Pete Davis (pmd734)

Predictive Models Take Home Exam

Book Problems:

Chapter 2: #10

a. There are 506 rows that represent the number of observations, each one being a given neighborhood in Boston. There are 14 columns that represent the 14 attributes or predictor variables that we are collecting data about, such as per capita crime rate by town (CRIM), average number of rooms per dwelling (RM), proportion of non-retail business acres per town (INDUS), etc.
b. The pairs(Boston) function in R produced a 14 x 14 matrix of scatterplots of every possible relationship of two predictors. The findings revealed that many variables seem to have a correlation with one another.
c. Yes, all of the predictors have a significant relationship with the per capita crime rate. By forming a cor.test() function for the crim with each of the other variables, the correlation coefficient proved to be significant with a t-value greater than 2 and a p-value below .05.
    a. Positive relationships
        i. Crim and indus → correlation coefficient = .4065834
        ii. Crim and nox → correlation coefficient = .4209717
        iii. Crim and age → correlation coefficient = .3527343
        iv. Crim and rad → correlation coefficient = .6255052
        v. Crim and tax → correlation coefficient = .5827643
        vi. Crim and ptratio → correlation coefficient = .2899456
        vii. Crim and lstat → correlation coefficient = .4556215
    b. Negative relationships

i. Crim and zn → correlation coefficient = -.2004692

ii. Crim and chas → correlation coefficient = -.0558916

iii. Crim and rm → correlation coefficient = -.2192467

iv. Crim and dis → correlation coefficient = -.3796701

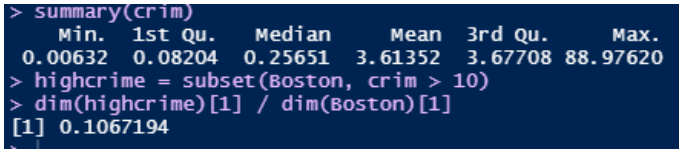
v. Crim and black → correlation coefficient = -.3850639

vi. Crim and medv → correlation coefficient = -.3883046

d.

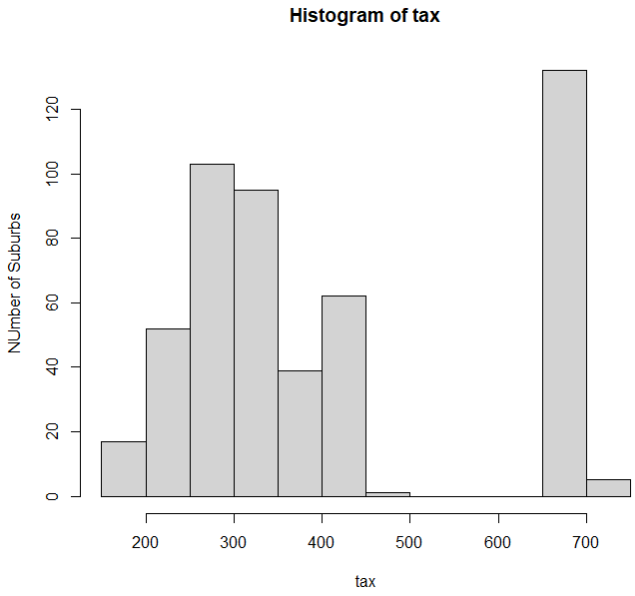
a. CRIME:

i. The median crime rate for a suburb was around .25651. However, from the summary, the max of 88.97620 showed us that we had some outliers with high crime rates. This means that there are 88.97 crimes per 1000 people every year. After further investigation, 10.67% of the suburbs had a crime rate that was greater than 10, meaning a pretty significant amount of the suburbs had particularly high crime rates. The range is from .00632 – 88.9762, which is very large.



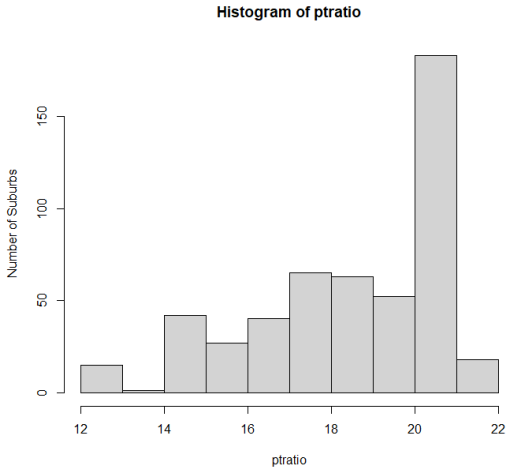
b. TAX:

i. The median tax value property-tax per $10,000 is $333. As you can see from the histogram on the right, there are a number of suburbs with tax at $666 and above. 27.08% of the total suburbs have high tax rates, those above $666 per $10,000. The range is $187 to $711, which is very large.



c. PTRATIO:

i. The median pupil-teacher ratio was 19.05, and the range was from 12.6-22.00. There are not many extreme outliers.



e. 35 suburbs bound the Charles river.

f. The median pupil-teacher ratio was 19.05, or 19 pupils per teacher.

g. Suburb 399 had the lowest median value of owner-occupied homes at 5. This suburb had a

```
> Boston[order(medv),][1,]
      crim zn indus chas   nox    rm age    dis rad tax ptratio black lstat medv
399 38.3518  0  18.1    0 0.693 5.453 100 1.4896  24 666    20.2 396.9 30.59    5
```

relatively high crime rate, above the third quartile. It had an average zn. It had a relatively high indus (proportion of non-retail business acres per town), right at the third quartile. It did not bind the Charles river. It had a high nitric oxide concentration, above the third quartile. It had a low average rooms per dwelling (rm), below the first quartile. It had the highest proportion of owner-occupied units built prior to 1940. The distance from Boston employment centers is

small, below the first quartile. It has the highest accessibility to radial highways (rad). It has a high tax rate. It has a high pupil-teacher ratio, right at the third quartile. It has the highest proportion of black people by town. It has a high percentage of lower-income status (lstat), above the third quartile. Based on this data, suburb 399 can be seen as one of the least desirable places to live in Boston.

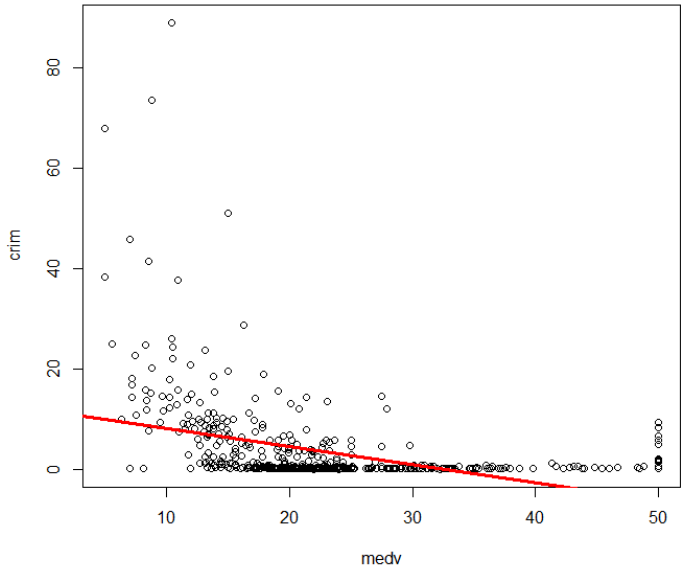
```
> summary(Boston)
      crim                zn             indus            chas              nox                rm             age              dis        
 Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000   Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130  
 1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000   1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100  
 Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000   Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207  
 Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917   Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795  
 3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000   3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188  
 Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000   Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127  
      rad              tax           ptratio          black            lstat            medv      
 Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32   Min.   : 1.73   Min.   : 5.00  
 1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38   1st Qu.: 6.95   1st Qu.:17.02  
 Median : 5.000   Median :330.0   Median :19.05   Median :391.44   Median :11.36   Median :21.20  
 Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67   Mean   :12.65   Mean   :22.53  
 3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23   3rd Qu.:16.95   3rd Qu.:25.00  
 Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90   Max.   :37.97   Max.   :50.00  
```

