg.summary$adjr2))
```

```
## (Intercept)          V2          V3          V6          V8         V10
##  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)?

```
fwd  <- regsubsets(y ~ poly(x, 10, raw = T), data = df2, nvmax = 10, method = "forward" )
bwd <- regsubsets(y ~ poly(x, 10, raw = T), data = df2, nvmax = 10, method = "backward")
fwdsummary  <- summary(fwd)
bwdsummary <- summary(bwd)

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

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

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

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

plot(fwdsummary$adjr2, ylab = "Forward Adjusted R2",xlab="Number of Variables")
points(which.max(fwdsummary$adjr2),
```

```r
       fwdsummary$adjr2[which.max(fwdsummary$adjr2)],
       pch = 20, col = "red", lwd = 7)

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



From this subplots, we can see that forward method is selected for building model according to CP, BIC. While for R^2, backword method is applied.

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

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

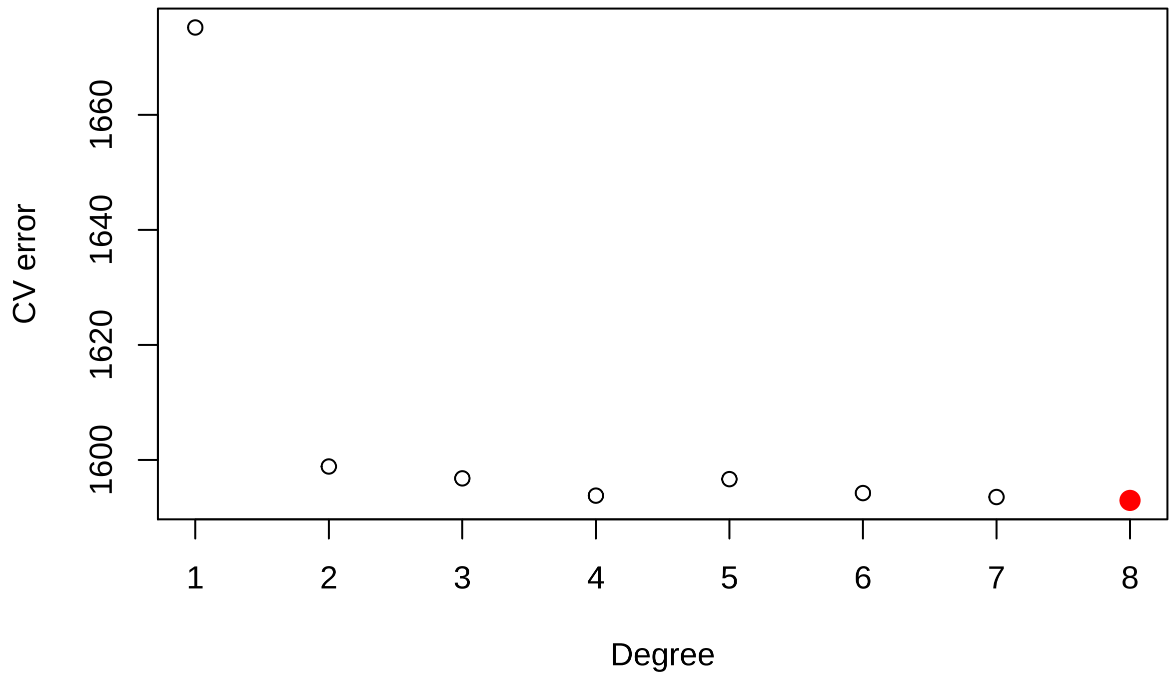
(a) Perform polynomial regression to predict `wage` using `age`. Use cross-validation to select the optimal degree d for the polynomial. What degree was chosen? Make a plot of the resulting polynomial fit to the data.

