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..

2022-04-10

I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination. I further pledge that I have not copied any material from a book, article, the Internet or any other source except where I have expressly cited the source.

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.

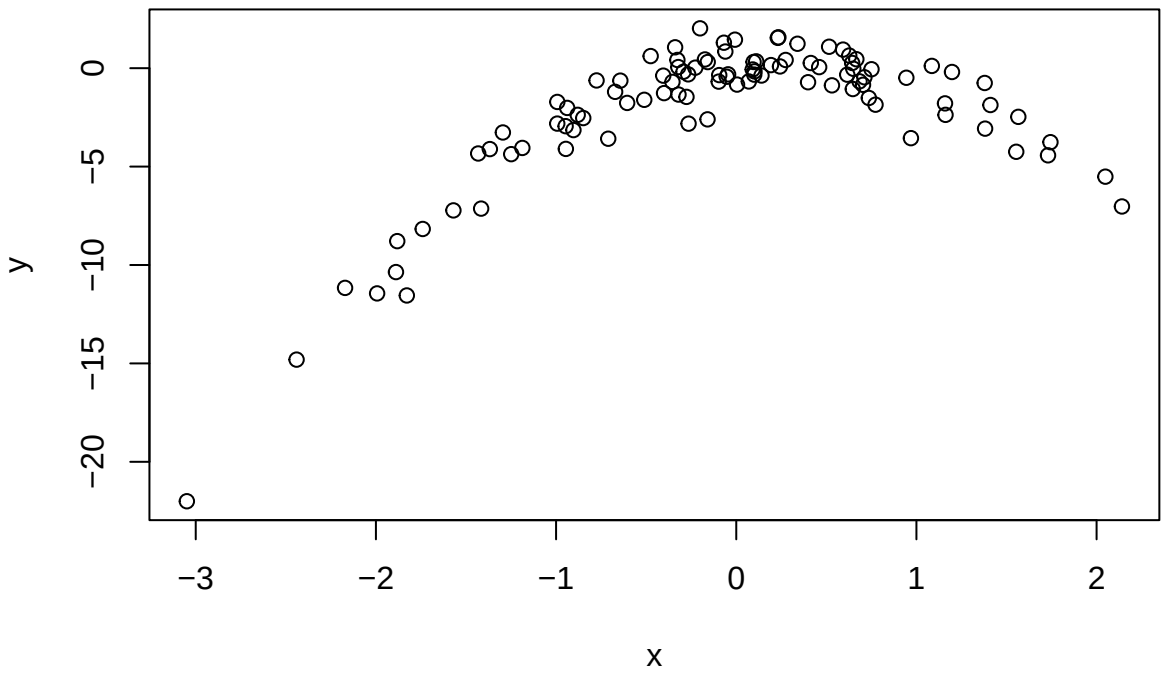
You will use this personalized simulated data set for this problem:

```
CWID = 10479206
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? n is 100, p is 1

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

```
#(a)
plot(x,y)
```



This is a parabola

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

```
library(boot)
set.seed(10086)
data1 <- data.frame(y, x)
cv_error <- seq(1:4)
for (i in cv_error) {
glm.fit <- glm(y~poly(x, i), data = data1)
cv_error[i] <- cv.glm(data1, glm.fit)$delta[1]
}
names(cv_error) <- c("poly1","poly2","poly3","poly4")
print(cv_error)
```

```
##     poly1     poly2     poly3     poly4
## 10.595201  1.026850  1.063247  1.046519
```

(d) Which of the models in (c) had the smallest LOOCV error? Is this what you expected? Explain your answer. poly=2 had the smallest LOOCV error. Yes because the formula is y <- x - 2*xˆ2 + rnorm(100)

(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?

```
for (i in 1:4){
  cof <- glm(y ~ poly(x,i), data = data1)$coefficients
  cof.app <- cbind(cof,cof)
}
cof.app <- as.data.frame(cof.app)
cof.app
```

```
##                      cof         cof
## (Intercept)  -2.1100471  -2.1100471
## poly(x, i)1  20.8428061  20.8428061
## poly(x, i)2 -29.3380422 -29.3380422
## poly(x, i)3   0.3628506   0.3628506
## poly(x, i)4   1.2016413   1.2016413
```

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).