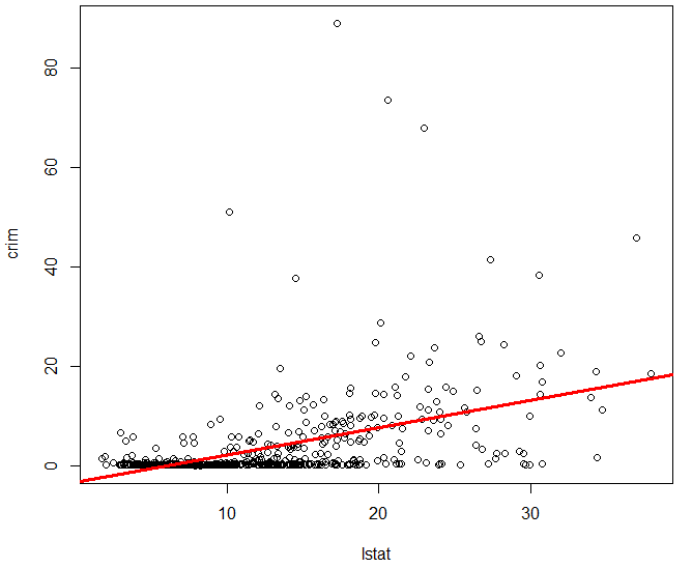
h. There are 64 suburbs that average more than seven rooms per dwelling. There are 13 suburbs that average more than eight rooms per dwelling. These are higher income suburbs with the median % of lower income status below the first quartile for that of the whole data set. The median value of owner-occupied homes is also above the third quartile for Boston suburbs as a whole.

Chapter 3: #15

a. After running a linear regression on crime rate as a function of each of the predictors the only predictor that did not have a statistically significant association with the crime rate was chas (whether or not the suburb bounded the Charles River). Every other predictor had a significant relationship with the crime rate, with a t-statistics above 2 and a p-value below .05. Below, I have included two plots to demonstrate the relationship between crim and two of the predictors. Crime rate and median value of owner-occupied homes have a negative relationship. For every $1000 increase in

median value of homes in the suburb, the crime rate goes down .363. Crime rate and percentage of lower status of the population have a positive relationship. For every 1% increase in lower income status of the suburb, the crime rate goes up 0.548.

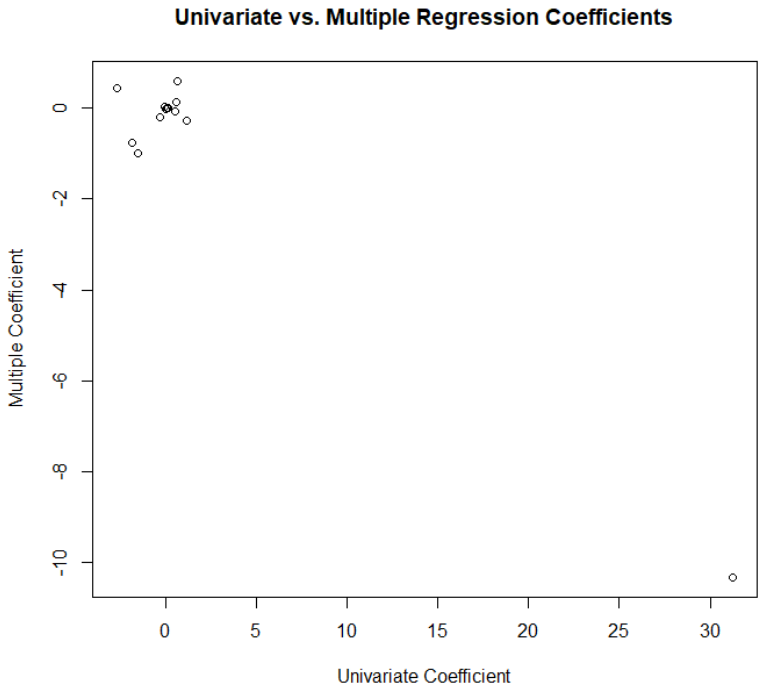




b. The multiple linear regression seems to expose what the individual linear regression couldn't, that some of the variables are related through correlation and not causation. This MLR revealed that we could certainly reject the null hypothesis for dis and rad, and we could possibly reject the null hypothesis for medv, black, and zn. All of these variables had coefficients with t-statistics greater than 2 and p-values less than 0.05.

```
lm(formula = crim ~ ., data = Boston)

Residuals:
   Min     1Q Median     3Q    Max
-9.924 -2.120 -0.353  1.019 75.051

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  17.033228   7.234903   2.354 0.018949 *
zn            0.044855   0.018734   2.394 0.017025 *
indus        -0.063855   0.083407  -0.766 0.444294
chas         -0.749134   1.180147  -0.635 0.525867
nox         -10.313535   5.275536  -1.955 0.051152 .
rm            0.430131   0.612830   0.702 0.483089
age           0.001452   0.017925   0.081 0.935488
dis          -0.987176   0.281817  -3.503 0.000502 ***
rad           0.588209   0.088049   6.680 6.46e-11 ***
tax          -0.003780   0.005156  -0.733 0.463793
ptratio      -0.271081   0.186450  -1.454 0.146611
black        -0.007538   0.003673  -2.052 0.040702 *
lstat         0.126211   0.075725   1.667 0.096208 .
medv         -0.198887   0.060516  -3.287 0.001087 **
```

c. Like I said in part (b), the MLR revealed that there were not as many variables that had a significant relationship with the crime rate as the individual linear regression models indicated.



d. The variables ndus, nox, dis, ptratio, and medv all appear to have a non-linear relationship. The squared and cubed coefficients the regressions with these variables had t-statistics that were greater than 2 and a p-values that were less than 0.05. Age appeared to have a non-linear relationship as well. The squared coefficient had a t-stat slightly below 2, but the cubed coefficient had a t-stat that was above 2 and a p-value that was below 0.05. For these variables, we would reject the null hypothesis that the non-linear coefficients = 0.