```r
library(ISLR)
attach(Wage)
lm.fit <- lm(wage ~ poly(age,8))
coef(summary(lm.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
```

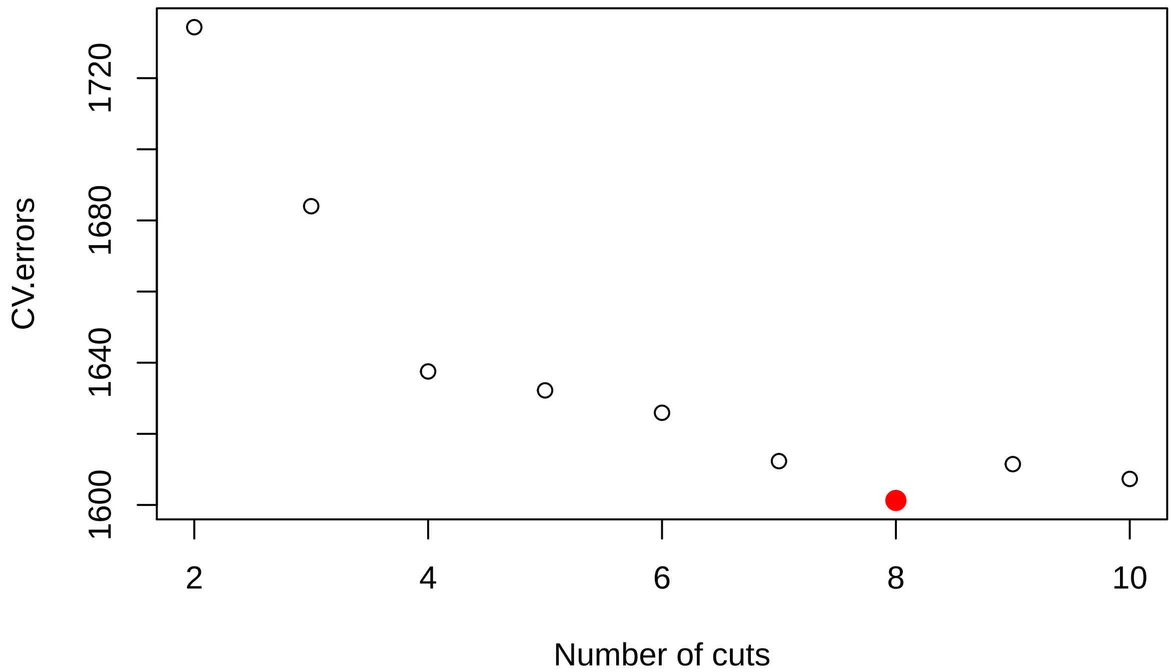
```r
# cross-validation
cv.error <- rep(0, 8)
for (i in 1:8) {
  glm.fit <- glm(wage~poly(age, i))
  cv.error[i] = cv.glm(Wage, glm.fit, K=8)$delta[1]
}
par(mfrow = c(1, 1))
plot(1:8, cv.error, xlab="Degree", ylab="CV error")
n <- which.min(cv.error)
points(n, cv.error[n], col="red", cex=2, pch=20)
```



Degree 8 was chosen.

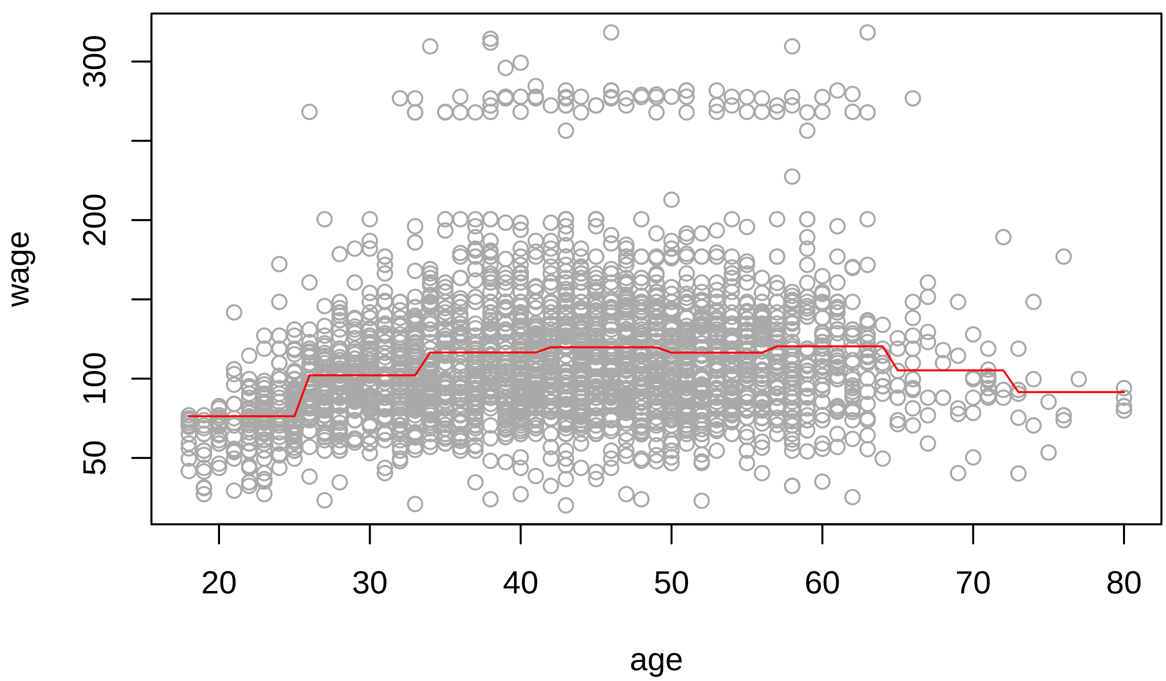
(b) Fit a step function to predict `wage` using `age`, and perform cross-validation to choose the optimal number of cuts. Make a plot of the fit obtained.