```
library(ISLR)
library(leaps)
data(College)
train <- sample(nrow(College),nrow(College)*0.7)
train.set <- College[train,]
test.set <- College[-train,]
regit.full <- regsubsets(Outstate~.,data=College,method = "backward")
reg.sum <- summary(regit.full)
```

(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.

```
library(gam)
gam.1 <- gam(Outstate~Private+s(Expend,5)+s(perc.alumni,5)+s(Grad.Rate,5)+s(Room.Board,5)+s(F.Undergrad
```

(c) Evaluate the model obtained on the test set, and explain the results obtained

```
preds2 <- predict(gam.1, newdata = test.set)
mean(preds2==mean(test.set$Outstate))
```

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

(d) For which variables, if any, is there evidence of a non-linear relationship with the response? Which are probably linear? Justify your answers.

```
summary(gam.1)$anova
```

```
## Anova for Nonparametric Effects
##                   Npar Df  Npar F     Pr(F)
## (Intercept)
## Private
## s(Expend, 5)            4 15.2662 1.684e-11 ***
## s(perc.alumni, 5)       4  1.5010   0.20146
## s(Grad.Rate, 5)         4  2.2474   0.06366 .
## s(Room.Board, 5)        4  0.7387   0.56611
## s(F.Undergrad, 5)       4  1.7786   0.13265
## s(Terminal, 5)          4  0.9135   0.45614
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Anova shows Expend has strong non-linear relationshop with the Outstate. Grad.Rate and PhD have moderate non-linear relationship with the Outstate.

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

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

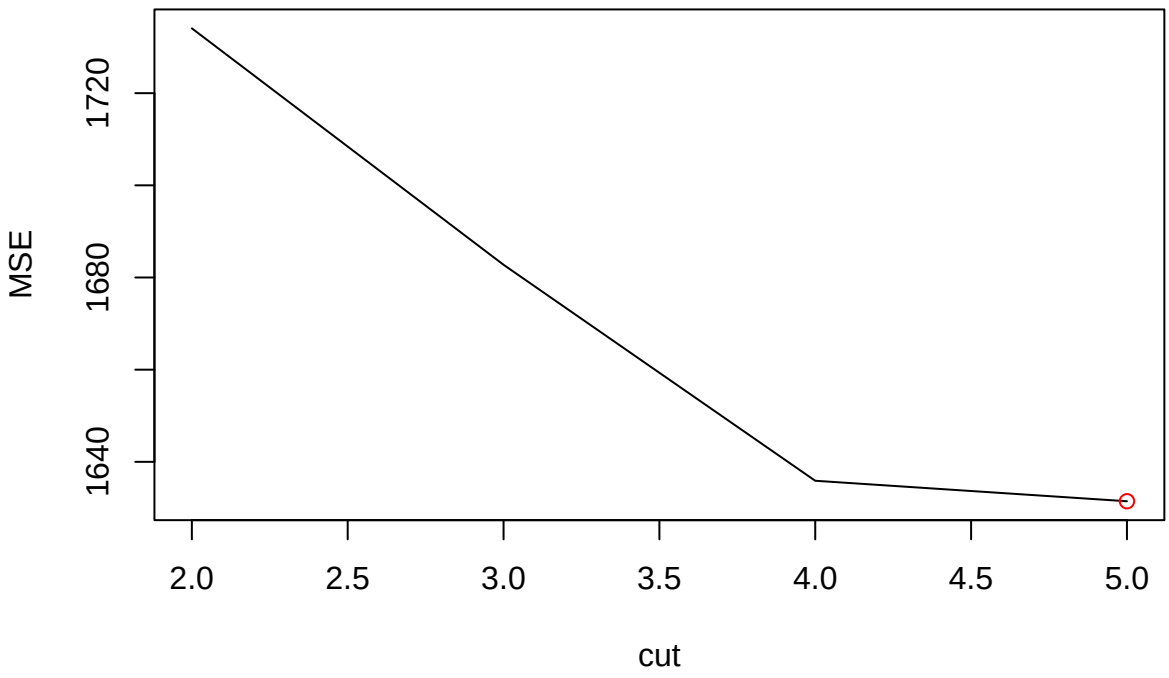
(a) Perform polynomial regression to predict `wage` using `age`. Use cross-validation to select the optimal degree d for the polynomial. What degree was chosen? Make a plot of the resulting polynomial fit to the data.

```
data(Wage)
library(boot)
set.seed(1)
cv.error <- rep(NA, 5)
for (i in 1:5) {
  fit <- glm(wage ~ poly(age, i), data = Wage)
  cv.error[i] <- cv.glm(Wage, fit)$delta[1]
}
cv.error
```

```
## [1] 1676.235 1600.529 1595.960 1594.596 1594.879
```

