FE590. Assignment #3.

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

2017-11-11

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.

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 2.

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

We can see from the plot that the relationship between x and y is more likly a curve, in other word, x and y are more likly a quadratic relationship.

(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. $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \beta_4 X^4 + \epsilon$

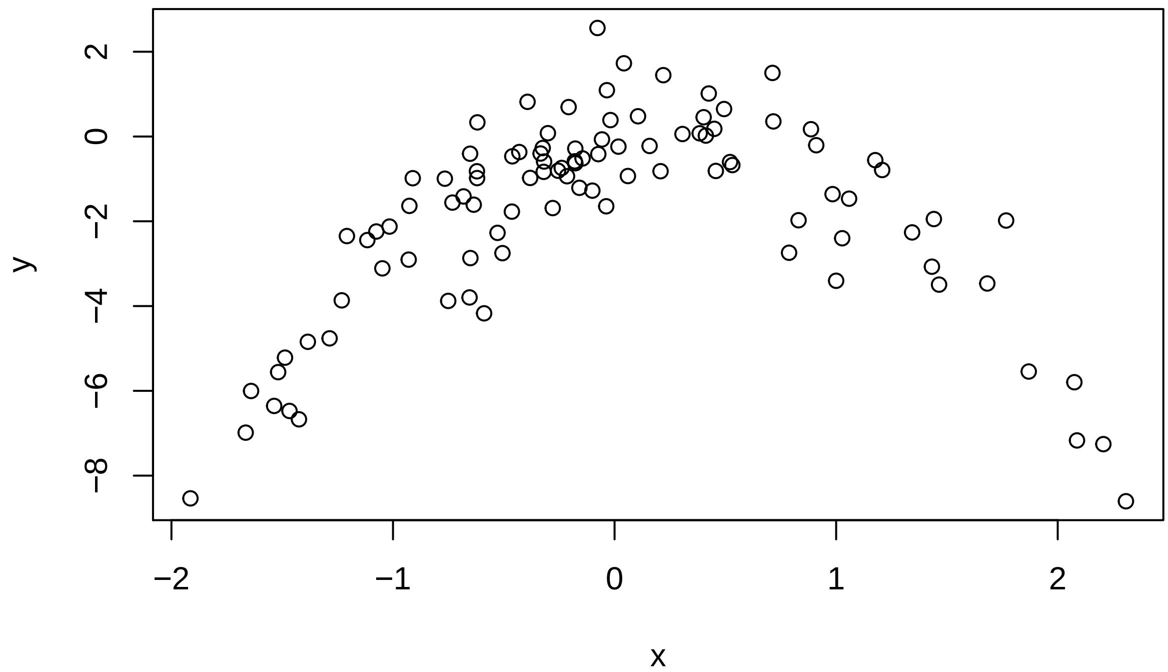
(d) Which of the models in (c) had the smallest LOOCV error? Is this what you expected? Explain your answer.

The number 2 modle has the lowest loocv error. This is what I expected. After we plot the graph, we can estimate that x and y are quadratic relationship, which means a qudratic function will be the best fit to explain the relationship between x and y.

(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 drawn based on the cross-validation results?

The result do agree with conclusion drawn based on the cross-validation results. From the summary we can see the most significant variable is Xˆ2, which correspond to the conclusion.

```
# Enter your R code here!
plot(x,y)
```




```
library(boot)
set.seed(2)
data=data.frame(x,y)
glm.fit1=glm(y~x)
cv.err1=cv.glm(data,glm.fit1)$delta[1]
cv.err1
```

```
## [1] 5.890979
```

```
glm.fit2=glm(y~poly(x,2))
cv.err2=cv.glm(data,glm.fit2)$delta[1]
cv.err2
```

```
## [1] 1.086596
```

```
glm.fit3=glm(y~poly(x,3))
cv.err3=cv.glm(data,glm.fit3)$delta[1]
cv.err3
```

```
## [1] 1.102585
```

```
glm.fit4=glm(y~poly(x,4))
cv.err4=cv.glm(data,glm.fit4)$delta[1]
cv.err4
```

```
## [1] 1.114772
```

```r
summary(glm.fit4)
```

```
##
## Call:
## glm(formula = y ~ poly(x, 4))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8914  -0.5244   0.0749   0.5932   2.7796
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.8277     0.1041 -17.549   <2e-16 ***
## poly(x, 4)1   2.3164     1.0415   2.224   0.0285 *
## poly(x, 4)2 -21.0586     1.0415 -20.220   <2e-16 ***
## poly(x, 4)3  -0.3048     1.0415  -0.293   0.7704
## poly(x, 4)4  -0.4926     1.0415  -0.473   0.6373
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.084654)
##
##     Null deviance: 552.21  on 99  degrees of freedom
## Residual deviance: 103.04  on 95  degrees of freedom
## AIC: 298.78
##
## Number of Fisher Scoring iterations: 2
```

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`.

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

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

(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 .

3-variables model that contains x,xˆ2 and xˆ3 is the best according to CP and BIC, however, for adjusted Rsq,we pick 5-variable model that includs x,x^{2,}^{x}5,xˆ7 and xˆ9 as the best model.

