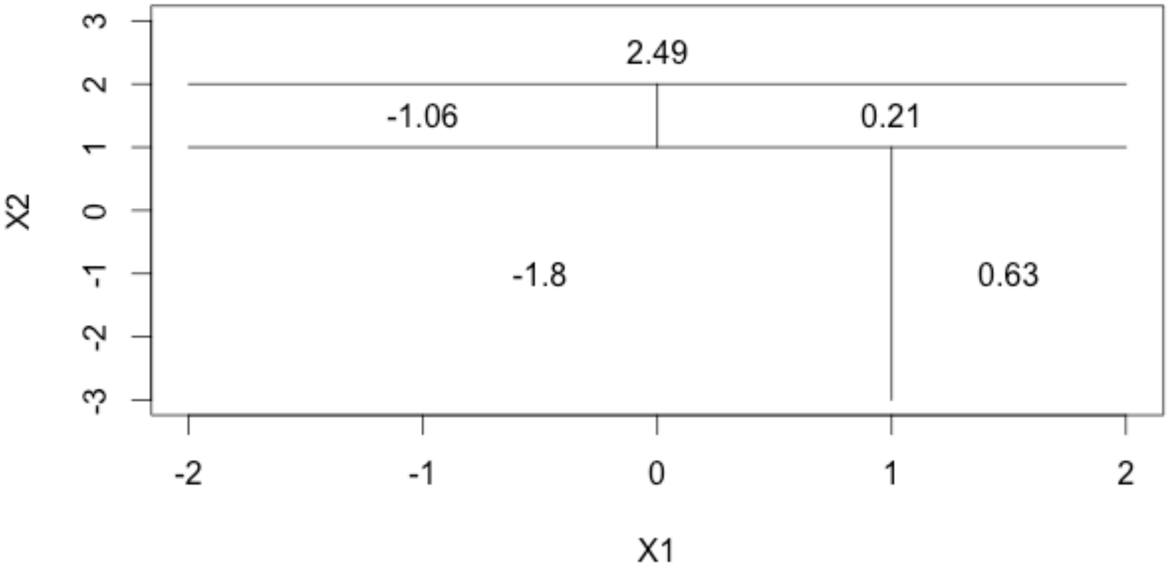
Assignment 11
Shaheed Shihan
Modern Applied Statistics

Chapter 8

4. a. If $X_1 > 1$ then Y is 5. Otherwise if $X_2 < 1$ then its 15.
If $0 < X_1 < 1$ then Y is 0 and 10. Otherwise if $X_2 < 1$ and $X_1 < 0$ then Y is 3.

b.

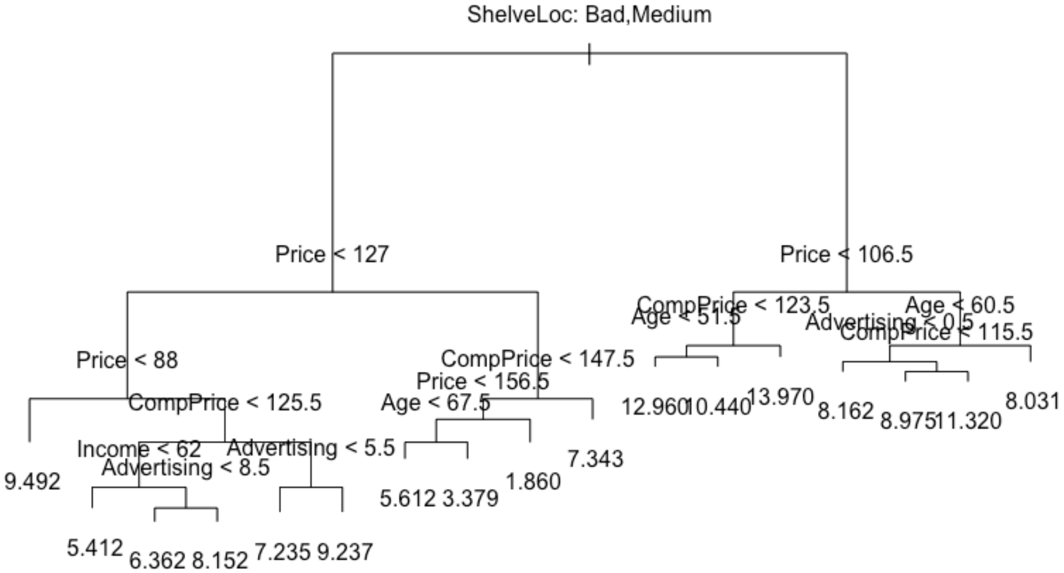


8. In the lab, a classification tree was applied to the Carseats data set after converting Sales into a qualitative response variable. Now we will seek to predict Sales using regression trees and related approaches, treating the response as a quantitative variable.

