Final Exam

Brett Scroggins

8/2/2017

Chapter 2 - Question 10

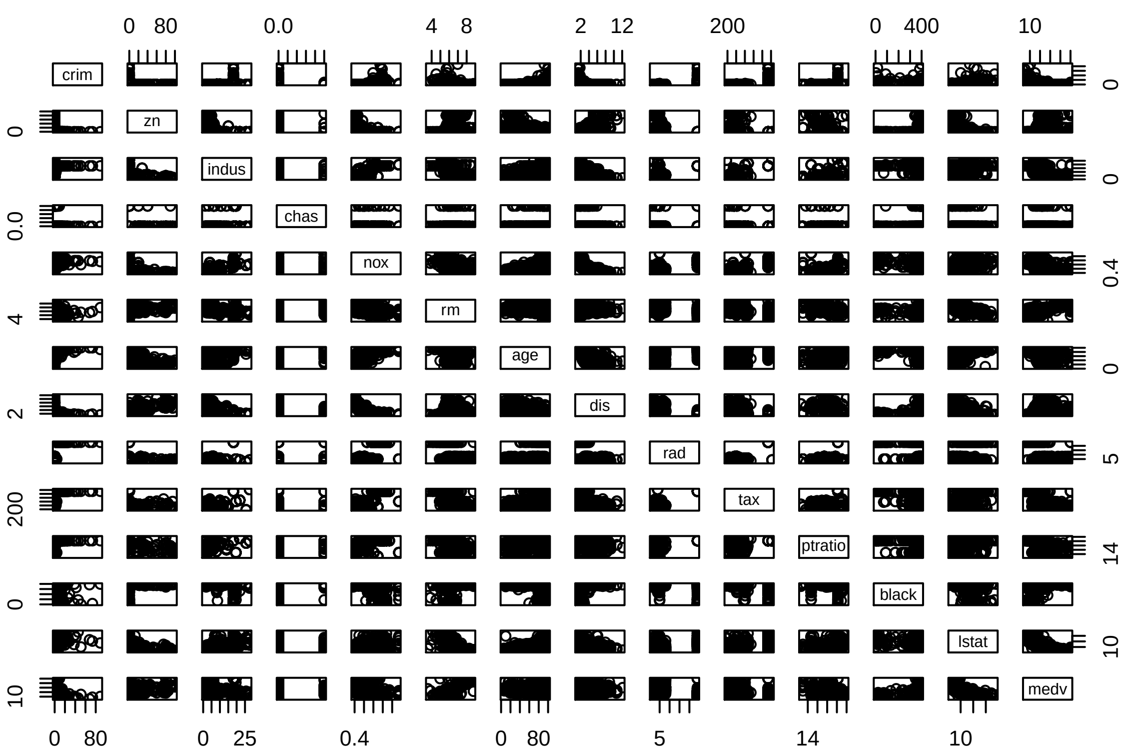
```
##########################################################
# Chapter 2 - Question 10
##########################################################

library(MASS)
attach(Boston)

# A - size of data set
dim(Boston)
```

```
## [1] 506  14
```

```
#result = 506 x 14
```

a. The Boston data sets contains 506 rows and 14 columns. The rows represent the 506 housing entries in the set, and the 14 columns represent different descriptive housing qualities.

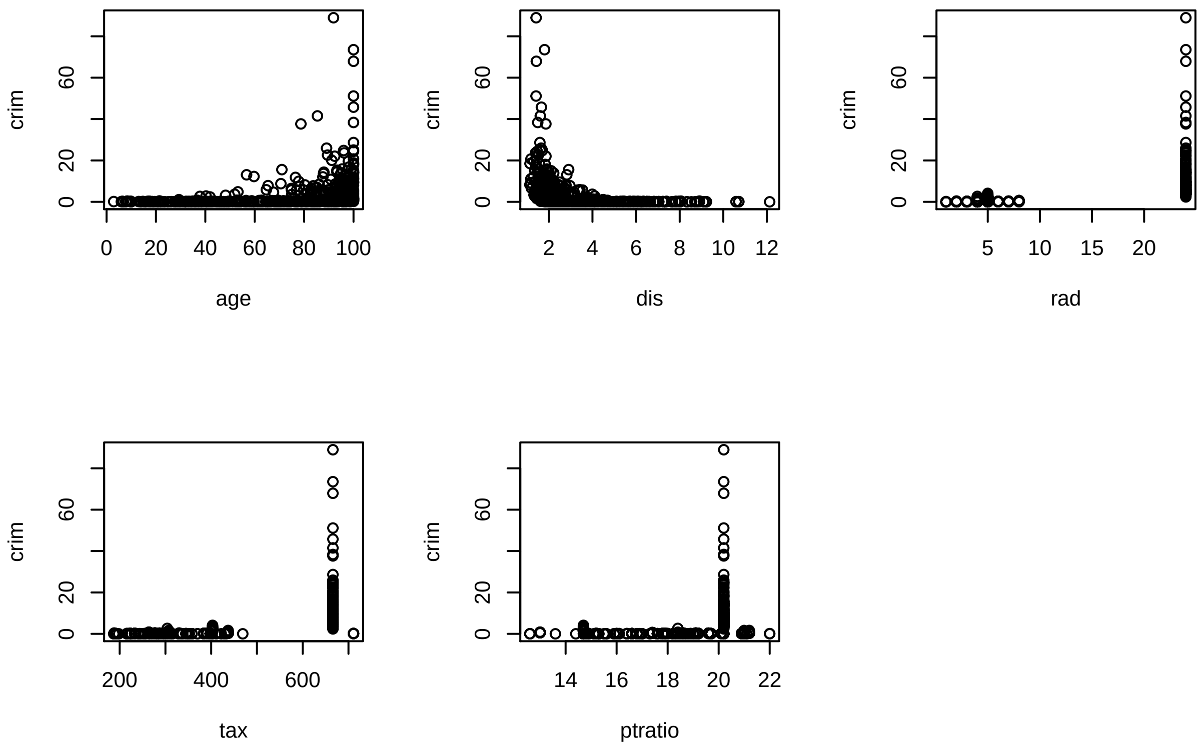
```
# B - pairwise scatter plots
pairs(Boston)
```



b. Upon review the pairs scatterplots, the following relationships were deemed to have correlation. The scatterplots have been attached on the page to follow.

Per capita crime rate (crim) correlates with the following: zn, age, dis, rad, and tax Proportion of residential land zoned for lots over 25,000 sq.ft. (zn) correlates with: indus, nox, black Proportion of non-retail business acres per town (indus) correlates with: dis, rad Nitrogen oxides concentration (nox) correlates with: age, dis Average number of rooms per dwelling (rm) correlates with: lstat, medv

```r
# C - per capita crime rate associations
par(mfrow=c(2,3))
plot(age,crim)
plot(dis,crim)
plot(rad,crim)
plot(tax,crim)
plot(ptratio,crim)
```



c. The per capita crime rate has a few different predictors that it associates with. • Age vs. Crim – positive • Dis vs. Crim – negative • Rad vs. Crim – positive • Tax vs. Crim – positive • PTRatio vs. Crim – positive

```r
# D - any suburbs with high crime rates? tax rates? puple-teacher ratio?
# comment on range of each precictor.
par(mfrow=c(3,1))
hist(crim,breaks = 10,col='yellow')
crimhi = Boston[crim >=20,]
nrow(crimhi)
```

```
## [1] 18
```

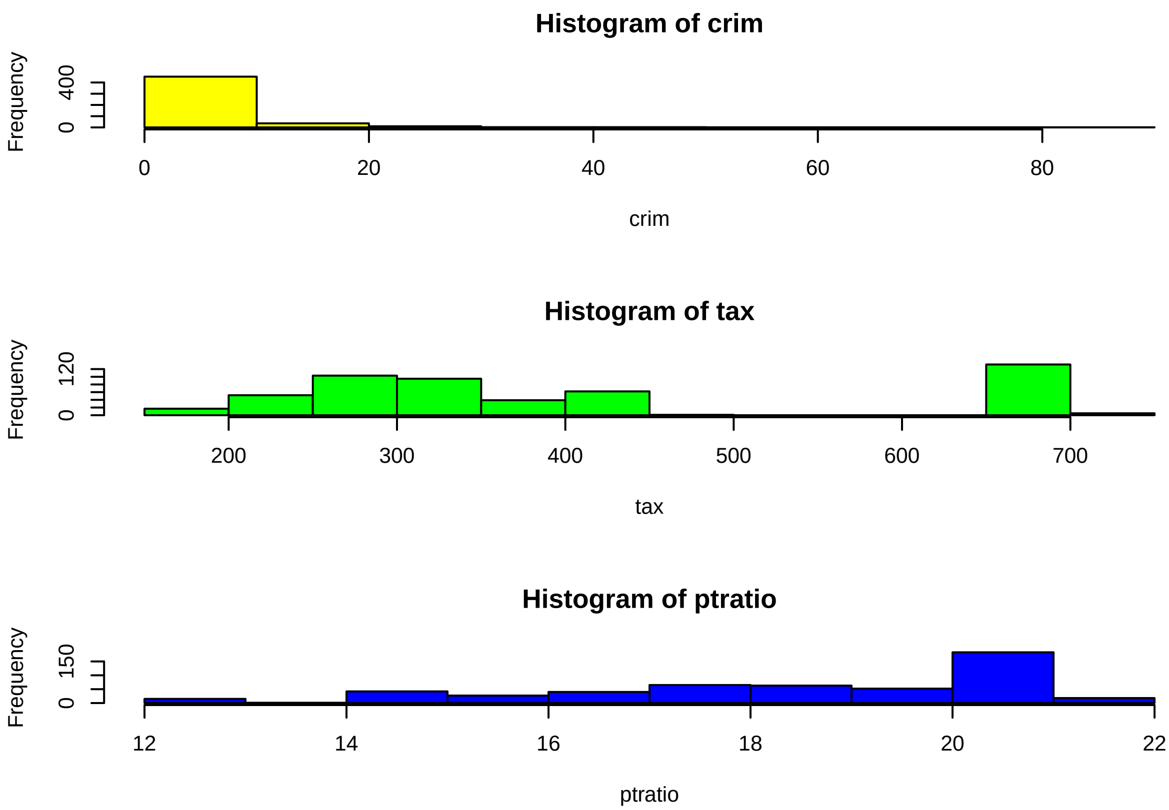
```r
hist(tax, breaks=10,col='green')
taxmid1 = Boston[tax>=200,]
taxmid2 = Boston[tax>=450,]
nrow(taxmid1) - nrow(taxmid2)
```

```
## [1] 351
```

```r
taxhi = Boston[tax>=650,]
nrow(taxhi)
```

```
## [1] 137
```

```r
hist(ptratio,breaks=10,col='blue')
```




```r
majlo = Boston[ptratio > 14,]
num20hi = Boston[ptratio > 20.51,]
num20lo = Boston[ptratio > 19.51,]
nrow(num20lo) - nrow(num20hi)
```

```
## [1] 161
```

```r
nrow(majlo) - nrow(num20hi)
```

```
## [1] 434
```

d. The crime rate in majority of suburbs is low (<10%), with a very small amount (18) reaching about the 20% crime rate.