(d) Repeat (c), using forward stepwise selection and also using backwards stepwise selection. How does your answer compare to the results in (c)? The results are different. when using forward selecion method, 3-variable model that contains x,xˆ2 and xˆ3 is the best for all of the approach. When using backward selection, the best model that contains x,x^{2,}^{x}5,xˆ7 and xˆ9 for CP and Adjrst rsq approaches is 5-variable model. And for Bic, the best model is 4-variable model; the most important variable arex,x^{2,}^{x}3 and xˆ4.

```r
# Enter your R code here!
#(a)
set.seed(10)
X=rnorm(100)
epsilion=rnorm(100)

#(b)
Y=rnorm(100)
Y=4+3*X+2*X^2+X^3+epsilion

#(c)
library(leaps)
data2=data.frame(x=X,y=Y)
regfit.full=regsubsets(y~ x+I(x^2)+I(x^3)+I(x^4)+I(x^5)+I(x^6)+I(x^7)+I(x^8)+I(x^9)+I(x^10),data = data2
reg.summary=summary(regfit.full)
par(mfrow=c(2,2))
plot(reg.summary$cp, xlab = "Number of variables", ylab = "C_p", type = "l")
which.min(reg.summary$cp)
```

```
## [1] 3
```

```r
points (3, reg.summary$cp[3], col ="red",cex =2, pch =20)
plot(reg.summary$bic, xlab = "Number of variables", ylab = "BIC", type = "l")
which.min(reg.summary$bic)
```

```
## [1] 3
```

```r
points (3, reg.summary$bic[3], col ="red",cex =2, pch =20)
plot(reg.summary$adjr2, xlab = "Number of variables", ylab = "Adjusted R^2", type = "l")
which.max(reg.summary$adjr2)
```

```
## [1] 5
```

```r
points (5, reg.summary$adjr2[5], col ="red",cex =2, pch =20)
coef(regfit.full,which.min(reg.summary$cp))
```

```
## (Intercept)           x      I(x^2)      I(x^3)
##    3.928974    2.884212    1.963622    1.021113
```

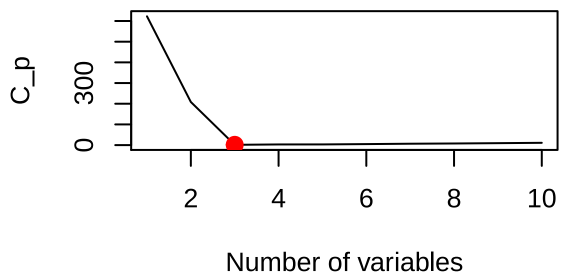
```r
coef(regfit.full,which.min(reg.summary$bic))
```

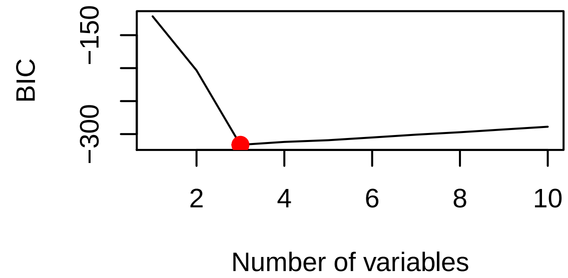
```
## (Intercept)           x      I(x^2)      I(x^3)
##    3.928974    2.884212    1.963622    1.021113
```

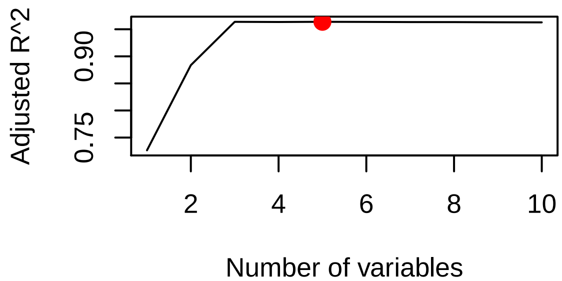
```r
coef(regfit.full,which.max(reg.summary$adjr2))
```

```
## (Intercept)           x      I(x^2)      I(x^5)      I(x^7)      I(x^9)
##  3.95409012  3.12434155  1.93068856  1.04218301 -0.36785066  0.04036764
```

```r
#(d)
regfit.fwd=regsubsets(y~ x+I(x^2)+I(x^3)+I(x^4)+I(x^5)+I(x^6)+I(x^7)+I(x^8)+I(x^9)+I(x^10),data = data2
reg.summaryfwd=summary(regfit.fwd)
par(mfrow = c(2, 2))
```








```
plot(reg.summaryfwd$cp, xlab = "Number of variables,Forward", ylab = "C_p", type = "l")
which.min(reg.summaryfwd$cp)
```

```
## [1] 3
```

```
points (3, reg.summaryfwd$cp[3], col ="red",cex =2, pch =20)
plot(reg.summaryfwd$bic, xlab = "Number of variables,Forward", ylab = "BIC", type = "l")
which.min(reg.summaryfwd$bic)
```

```
## [1] 3
```

```
points (3, reg.summaryfwd$bic[3], col ="red",cex =2, pch =20)
plot(reg.summaryfwd$adjr2, xlab = "Number of variables,Forward", ylab = "Adjusted R^2", type = "l")
which.max(reg.summaryfwd$adjr2)
```

```
## [1] 3
```

```
points (3, reg.summaryfwd$adjr2[3], col ="red",cex =2, pch =20)
coef(regfit.fwd,which.min(reg.summaryfwd$cp))
```

```
## (Intercept)           x      I(x^2)      I(x^3)
##    3.928974    2.884212    1.963622    1.021113
```