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

See R code.

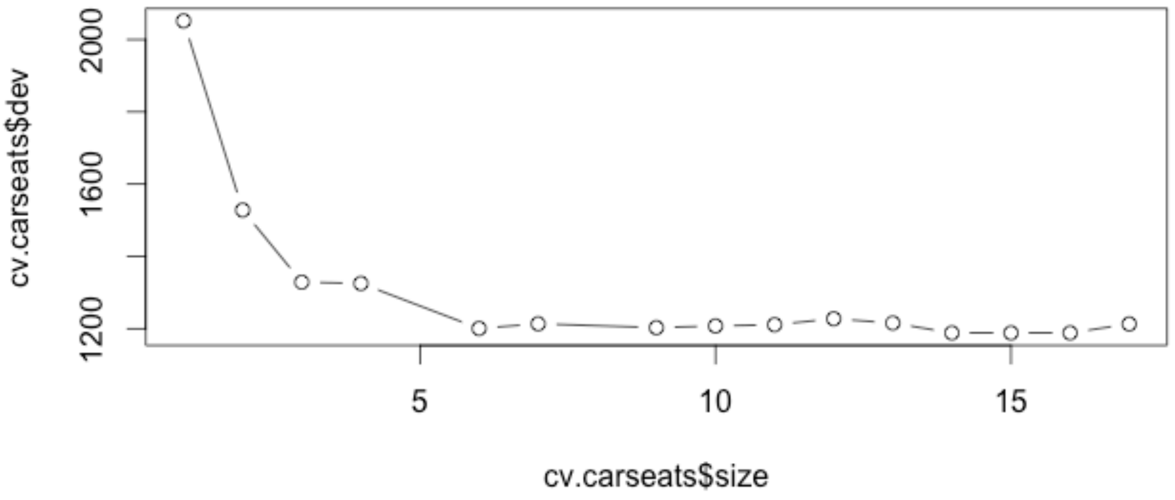
(b) Fit a regression tree to the training set. Plot the tree, and interpret the results. What test MSE do you obtain?



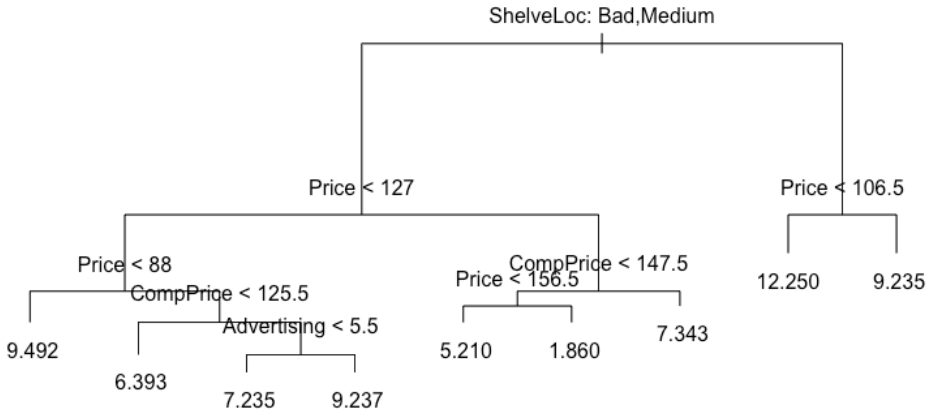
```
> mean((yhat - d.test$Sales)^2)
[1] 4.883877
```

Test MSE is around 4.88.

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



Minimum occurs at 8.