Chapter 6: #9

a. The data is now split into a training set of 600 observations and a test set of 177 observations.

```
library(ISLR)
library(caret)
attach(College)

train = data.frame(College)
test = data.frame(College)

n = dim(train)[1]

#Sample (in this case with uniform distribution)
tr = sample(1:777, #The values that will be sampled
            size = 600, #The size of the sample
            replace = FALSE) #without replacement

train = train[tr,] #the rows of train will be the ones sampled
test = test[-tr,] #and test will be everything else (thus, out-of-sample)

preObj <- preProcess(train, method = c('center', 'scale'))

train <- predict(preObj, train)
test <- predict(preObj, test)
```

b. The RMSE of the linear model prediction was 0.31098.

c. The MSE of this prediction from the ridge regression was 0.29464. The optimal lambda used from cross-validation was .09459.

d. The MSE of this prediction from the lasso regression was 0.3095178. The optimal lambda used from cross-validation was 0.0005048105. There are 15 non-zero coefficients and they are listed below. The intercept is not included as a non-zero coefficient.

```
> coef.L[coef.L != 0]
 (Intercept)   PrivateYes       Accept       Enroll    Top10perc    Top25perc  F.Undergrad  P.Undergrad     Outstate   Room.Board        Books 
0.078346018 -0.105635081  1.043147424 -0.246983852  0.231625523 -0.069057287  0.075301063  0.026230568 -0.082423691  0.041796920  0.008386538 
    Personal          PhD    S.F.Ratio  perc.alumni       Expend 
0.001448003 -0.043990552  0.018782824  0.001483850  0.081435171 
```

e. To the right, you can see the R code I used to perform the PCR. The root mean squared error was 0.3771579. Also, the summary of the fit shows that the cross-validation selected M = 10 components.

```
> pcr_fit <- train(x = xtrain, y = ytrain,
+                  method = 'pcr',
+                  trControl = trainControl(method = 'cv', number = 10),
+                  tuneGrid = expand.grid(ncomp = 1:10))
> #this will show the error of the prediction
> (pcr_info = postResample(predict(pcr_fit, xtest), ytest))
     RMSE  Rsquared       MAE 
0.3771579 0.8801608 0.2238508 
>
> # this will show a summary of the prediction with the number of components
> summary(pcr_fit)
Data:   X dimension: 600 17
        Y dimension: 600 1
Fit method: svdpc
Number of components considered: 10
TRAINING: % variance explained
         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps  10 comps
X          33.28    57.12    64.59    70.44    76.01    81.16    84.99    88.51    91.65     93.96
.outcome   10.07    73.52    73.77    80.76    83.68    83.73    83.88    84.86    85.28     85.45
>
```

f. To the right, you can see the R code I used to perform the PLS. The root mean squared error was 0.3091268. Also, the summary of the fit shows that the cross-validation selected M = 10 components.

```
> pls_fit <- train(x = xtrain, y = ytrain,
+                  method = 'pls',
+                  trControl = trainControl(method = 'cv', number = 10),
+                  tuneGrid = expand.grid(ncomp = 1:10))
> #this will show the error of the prediction
> (pls_info = postResample(predict(pls_fit, xtest), ytest))
     RMSE  Rsquared       MAE 
0.3091268 0.9139673 0.1723655 
>
> # this will show a summary of the prediction with the number of components
> summary(pls_fit)
Data:   X dimension: 600 17
        Y dimension: 600 1
Fit method: oscorespls
Number of components considered: 10
TRAINING: % variance explained
         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps  10 comps
X          25.91    53.00    62.65    65.36    69.71    74.32    77.89    81.16    83.29     86.22
.outcome   77.27    82.13    87.46    90.95    92.56    93.07    93.17    93.27    93.35     93.37
>
```

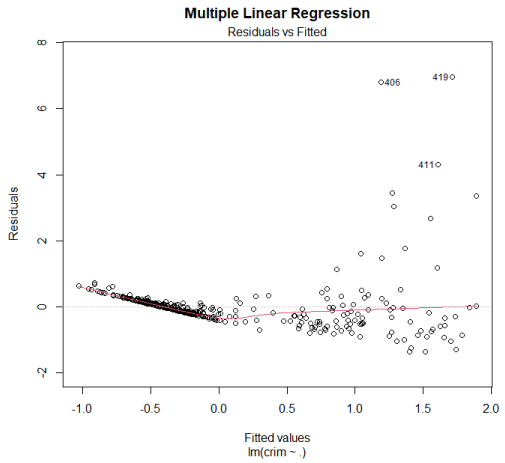
g. Below are the five methods and their corresponding RMSEs. The Ridge Regression shrinkage method had the lowest RMSE of 0.29464, and would thus be the best model to use for prediction.

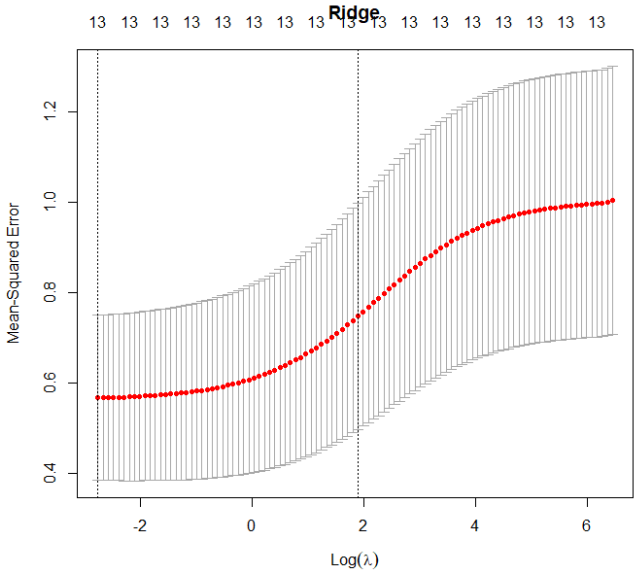
1. Ridge – RMSE: 0.29464
2. PLS – RMSE: 0.3091268
3. Lasso – RMSE: 0.3095178
4. Linear – RMSE: 0.31098