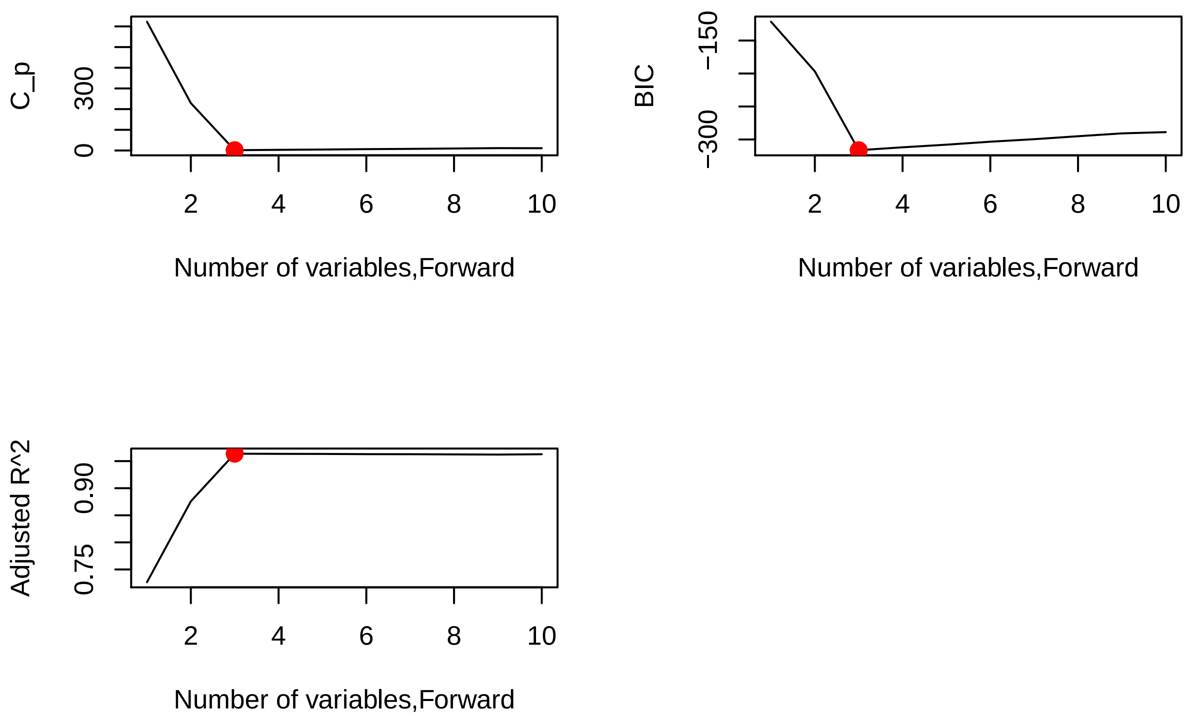
```
coef(regfit.fwd,which.min(reg.summaryfwd$bic))
```

```
## (Intercept)           x      I(x^2)      I(x^3)
##    3.928974    2.884212    1.963622    1.021113
```

```
coef(regfit.fwd,which.max(reg.summaryfwd$adjr2))
```

```
## (Intercept)           x      I(x^2)      I(x^3)
##    3.928974    2.884212    1.963622    1.021113
```

```
regfit.bwd=regsubsets(y~ x+I(x^2)+I(x^3)+I(x^4)+I(x^5)+I(x^6)+I(x^7)+I(x^8)+I(x^9)+I(x^10),data = data2
reg.summarybwd=summary(regfit.bwd)
par(mfrow = c(2, 2))
```




```
plot(reg.summarybwd$cp, xlab = "Number of variables,Backward", ylab = "C_p", type = "l")
which.min(reg.summarybwd$cp)
```

```
## [1] 5
```

```
points (5, reg.summarybwd$cp[5], col ="red",cex =2, pch =20)
plot(reg.summarybwd$bic, xlab = "Number of variables,Backward", ylab = "BIC", type = "l")
which.min(reg.summarybwd$bic)
```

```
## [1] 4
```

```
points (4, reg.summarybwd$bic[4], col ="red",cex =2, pch =20)
plot(reg.summarybwd$adjr2, xlab = "Number of variables,Backward", ylab = "Adjusted R^2", type = "l")
which.max(reg.summarybwd$adjr2)
```

```
## [1] 5
```

```
points (5, reg.summarybwd$adjr2[5], col ="red",cex =2, pch =20)
coef(regfit.bwd,which.min(reg.summarybwd$cp))
```