In this data, the property value of 351 houses fell between $2,000,000 and $4,500,000 in this range; however, there was a high-end to which the tail of the histogram with property value over $6,500,000 contained an additional 137 houses.

The student-teacher ratio in the suburbs falls between the approximate ratios of 14-1 and 20-1 for 434 of the entries. The most common student to teacher ratio was about 20-1, and this occurred in 161 data points.

```r
# E - how many of suburbs bound Charles river?
sum(Boston[,'chas'])
```

```
## [1] 35
```

```r
# 35 are on the river
```

e. Of the houses sampled, 35 lie on the Charles river.

```r
# F - median of pupil-teacher ratio of whole data set
# G - lowest median value of owner-occupied homes
summary(Boston)
```

```
##       crim                zn             indus            chas
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000
##  1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000
##       nox               rm             age              dis
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127
##       rad              tax           ptratio          black
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90
##      lstat            medv
##  Min.   : 1.73   Min.   : 5.00
##  1st Qu.: 6.95   1st Qu.:17.02
##  Median :11.36   Median :21.20
##  Mean   :12.65   Mean   :22.53
##  3rd Qu.:16.95   3rd Qu.:25.00
##  Max.   :37.97   Max.   :50.00
```

```r
# Median ptratio = 19.05
# Lowest medv = $5000 (5 but by \$1000's)

lmedv = Boston[medv==5,]
nrow(lmedv)
```

```
## [1] 2
```

```r
summary(lmedv)
```

```
##       crim             zn        indus            chas         nox
##  Min.   :38.35   Min.   :0   Min.   :18.1   Min.   :0   Min.   :0.693
##  1st Qu.:45.74   1st Qu.:0   1st Qu.:18.1   1st Qu.:0   1st Qu.:0.693
##  Median :53.14   Median :0   Median :18.1   Median :0   Median :0.693
##  Mean   :53.14   Mean   :0   Mean   :18.1   Mean   :0   Mean   :0.693
##  3rd Qu.:60.53   3rd Qu.:0   3rd Qu.:18.1   3rd Qu.:0   3rd Qu.:0.693
##  Max.   :67.92   Max.   :0   Max.   :18.1   Max.   :0   Max.   :0.693
##        rm             age           dis             rad          tax
##  Min.   :5.453   Min.   :100   Min.   :1.425   Min.   :24   Min.   :666
```

```
##  1st Qu.:5.511   1st Qu.:100   1st Qu.:1.441   1st Qu.:24   1st Qu.:666
##  Median :5.568   Median :100   Median :1.458   Median :24   Median :666
##  Mean   :5.568   Mean   :100   Mean   :1.458   Mean   :24   Mean   :666
##  3rd Qu.:5.625   3rd Qu.:100   3rd Qu.:1.474   3rd Qu.:24   3rd Qu.:666
##  Max.   :5.683   Max.   :100   Max.   :1.490   Max.   :24   Max.   :666
##     ptratio          black           lstat            medv
##  Min.   :20.2   Min.   :385.0   Min.   :22.98   Min.   :5
##  1st Qu.:20.2   1st Qu.:388.0   1st Qu.:24.88   1st Qu.:5
##  Median :20.2   Median :390.9   Median :26.79   Median :5
##  Mean   :20.2   Mean   :390.9   Mean   :26.79   Mean   :5
##  3rd Qu.:20.2   3rd Qu.:393.9   3rd Qu.:28.69   3rd Qu.:5
##  Max.   :20.2   Max.   :396.9   Max.   :30.59   Max.   :5
```

f. The median student-teacher ratio of all suburbs is 19.05 – 1.

g. The lowest median value of owner-occupied homes is $5,000. In the suburbs in which these occur in, there are other predictors that stand out in comparison with the overall range. Predictors such as indus, nox, tax, ptratio, and lstat all fell in the 3rd Quartile or higher and crim, age, and rad all contained the maximum value of the data set. The only qualifier that was below the mean was dist, which fell below the 1st quartile.

```
# H - 7 or more, and 8 or more room dwellings

seven = Boston[rm >= 7,]
nrow(seven)
```

```
## [1] 64
```

```
# 64 with 7 or more

eight = Boston[rm>=8,]
nrow(eight)
```

```
## [1] 13
```

```
# 13 with 8 or more rooms
```

h. In this data set, there are 64 dwellings that have 7 or more rooms; additionally, 13 of those have 8 or more rooms in the house.

Chapter 3 - Questions 15

```
#######################################################
# Chapter 3 - Question 15
#######################################################


#A - create regression between each variable

summary(Boston)
```

```
##       crim                zn             indus            chas
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000
##  1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000
```

```
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000
##       nox               rm             age              dis
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127
##       rad              tax           ptratio          black
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90
##      lstat             medv
##  Min.   : 1.73   Min.   : 5.00
##  1st Qu.: 6.95   1st Qu.:17.02
##  Median :11.36   Median :21.20
##  Mean   :12.65   Mean   :22.53
##  3rd Qu.:16.95   3rd Qu.:25.00
##  Max.   :37.97   Max.   :50.00
```

```r
attach(Boston)
```

```
## The following objects are masked from Boston (pos = 3):
##
##     age, black, chas, crim, dis, indus, lstat, medv, nox, ptratio,
##     rad, rm, tax, zn
```

```r
lm.zn = lm(crim~zn)
summary(lm.zn)
```

```
##
## Call:
## lm(formula = crim ~ zn)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.429 -4.222 -2.620  1.250 84.523
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.45369    0.41722  10.675  < 2e-16 ***
## zn          -0.07393    0.01609  -4.594 5.51e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.435 on 504 degrees of freedom
## Multiple R-squared:  0.04019,    Adjusted R-squared:  0.03828
## F-statistic:  21.1 on 1 and 504 DF,  p-value: 5.506e-06
```

```r
#yes

lm.indus = lm(crim~indus)
summary(lm.indus)
```

```
##
## Call:
## lm(formula = crim ~ indus)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.972  -2.698  -0.736   0.712  81.813
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.06374    0.66723  -3.093  0.00209 **
## indus        0.50978    0.05102   9.991  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.866 on 504 degrees of freedom
## Multiple R-squared:  0.1653, Adjusted R-squared:  0.1637
## F-statistic: 99.82 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
#yes

lm.chas = lm(crim~chas)
summary(lm.chas)
```

```
##
## Call:
## lm(formula = crim ~ chas)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.738 -3.661 -3.435  0.018 85.232
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7444     0.3961   9.453   <2e-16 ***
## chas         -1.8928     1.5061  -1.257    0.209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.597 on 504 degrees of freedom
## Multiple R-squared:  0.003124,   Adjusted R-squared:  0.001146
## F-statistic: 1.579 on 1 and 504 DF,  p-value: 0.2094
```

```r
#no

lm.nox = lm(crim~nox)
summary(lm.nox)
```

```
##
## Call:
## lm(formula = crim ~ nox)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.371  -2.738  -0.974   0.559  81.728
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -13.720      1.699  -8.073 5.08e-15 ***
## nox           31.249      2.999  10.419  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.81 on 504 degrees of freedom
## Multiple R-squared:  0.1772, Adjusted R-squared:  0.1756
## F-statistic: 108.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
#yes

lm.rm = lm(crim~rm)
summary(lm.rm)
```

```
##
## Call:
## lm(formula = crim ~ rm)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.604 -3.952 -2.654  0.989 87.197
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.482      3.365   6.088 2.27e-09 ***
## rm            -2.684      0.532  -5.045 6.35e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.401 on 504 degrees of freedom
## Multiple R-squared:  0.04807,    Adjusted R-squared:  0.04618
## F-statistic: 25.45 on 1 and 504 DF,  p-value: 6.347e-07
```

```
#yes

lm.age = lm(crim~age)
summary(lm.age)
```

```
##
## Call:
## lm(formula = crim ~ age)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.789 -4.257 -1.230  1.527 82.849
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.77791    0.94398  -4.002 7.22e-05 ***
## age          0.10779    0.01274   8.463 2.85e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.057 on 504 degrees of freedom
```

```
## Multiple R-squared:  0.1244, Adjusted R-squared:  0.1227
## F-statistic: 71.62 on 1 and 504 DF,  p-value: 2.855e-16
```

```r
#yes

lm.dis = lm(crim~dis)
summary(lm.dis)
```

```
##
## Call:
## lm(formula = crim ~ dis)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.708 -4.134 -1.527  1.516 81.674
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.4993     0.7304  13.006   <2e-16 ***
## dis          -1.5509     0.1683  -9.213   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.965 on 504 degrees of freedom
## Multiple R-squared:  0.1441, Adjusted R-squared:  0.1425
## F-statistic: 84.89 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
#yes

lm.rad = lm(crim~rad)
summary(lm.rad)
```

```
##
## Call:
## lm(formula = crim ~ rad)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.164  -1.381  -0.141   0.660  76.433
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.28716    0.44348  -5.157 3.61e-07 ***
## rad          0.61791    0.03433  17.998  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.718 on 504 degrees of freedom
## Multiple R-squared:  0.3913, Adjusted R-squared:   0.39
## F-statistic: 323.9 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
#yes

lm.tax = lm(crim~tax)
summary(lm.tax)
```

```
##
## Call:
## lm(formula = crim ~ tax)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.513  -2.738  -0.194   1.065  77.696
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.528369   0.815809  -10.45   <2e-16 ***
## tax          0.029742   0.001847   16.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.997 on 504 degrees of freedom
## Multiple R-squared:  0.3396, Adjusted R-squared:  0.3383
## F-statistic: 259.2 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
#yes

lm.ptratio = lm(crim~ptratio)
summary(lm.ptratio)
```

```
##
## Call:
## lm(formula = crim ~ ptratio)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.654 -3.985 -1.912  1.825 83.353
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.6469     3.1473  -5.607 3.40e-08 ***
## ptratio       1.1520     0.1694   6.801 2.94e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.24 on 504 degrees of freedom
## Multiple R-squared:  0.08407,    Adjusted R-squared:  0.08225
## F-statistic: 46.26 on 1 and 504 DF,  p-value: 2.943e-11
```

```r
#yes

lm.black = lm(crim~black)
summary(lm.black)
```

```
##
## Call:
## lm(formula = crim ~ black)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.756  -2.299  -2.095  -1.296  86.822
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.553529   1.425903  11.609   <2e-16 ***
## black       -0.036280   0.003873  -9.367   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.946 on 504 degrees of freedom
## Multiple R-squared:  0.1483, Adjusted R-squared:  0.1466
## F-statistic: 87.74 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
#yes

lm.lstat = lm(crim~lstat)
summary(lm.lstat)
```

```
##
## Call:
## lm(formula = crim ~ lstat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.925  -2.822  -0.664   1.079  82.862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.33054    0.69376  -4.801 2.09e-06 ***
## lstat        0.54880    0.04776  11.491  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.664 on 504 degrees of freedom
## Multiple R-squared:  0.2076, Adjusted R-squared:  0.206
## F-statistic:   132 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
#yes

lm.medv = lm(crim~medv)
summary(lm.medv)
```

```
##
## Call:
## lm(formula = crim ~ medv)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.071 -4.022 -2.343  1.298 80.957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.79654    0.93419   12.63   <2e-16 ***
## medv        -0.36316    0.03839   -9.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.934 on 504 degrees of freedom
```

```
## Multiple R-squared:  0.1508, Adjusted R-squared:  0.1491
## F-statistic: 89.49 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
#yes
```

a. While running a linear regression for each variable and crime rate, the following list of variable were found to have statistically significant correlation with crime rate.