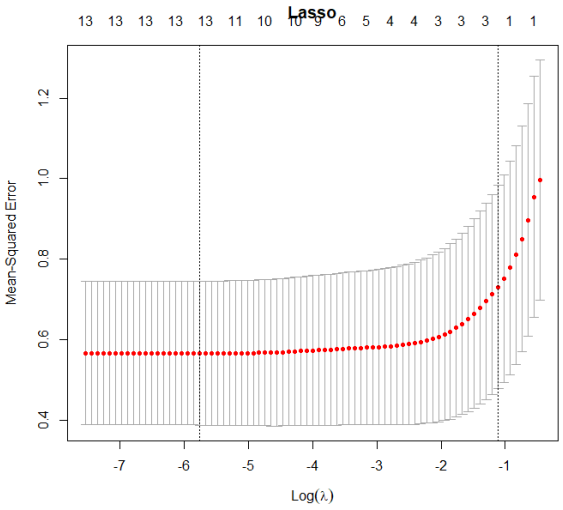
5. PCR – RMSE: 0.3771579

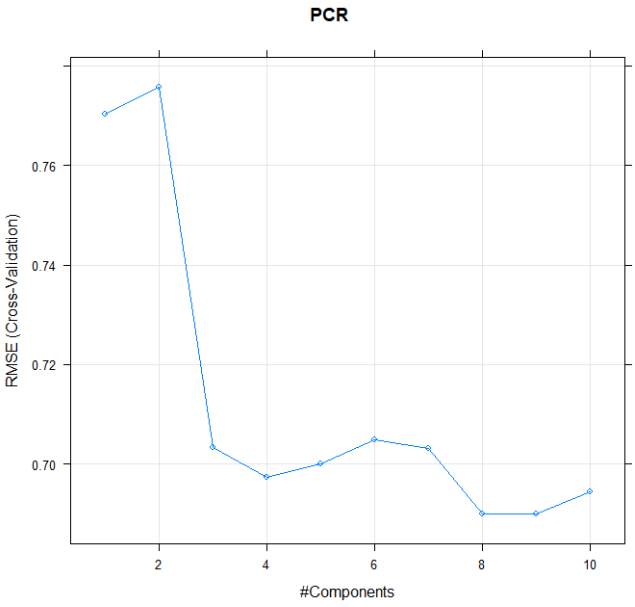
Chapter 6: Problem 11

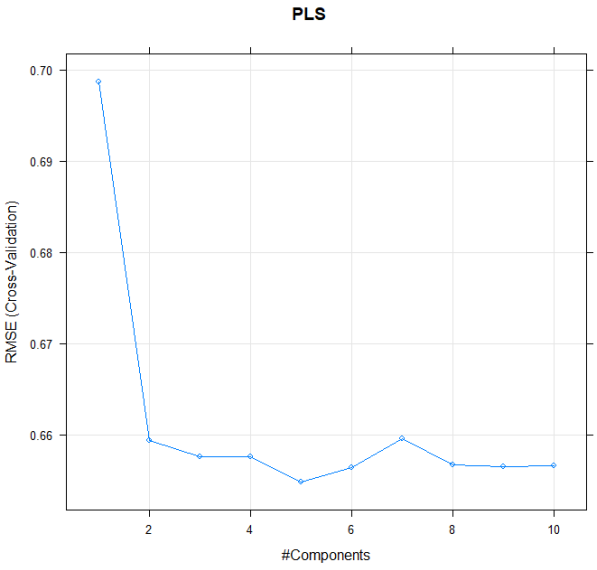
a. I ran a multiple linear regression, ridge regression, lasso regression, PCR, and PLS to try and predict the per capita crime rate based on the other predictors in the Boston data set.











b. To the right, a picture of my code illustrates the RMSEs of each model. They found using cross-validation. The numbers were all pretty similar, but the Multiple Linear Regression model had the lowest RMSE, 1.012721. Therefore, the best model is the Multiple Linear Regression model.

```
> print(RMSE_linear)
[1] 1.012721
> print(RMSE_ridge)
[1] 1.024805
> print(RMSE_lasso)
[1] 1.014697
> print(pcr_info)
     RMSE  Rsquared       MAE
1.0462918 0.3448423 0.3602000
> print(pls_info)
     RMSE  Rsquared       MAE
1.0172880 0.3843102 0.3478014
```

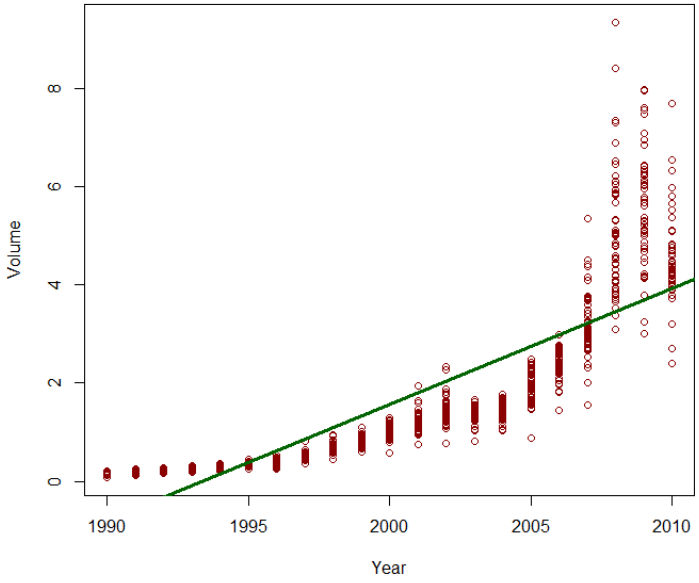
c. The chosen model, Multiple Linear Regression, includes all of the features in the data set. This is because I modeled the fit with every single predictor in the data set. However, you can see that only four of the predictors (dis, rad, black, and medv) are statistically significant. They are the only predictors that have coefficients with t-statistics greater than 2 in magnitude and p-values less than 0.05.

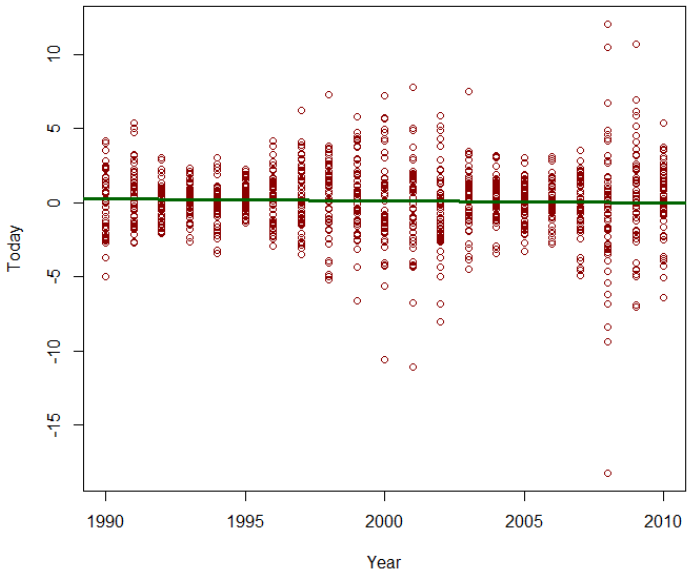
```
Call:
lm(formula = crim ~ ., data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-1.3645 -0.2236 -0.0413  0.1088  6.9664

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.775e-16  3.672e-02   0.000 1.000000
zn           1.019e-01  5.494e-02   1.854 0.064439 .
indus       -5.748e-02  7.185e-02  -0.800 0.424263
chas        -2.022e-02  3.882e-02  -0.521 0.602847
nox         -1.286e-01  7.754e-02  -1.658 0.098055 .
rm          -1.027e-02  5.613e-02  -0.183 0.854917
age          2.143e-02  6.510e-02   0.329 0.742151
dis         -2.014e-01  7.641e-02  -2.635 0.008749 **
rad          5.666e-01  9.713e-02   5.833 1.15e-08 ***
tax         -5.929e-02  1.087e-01  -0.546 0.585712
ptratio     -7.283e-02  5.185e-02  -1.405 0.160923
black       -1.672e-01  4.375e-02  -3.822 0.000154 ***
lstat        1.081e-01  6.838e-02   1.581 0.114783
medv        -1.550e-01  7.149e-02  -2.168 0.030745 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7344 on 386 degrees of freedom
Multiple R-squared:  0.4783,    Adjusted R-squared:  0.4607
F-statistic: 27.22 on 13 and 386 DF,  p-value: < 2.2e-16
```

Chapter 4: Problem 10

```
> summary(Weekly)
      Year           Lag1               Lag2               Lag3               Lag4               Lag5              Volume
 Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747
 1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580   1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202
 Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410   Median :  0.2380   Median :  0.2340   Median :1.00268
 Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472   Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462
 3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090   3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373
 Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821
     Today          Direction
 Min.   :-18.1950   Down:484
 1st Qu.: -1.1540   Up  :605
 Median :  0.2410
 Mean   :  0.1499
 3rd Qu.:  1.4050
 Max.   : 12.0260
```

a. Above, I have included a summary of each variable within the weekly data. After further inspection of the variables, a couple of patterns emerged. Overtime, the volume of shares being traded daily trended from an average of around $1 billion in 2000 to an average of around $4 billion in 2010. Also, over time, the average weekly return remains pretty constant, slightly above 0%. These trends are represented below.