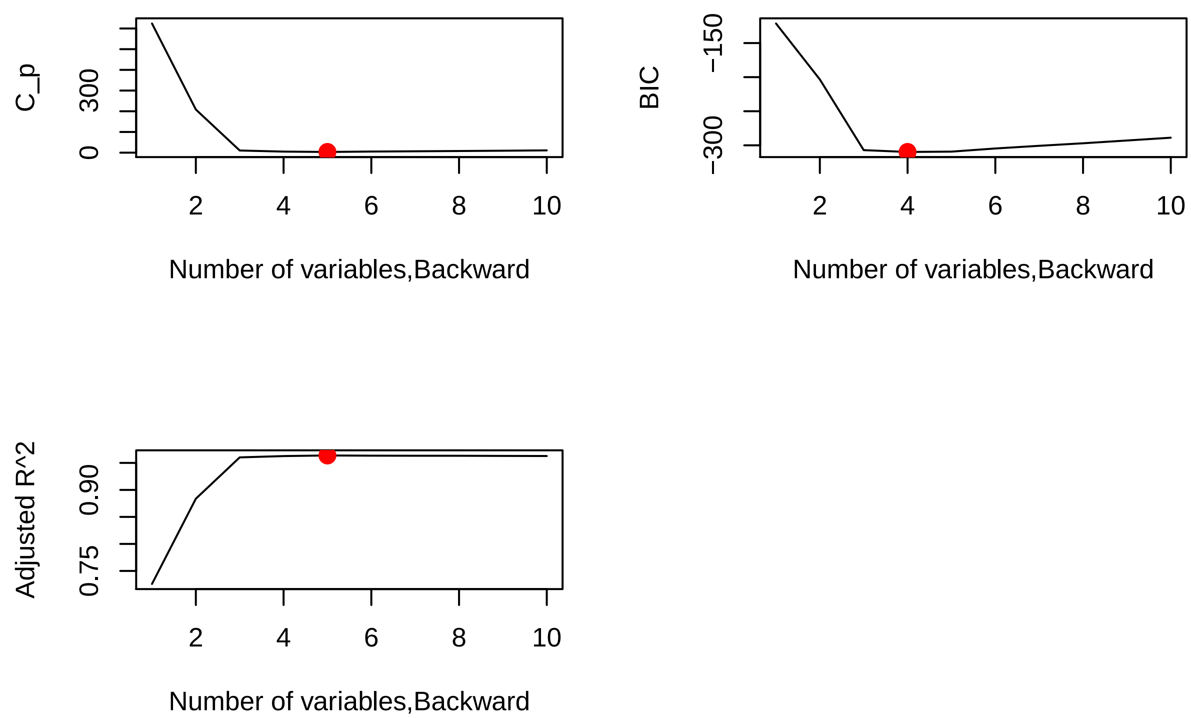
```
## (Intercept)           x      I(x^2)      I(x^5)      I(x^7)      I(x^9)
##  3.95409012  3.12434155  1.93068856  1.04218301 -0.36785066  0.04036764
```

```r
coef(regfit.bwd,which.min(reg.summarybwd$bic))
```

```
## (Intercept)           x      I(x^2)      I(x^5)      I(x^7) 
##  3.92543184  3.42357607  1.97564883  0.46966736 -0.05868386
```

```r
coef(regfit.bwd,which.max(reg.summarybwd$adjr2))
```

```
## (Intercept)           x      I(x^2)      I(x^5)      I(x^7)      I(x^9) 
##  3.95409012  3.12434155  1.93068856  1.04218301 -0.36785066  0.04036764
```



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.

The optimal degree d for the polynomial is 10.

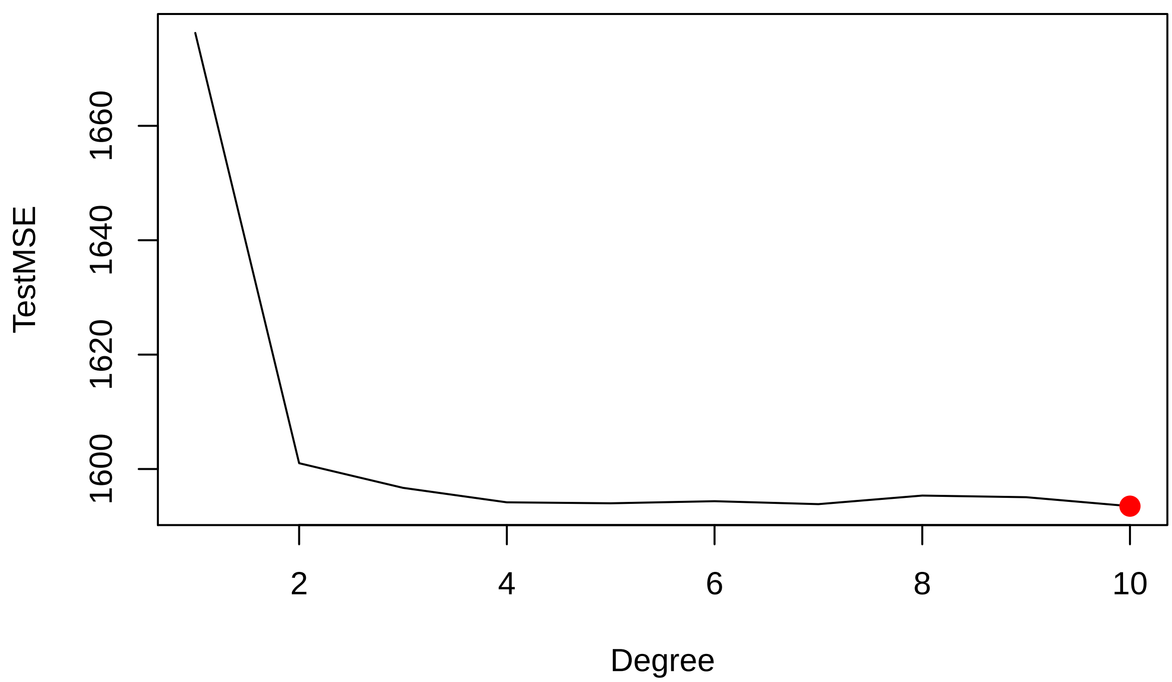
(b) Fit a step function to predict `wage` using `age`, and perform cross-validation to choose the optimal number of cuts. Make a plot of the fit obtained.

```r
# Enter your R code here!
#(a)
library(ISLR)
library(boot)
```

```r
set.seed(3)
error10=rep(0,10)
for (i in 1:10) {
fit=glm(wage~ poly(age,i),data=Wage)
error10[i]=cv.glm (Wage,fit,K=10)$delta [1]
}
plot(1:10,error10,xlab = "Degree",ylab = "TestMSE",type = "l")
which.min(error10)
```