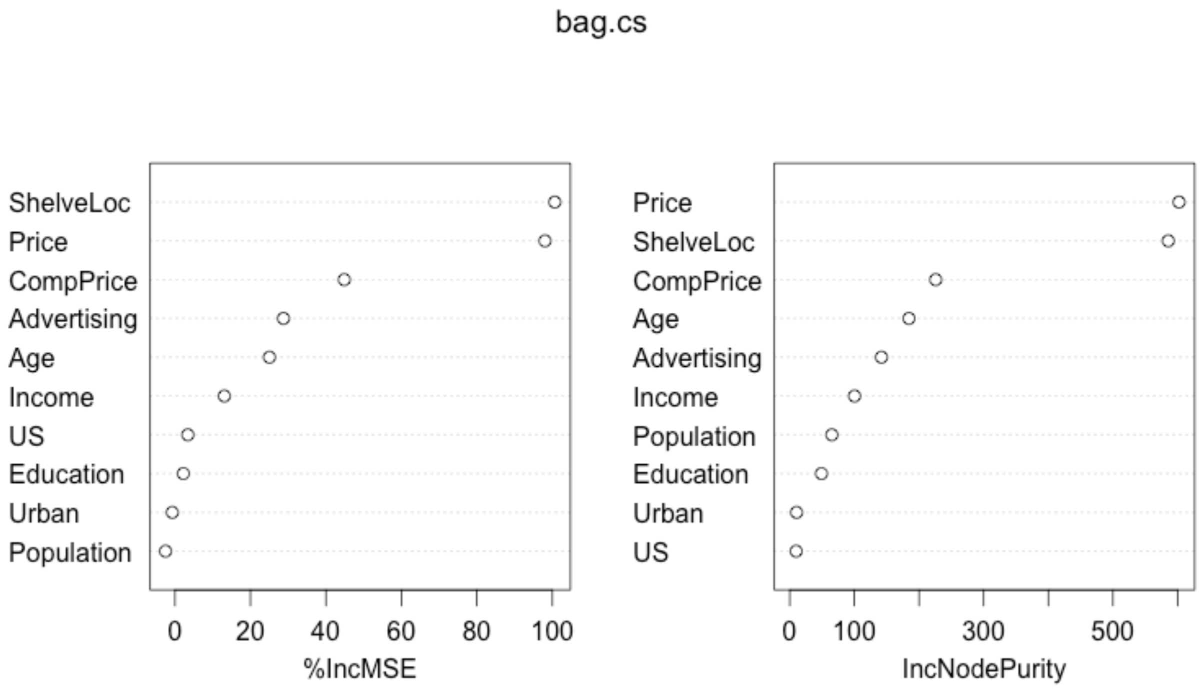


```
> mean((yhat2 - d.test$Sales)^2)
[1] 4.900529
```

Pruning the tree increases the MSE to 4.90.

(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.

```
> mean((yhat3 - d.test$Sales)^2)
[1] 3.058153
```