zn, indus, nox, rm, age, dis, rad, tax, ptratio, black, lstat, and medv

```r
# B - create multiple regression model between crim and all others

lm.all = lm(crim~.,data = Boston)
summary(lm.all)
```

```
##
## Call:
## 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 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

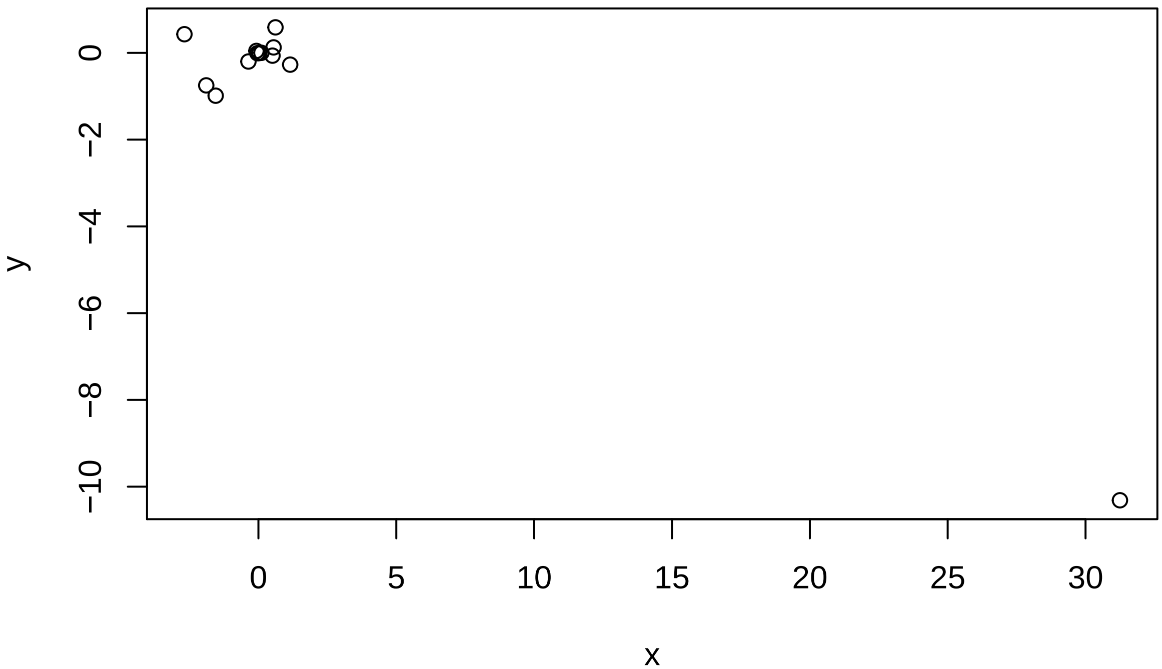
```r
# Yes = zn, dis, rad, black, medv
# No = indus, chas, nox, rm, age, tax, ptratio, lstat
# y = 17.033 + 0.045(zn) - 0.987(dis) + 0.588(rad) - 0.008(black) - 0.199(medv)
```

b. The multiple regression for the data set showed that the variable with significant impact on crime rate are as follows.

zn, dis, rad, black, and medv

```r
# C - create a plot comparing coefficients from linear to multiple

x = c(coefficients(lm.zn)[2],coefficients(lm.indus)[2],coefficients(lm.chas)[2],coefficients(lm.nox)[2]
```

```
      coefficients(lm.rm)[2],coefficients(lm.age)[2],coefficients(lm.dis)[2],coefficients(lm.rad)[2],
      coefficients(lm.tax)[2],coefficients(lm.ptratio)[2],coefficients(lm.black)[2], coefficients(lm.ls
      coefficients(lm.medv)[2])

y = coefficients(lm.all)[2:14]

par(mfrow=c(1,1))
plot (x,y)
```



c. A dot-plot of the data based on the value of coefficients from linear regression (x) and multiple regression (y).

```
# D - check all functions with polynomial regression (max x^3)

lm.zn3 = lm(crim~poly(zn,3))
summary(lm.zn3)
```

```
##
## Call:
## lm(formula = crim ~ poly(zn, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.821 -4.614 -1.294  0.473 84.130
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3722   9.709  < 2e-16 ***
## poly(zn, 3)1 -38.7498     8.3722  -4.628  4.7e-06 ***
## poly(zn, 3)2  23.9398     8.3722   2.859  0.00442 **
## poly(zn, 3)3 -10.0719     8.3722  -1.203  0.22954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 8.372 on 502 degrees of freedom
## Multiple R-squared:  0.05824,    Adjusted R-squared:  0.05261
## F-statistic: 10.35 on 3 and 502 DF,  p-value: 1.281e-06
```

```r
# 1 and 2

lm.indus3 = lm(crim~poly(indus,3))
summary(lm.indus3)
```

```
##
## Call:
## lm(formula = crim ~ poly(indus, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.278 -2.514  0.054  0.764 79.713
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.614      0.330  10.950  < 2e-16 ***
## poly(indus, 3)1   78.591      7.423  10.587  < 2e-16 ***
## poly(indus, 3)2  -24.395      7.423  -3.286  0.00109 **
## poly(indus, 3)3  -54.130      7.423  -7.292  1.2e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.423 on 502 degrees of freedom
## Multiple R-squared:  0.2597, Adjusted R-squared:  0.2552
## F-statistic: 58.69 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
# 1, 2, and 3

# Chase river will not be a valid predictor with polynomials

lm.nox3 = lm(crim~poly(nox,3))
summary(lm.nox3)
```

```
##
## Call:
## lm(formula = crim ~ poly(nox, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.110 -2.068 -0.255  0.739 78.302
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3216  11.237  < 2e-16 ***
## poly(nox, 3)1  81.3720     7.2336  11.249  < 2e-16 ***
## poly(nox, 3)2 -28.8286     7.2336  -3.985 7.74e-05 ***
## poly(nox, 3)3 -60.3619     7.2336  -8.345 6.96e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.234 on 502 degrees of freedom
```

```
## Multiple R-squared:  0.297,  Adjusted R-squared:  0.2928
## F-statistic: 70.69 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
# 1, 2, and 3

lm.rm3 = lm(crim~poly(rm,3))
summary(lm.rm3)
```

```
##
## Call:
## lm(formula = crim ~ poly(rm, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.485  -3.468  -2.221  -0.015  87.219
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.6135     0.3703   9.758  < 2e-16 ***
## poly(rm, 3)1 -42.3794     8.3297  -5.088 5.13e-07 ***
## poly(rm, 3)2  26.5768     8.3297   3.191  0.00151 **
## poly(rm, 3)3  -5.5103     8.3297  -0.662  0.50858
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.33 on 502 degrees of freedom
## Multiple R-squared:  0.06779,    Adjusted R-squared:  0.06222
## F-statistic: 12.17 on 3 and 502 DF,  p-value: 1.067e-07
```

```r
# 1 and 2

lm.age3 = lm(crim~poly(age,3))
summary(lm.age3)
```

```
##
## Call:
## lm(formula = crim ~ poly(age, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.762 -2.673 -0.516  0.019 82.842
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3485  10.368  < 2e-16 ***
## poly(age, 3)1  68.1820     7.8397   8.697  < 2e-16 ***
## poly(age, 3)2  37.4845     7.8397   4.781 2.29e-06 ***
## poly(age, 3)3  21.3532     7.8397   2.724  0.00668 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.84 on 502 degrees of freedom
## Multiple R-squared:  0.1742, Adjusted R-squared:  0.1693
## F-statistic: 35.31 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
# 1, 2, and 3

lm.dis3 = lm(crim~poly(dis,3))
summary(lm.dis3)
```

```
##
## Call:
## lm(formula = crim ~ poly(dis, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.757  -2.588   0.031   1.267  76.378
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.6135     0.3259  11.087  < 2e-16 ***
## poly(dis, 3)1 -73.3886     7.3315 -10.010  < 2e-16 ***
## poly(dis, 3)2  56.3730     7.3315   7.689 7.87e-14 ***
## poly(dis, 3)3 -42.6219     7.3315  -5.814 1.09e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.331 on 502 degrees of freedom
## Multiple R-squared:  0.2778, Adjusted R-squared:  0.2735
## F-statistic: 64.37 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
# 1, 2, and 3

lm.rad3 = lm(crim~poly(rad,3))
summary(lm.rad3)
```

```
##
## Call:
## lm(formula = crim ~ poly(rad, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.381  -0.412  -0.269   0.179  76.217
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.2971  12.164  < 2e-16 ***
## poly(rad, 3)1 120.9074     6.6824  18.093  < 2e-16 ***
## poly(rad, 3)2  17.4923     6.6824   2.618  0.00912 **
## poly(rad, 3)3   4.6985     6.6824   0.703  0.48231
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.682 on 502 degrees of freedom
## Multiple R-squared:    0.4,  Adjusted R-squared:  0.3965
## F-statistic: 111.6 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
# 1 and 2

lm.tax3 = lm(crim~poly(tax,3))
```

```r
summary(lm.tax3)
```

```
##
## Call:
## lm(formula = crim ~ poly(tax, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.273  -1.389   0.046   0.536  76.950
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3047  11.860  < 2e-16 ***
## poly(tax, 3)1 112.6458     6.8537  16.436  < 2e-16 ***
## poly(tax, 3)2  32.0873     6.8537   4.682 3.67e-06 ***
## poly(tax, 3)3  -7.9968     6.8537  -1.167    0.244
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.854 on 502 degrees of freedom
## Multiple R-squared:  0.3689, Adjusted R-squared:  0.3651
## F-statistic:  97.8 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
# 1 and 2

lm.ptratio3 = lm(crim~poly(ptratio,3))
summary(lm.ptratio3)
```

```
##
## Call:
## lm(formula = crim ~ poly(ptratio, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.833 -4.146 -1.655  1.408 82.697
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.614      0.361  10.008  < 2e-16 ***
## poly(ptratio, 3)1   56.045      8.122   6.901 1.57e-11 ***
## poly(ptratio, 3)2   24.775      8.122   3.050  0.00241 **
## poly(ptratio, 3)3  -22.280      8.122  -2.743  0.00630 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.122 on 502 degrees of freedom
## Multiple R-squared:  0.1138, Adjusted R-squared:  0.1085
## F-statistic: 21.48 on 3 and 502 DF,  p-value: 4.171e-13
```