(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.error1 <- rep(NA, 4)
for (i in 2:5) {
  Wage$age.cut <- cut(Wage$age, i)
  fit <- glm(wage ~ age.cut, data = Wage)
  cv.error[i] <- cv.glm(Wage, fit)$delta[1]
}
plot(2:5, cv.error[-1], xlab = 'cut', ylab = 'MSE',type = 'l')
deg.min <- which.min(cv.error)
points(deg.min, cv.error[deg.min], col = 'red', cex = 1, pch = 1)
```



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(MASS)
library(randomForest)
```

```
## randomForest 4.7-1
```

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

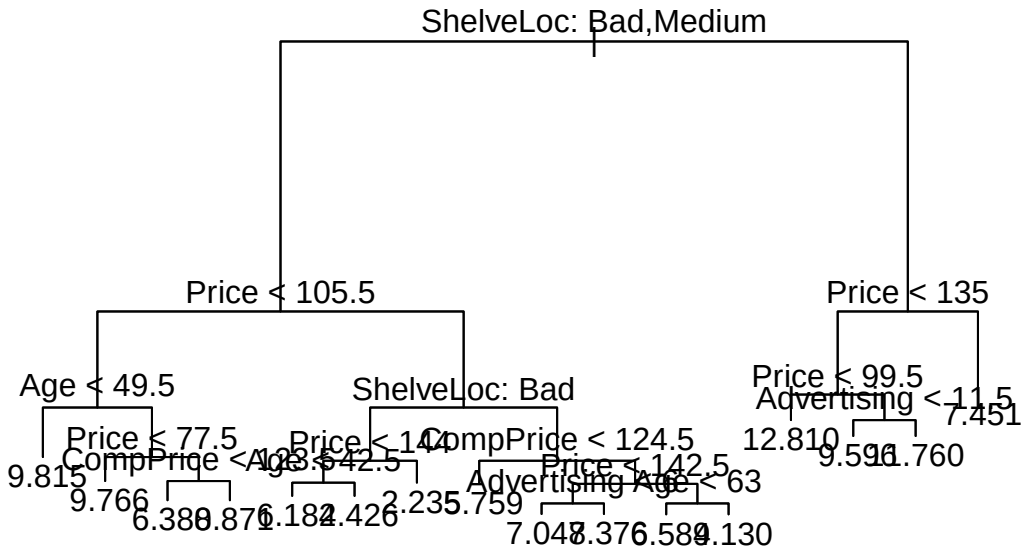
```
train.num <- sample(nrow(Carseats),nrow(Carseats)*0.7)
train.data <- Carseats[train.num,]
test.data <- Carseats[-train.num,]
```

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

```
library(tree)
```

```
## Registered S3 method overwritten by 'tree':
##   method     from
##   print.tree cli
```

```
tree.car <- tree(Sales~.,train.data)
plot(tree.car)
text(tree.car ,pretty =0)
```