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

```r
points (10, error10[10], col ="red",cex =2, pch =20)
```

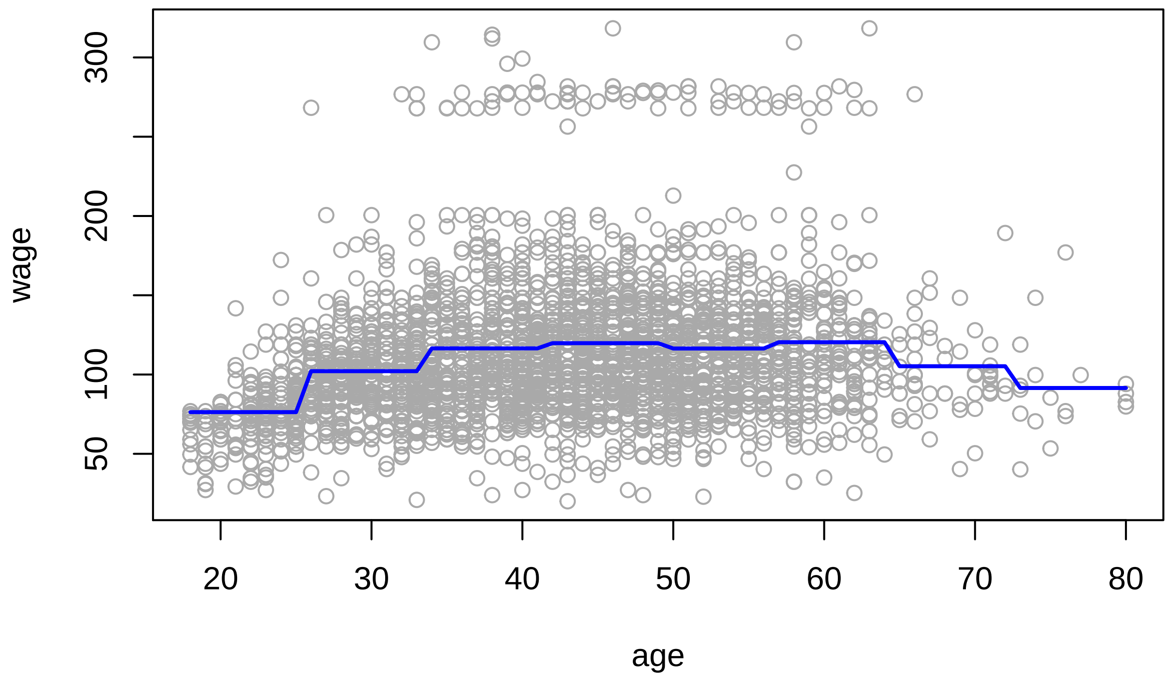



```r
#(b)
error9=rep(0, 10)
for (i in 2:10) {
Wage$age.cut=cut(Wage$age, i)
fit=glm(wage ~ age.cut, data = Wage)
error9[i]=cv.glm(Wage, fit, K = 10)$delta[1]
}
plot(2:10, error9[-1], xlab = "Bestcuts", ylab = "Test MSE", type = "l")
```


```
which.min(error9)
```

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

```
agelims =range(Wage$age)
age.grid=seq (from=agelims [1], to=agelims [2])
fit= glm(wage~ cut(age,8),data=Wage)
lm.preds=predict (fit ,newdata =list(age=age.grid))
plot(wage~age,data = Wage,col =" darkgrey ")
lines(age.grid,lm.preds,lwd =2, col =" blue")
```



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?

The test MSE is 4.148897

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

We can see that the test MSE increase to 5.09. So it doesn't 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.

The test MSE is 2.55 and the most important variable are price, second most important variable is shelveLoc.

```
# Enter your R code here!
#(a)
library(ISLR)
set.seed(1)
train=sample(1:nrow(Carseats), nrow(Carseats) / 2)
Carseats.test= Carseats[-train, ]
```

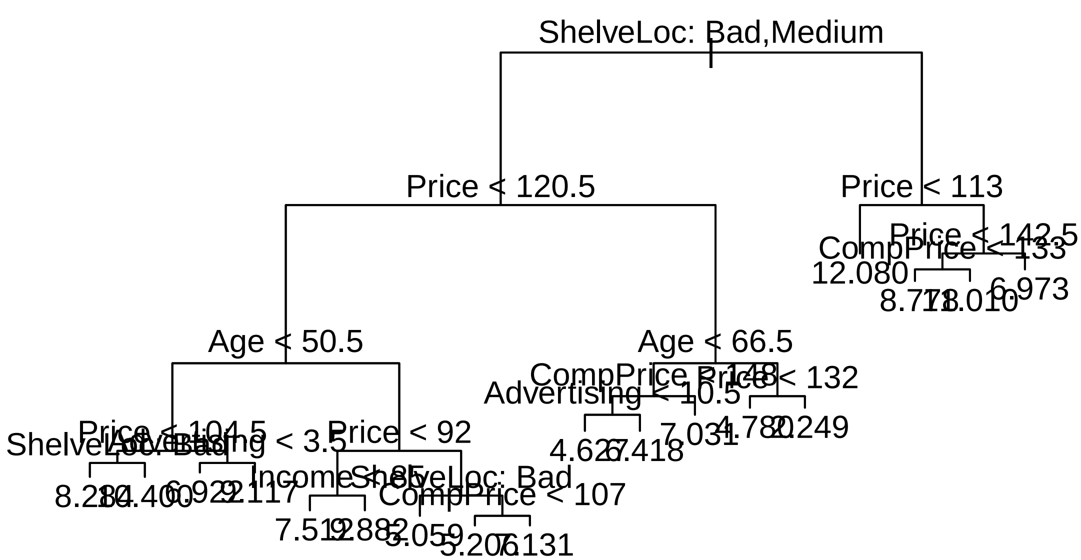
```r
#(b)
library (tree)
```

```
## Warning: package 'tree' was built under R version 3.4.2
```

```r
tree.Carseats=tree(Sales ~ ., Carseats,subset=train)
summary(tree.Carseats)
```

```
##
## Regression tree:
## tree(formula = Sales ~ ., data = Carseats, subset = train)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "Age"         "Advertising" "Income"
## [6] "CompPrice"
## Number of terminal nodes:  18
## Residual mean deviance:  2.36 = 429.5 / 182
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -4.2570 -1.0360  0.1024  0.0000  0.9301  3.9130
```