```r
# 1, 2, and 3

lm.black3 = lm(crim~poly(black,3))
summary(lm.black3)
```

```
##
```

```
## Call:
## lm(formula = crim ~ poly(black, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.096  -2.343  -2.128  -1.439  86.790
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.6135     0.3536  10.218   <2e-16 ***
## poly(black, 3)1 -74.4312     7.9546  -9.357   <2e-16 ***
## poly(black, 3)2   5.9264     7.9546   0.745    0.457
## poly(black, 3)3  -4.8346     7.9546  -0.608    0.544
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.955 on 502 degrees of freedom
## Multiple R-squared:  0.1498, Adjusted R-squared:  0.1448
## F-statistic: 29.49 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
# 1

lm.lstat3 = lm(crim~poly(lstat,3))
summary(lm.lstat3)
```

```
##
## Call:
## lm(formula = crim ~ poly(lstat, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.234  -2.151  -0.486   0.066  83.353
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.6135     0.3392  10.654   <2e-16 ***
## poly(lstat, 3)1  88.0697     7.6294  11.543   <2e-16 ***
## poly(lstat, 3)2  15.8882     7.6294   2.082   0.0378 *
## poly(lstat, 3)3 -11.5740     7.6294  -1.517   0.1299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.629 on 502 degrees of freedom
## Multiple R-squared:  0.2179, Adjusted R-squared:  0.2133
## F-statistic: 46.63 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
# 1 and 2

lm.medv3 = lm(crim~poly(medv,3))
summary(lm.medv3)
```

```
##
## Call:
## lm(formula = crim ~ poly(medv, 3))
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -24.427  -1.976  -0.437   0.439  73.655
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.614      0.292  12.374  < 2e-16 ***
## poly(medv, 3)1  -75.058      6.569 -11.426  < 2e-16 ***
## poly(medv, 3)2   88.086      6.569  13.409  < 2e-16 ***
## poly(medv, 3)3  -48.033      6.569  -7.312 1.05e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.569 on 502 degrees of freedom
## Multiple R-squared:  0.4202, Adjusted R-squared:  0.4167
## F-statistic: 121.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
# 1, 2, and 3
```

d. Lastly, the crime rate was regressed against each additional variable with a new constraint that each regression could have up to a 3rd degree polynomial term. The following are the highest order relationship between crime rate and the specified variable.

zn – 2nd indus – 3rd nox – 3rd rm – 2nd age – 3rd dis – 3rd rad – 2nd tax – 2nd ptratio – 3rd black – 1st lstat – 2nd medv – 3rd ***Note: There was no statistical significance of a higher order for the location by Charles river as this was not a numeric variable.

Chapter 6 - Questions 9

```
#######################################################
# Chapter 6 - Question 9
#######################################################

library(ISLR)
library(glmnet)
```

```
## Loading required package: Matrix
## Loading required package: foreach
## Loaded glmnet 2.0-10
```

```
library(pls)
```

```
##
## Attaching package: 'pls'
## The following object is masked from 'package:stats':
##
##     loadings
```

```
# Part A - seperate to test and train data set

set.seed(1)
train = sample(1:dim(College)[1], dim(College)[1]*.8)
test = -train
train = College[train, ]
test = College[test, ]
```

a. The College data was divided into two subgroups. 80% of the data was randomly selected to be put into the train set, and the remaining 20% were stored in the test set.

```r
# Part B - least squares regression for train set

least.fit = lm(Apps~., data = train)
least.pred = predict(least.fit, test)
mean((least.pred - test[,'Apps'])^2)
```

```
## [1] 1075064
```

```r
# MSE = 1075064
sqrt(1075064)
```

```
## [1] 1036.853
```

b. The linear model least squares regression output a MSE of 1,075,064 (or RMSE of 1,036.85).

```r
# Part C - Ridge regression with lambda chosen with CV

train.mat = model.matrix(Apps~., data=train)
test.mat = model.matrix(Apps~., data=test)
grid = 10 ^ seq(10, -2, length=100)
ridge = cv.glmnet(train.mat, train[, "Apps"], alpha=0, lambda=grid, thresh=1e-12)
bestlam = ridge$lambda.1se
bestlam
```

```
## [1] 174.7528
```

```r
# Best lambda = 174.75

ridge.pred = predict(ridge, newx=test.mat, s=bestlam)
mean((test[, "Apps"] - ridge.pred)^2)
```

```
## [1] 1115737
```

```r
# MSE = 1115735
sqrt(1115735)
```

```
## [1] 1056.284
```

c. After cross validation, the minimum lambda found was 174.75. This produced a ridge regression model that output an MSE of 1,115,735 (or RMSE of 1,056.28).

```r
# Part D - Lasso method with lambda chosen with cv

lasso = cv.glmnet(train.mat, train[, "Apps"], alpha=1, lambda=grid, thresh=1e-12)
bestlam = lasso$lambda.min
bestlam
```

```
## [1] 18.73817
```

```r
# Best lambda = 18.74

lasso.pred = predict(lasso, newx=test.mat, s=bestlam)
mean((test[, "Apps"] - lasso.pred)^2)
```

```
## [1] 1112058
```

```r
# MSE = 1112058
sqrt(1112058)
```

```
## [1] 1054.542
```

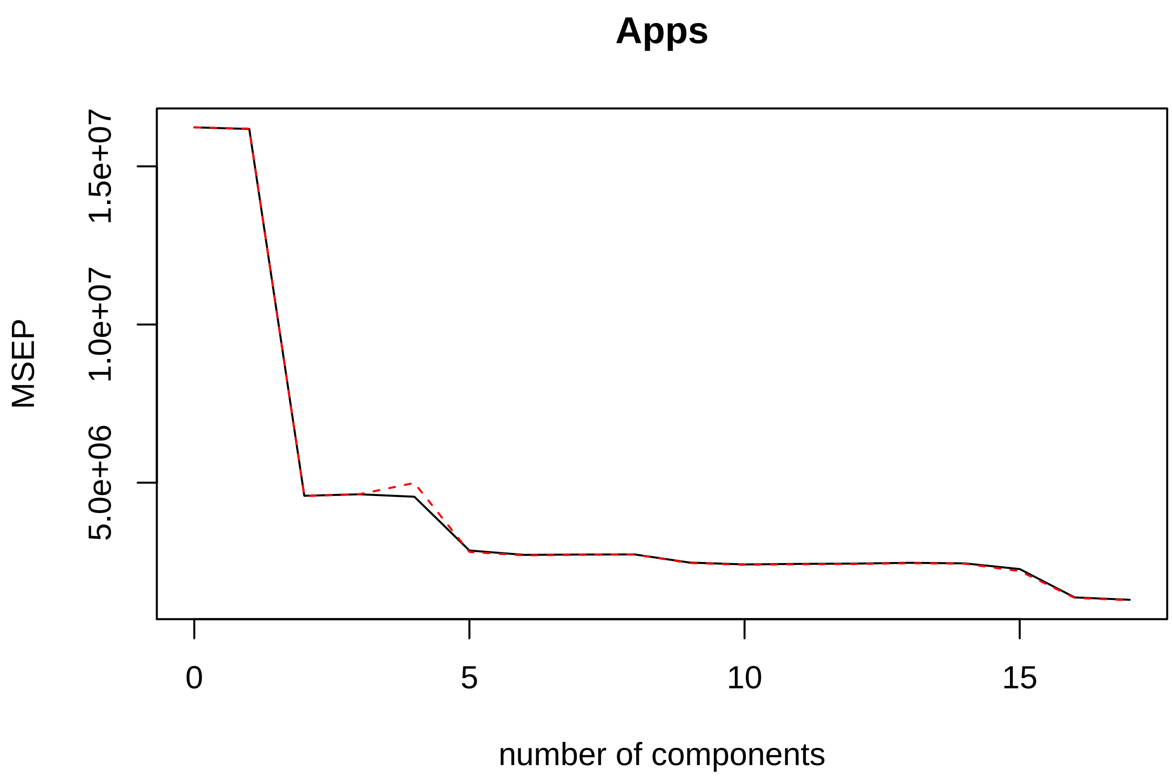
```
lasso = glmnet(model.matrix(Apps~., data=College), College[, "Apps"], alpha=1)
predict(lasso, s=bestlam, type="coefficients")
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                         1
## (Intercept) -5.813707e+02
## (Intercept)  .
## PrivateYes  -4.368911e+02
## Accept       1.471991e+00
## Enroll      -2.541925e-01
## Top10perc    3.574808e+01
## Top25perc   -3.838790e+00
## F.Undergrad  .
## P.Undergrad  2.596168e-02
## Outstate    -6.179537e-02
## Room.Board   1.284644e-01
## Books        .
## Personal     2.520096e-03
## PhD         -6.025350e+00
## Terminal    -3.256552e+00
## S.F.Ratio    6.026031e+00
## perc.alumni -8.955597e-01
## Expend       7.076914e-02
## Grad.Rate    5.580536e+00
```

```
# Books and F.Undergrad are the only two with zero coeffiencents
```

d. Cross validation for the Lasso method gave an ideal lambda of 18.74. This gave an out of sample MSE of 1,112,058 (or RMSE of 1,054.54).

```
# Part E - PCR model with M chosen with CV

pcr.fit = pcr(Apps~., data=train, scale=T, validation="CV")
validationplot(pcr.fit, val.type="MSEP")
```




```
# 7 looks to be the approximate ideal number of components

pcr.pred = predict(pcr.fit, test, ncomp=7)
mean((test[, "Apps"] - data.frame(pcr.pred))^2)
```

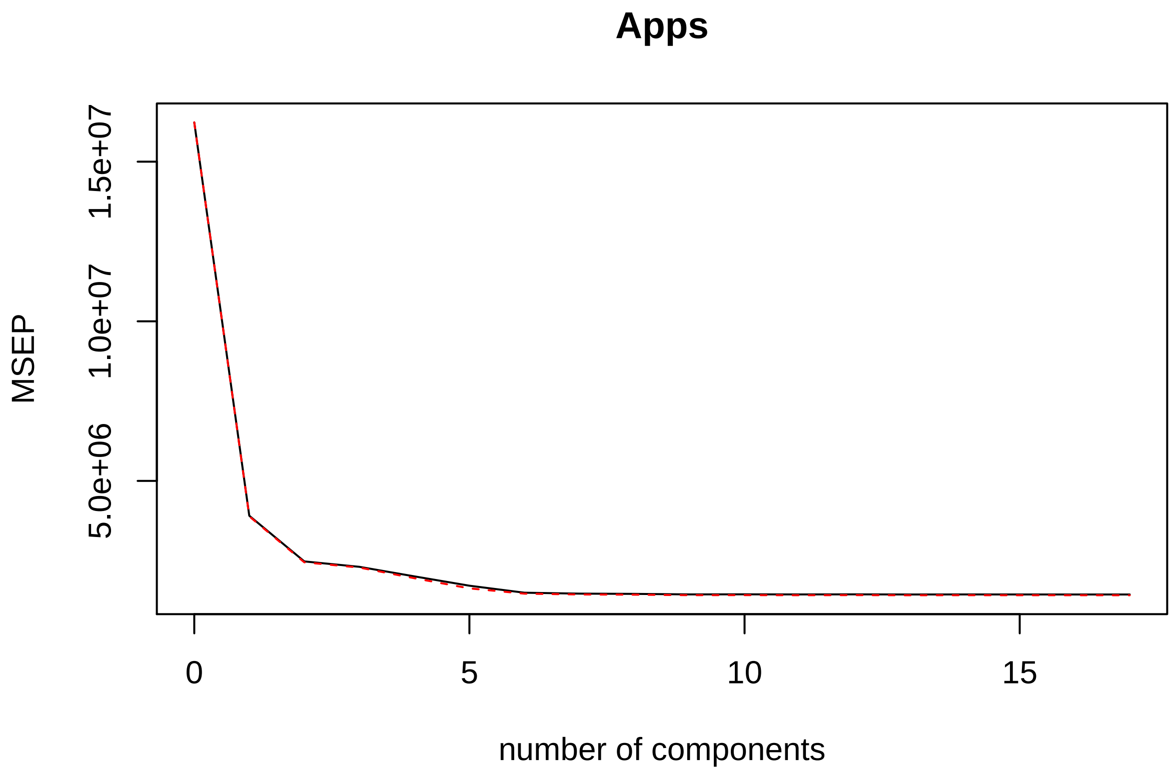
```
## [1] 1862757
```

```
# MSE = 1541736
sqrt(1541736)
```

```
## [1] 1241.667
```

e. For a PCR for this data set, an ideal M value was found to be 7. Using this, the fit analyzing the test set produces an MSE of 1,541,736 (or RMSE of 1,241.67).