```r
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")
n <- which.min(cv.errors)
points(n + 1, cv.errors[n], col="red", cex=2, pch=20)
```



The optimal number of cuts is 8 as showed in the plot.

```
glm.fit <- glm(wage~cut(age, 8))
age.range <- range(Wage$age)
age.grid <- seq(from=age.range[1], to=age.range[2])
glm.pred <- predict(glm.fit, data.frame(age=age.grid))
par(mfrow = c(1, 1))
plot(wage~age, data=Wage, col="darkgrey")
lines(age.grid, glm.pred, col="red")
```



```
detach(Wage)
```

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.

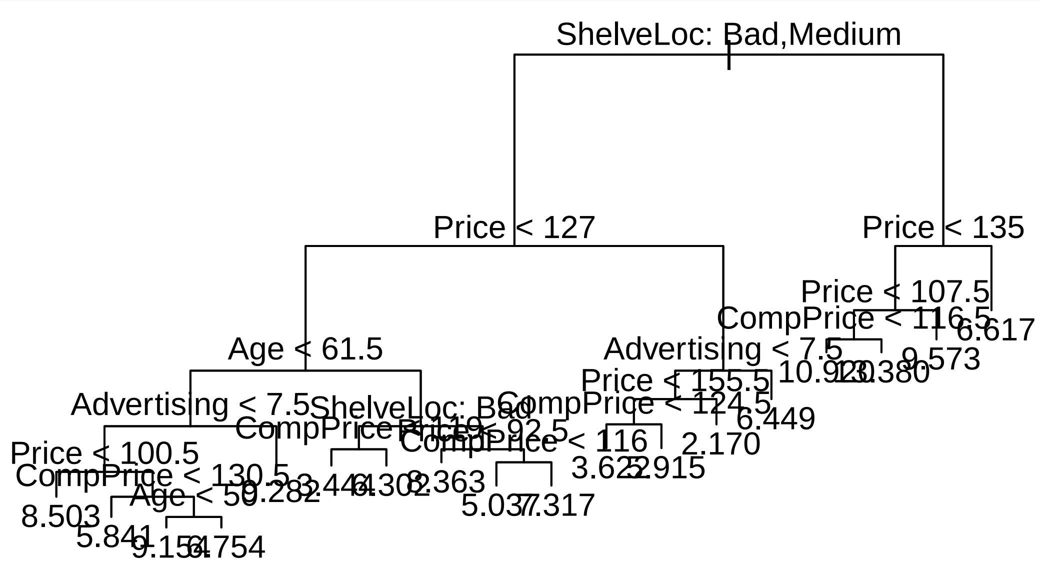
```
library(ISLR)
attach(Carseats)
Carseats.train <- sample(dim(Carseats)[1], dim(Carseats)[1]/2)
Carseats.test <- Carseats[-Carseats.train,]
```

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