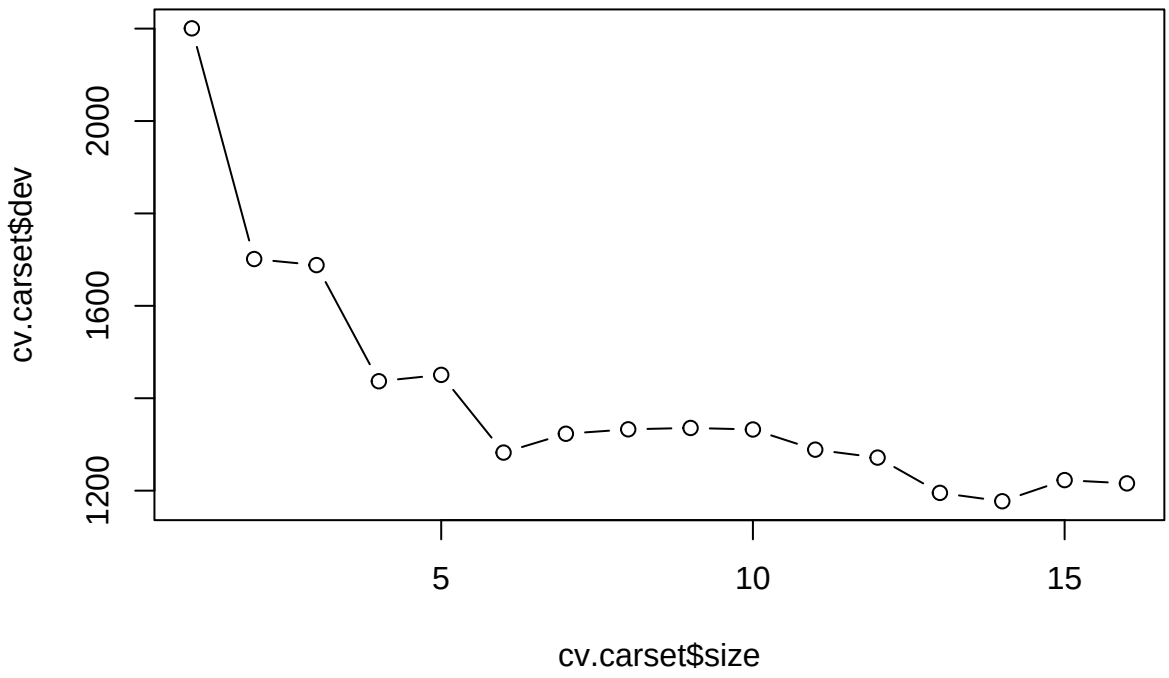

```
tree.pred <- predict(tree.car,newdata = test.data)
mean((tree.pred - test.data$Sales)^2)
```

```
## [1] 4.657927
```

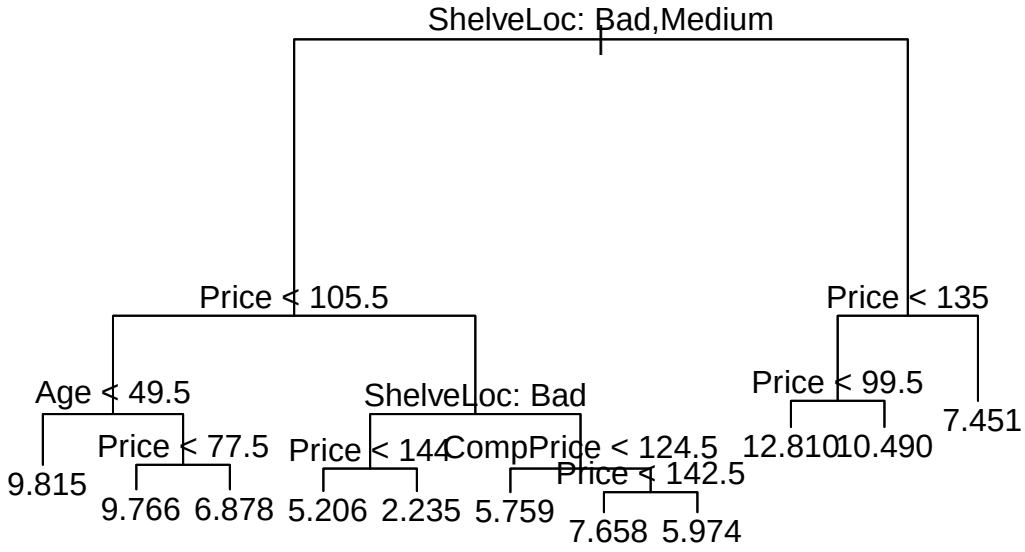
MSE is 2.399161

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

```
set.seed(10086)
cv.carset = cv.tree(tree.car)
plot(cv.carset$size, cv.carset$dev, type = "b")
```




```
prune.carset = prune.tree(tree.car, best = 11)
plot(prune.carset)
text(prune.carset,pretty=0)
```




```
tree.pred1 <- predict(prune.carset,newdata = test.data)
mean((tree.pred1 - test.data$Sales)^2)
```

```
## [1] 5.051975
```

Its not getting better acutally it increases MSE to 3.034282. (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)
set.seed(1)
#eleven variales so mttry = 10
ba.tre = randomForest(Sales~.,data=train.data,mtry = 10, importance = TRUE)
ba.pred = predict(ba.tre,newdata=test.data)
mean((ba.pred-test.data$Sales)^2)
```