```
# F - PLS Model with M chosen by CV

pls.fit = plsr(Apps~., data=train, scale=T, validation="CV")
validationplot(pls.fit, val.type="MSEP")
```




```
pls.pred = predict(pls.fit, test, ncomp=10)
mean((test[, "Apps"] - data.frame(pls.pred))^2)
```

```
## [1] 1075376
```

```
# MSE = 1075376
sqrt(1075376)
```

```
## [1] 1037.003
```

f. After finding the ideal number of variables to comparable to be 10, this yielded an MSE of 1,075,376 (or RMSE of 1,037.00).

```
# G - Comment on results obtained. How accurately can we predict number
# of college apps? Is there much difference amongst tests?

summary(College$Apps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      81     776    1558    3002    3624   48094
```

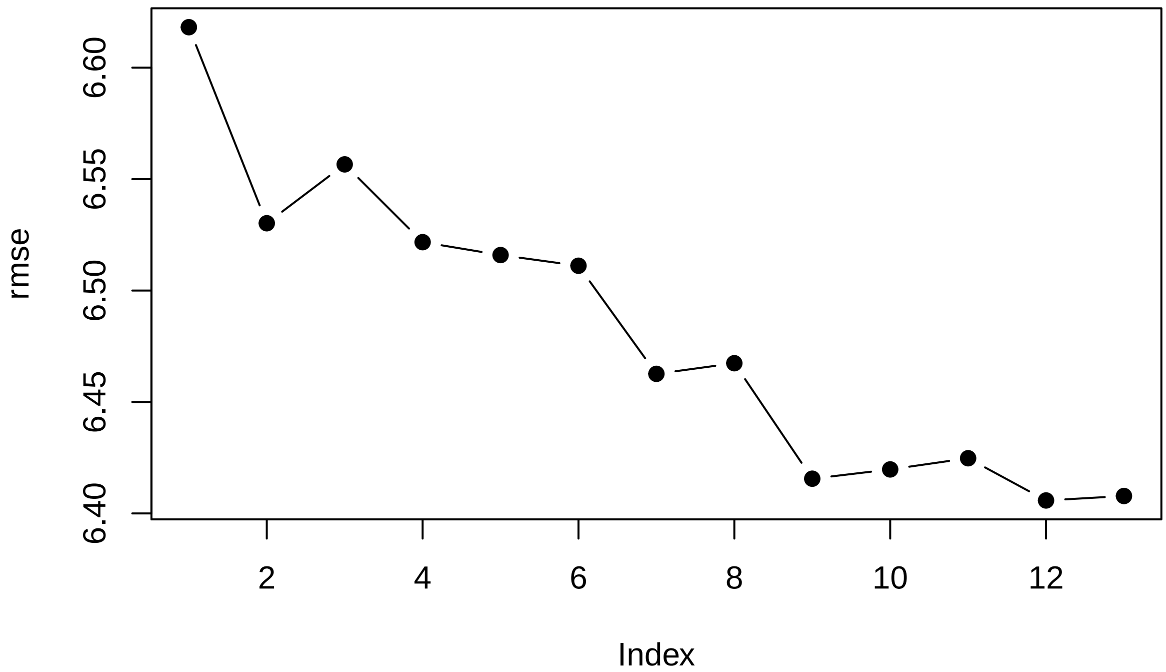
g. The accuracy of predicting college apps for every method tested all produced a RMSE of over 1,000. This means that to obtain an approximate range with a 95% confidence that the actual value will fall in that range, you must take the expected value of and add/subtract two times that RMSE. This produces a very wide range that is not very useful in this case. With on average colleges in the data set only receive 3,002 and median value of 1,558, a range of +- over 2,000 is not the most descriptive. There seem to be outliers affecting the data (one example being the maximum of 48,094 applications received) that would need to be cleaned before more accurate regressions can be formed.

Chapter 6 - Questions 11

```
########################################################
# Chapter 6 - Question 11
########################################################

library(MASS)
library(leaps)
library(glmnet)
library(pls)

# A - Do subset selection, lasso, ridge, and PCR

# Subset selection
predict.regsubsets = function(object, newdata,id,...){
  form = as.formula(object$call[[2]])
  mat = model.matrix(form,newdata)
  coefi = coef(object, id=id)
  xvars = names(coefi)
  mat[,xvars]%*%coefi
}
regfit.best = regsubsets(crim~., data = Boston, nvmax = 13)
coef(regfit.best,10)
```

```
##  (Intercept)           zn        indus          nox           rm
##  16.38579874   0.04186311  -0.09330371 -10.62174498   0.44828268
##          dis          rad      ptratio        black        lstat
##  -0.99077268   0.53566370  -0.26956103  -0.00759461   0.13084608
##         medv
##  -0.19800062
```

```
k = 10
set.seed(1)
folds = sample(1:k, nrow(Boston),replace = T)
cv.errors = matrix(NA, k, 13, dimnames = list(NULL, paste(1:13)))
for(j in 1:k){
  best.fit = regsubsets(crim~., data = Boston[folds != j,],nvmax = 13)
  for(i in 1:13){
    pred = predict(best.fit,Boston[folds == j,], id=i)
    cv.errors[j,i] = mean((Boston$crim[folds ==j] - pred)^2)
  }
}
mean.cv.errors = apply(cv.errors,2,mean)
rmse = sqrt(mean.cv.errors)
par(mfrow=c(1,1))
plot(rmse, pch = 19, type = "b")
```



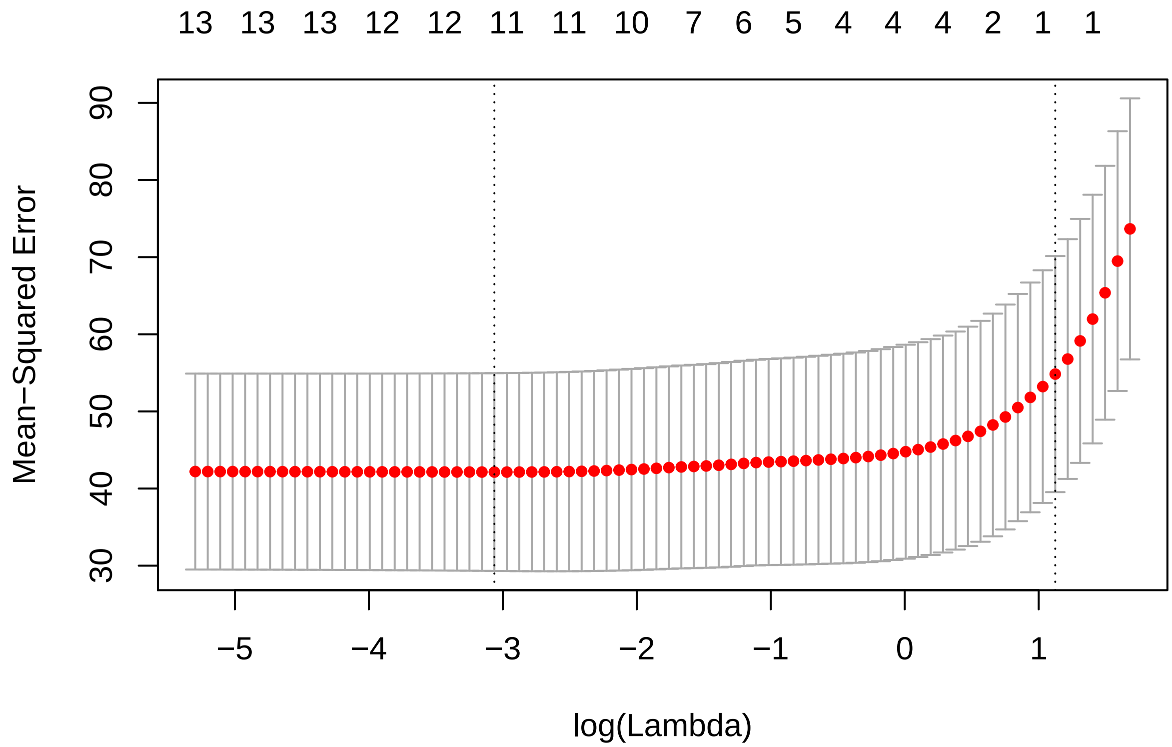

```r
min.ind = which.min(rmse)
min.ind
```

```
## 12
## 12
```

```r
# 12 is minimum index
rmse[min.ind]
```

```
##       12
## 6.405823
```

```r
# RMSE = 6.57

# Lasso
x = model.matrix(crim~., data = Boston)
y = Boston$crim
lasso = cv.glmnet(x,y,type.measure = 'mse')
plot(lasso)
```



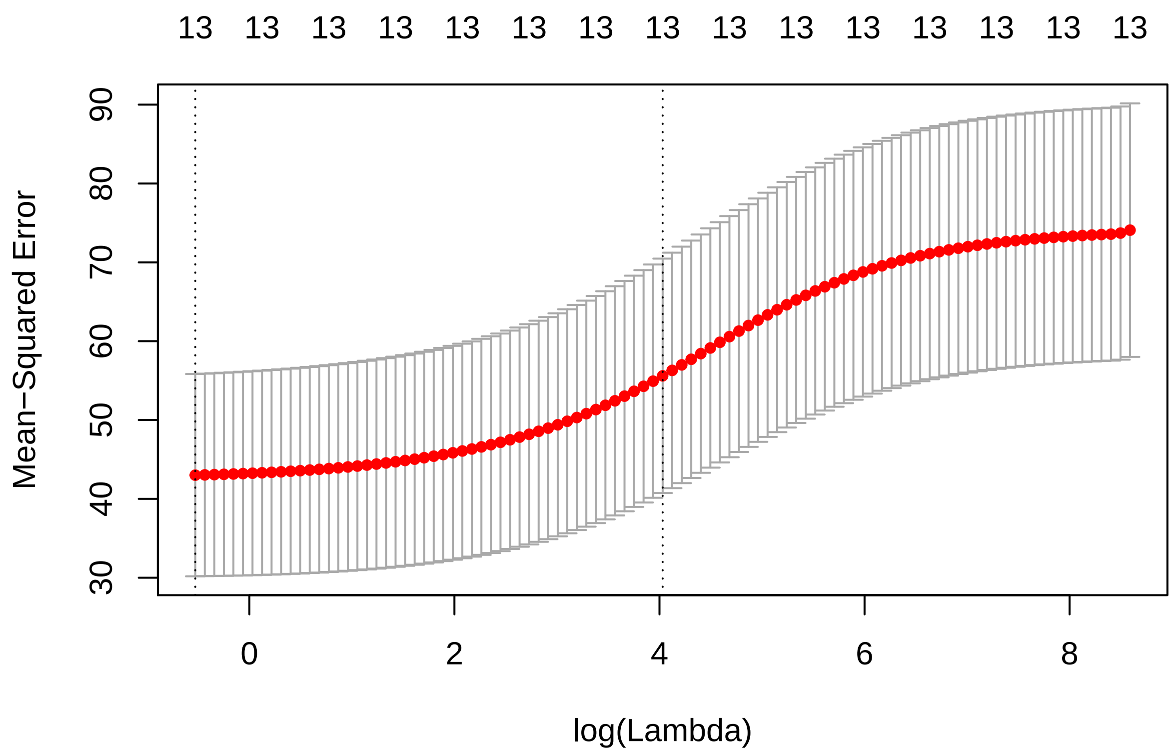