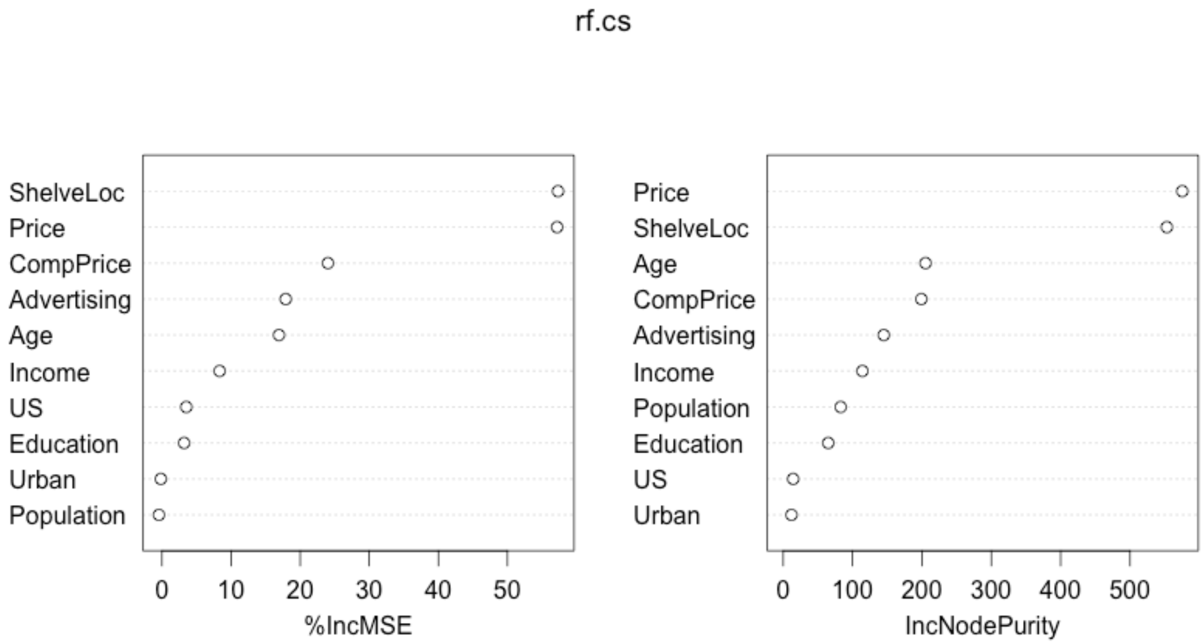


Therefore, price and shelve location are the two most important variables.

(e) Use random forests to analyze this data. What test MSE do you obtain? Use the importance () function to determine which variables are most important. Describe the effect of m, the number of variables considered at each split, on the error rate obtained.

```
> tab
    [,1]
2 3.824597
3 3.317322
4 3.153128
5 3.118554
6 3.010419
```

We can see that a m = 6 gives us the lowest MSE. The MSE goes down as the number of variables goes up.



Price and shelve location are still the two most important variables.

9. This problem involves the OJ data set which is part of the ISLR package.

(a) Create a training set containing a random sample of 800 observations, and a test set containing the remaining observations.

See R Code.

(b) Fit a tree to the training data, with Purchase as the response and the other variables as predictors. Use the summary() function to produce summary statistics about the tree, and describe the results obtained. What is the training error rate? How many terminal nodes does the tree have?

```
> tree.oj <- tree(Purchase~., data = train.oj)
> summary(tree.oj)

Classification tree:
tree(formula = Purchase ~ ., data = train.oj)
Variables actually used in tree construction:
[1] "LoyalCH"      "PriceDiff"     "ListPriceDiff"
Number of terminal nodes:  6
```

```
Residual mean deviance:  0.7428 = 589.8 / 794
Misclassification error rate: 0.1675 = 134 / 800
```

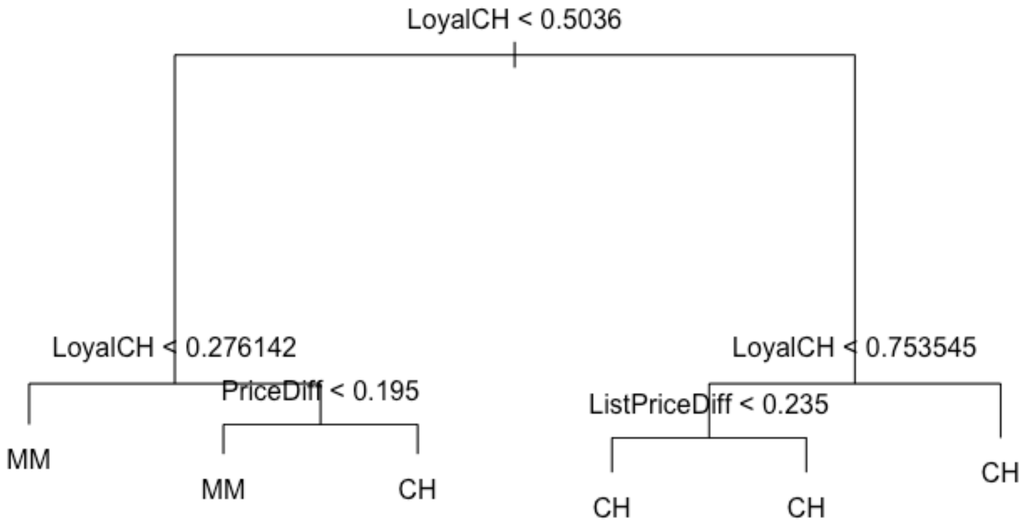
The number of terminal nodes is 6 and the misclassification rate is 16.75%.