```
## [1] 2.626902
```

```
importance(ba.tre)
```

```
##               %IncMSE IncNodePurity
## CompPrice   27.879430     194.99703
## Income       7.291594     109.74408
```

```
## Advertising 14.753101     138.89389
## Population  -1.930149      63.70710
## Price       67.658308     637.87761
## ShelveLoc   76.461331     699.23720
## Age         20.927646     193.10387
## Education    2.664442      57.78174
## Urban       -1.228518      15.19695
## US           3.453157      11.62414
```

```
library(randomForest)
set.seed(1)
car.im = randomForest(Sales~Price+ShelveLoc+CompPrice,data=train.data,mtry = 3)
pred.car2 = predict(car.im,newdata=test.data)
mean((pred.car2-test.data$Sales)^2)
```

```
## [1] 3.402758
```

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

```
data(Hitters)
Hitters <- na.omit(Hitters)
Hitters$Salary <- log(Hitters$Salary)
```

(b) Split the data into training and testing sets for cross validation purposes.

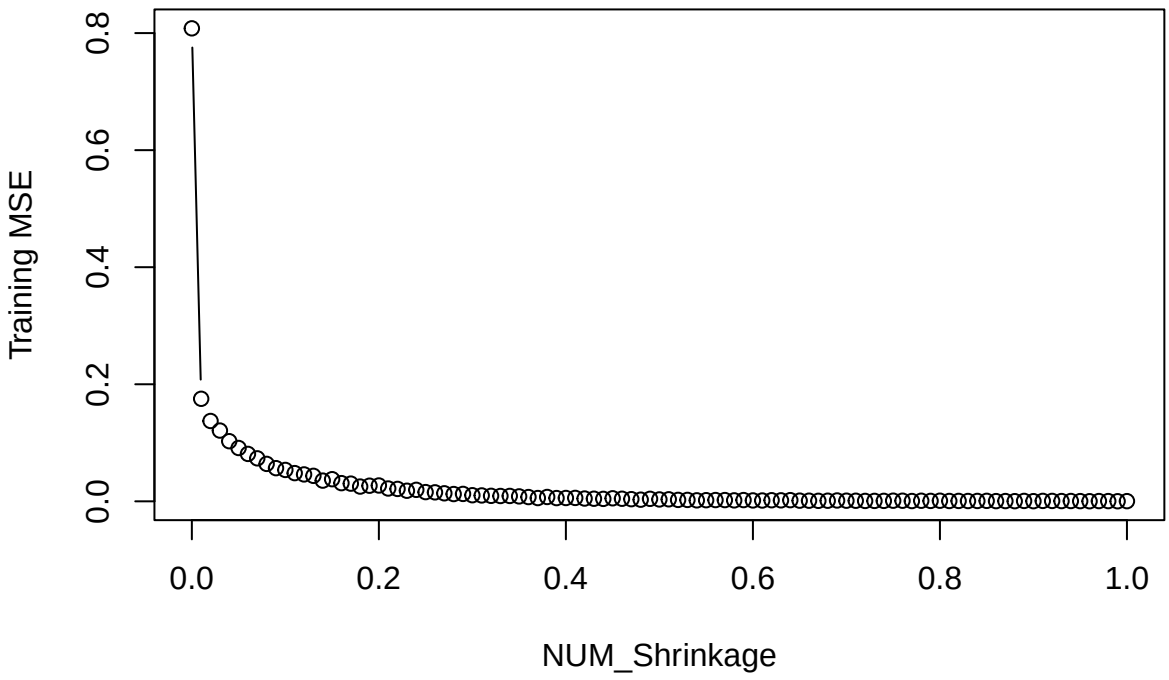
```
train.5 <- sample(nrow(Hitters),nrow(Hitters)*0.7)
train.set5 <- Hitters[train.5, ]
test.set5 <- Hitters[-train.5, ]
```

(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