```
lasso$lambda.min
```

```
## [1] 0.04674894
```

```
# Best lambda = 0.047
lasso.mse = lasso$cvm[lasso$lambda == lasso$lambda.min]
sqrt(lasso.mse)
```

```
## [1] 6.491124
```

```
# RMSE = 6.56

# Ridge
ridge = cv.glmnet(x,y, type.measure = 'mse',alpha = 0)
plot(ridge)
```




```r
ridge$lambda.min
```

```
## [1] 0.5899047
```

```r
# Best Lambda = 0.59
ridge.mse = ridge$cvm[ridge$lambda == ridge$lambda.min]
sqrt(ridge.mse)
```

```
## [1] 6.558329
```

```r
# RMSE = 6.55

# PCR
pcr.fit = pcr(crim~., data=Boston, scale=T, validation = 'CV')
summary (pcr.fit)
```

```
## Data:    X dimension: 506 13
##  Y dimension: 506 1
## Fit method: svdpc
## Number of components considered: 13
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            8.61    7.170    7.163    6.733    6.728    6.740    6.765
## adjCV         8.61    7.169    7.162    6.730    6.723    6.737    6.760
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       6.760    6.634    6.652     6.642     6.652     6.607     6.546
## adjCV    6.754    6.628    6.644     6.635     6.643     6.598     6.536
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
```

```
## X       47.70    60.36    69.67    76.45    82.99    88.00    91.14
## crim    30.69    30.87    39.27    39.61    39.61    39.86    40.14
##       8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## X       93.45    95.40     97.04     98.46     99.52     100.0
## crim    42.47    42.55     42.78     43.04     44.13      45.4
```

```
# 13 Variable RMSE = 6.52
```

a. While trying to analyze the Boston data set for influences on the crime rate, regressions were run with subset selection, Lasso regression, ridge regression, and PCR.

```
# B - Which model to use? Include validation error, MSE, etc.
```

b. For the subset selection method, 12 was found as the minimum index and that produced an RMSE of 6.57. For the Lasso regression, the minimum lambda of 0.047 was found and used to result an RMSE of 6.56. With the ridge regression, the minimum lambda was found to be 0.59; this produced a RMSE of 6.55. Lastly, using PCR, the 13-variable RMSE was output to be 6.52.

```
# C - Does the chosen model involve all features? Why or why not?
```

c. For this question, the best model to use would be the PCR using 13-variables because it had the lower RMSE of the entire set. However, all models produced very similar RMSE, so the easiest explained would also have merit for this prediction.

Chapter 8 - Questions 8

```
###########################################################
# Chapter 8 - Question 8
###########################################################

library (ISLR)
library (tree)
library(randomForest)
```

```
## randomForest 4.6-12

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

```
summary(Carseats)
```

```
##      Sales          CompPrice       Income        Advertising
##  Min.   : 0.000   Min.   : 77   Min.   : 21.00   Min.   : 0.000
##  1st Qu.: 5.390   1st Qu.:115   1st Qu.: 42.75   1st Qu.: 0.000
##  Median : 7.490   Median :125   Median : 69.00   Median : 5.000
##  Mean   : 7.496   Mean   :125   Mean   : 68.66   Mean   : 6.635
##  3rd Qu.: 9.320   3rd Qu.:135   3rd Qu.: 91.00   3rd Qu.:12.000
##  Max.   :16.270   Max.   :175   Max.   :120.00   Max.   :29.000
##    Population        Price        ShelveLoc        Age
##  Min.   : 10.0   Min.   : 24.0   Bad   : 96   Min.   :25.00
##  1st Qu.:139.0   1st Qu.:100.0   Good  : 85   1st Qu.:39.75
##  Median :272.0   Median :117.0   Medium:219   Median :54.50
##  Mean   :264.8   Mean   :115.8                Mean   :53.32
##  3rd Qu.:398.5   3rd Qu.:131.0                3rd Qu.:66.00
##  Max.   :509.0   Max.   :191.0                Max.   :80.00
##    Education     Urban       US
##  Min.   :10.0   No :118   No :142
##  1st Qu.:12.0   Yes:282   Yes:258
```

```
##  Median :14.0
##  Mean   :13.9
##  3rd Qu.:16.0
##  Max.   :18.0
```

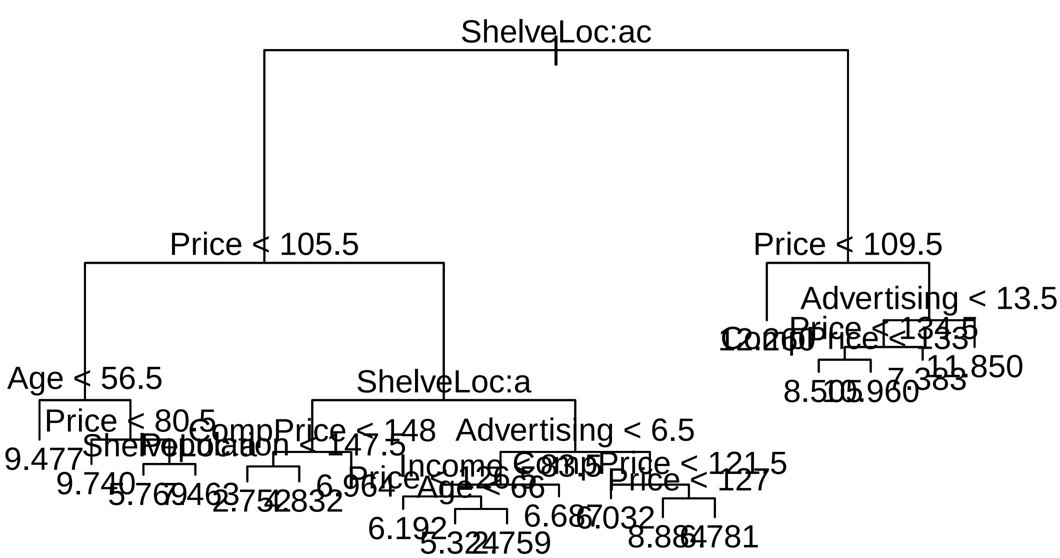
```
# A - separate into test and sample set.

set.seed(1)
train = sample(1:nrow(Carseats), nrow(Carseats)*0.8)
carseats.test = Carseats[-train,]
```

a. The Carseats dataset was divided into two subgroups. 80% of the data was randomly selected to be put into the train set, and the remaining 20% were stored in the test set.

```
# B - Regression tree

tree.carseats = tree(Sales~., data = Carseats[train,])
summary(tree.carseats)
```

```
##
## Regression tree:
## tree(formula = Sales ~ ., data = Carseats[train, ])
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "Age"         "CompPrice"   "Population"
## [6] "Advertising" "Income"
## Number of terminal nodes:  19
## Residual mean deviance:  2.452 = 738.2 / 301
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -3.80300 -0.97550 -0.06679  0.00000  0.95970  5.30800
```

```
plot(tree.carseats)
text(tree.carseats)
```



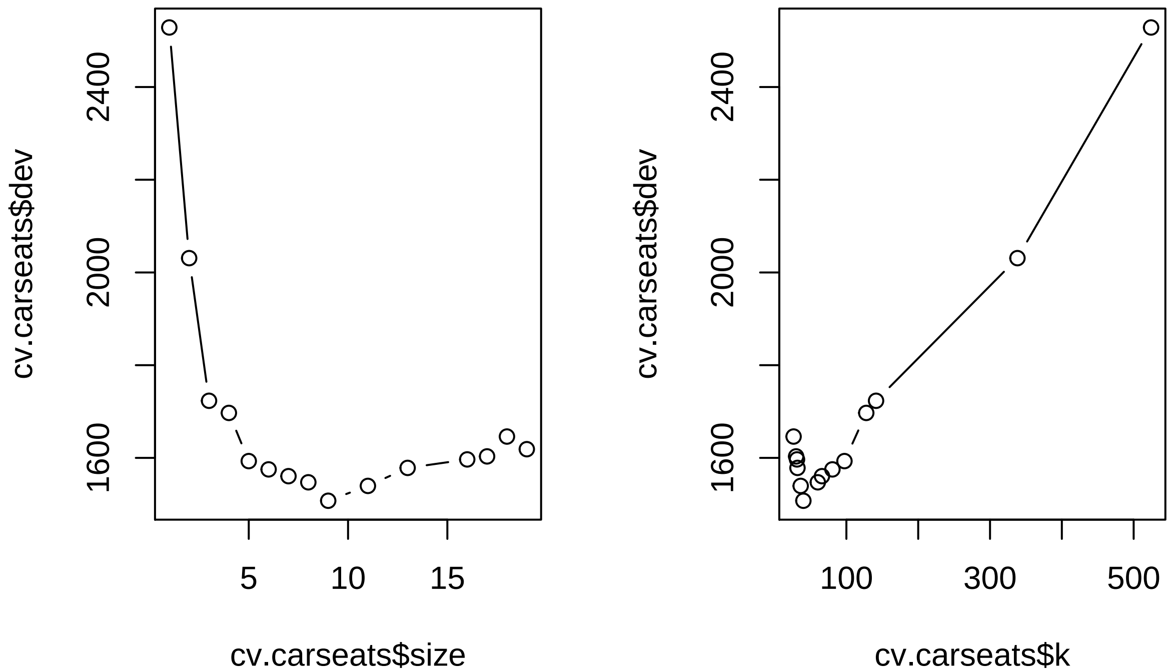

```
yhat.carseats = predict(tree.carseats, carseats.test)
mean((yhat.carseats - carseats.test$Sales)^2)
```

```
## [1] 4.817033
```

```r
# MSE = 4.817
```

b. The following tree diagram shows the regression tree based on the training set of the dataset. The MSE that was produced with this regression was 4.817.

```r
# C - Cross-validation to determine optimal tree complexity

cv.carseats = cv.tree(tree.carseats, FUN = prune.tree)
par(mfrow = c(1,2))
plot(cv.carseats$size, cv.carseats$dev, type = 'b')
plot(cv.carseats$k, cv.carseats$dev, type = 'b')
```




```r
# Looks like approximately 8 - 10 is the best choice
pruned.carseats8 = prune.tree(tree.carseats, best = 8)
summary(pruned.carseats8)
```

```
##
## Regression tree:
## snip.tree(tree = tree.carseats, nodes = c(14L, 22L, 10L, 23L,
## 9L))
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "Age"         "Advertising"
## Number of terminal nodes:  8
## Residual mean deviance:  3.636 = 1135 / 312
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -5.6870 -1.2370  0.0211  0.0000  1.2320  5.6890
```

```r
yhat8 = predict(pruned.carseats8, carseats.test)
mean((yhat8 - carseats.test$Sales)^2)
```

```
## [1] 4.97754
```