(c) Type in the name of the tree object in order to get a detailed text output. Pick one of the terminal nodes, and interpret the information displayed.

```
> tree.oj
node), split, n, deviance, yval, (yprob)
     * denotes terminal node

 1) root 800 1068.00 CH ( 0.61250 0.38750 )
   2) LoyalCH < 0.5036 351  411.90 MM ( 0.27350 0.72650 )
     4) LoyalCH < 0.276142 167  122.40 MM ( 0.11976 0.88024 ) *
     5) LoyalCH > 0.276142 184  249.50 MM ( 0.41304 0.58696 )
      10) PriceDiff < 0.195 81   80.51 MM ( 0.19753 0.80247 ) *
      11) PriceDiff > 0.195 103  140.00 CH ( 0.58252 0.41748 ) *
   3) LoyalCH > 0.5036 449  333.90 CH ( 0.87751 0.12249 )
     6) LoyalCH < 0.753545 184  209.10 CH ( 0.74457 0.25543 )
      12) ListPriceDiff < 0.235 75  103.90 CH ( 0.52000 0.48000 ) *
      13) ListPriceDiff > 0.235 109   71.31 CH ( 0.89908 0.10092 ) *
     7) LoyalCH > 0.753545 265   71.76 CH ( 0.96981 0.03019 ) *
```

The ones with the asterisk are terminal nodes. I'll choose terminal node labelled 12 which is List price difference. It has a split criterion of ListPriceDiff < 0.235 and a deviation of 90. There are 75 observations in this split out of which 52% of the observations take that value and the rest take the value of CH.

(d) Create a plot of the tree, and interpret the results.



The most significant predictor appears to be LoyalCH since that is used as the primary split as well as for the splitting criterion for the branches.

(e) Predict the response on the test data, and produce a confusion matrix comparing the test labels to the predicted test labels. What is the test error rate?

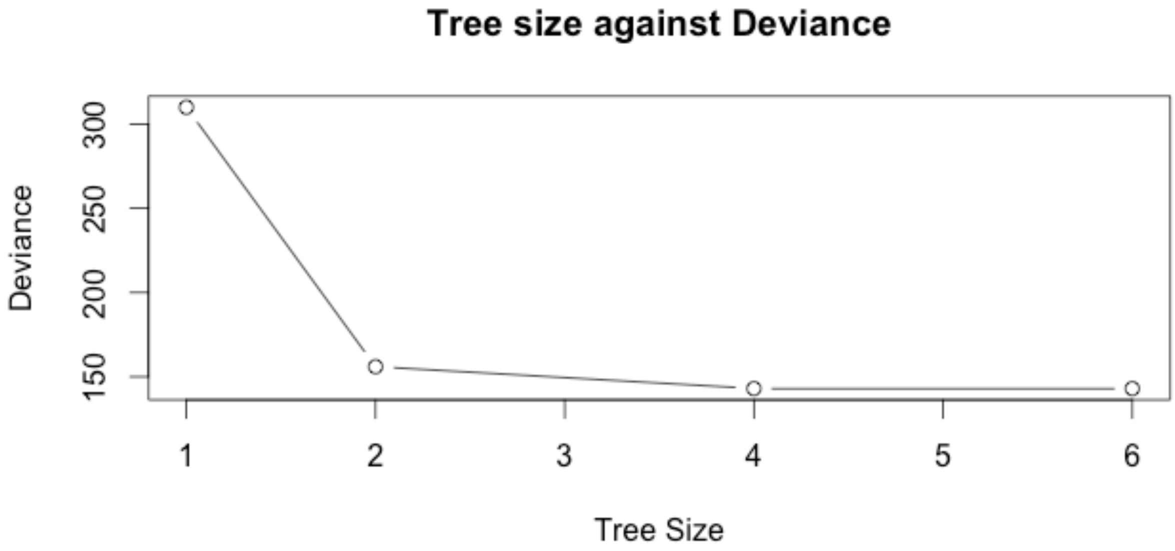
```
> table(pred, test.oj$Purchase)

pred  CH  MM
  CH 145  40
  MM  18  67

> 1-sum(diag(table(pred,test.oj$Purchase)))/nrow(test.oj)
[1] 0.2148148
```

The test error rate is 21.48%.