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



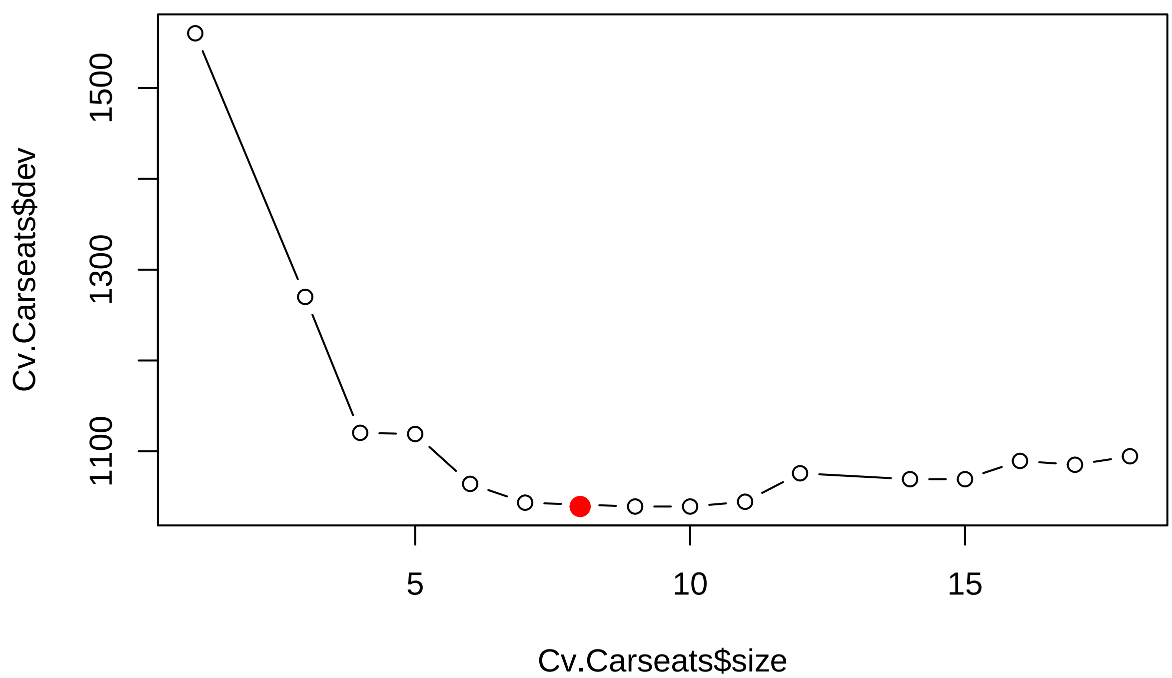

```r
yhat=predict(tree.Carseats, newdata = Carseats.test)
mean((yhat - Carseats.test$Sales)^2)
```

```
## [1] 4.148897
```

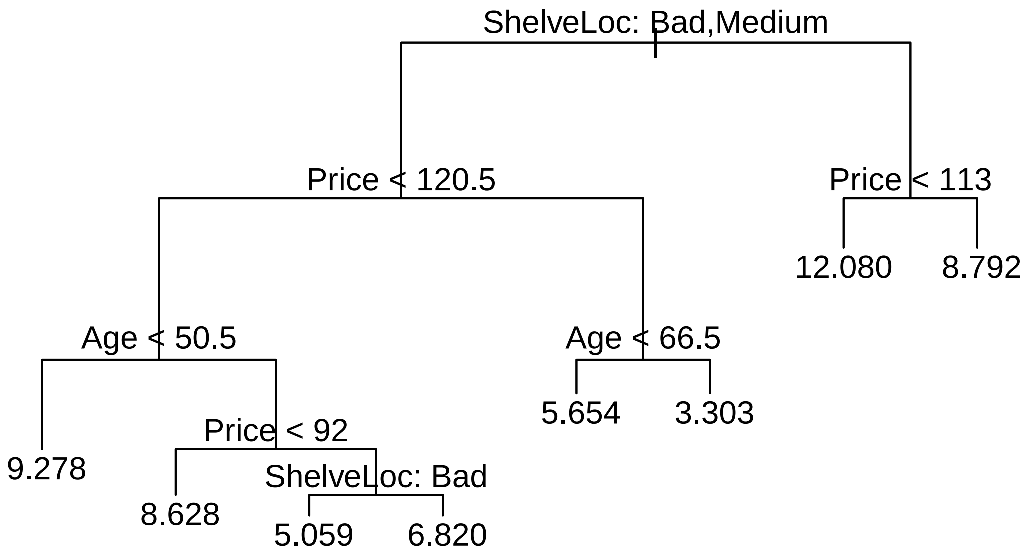
```r
#(c)
Cv.Carseats=cv.tree(tree.Carseats)
plot(Cv.Carseats$size, Cv.Carseats$dev, type = "b")
which.min(Cv.Carseats$dev)
```