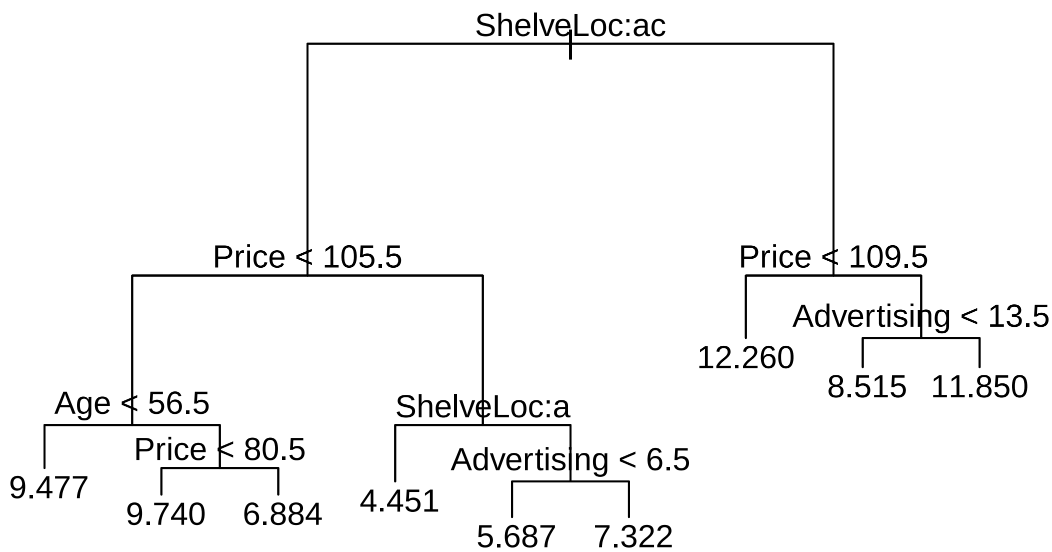
```r
pruned.carseats9 = prune.tree(tree.carseats, best = 9)
summary(pruned.carseats9)
```

```
##
## Regression tree:
## snip.tree(tree = tree.carseats, nodes = c(19L, 14L, 22L, 10L,
## 23L))
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "Age"         "Advertising"
## Number of terminal nodes:  9
## Residual mean deviance:  3.455 = 1074 / 311
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -5.68700 -1.29200  0.01451  0.00000  1.20300  5.68900
```

```
yhat9 = predict(pruned.carseats9, carseats.test)
mean((yhat9 - carseats.test$Sales)^2)
```

```
## [1] 4.831068
```

```
# 9 produces slightly lower MSE (4.831) than 8 nodes (4.978),
# but both were higher than before

par(mfrow = c(1,1))
plot(pruned.carseats9)
text(pruned.carseats9)
```



c. Upon using cross validation of the train and test data set, the optimal level of pruning was 9 terminal nodes. See the plots below to determine where those values were found.

When selecting the pruning method to stop at 9 terminal nodes, the MSE that was produced was 4.831. This was however not an improvement from the tree diagram that incorporated for all variables. The tree diagram for the 9-node pruned tree is shown below.

```
# D - Bagging

bag.carseats = randomForest(Sales~.,
                            data = Carseats[train,],
                            mtry = 10,
                            ntree = 500,
```

```
                                 importance = T)
yhat.bag = predict(bag.carseats, carseats.test)
mean((yhat.bag - carseats.test$Sales)^2)
```

```
## [1] 2.066403
```

```
# Bagging reduces RME to 2.077

importance(bag.carseats)
```

```
##                %IncMSE IncNodePurity
## CompPrice   31.5650048     255.53201
## Income       8.9309599     135.78725
## Advertising 24.7524962     203.48475
## Population  -2.1969232      90.53550
## Price       72.2076071     781.44000
## ShelveLoc   71.1827145     665.42105
## Age         23.1708502     232.03818
## Education    2.3070653      63.58163
## Urban       -0.4741838      10.25325
## US           2.3312105      11.06336
```

```
# Top three most important in terms of % Increase of MSE are
# Price, ShelveLoc, and Age with Advertising and CompPrice next.
```

d. In the bagging approach for this method, with a 500-tree set. This method produced a test error rate that was lower than a single tree diagram and the cross-validation method as the MSE for this method was 2.077. After utilizing the importance function, Price, ShelveLoc, and Age are the top three most important variables for prediction.

```
# E - Random Forest

rf.carseats = randomForest(Sales~.,
                           data=Carseats[train,],
                           mtry = 5,
                           ntree = 500,
                           importance = T)
yhat.rf = predict(rf.carseats, carseats.test)
mean((yhat.rf - carseats.test$Sales)^2)
```

```
## [1] 2.161526
```

```
# Random forest produced MSE = 2.175

importance(rf.carseats)
```

```
##               %IncMSE IncNodePurity
## CompPrice   20.925835     232.45921
## Income       9.115387     175.09659
## Advertising 22.522893     239.29988
## Population  -1.514122     116.00575
## Price       60.213810     684.93441
## ShelveLoc   67.218882     610.56365
## Age         20.874616     245.75551
## Education    2.474034      85.85813
## Urban       -1.563751      14.30754
## US           4.389134      23.56289
```

```r
# Top two are clearly Price and ShelveLoc, and next three are
# CompPrice, Advertising, and Age
```

e. Utilizing the random forest function with 500 trees and depth of 5 nodes per tree, the MSE produced from training and testing data sets was 2.175. Again, using the importance function, the top two predicting factors for the random forest were Price and ShelveLoc; with CompPrice, Advertising and Age also in the top five.

Chapter 8 - Questions 11

```r
######################################################
# Chapter 8 - Question 11
######################################################

library(gbm)
```

```
## Loading required package: survival
## Loading required package: lattice
## Loading required package: splines
## Loading required package: parallel
## Loaded gbm 2.1.3
```

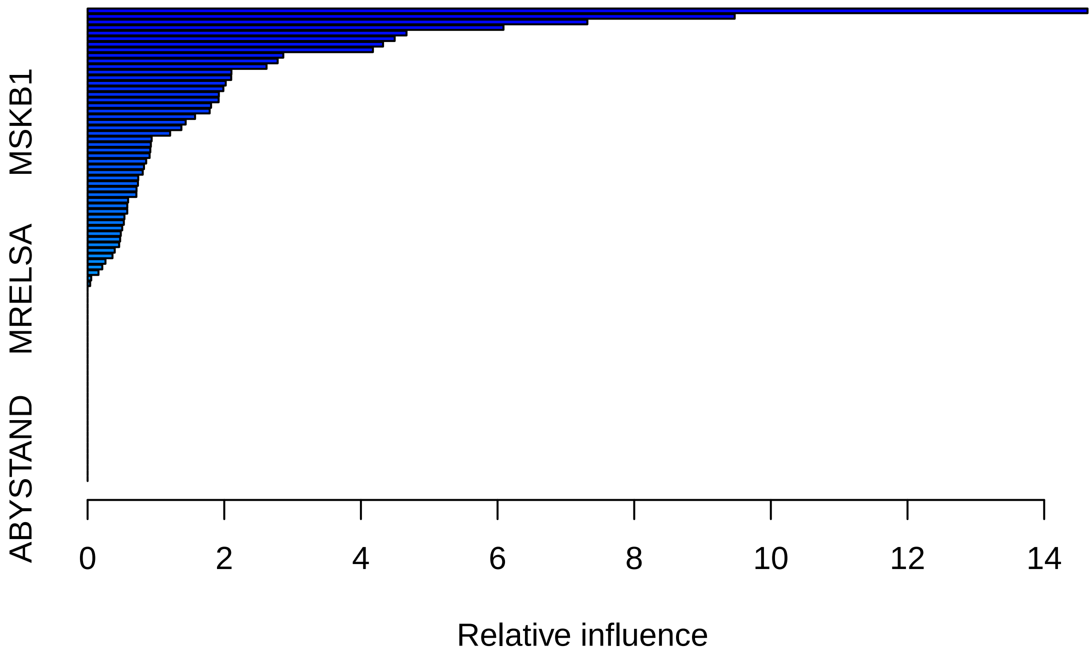
```r
# A - 1000 observation training set
train = c(1:1000)

Caravan$Purchase = ifelse (Caravan$Purchase == 'Yes',1,0)
caravan.train = Caravan[train,]
caravan.test = Caravan[-train,]
```