(f) Apply the cv.tree() function to the training set in order to determine the optimal tree size.

See R code.

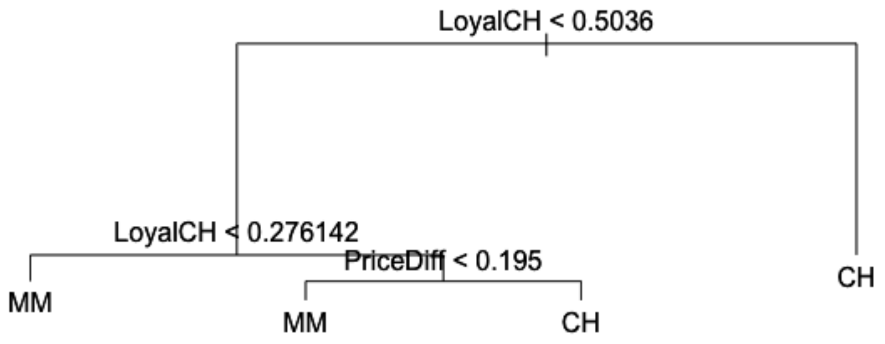
(g) Produce a plot with tree size on the x-axis and cross-validated classification error rate on the y-axis.



(h) Which tree size corresponds to the lowest cross-validated classification error rate?

According to the plot, a 4 node tree produces the lowest cross validated classification error rate.

(i) Produce a pruned tree corresponding to the optimal tree size obtained using cross-validation. If cross-validation does not lead to selection of a pruned tree, then create a pruned tree with five terminal nodes.



(j) Compare the training error rates between the pruned and unpruned trees. Which is higher?

```
> summary(oj.prune)

Classification tree:
snip.tree(tree = tree.oj, nodes = 3L)
Variables actually used in tree construction:
[1] "LoyalCH"   "PriceDiff"
Number of terminal nodes:  4
Residual mean deviance:  0.8503 = 676.8 / 796
Misclassification error rate: 0.1675 = 134 / 800

> summary(tree.oj)
Classification tree:
tree(formula = Purchase ~ ., data = train.oj)
Variables actually used in tree construction:
[1] "LoyalCH"       "PriceDiff"     "ListPriceDiff"
Number of terminal nodes:  6
Residual mean deviance:  0.7428 = 589.8 / 794
Misclassification error rate: 0.1675 = 134 / 800
```

It appears that the misclassification rate for both the pruned and unpruned tree is the same, i.e 16.75%

(k) Compare the test error rates between the pruned and unpruned trees. Which is higher?

```
> table(pred1, test.oj$Purchase)

pred1  CH  MM
   CH 145  40
   MM  18  67
> 1-sum(diag(table(pred1,test.oj$Purchase)))/nrow(test.oj)
[1] 0.2148148
```

The error rates stay the same for both. This led me to wonder if there was an error in my code but I couldn’t find one.

10. We now use boosting to predict Salary in the Hitters data set.

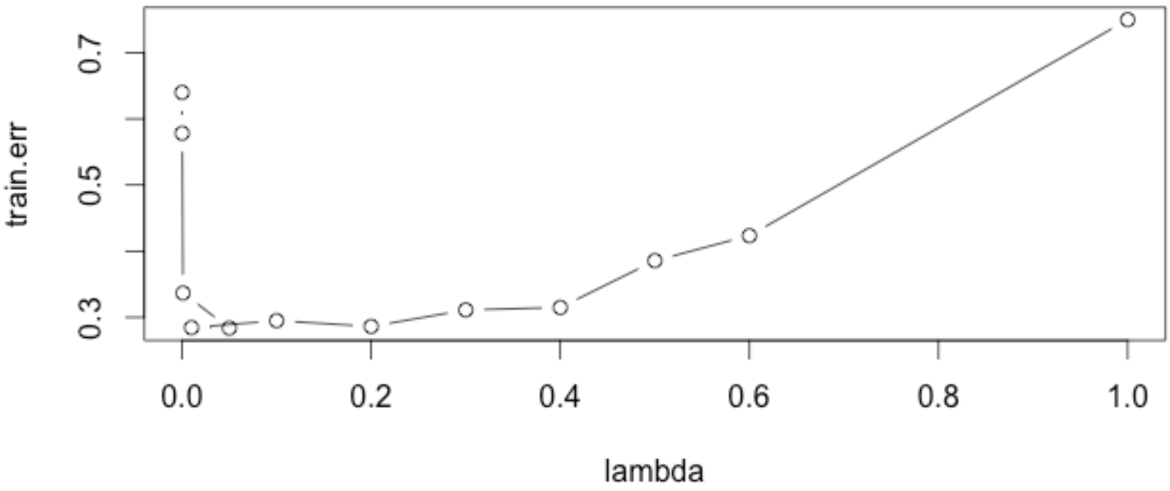
(a) Remove the observations for whom the salary information is unknown, and then log-transform the salaries.