```
library(tree)
Carseats.tree <- tree(Sales ~ ., Carseats, subset= Carseats.train)
summary(Carseats.tree)
```

```
##
## Regression tree:
## tree(formula = Sales ~ ., data = Carseats, subset = Carseats.train)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "Age"         "Advertising" "CompPrice"
## Number of terminal nodes:  18
## Residual mean deviance:  2.229 = 405.7 / 182
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3.6110 -0.8899  0.0030  0.0000  0.8219  4.5080
```

```
plot(Carseats.tree)
text(Carseats.tree, pretty = 0)
```



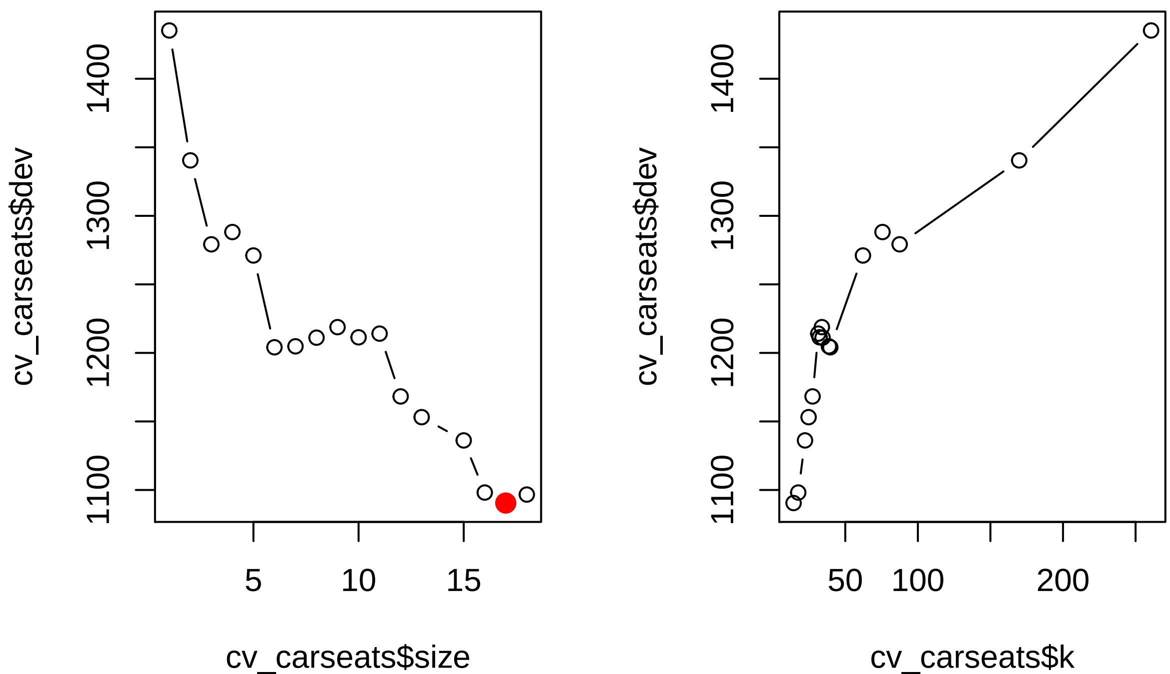
```r
Carseats_pred = predict(Carseats.tree, Carseats.test)
a <- mean((Carseats.test$Sales - Carseats_pred)^2)
```

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

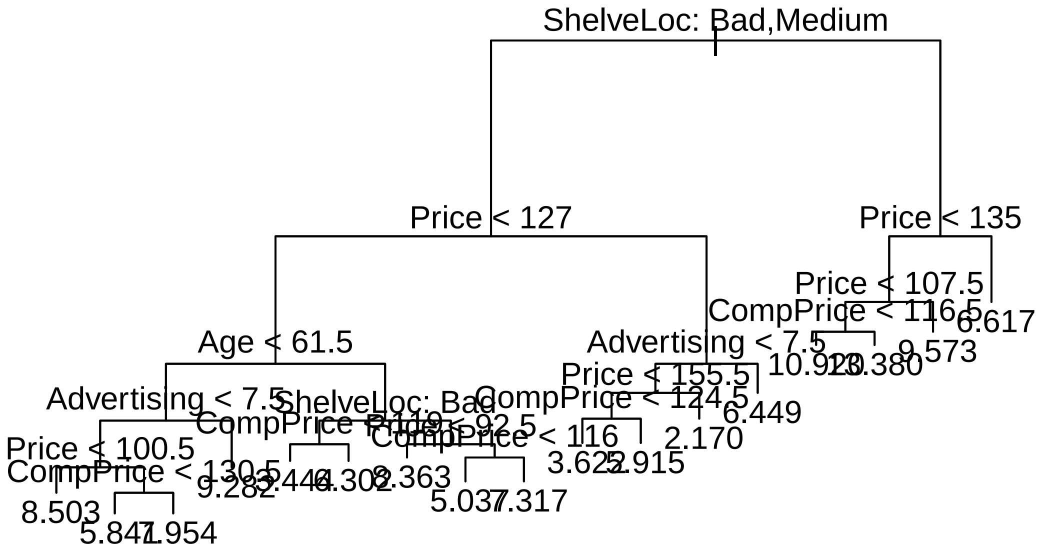
```r
cv_carseats <- cv.tree(Carseats.tree, FUN=prune.tree)
par(mfrow = c(1, 2))
plot(cv_carseats$size, cv_carseats$dev, type = "b")
n <- which.min(cv_carseats$dev)
cv_carseats$size[n]
```