b. In the logistic regression, the only variable that was statistically significant in predicting the direction was Lag2, which is the percentage return for the two previous weeks. The coefficient for Lag2 had a t-statistic that was greater than 2 and a p-value that was less than 0.05.

```
> summary(log_reg)

Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = "binomial", data = Weekly)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6949  -1.2565   0.9913   1.0849   1.4579

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.26686    0.08593   3.106   0.0019 **
Lag1        -0.04127    0.02641  -1.563   0.1181
Lag2         0.05844    0.02686   2.175   0.0296 *
Lag3        -0.01606    0.02666  -0.602   0.5469
Lag4        -0.02779    0.02646  -1.050   0.2937
Lag5        -0.01447    0.02638  -0.549   0.5833
Volume      -0.02274    0.03690  -0.616   0.5377
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)
```

c. From the confusion matrix, you see that we predict most cases to go UP. We predict (430 + 557/1089) = 90.6% of the cases to be UP. Of the 90.6% that we are predicting to be UP, we are predicting up 430/(430+557) = 43.57% of those has false positives. This is a huge issue. Of the negative predictions, we are predicting (48/(48+54)) = 47.06% as false negatives. Overall, we are predicting (430 + 48)/(1089) = 43.89% of our predictions falsely. We are predicting 56.11% correctly with logistic regression. We are not being very accurate in our predictions.

```
> p = predict(log_reg, type = "response")
> prediction = rep("Down", 1089)
> prediction[p > 0.5] = "Up"
> table(prediction, Direction)
          Direction
prediction Down  Up
      Down   54  48
      Up    430 557
```

d. The confusion matrix for the logistic regression of Direction as a function of the Lag2 predictor variable is on the right. The overall fraction of correct predictions is (23+583)/(1089) = 55.65%.

```
> p2 = predict(log_reg2, type = "response")
> prediction2 = rep("Down", 1089)
> prediction2[p2 > 0.5] = "Up"
> table(prediction2, Direction)
           Direction
prediction2 Down  Up
       Down   23  22
       Up    461 583
```

e. Omit

f. Omit

g. The confusion matrix for the k-nearest neighbors test with k = 1 is on the right. The overall fraction of correct predictions is (21+31)/104 = 50%.

```
> x.train = cbind(train$Lag2)
> y.train = cbind(train$Direction)
> x.test = cbind(test$Lag2)
> prediction_knn = knn(x.train, x.test, y.train, k = 1)
> table(prediction_knn, test$Direction)

prediction_knn Down Up
             1   21 30
             2   22 31
```

h. This correct prediction rate is lower for the KNN model with k =1 than that of the logistic regression. However, of the times that the direction was actually down, the logistic regression predicted a false positive (461/(461+23) = 95.24% of the time. This is alarmingly high. In contrast, of the times that the direction was actually down, the KNN model predicted a false

positive (22)/(22+21) = 51.16%. This is a much lower false positive rate. For this reason, the KNN model is a better model for prediction than the logistic regression.

i. Below, I have included the matrices for a logistic regression with three preditors, a logistic regression with two predictors, a k=3 KNN model, a k=5 KNN model, and a k=7 KNN model. Below are the overall correct prediction percentages and false positive percentages for each respecive model.

   a. Logistic regression → Predictors are Lag1, Lag2, and Lag4
      i. Correct Prediction: (58+552)/(1089) = 56.01%
      ii. False positive (426)/(426+58) = 88.02%
   b. Logistic regression → Predictors are Lag1 and Lag2
      i. Correct Prediction: (47+556)/(1089) = 55.37%
      ii. False positive: (437)/(437+47) = 90.29%
   c. KNN model with k = 3
      i. Correct Prediction: (15+42)/(104) = 54.81%
         1. 104 is the size of the test set (rows with year >= 2009)
      ii. False positive: (28)/(28+15) = 65.12%
   d. KNN model with k = 5
      i. Correct Prediction: (16+40)/(104) = 53.85%
      ii. False positive: (27)/(27+16) = 62.79%
   e. KNN model with k = 7
      i. Correct Prediction: (15+42)/(104) = 54.81%
      ii. False positive: (28)/(28+15) = 65.12%

- The logistic regression with predictors Lag1, Lag2, and Lag4 has the highest correct prediction percentage. However, of the times in which direction is down, this regression predicts a false

positive 88.02% of the time. For this reason, the best model is the KNN model with k=3 or with k= 7. The correct prediction rate is only slightly lower at 54.81%, but the false positive rate is much lower at 65.12%.

```
> # logistic regression with lag1, lag2, and lag4
> log_reg3 = glm(Direction ~ Lag1 + Lag2 + Lag4, family = "binomial", data = train)
> # summary(log_reg3)
> p3 = predict(log_reg3, type = "response")
> prediction3 = rep("Down", 1089)
> prediction3[p3 > 0.5] = "Up"
> table(prediction3, Direction)
           Direction
prediction3 Down  Up
       Down   58  53
       Up    426 552
```

```
> # logistic regression with lag1 and lag2
> log_reg4 = glm(Direction ~ Lag1 + Lag2, family = "binomial", data = train)
> # summary(log_reg4)
> p4 = predict(log_reg4, type = "response")
> prediction4 = rep("Down", 1089)
> prediction4[p4 > 0.5] = "Up"
> table(prediction4, Direction)
           Direction
prediction4 Down  Up
       Down   47  49
       Up    437 556
```

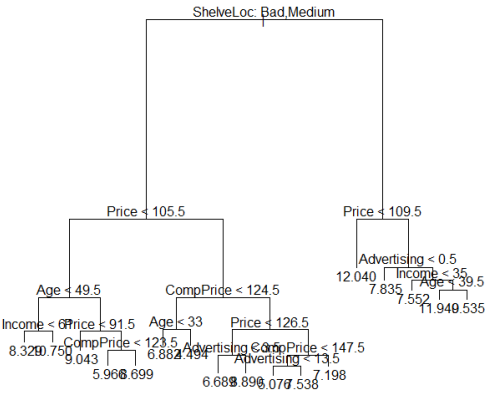
```
> prediction_knn3 = knn(x.train, x.test, y.train, k = 3)
> table(prediction_knn3, test$Direction)

prediction_knn3 Down Up
              1   15 19
              2   28 42
>
> # k = 5
> prediction_knn5 = knn(x.train, x.test, y.train, k = 5)
> table(prediction_knn5, test$Direction)

prediction_knn5 Down Up
              1   16 21
              2   27 40
>
> # k = 7
> prediction_knn7 = knn(x.train, x.test, y.train, k = 7)
> table(prediction_knn7, test$Direction)

prediction_knn7 Down Up
              1   15 19
              2   28 42
```

Chapter 8: Problem 8

a. I split the Carseats data into a test set of 300 observations and a training set of 100 observations.

```
n = dim(Carseats)[1]
#Sample (in this case with uniform distribution)
tr = sample(1:400, #The values that will be sampled
            size = 300, #The size of the sample
            replace = FALSE) #without replacement

# train and test set of the Carseats data
train = Carseats[tr,]
test = Carseats[-tr,]
```

b. In this tree diagram, we are trying to predict sales using all of the other predictor variables. I have plotted the resulting tree on the right. The root node asks whether the test observation has a ShelveLoc of Bad or Medium. If the answer is yes, then it travels left down the tree. If the answer is no, then you travel right down the tree. You repeat this at each level of the tree until you reach a leaf node, which gives you the prediction for