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

```r
carmin=which.min(Cv.Carseats$dev)
points(carmin,Cv.Carseats$dev[carmin],col ="red",cex =2, pch =20)
```




```r
prune.Carseats=prune.tree(tree.Carseats, best = carmin)
plot(prune.Carseats)
text(prune.Carseats, pretty = 0)
```




```
yhat.prune=predict(prune.Carseats, newdata = Carseats.test)
mean((yhat.prune - Carseats.test$Sales)^2)
```

```
## [1] 5.09085
```

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

```
## Warning: package 'randomForest' was built under R version 3.4.2
```

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

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

```
set.seed (1)
names(Carseats)
```

```
##  [1] "Sales"       "CompPrice"   "Income"      "Advertising" "Population"
##  [6] "Price"       "ShelveLoc"   "Age"         "Education"   "Urban"
## [11] "US"
```

```
bag.Carseats=randomForest(Sales ~ ., data = Carseats,subset=train, mtry = 10,importance = TRUE)
yhat.bag=predict(bag.Carseats, newdata = Carseats.test)
mean((yhat.bag - Carseats.test$Sales)^2)
```

```
## [1] 2.554292
```

```
importance(bag.Carseats)
```

```
##              %IncMSE IncNodePurity
```

```
## CompPrice   14.032030    129.568747
## Income       5.523038     75.448682
## Advertising 13.571285    131.246840
## Population   1.968853     63.042648
## Price       56.863812    504.158108
## ShelveLoc   44.720455    323.055042
## Age         22.225468    194.915976
## Education    4.823966     40.810991
## Urban       -1.902185      8.746566
## US           6.632887     14.599565
```

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. compare to the MSE of boosted trees, the other regression techniques have higher MSE.

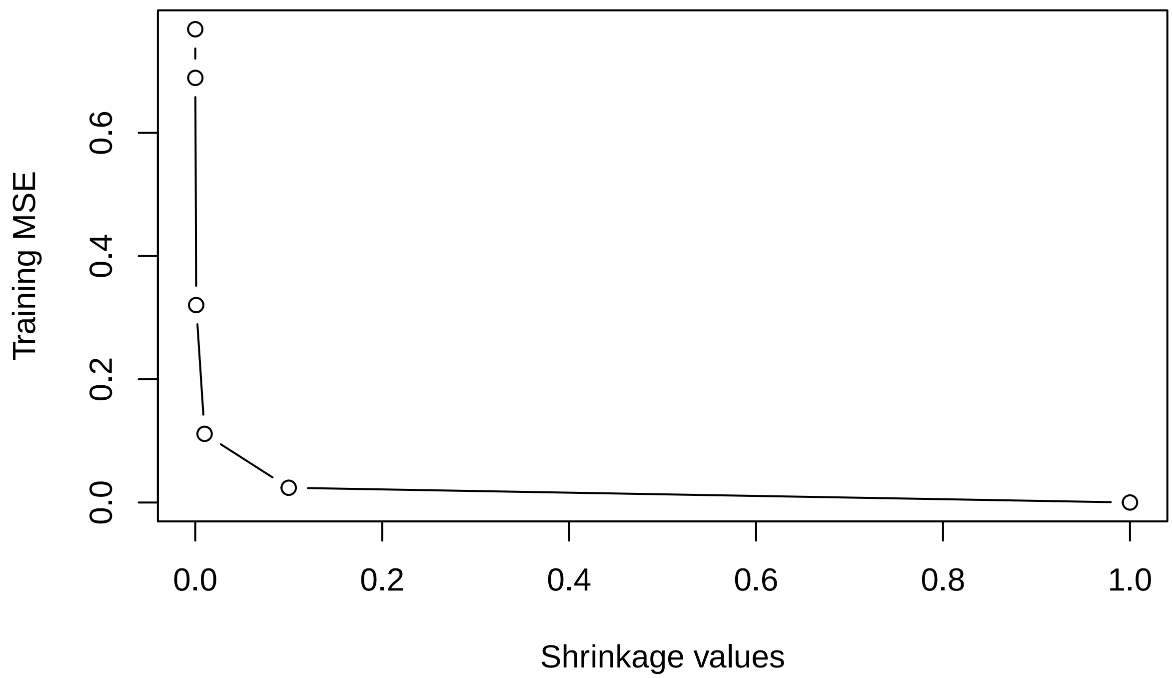
(f) Reproduce (c) and (d), but this time use bagging instead of boosting and compare to the boosted MSE's and the MSE's from (e)

```r
# Enter your R code here!
#(a)
Hitters=na.omit(Hitters)
Hitters$Salary=log(Hitters$Salary)

#(b)
set.seed(1)
train=sample (c(TRUE ,FALSE), nrow(Hitters ),rep=TRUE)
Hitters.train=Hitters[train,]
Hitters.test =Hitters[-train,]

#(c)
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 3.4.2

## Loading required package: survival

##
## Attaching package: 'survival'

## The following object is masked from 'package:boot':
##
##     aml

## Loading required package: lattice
```

```
##
## Attaching package: 'lattice'

## The following object is masked from 'package:boot':
##
##     melanoma

## Loading required package: splines

## Loading required package: parallel

## Loaded gbm 2.1.3
```

```
set.seed(1)
lambdas=c(0.00001,0.0001,0.001,0.01,0.1,1)
trainset=rep(0, length(lambdas))
for (i in 1:length(lambdas)) {
boost.Hitters=gbm(Salary ~ ., data = Hitters.train, distribution = "gaussian", n.trees = 1000, shrinkage
pred.train=predict(boost.Hitters, newdata=Hitters.train, n.trees = 1000)
trainset[i]=mean((pred.train - Hitters.train$Salary)^2)
}
plot(lambdas,trainset, type = "b", xlab = "Shrinkage values", ylab = "Training MSE")
```