```
library(gbm)
```

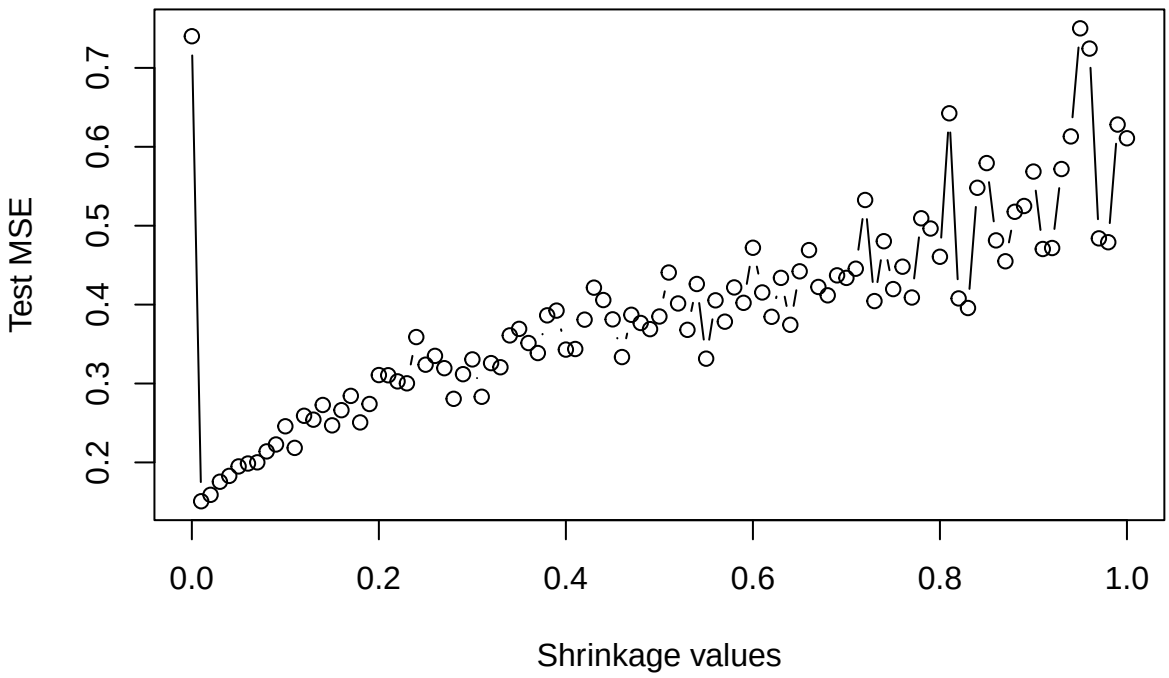
```
## Loaded gbm 2.1.8
```

```
#set.seed(10086)
num5 = seq(0, 1, by = 0.01)
train.error = rep(NA, length(num5))
for (i in 1:length(num5)) {
boo.hitters = gbm(Salary ~ ., data = train.set5, distribution = "gaussian", n.trees = 1000, shrinkage =
pred.5 = predict(boo.hitters, train.set5, n.trees = 1000)
train.error[i] = mean((pred.5 - train.set5$Salary)^2)
}
plot(num5, train.error, type = "b", xlab = "NUM_Shrinkage", ylab = "Training MSE")
```



(d) Produce a plot similar to the last one, but this time using the test set MSE

```
test.error <- rep(NA, 101)
set.seed(10086)
for (i in 1:101) {
boost.d = gbm(Salary ~ ., data = train.set5, distribution = "gaussian", n.trees = 1000, shrinkage = num5
pred.5d = predict(boost.d, test.set5, n.trees = 1000)
test.error[i] = mean((pred.5d - test.set5$Salary)^2)
}
plot(num5, test.error, type = "b", xlab = "Shrinkage values", ylab = "Test 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.

```
lin5.fit = lm(Salary ~ ., data = train.set5)
pred.5 = predict(lin5.fit, test.set5)
mean((pred.5 - test.set5$Salary)^2)
```

```
## [1] 0.3407823
```

```
glm.fit <- glm(Salary~poly(AtBat+Hits+HmRun+Runs+RBI+Walks+Years+CAtBat+CHits, 4), data = train.set5)
pred.5.2 <- predict(glm.fit, test.set5)
mean((pred.5.2 - test.set5$Salary)^2)
```

```
## [1] 0.2727254
```

(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)

```
set.seed(10086)
#train.error
tree.5.3=tree(Salary~.,data=Hitters,subset=train.5)
pre5.3=predict(tree.5.3,newdata=train.set5)
table1 <- table(pre5.3,Hitters$Salary[train.5])
mean((pre5.3 - train.set5$Salary)^2)
```

```
## [1] 0.1807587
```

```
#test.error
pre5.3.tra=predict(tree.5.3,newdata=test.set5)
mean((pre5.3.tra - test.set5$Salary)^2)
```

```
## [1] 0.3104104
```

boosting is better