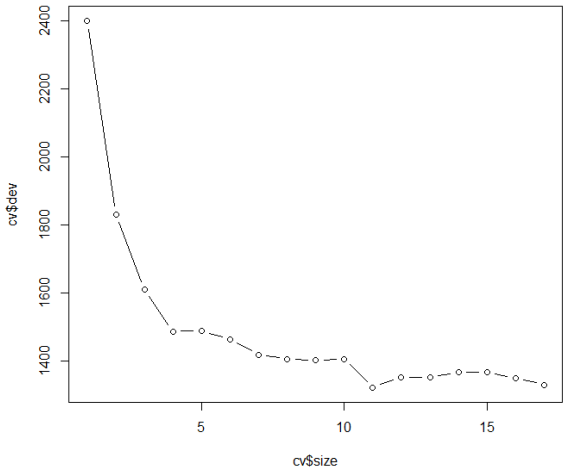


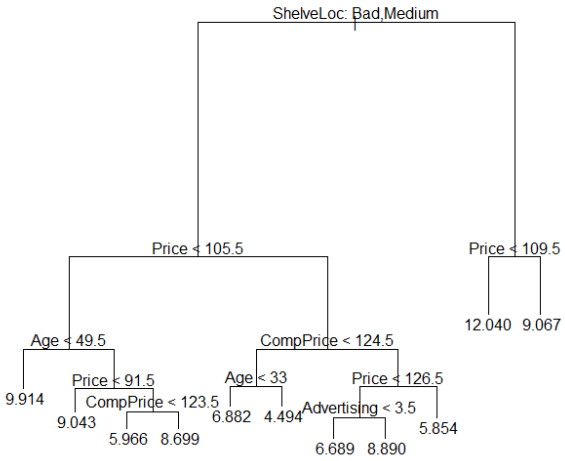

```
> # predicts the validation set
> tree.pred = predict(tree, newdata = test)
>
> #finds the MSE
> mean((tree.pred - test$Sales)^2)
[1] 4.910268
```

sales. We were able to predict sales using all of the other predictor variables using this tree method with an MSE of 4.910268.

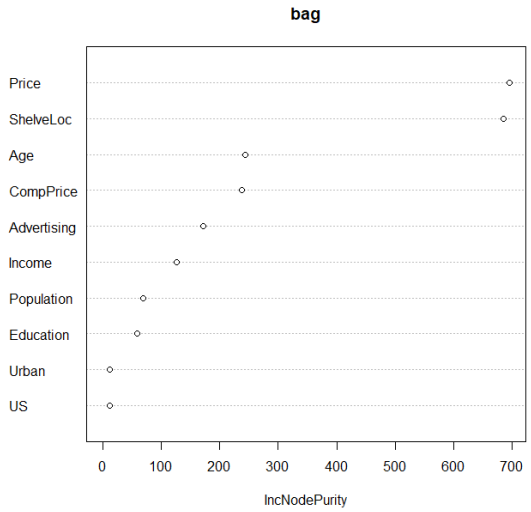
c. I used cross-validation to find the number of leaves that contributed to the lowest complexity. Below, you can see that the model with 11 leaves contributed to the lowest dev value in the cross-validation. Based on this knowledge, I created a new tree using cross validation with 11 leaves. However, this did not improve the error. The MSE of this model was 5.236696.

```
> pruned.pred = predict(pruned, newdata = test)
>
> #finds the MSE
> mean((pruned.pred - test$Sales)^2)
[1] 5.236696
```





d. Using the bagging method to analyze the data, sales was able to be predicted using the other 10 predictor variables with an MSE of 2.89051. Using the varImpPlot() function on the bagging method, we were able to see most important variables on the basis of their ability to increase the none purity. Price and ShelveLoc were the most important variables by a big margin, Age and CompPrice were third and fourth.



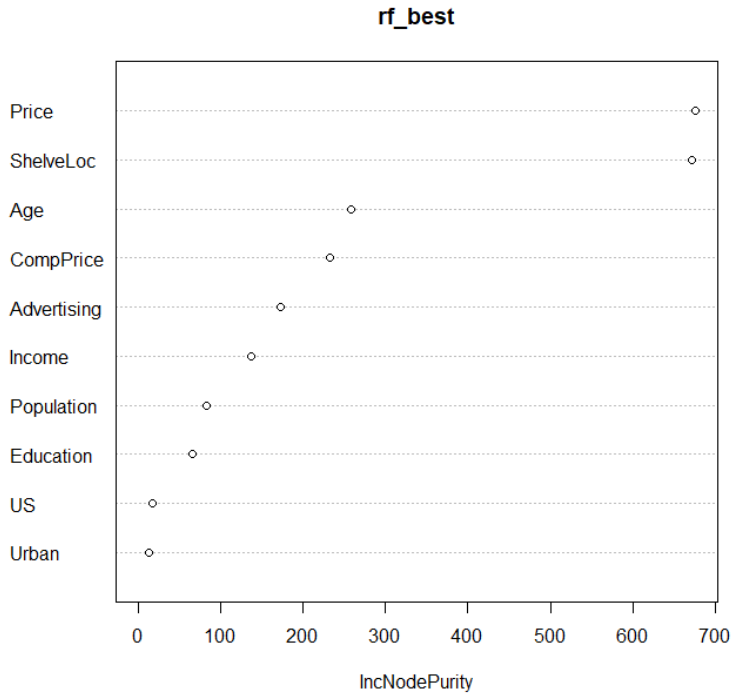

```
> bag = randomForest(Sales ~ ., data = train, mtry = 10, importance = TRUE)
> ?randomForest()
> prediction_bag = predict(bag, newdata = test)
> #Find the MSE of the bagging prediction
> mean((prediction_bag - test$Sales)^2)
[1] 2.89051
```

e. Using random forest, I was able to find the MSE's for models with from mtry = 1 to mtry = 10 to determine its effect on the MSE. To the right, you can see the MSE for each varying mtry value. The MSE went down in each incremental increase from 1:7, but it never went below the MSE value for the random forest model using mtry = 7. This MSE value was 2.793451. From here, I ran another random forest using mtry = 7, and plotted the variable importance. Price, ShelveLoc, Age and ComPrice (in order) remained the most important contributors in predicting Sales.

```
# random forest with different numbers of mtry
m = c(1:10)
for (i in m){
  rf = randomForest(Sales ~ ., data = train, mtry = i, importance = TRUE)
  prediction_rf = predict(rf, newdata = test)
  print(i)
  MSE = (mean((prediction_rf - test$Sales)^2))
  print(MSE)
}
```

```
[1] 1
[1] 4.914603
[1] 2
[1] 3.564023
[1] 3
[1] 3.070238
[1] 4
[1] 2.90339
[1] 5
[1] 2.812403
[1] 6
[1] 2.801869
[1] 7
[1] 2.793451
[1] 8
[1] 2.811897
[1] 9
[1] 2.81508
[1] 10
[1] 2.88106
```

```
> rf_best = randomForest(Sales ~ ., data = train, mtry = 7, importance = TRUE)
> importance = importance(rf_best)
> varImpPlot(rf_best, type = 2)
```

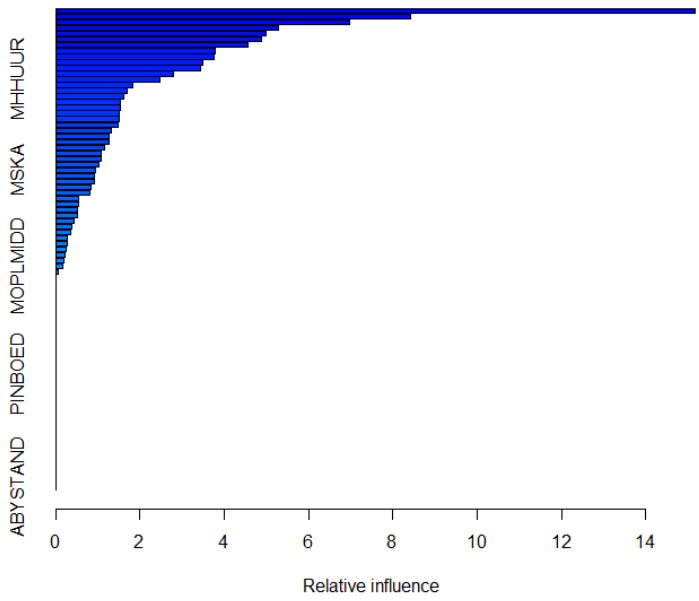


Chapter 8: Problem 11

a. After this step, the Caravan data is set is split into a training set of 1000 observations and a test set of the remaining 4822 observations.

```
n = dim(Caravan)[1]
#Sample (in this case with uniform distribution)
tr = sample(1:5822, #The values that will be sampled
            size = 1000, #The size of the sample
            replace = FALSE) #without replacement

# train and test set of the Carseats data
train = Caravan[tr,]
test = Caravan[-tr,]
```

b. To the right, you can see the summary of the relevant predictors in the boosting method. The variables that contribute the most in the prediction of Purchase are MHHUR, MSKA, and MOPLMIDD, in that order.