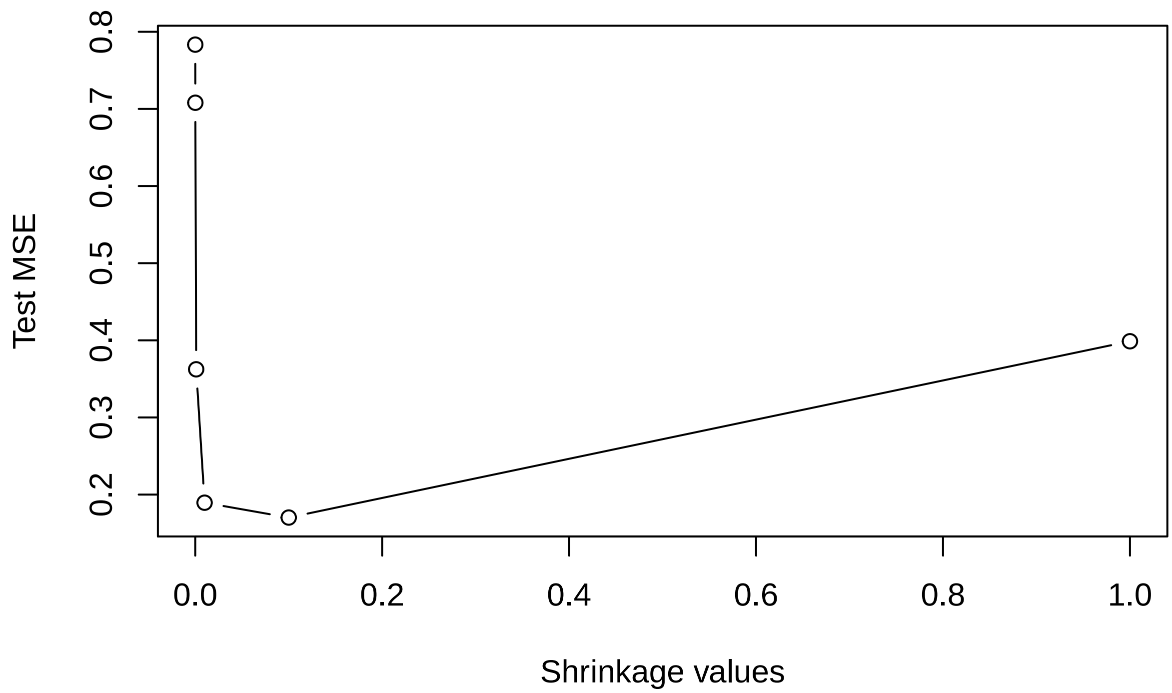



```
min(trainset)
```

```
## [1] 4.388678e-05
```

```
#(d)
set.seed(1)
lambdas=c(0.00001,0.0001,0.001,0.01,0.1,1)
```

```r
testset=rep(0, length(lambdas))
for (i in 1:length(lambdas)) {
boost.Hitters=gbm(Salary ~ ., data = Hitters.train, distribution = "gaussian", n.trees = 1000, shrinkag
pred.test=predict(boost.Hitters, newdata=Hitters.test, n.trees = 1000)
testset[i]=mean((pred.test - Hitters.test$Salary)^2)
}
plot(lambdas,testset, type = "b", xlab = "Shrinkage values", ylab = "Test MSE")
```




```r
min(testset)
```

```
## [1] 0.1702429
```

```r
#(e)
library(glmnet)
```

```
## Loading required package: Matrix
## Loading required package: foreach
## Loaded glmnet 2.0-12
```

```r
#linear regression
lmfit=lm(Salary ~ ., data = Hitters.train)
pred.lmfit=predict(lmfit, Hitters.test)
mean((pred.lmfit - Hitters.test$Salary)^2)
```

```
## [1] 0.4023807
```

```r
#Ridge Regression
x=model.matrix(Salary ~ ., data = Hitters.train)
x.test= model.matrix(Salary ~ ., data = Hitters.test)
y=Hitters.train$Salary
ridge.mod=glmnet(x, y, alpha = 0)
ridge.pred= predict(ridge.mod, s =0.01, newx = x.test)
mean((ridge.pred - Hitters.test$Salary)^2)
```

```
## [1] 0.3827933
```

```r
#(f)
#training set
set.seed (1)
xvalue=(1:100) * 10
for (i in 1:100) {
bag.Hitters=randomForest(Salary ~ ., data = Hitters.train, mtry = 19,importance = TRUE,ntree=i*10)
yhat.bag=predict(bag.Hitters, newdata = Hitters.test)
mean((yhat.bag - Hitters.test$Salary)^2)

}
```