See R code

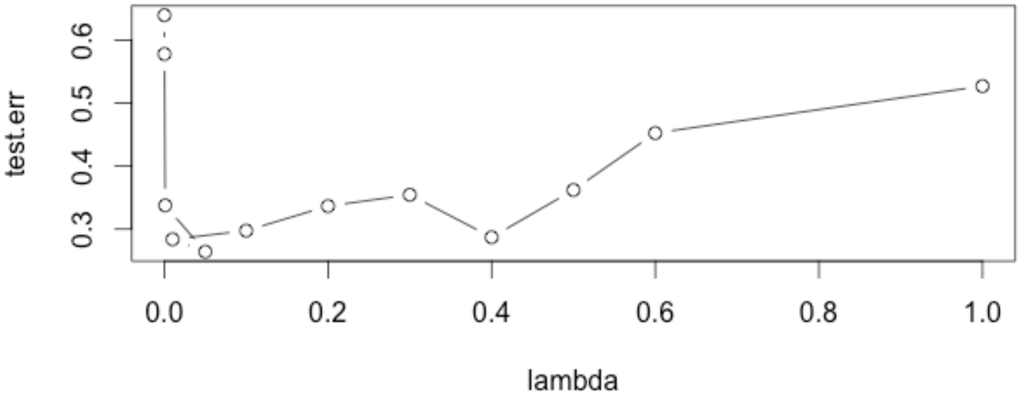
(b) Create a training set consisting of the first 200 observations, and a test set consisting of the remaining observations.

See R code

(c) Perform boosting on the training set with 1,000 trees for a range of values of the shrinkage parameter λ. Produce a plot with different shrinkage values on the x-axis and the corresponding training set MSE on the y-axis.



(d) Produce a plot with different shrinkage values on the x-axis and the corresponding test set MSE on the y-axis.



```
> min(test.err)
[1] 0.2641891
> lambda[which.min(test.err)]
[1] 0.05
```

The test error rate is 26.4% for a lambda of 0.05.

(e) Compare the test MSE of boosting to the test MSE that results from applying two of the regression approaches seen in Chapters 3 and 6.

I used glm and PCR regression. Both had test error rates higher than boosting.

```
For glm
> mean((pred2-H.test$Salary)^2)
[1] 0.4917959
for PCR
> mean((pred3-H.test$Salary)^2)
[1] 0.498467
```

(f) Which variables appear to be the most important predictors in the boosted model?

```
> summary(hitters.boost)
           var    rel.inf
CAtBat      CAtBat 17.7994950
CRBI         CRBI 12.1266542
```

```
CWalks      CWalks  8.8990728
CRuns        CRuns  7.5042930
PutOuts    PutOuts  6.8660861
CHits        CHits  6.3125894
Years        Years  5.8030810
Walks        Walks  5.4139487
CHmRun      CHmRun  5.2402823
RBI            RBI  5.2232102
AtBat        AtBat  3.8240851
Assists    Assists  3.4072687
Hits          Hits  3.3446258
HmRun        HmRun  2.5366892
Runs          Runs  2.2428065
Errors      Errors  2.1008703
Division  Division  0.5740459
NewLeague NewLeague  0.4916071
League      League  0.2892886
```

CAtBat is the most important variable thus far.

(g) Now apply bagging to the training set. What is the test set MSE for this approach?

```
> mean((yhatbag-H.test$Salary)^2)
[1] 0.2319783
```

The MSE is 23.20% which is slightly lower than the boosting method.