c. The confusion matrix only includes probabilities that were greater than 20% from the test set, as there were 4822 total observations in the training set and 4822 in this confusion matrix that does not include a 0 column. The probability that a person makes a purchase if the estimated probability is greater than 20% is 296/(296 + 4526) = 6.1385%.

```
> probs <- predict(boost, test, n.trees = 1000, type = "response")
>
> pred_boost <- ifelse(probs > 0.2, 1, 0)
> table(test$Purchase, pred_boost)
     pred_boost
          1
  No   4526
  Yes   296
```

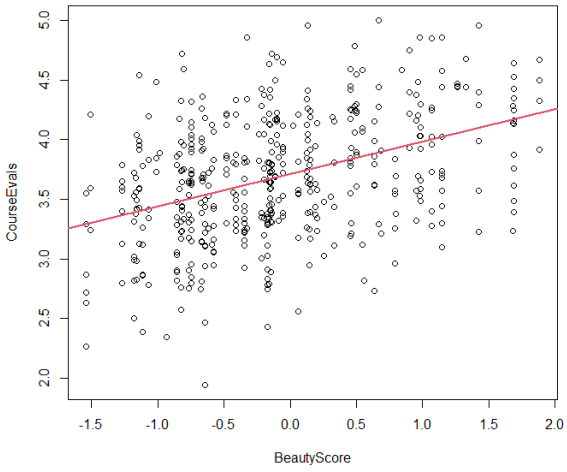
Logistical regression yielded slightly better results, a probability of 52/(53+319) = 14.0162%.

```
> pred2 =  ifelse(probs2 > .2, 1, 0)
> table(test$Purchase, pred2)
     pred2
          0    1
  No   4207  319
  Yes   244   52
```

Non-Book Problems:

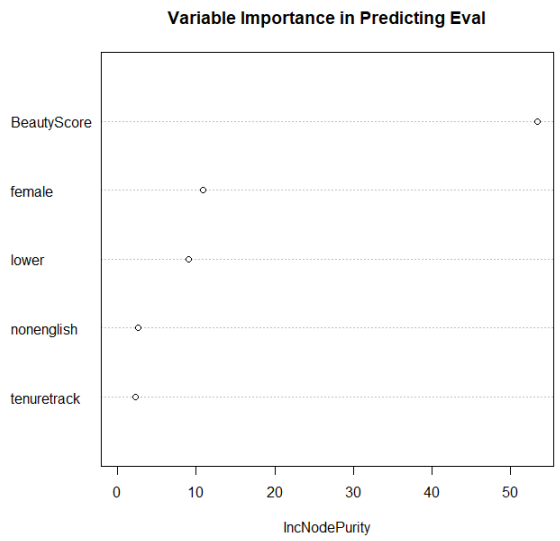
Problem 1: Beauty Pays

1. After running a linear regression with BeautyScore as the x-variable and CourseEvals as the y-variable, there is an obvious positive relationship. BeautyScore has a coefficient that is a statistically significant predictor of CourseEvals, as the t-statistic is much greater than 2 and the p-value is much lower than 0.05. However, it is interesting to see that the R-squared is .1657. This means that only 16.57% of the variance in CourseEvals can be explained by the variance of BeautyScore. So, I introduce more variable to see if there were other contributors. After running a multiple linear regression with CourseEvals as a function of all of the variables in the data set, every single variable had a statically significant coefficient (t-stat >2). The MLR had a higher R-squared of .3471. This reveals that beauty score is not the only variable that determines the course evaluation ratings of teachers because gender, class type, native language, and tenure track status all have something to do with it as well. However, the variable importance plot used in the bagging method revealed that BeautyScore had the highest effect on the outcome of the course evaluation among all predictors by a large margin. In conclusion, BeautyScore has a significant on the outcome of CourseEvals, even more significant than that of the other predictors in the data set, but it cannot completely explain the variance of CourseEvals.




```
Call:
lm(formula = CourseEvals ~ BeautyScore, data = beautyData)

Residuals:
    Min      1Q  Median      3Q     Max
-1.5936 -0.3346  0.0097  0.3702  1.2321

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.71340    0.02249 165.119   <2e-16 ***
BeautyScore  0.27148    0.02837   9.569   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4809 on 461 degrees of freedom
Multiple R-squared:  0.1657,    Adjusted R-squared:  0.1639
F-statistic: 91.57 on 1 and 461 DF,  p-value: < 2.2e-16
```

```
Call:
lm(formula = CourseEvals ~ ., data = beautyData)

Residuals:
     Min       1Q   Median       3Q      Max
-1.31385 -0.30202  0.01011  0.29815  1.04929

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.06542    0.05145  79.020  < 2e-16 ***
BeautyScore  0.30415    0.02543  11.959  < 2e-16 ***
female      -0.33199    0.04075  -8.146 3.62e-15 ***
lower       -0.34255    0.04282  -7.999 1.04e-14 ***
nonenglish  -0.25808    0.08478  -3.044  0.00247 **
tenuretrack -0.09945    0.04888  -2.035  0.04245 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4273 on 457 degrees of freedom
Multiple R-squared:  0.3471,    Adjusted R-squared:  0.3399
F-statistic: 48.58 on 5 and 457 DF,  p-value: < 2.2e-16
```



2. Dr. Hamermesh says, “Disentangling whether this outcome represents productivity or discrimination is, as with the issue generally, probably impossible.” By saying this, he is referring to the fact that there are certainly lurking variables at play. In problem 1, we discovered that only 16.57% of the variance in the course instruction rating could be explained by the beauty score. Only 34.71% of the variance in the course instruction rating could be explained by all of the predictors in the data set. In order to determine whether or not productivity or discrimination have an effect on the course instruction rating, we would need to have data that measured these things. Even if we had some sort of data on these, the data will not be 100% reliable, as these are very subjective metrics. Even if we had all variables in the world, it is probably impossible to create a model in which the y variable is objectively 100% explained by the predictor variables.