```
## [1] 17
```

```r
points(cv_carseats$size[n],cv_carseats$dev[n],col="red", cex=2, pch=20)
plot(cv_carseats$k, cv_carseats$dev, type = "b")
```




```r
carseats_prune <- prune.tree(Carseats.tree, best = cv_carseats$size[n])
par(mfrow = c(1, 1))
plot(carseats_prune)
text(carseats_prune, pretty = 0)
```




```r
pruned_pred <- predict(carseats_prune, Carseats.test)
b <- mean((Carseats.test$Sales - pruned_pred)^2)

print(a) # the test MSE before pruning
```

```
## [1] 4.632195
```

```r
print(b) # the test MSE after pruning
```

```
## [1] 4.581675
```

The optimal level of tree complexity is 17. Pruning the tree does 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
library(randomForest)
```

```
## randomForest 4.6-12

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

```r
set.seed(6)
rf.carseats <- randomForest(Sales ~ .,Carseats, subset= Carseats.train, mtry = 10, ntree = 500, importa
rf.pred <- predict(rf.carseats, Carseats.test)
mean((Carseats.test$Sales - rf.pred)^2) # test MSE
```

```
## [1] 2.671396
```

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

```
##               %IncMSE IncNodePurity
## CompPrice   25.129819    168.948797
## Income       5.852185     73.875505
## Advertising 15.632316    116.057151
## Population   1.419306     51.485940
## Price       47.389802    424.277455
## ShelveLoc   45.579720    299.363091
## Age         15.965051    144.992291
## Education    3.010010     56.894052
## Urban       -2.193872      8.815258
```

```
## US            3.126488      6.722952
```

From the result, we can see that `Price` and 'ShelveLOv' are the most important