Problem 2: Housing Price Structure

1. After making the brick a dummy variable, I ran a multiple linear regression with Price as a function of all the other variables. If all else is held constant, you can see that the price of a house goes up $15601.82 based

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -9814.663   9858.884  -0.996  0.32149
Home            6.187     28.973   0.214  0.83128
Nbhd         9832.281   1821.869   5.397 3.47e-07 ***
Offers      -8351.794   1267.428  -6.590 1.24e-09 ***
SqFt           49.811      6.769   7.359 2.53e-11 ***
Bedrooms     5671.911   1840.979   3.081  0.00256 **
Bathrooms    8243.545   2449.897   3.365  0.00103 **
brick       15601.818   2261.896   6.898 2.66e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11540 on 120 degrees of freedom
Multiple R-squared:  0.8256,    Adjusted R-squared:  0.8154
F-statistic: 81.15 on 7 and 120 DF,  p-value: < 2.2e-16
```

on whether or not the house is made of brick. This shows that there is a premium for brick houses.

2. I made new dummy variable column (nbhd_3) in which houses in neighborhood 3 are assigned a value of 1 and all of the other houses are assigned a value of 0. I deleted the original nbhd column and ran a multiple linear regression with price as a function of all the other predictors. The coefficient for this dummy variable column is 21929.82, meaning that if all else is held constant, houses in neighborhood 3 have a premium of $21929.82. It makes sense that this is a newer and more prestigious part of town because the houses seem to be much more expensive.

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3049.72    8778.83   0.347  0.72890
Home           -8.68      25.03  -0.347  0.72940
Offers      -8061.22    1023.98  -7.872 1.73e-12 ***
SqFt           52.56       5.72   9.190 1.46e-15 ***
Bedrooms     3971.91    1601.85   2.480  0.01454 *
Bathrooms    7874.35    2124.72   3.706  0.00032 ***
brick       17051.46    1950.02   8.744 1.64e-14 ***
nbhd_3      21929.82    2491.57   8.802 1.20e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10030 on 120 degrees of freedom
Multiple R-squared:  0.8683,    Adjusted R-squared:  0.8606
F-statistic:   113 on 7 and 120 DF,  p-value: < 2.2e-16
```

3. After selecting a subset of the data that only included houses in neighborhood 3, I ran a multiple linear regression with price as a function of all of the other predictor variables, the brick dummy variable column from question 1 included. The results showed that the brick coefficient was 24262.18, which means that brick houses within neighborhood 3 had a premium of $24262.18.

```
Coefficients: (1 not defined because of singularities)
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 15446.321  15272.354   1.011   0.3194
Home          -63.735     48.568  -1.312   0.1988
Offers      -8552.987   1863.469  -4.590 6.52e-05 ***
SqFt           56.647      9.383   6.037 9.75e-07 ***
Bedrooms     5280.851   2526.808   2.090   0.0447 *
Bathrooms    7158.602   3348.565   2.138   0.0403 *
brick       24262.184   3205.391   7.569 1.27e-08 ***
nbhd_3             NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8388 on 32 degrees of freedom
Multiple R-squared:  0.8486,    Adjusted R-squared:  0.8202
F-statistic: 29.89 on 6 and 32 DF,  p-value: 8.596e-12
```

4. For the purpose of prediction, I could definitely combine neighborhoods 1 and 2 into a single “older” neighborhood. In fact, I already did this in question 2. By using an ifelse function, I created a new neighborhood column called nbhd_3 that made the value 1 if the house was in the newer neighborhood (neighborhood 3) and 0 if the house was in the “older” neighborhood

(neighborhood 1 and 2). This allowed me to find the premium of the "newer" neighborhood houses versus the "older" neighborhood houses.

```
# makes new column with dummy variable for houses in neighborhood 3
nbhd_3 = ifelse(mc.data$Nbhd == 3, 1, 0)
mc.data = mc.data[-2]
mc.data = cbind(mc.data, nbhd_3)

lm.mc3 = lm(Price ~ ., data = mc.data)
summary(lm.mc3)
```

Problem 3: What causes what??

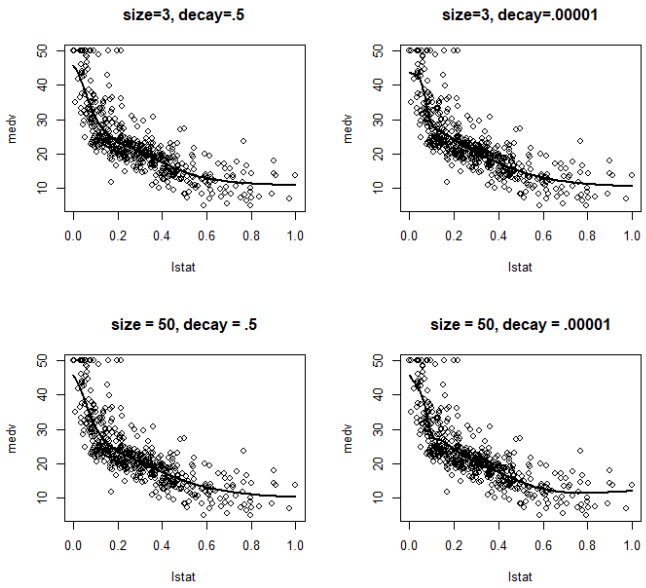
1. The reason you can't just get data from a few different cities and run the regression of "Crime" on "Police" is because high crime rate cities have an incentive to hire a lot of cops, so the data is very messy. A test such as this might suggest that more cops leads to higher crime rates, which we know does not make sense.
2. In order to isolate this effect, the researchers needed to find an example that a city had a large number of police for reasons other than street crime. They found such an example to study in Washington D.C., where more police were brought on in instance that the terrorism alert system was at level orange. Extra police were put on the street in order to protect against these terrorists. They then ran numbers on this instance against the crime rate in the city, and they concluded that more cops on the street did in fact result in lower street crime. Referring to Table 2, total daily crime reduced by 7.316 on high-alert days. However, only 14% of the variance in the total daily crime could be explained by the status of the alert level.
3. They had to control for METRO ridership because the question arose that maybe tourists or people in D.C. were less likely to be out and about on the city on these high alert days, which would limit the number of potential victims of crime on the street. In order to address this, they tested the METRO ridership to see if the numbers were different than those of regular, non-alert days, and they determined that there was not a significant difference in the number of victims.

This is a clever way to build a more convincing case of the direct effect of police presence on the crime rate.

4. The model being estimated is crime rate in the national mall (District 1) as a function of the high alert level in District 1 and other districts and the METRO ridership data. The model shows that the coefficient for high alert days is significant. On high alert days in District 1, the crime rate in District 1 is reduced by 2.62. To further isolate the specific location, the model includes high alert days for other districts, and there is not a significant relationship to the reduction in crime rate for the National Mall on these days. The METRO ridership data is also included to ensure there are significant number of potential victims on these high alert days. This is validated by the fact that the coefficient for Log(midday ridership) is statistically significant. This strengthens the conclusion that the number of police in a specific location will cause a reduction in street crime in that specific location.

Problem 4: Neural Nets

➔ **See Code**



Problem 5: Final Project

For our final project, we analyzed a set of data relating to NBA team game statistics from 20014 to 2018. Our goal was to determine the key statistics that contributed most to the outcome of a basketball game (win or loss) in order to present strategies and insights to a general manager to help him create a team that wins more. In this project, I took on a lot of responsibilities. I downloaded the data from Kaggle into an Excel file. I reduced the number of x variables from around 40 to 20 based on variables I considered to contribute towards winning. From there, I created a dummy variable in the excel file for the win/loss column. I made win 1 and loss 0. This allowed us to use the data more effectively when running different models in R. I was responsible for running bagging and boosting to find the variable importance, which ended up being our key conclusions. In addition, every time our team met via zoom, I shared my screen and created our final R-script with input from the team. I tended to organize a lot of the meetings and somewhat lead the meetings. When it came time to make slides, I was the one making the slides and sharing my screen in zoom meetings to hear input from the team. I also was one of three to present our final presentation in front of the class. Overall, I was one of the main contributors and leaders of the group.