a. A training set of the first 1000 responses was set and the remaining responses were set in the test set.

```r
# B - Boosting with purchasing. 1000 trees and .01 shrink

set.seed(1)
boost.caravan = gbm(Purchase~., data = caravan.train,
                    distribution = 'bernoulli', n.trees=1000,
                    shrinkage = 0.01)
```

```
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 50: PVRAAUT has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 71: AVRAAUT has no variation.
```

```r
summary(boost.caravan)
```




```
##                var     rel.inf
## PPERSAUT PPERSAUT 14.63504779
## MKOOPKLA MKOOPKLA  9.47091649
## MOPLHOOG MOPLHOOG  7.31457416
## MBERMIDD MBERMIDD  6.08651965
## PBRAND     PBRAND  4.66766122
## MGODGE     MGODGE  4.49463264
## ABRAND     ABRAND  4.32427755
## MINK3045 MINK3045  4.17590619
## MOSTYPE   MOSTYPE  2.86402583
## PWAPART   PWAPART  2.78191075
## MAUT1       MAUT1  2.61929152
## MBERARBG MBERARBG  2.10480508
## MSKA         MSKA  2.10185152
## MAUT2       MAUT2  2.02172510
## MSKC         MSKC  1.98684345
## MINKGEM   MINKGEM  1.92122708
## MGODPR     MGODPR  1.91777542
## MBERHOOG MBERHOOG  1.80710618
## MGODOV     MGODOV  1.78693913
## PBYSTAND PBYSTAND  1.57279593
## MSKB1       MSKB1  1.43551401
## MFWEKIND MFWEKIND  1.37264255
## MRELGE     MRELGE  1.20805179
## MOPLMIDD MOPLMIDD  0.93791970
## MINK7512 MINK7512  0.92590720
## MINK4575 MINK4575  0.91745993
## MGODRK     MGODRK  0.90765539
## MFGEKIND MFGEKIND  0.85745374
## MZPART     MZPART  0.82531066
## MRELOV     MRELOV  0.80731252
## MINKM30   MINKM30  0.74126812
## MHKOOP     MHKOOP  0.73690793
```

```
## MZFONDS   MZFONDS  0.71638323
## MAUTO        MAUTO  0.71388052
## MHHUUR      MHHUUR  0.59287247
## APERSAUT APERSAUT  0.58056986
## MOSHOOFD MOSHOOFD  0.58029563
## MSKB2        MSKB2  0.53885275
## PLEVEN      PLEVEN  0.53052444
## MINK123M MINK123M  0.50660603
## MBERARBO MBERARBO  0.48596479
## MGEMOMV   MGEMOMV  0.47614792
## PMOTSCO   PMOTSCO  0.46163590
## MSKD          MSKD  0.39735297
## MBERBOER MBERBOER  0.36417546
## MGEMLEEF MGEMLEEF  0.26166240
## MFALLEEN MFALLEEN  0.21448118
## MBERZELF MBERZELF  0.15906143
## MOPLLAAG MOPLLAAG  0.05263665
## MAANTHUI MAANTHUI  0.03766014
## MRELSA      MRELSA  0.00000000
## PWABEDR   PWABEDR  0.00000000
## PWALAND   PWALAND  0.00000000
## PBESAUT   PBESAUT  0.00000000
## PVRAAUT   PVRAAUT  0.00000000
## PAANHANG PAANHANG  0.00000000
## PTRACTOR PTRACTOR  0.00000000
## PWERKT      PWERKT  0.00000000
## PBROM        PBROM  0.00000000
## PPERSONG PPERSONG  0.00000000
## PGEZONG   PGEZONG  0.00000000
## PWAOREG   PWAOREG  0.00000000
## PZEILPL   PZEILPL  0.00000000
## PPLEZIER PPLEZIER  0.00000000
## PFIETS      PFIETS  0.00000000
## PINBOED   PINBOED  0.00000000
## AWAPART   AWAPART  0.00000000
## AWABEDR   AWABEDR  0.00000000
## AWALAND   AWALAND  0.00000000
## ABESAUT   ABESAUT  0.00000000
## AMOTSCO   AMOTSCO  0.00000000
## AVRAAUT   AVRAAUT  0.00000000
## AAANHANG AAANHANG  0.00000000
## ATRACTOR ATRACTOR  0.00000000
## AWERKT      AWERKT  0.00000000
## ABROM        ABROM  0.00000000
## ALEVEN      ALEVEN  0.00000000
## APERSONG APERSONG  0.00000000
## AGEZONG   AGEZONG  0.00000000
## AWAOREG   AWAOREG  0.00000000
## AZEILPL   AZEILPL  0.00000000
## APLEZIER APLEZIER  0.00000000
## AFIETS      AFIETS  0.00000000
## AINBOED   AINBOED  0.00000000
## ABYSTAND ABYSTAND  0.00000000
```

```r
# Three most impactful are Ppersaut, Mkoopkla, and Moplhoog
```

b. While using the boosting model with 1000 trees and shrinkage of 0.01, it was discovered that the most important factors were Ppersaut, Mkoopkla, and Moplhoog.

```r
# C - Boosting to predict data

boost.prob = predict(boost.caravan, caravan.test,
                            n.trees = 1000, type = 'response')
boost.predict = ifelse(boost.prob > .2, 1, 0)
table(caravan.test$Purchase, boost.predict)
```

```
##    boost.predict
##        0    1
##   0 4410  123
##   1  256   33
```

```r
# Predicted who end up buying diveded by total predicted
33 / (123 + 33)
```

```
## [1] 0.2115385
```

```r
# 21.15% predicted to buy actually do.

# Compare to binomial logistic regression predictor
lm.caravan = glm(Purchase~., data = caravan.train, family = 'binomial')
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
lm.prob = predict(lm.caravan, caravan.test)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```r
lm.predict = ifelse(lm.prob>.2,1,0)
table(caravan.test$Purchase,lm.predict)
```

```
##    lm.predict
##        0    1
##   0 4451   82
##   1  274   15
```

```r
# Predicted who end up buying divided by total predicted
15 / (15 + 82)
```

```
## [1] 0.1546392
```

```r
# 15.46% predicted to buy actually do logistic regression
```

c. To determine the probability that a consumer would buy a good, boosting was again used to predict the number of likely consumers (those with greater than 20% chance of purchasing). Of those that were predicted to buy (156), only 33 of those ended up purchasing. This gives the fraction of 33/156 who in fact made the purchase – approximately 21.15%. In comparison, if we were to do a binomial logistic regression, the predicted number of 97 was lower. Of those, only 15 ended up making the purchase, which yielded a 15/97 fraction, or 15.46%.

Problem 1

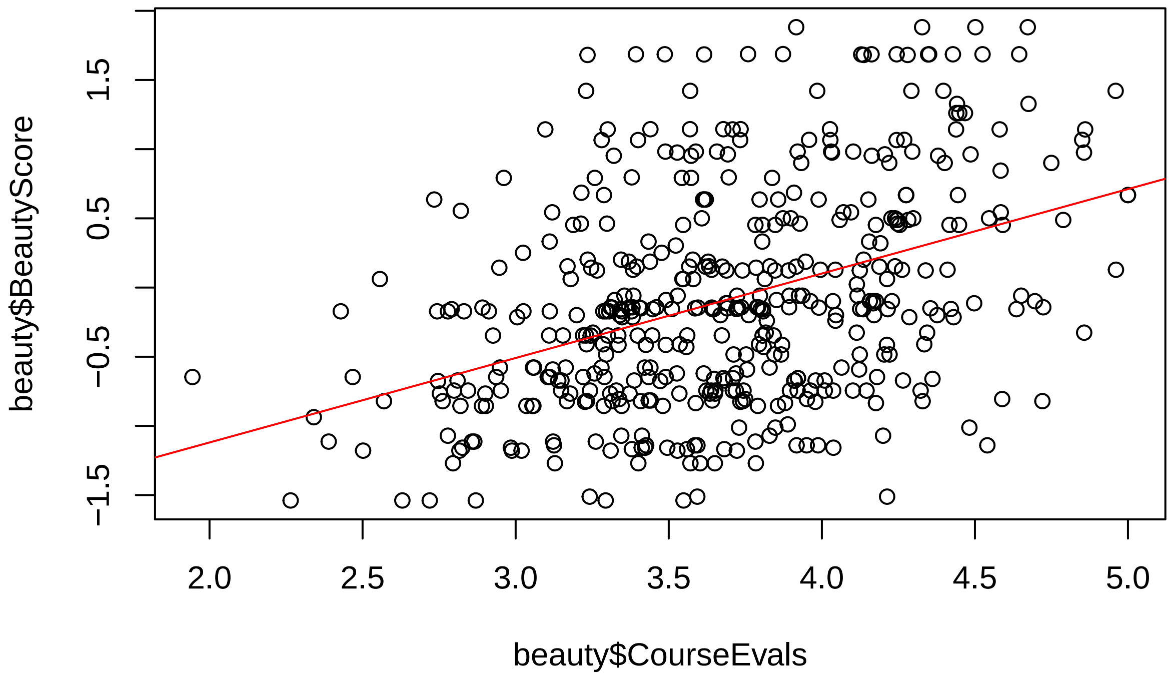
```r
########################################################
# Beauty Pays
########################################################

beauty = read.csv("~/Downloads/BeautyData.csv",header=T)
summary(beauty)
```

```
##   CourseEvals     BeautyScore           female           lower
##  Min.   :1.944   Min.   :-1.53884   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:3.326   1st Qu.:-0.74462   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :3.682   Median :-0.15636   Median :0.0000   Median :0.0000
##  Mean   :3.689   Mean   :-0.08835   Mean   :0.4212   Mean   :0.3391
##  3rd Qu.:4.067   3rd Qu.: 0.45725   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :5.000   Max.   : 1.88167   Max.   :1.0000   Max.   :1.0000
##    nonenglish        tenuretrack
##  Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:1.0000
##  Median :0.00000   Median :1.0000
##  Mean   :0.06048   Mean   :0.7797
##  3rd Qu.:0.00000   3rd Qu.:1.0000
##  Max.   :1.00000   Max.   :1.0000
```

```r
# 1 - Estimate effect of beauty into course ratings

lm.beauty = lm(BeautyScore ~ CourseEvals, data = beauty)
summary(lm.beauty)
```

```
##
## Call:
## lm(formula = BeautyScore ~ CourseEvals, data = beauty)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.74275 -0.54333 -0.08121  0.44552  2.04700
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.34054    0.23773  -9.846   <2e-16 ***
## CourseEvals  0.61045    0.06379   9.569   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7211 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
```

```r
plot(beauty$CourseEvals, beauty$BeautyScore)
abline(lm.beauty, col='red')
```




```r
lm.beauty2 = lm(BeautyScore ~., data = beauty)
summary(lm.beauty2)
```

```
##
## Call:
## lm(formula = BeautyScore ~ ., data = beauty)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.59343 -0.48966 -0.07571  0.44347  2.03803
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.33173    0.27516 -12.108  < 2e-16 ***
## CourseEvals  0.78367    0.06553  11.959  < 2e-16 ***
## female       0.41515    0.06726   6.173 1.48e-09 ***
## lower        0.32081    0.07184   4.466 1.01e-05 ***
## nonenglish   0.24398    0.13699   1.781   0.0756 .
## tenuretrack  0.06889    0.07874   0.875   0.3821
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6859 on 457 degrees of freedom
## Multiple R-squared:  0.2518, Adjusted R-squared:  0.2436
## F-statistic: 30.76 on 5 and 457 DF,  p-value: < 2.2e-16
```

```r
# There is an influence on beauty with course evaluations.
```

1. The first step in this problem was to see if there was a significant coefficient for BeautyScore and CourseEvals. To do this, I created a linear regression with only these two variables and saw a t value of over 9.5, meaning that there was essentially no chance for this not to be significant. The lead coefficient for this variable of 0.61 (+/- 0.12 for a 95% confidence intervale) shows that this is a positively correlated relationship, meaning that if the professor was more attractive, it is likely that their course evaluations

were higher. After seeing this, a second regression was done with all variables in comparison with price. What resulted was that, again, BeautyScore and CourseEvals were correlated. In addition though, it turned out the teaching lower classes, and whether a professor was male or female also had significant correlation as well.

```
# 2 - See Solution Below
```

2. Dr. Hamermesh's statement given about this study is very accurate. Based on this study, there is no way to tell if a teacher with good looks is rated higher because of their looks, or they are simple a better teacher. If one were looking to answer that question, a different study would have to be designed to do so.

Problem 2

```
#######################################################
# Housing Price Structure
#######################################################

houses = read.csv("~/Downloads/MidCity.csv",header=T)
summary (houses)
```

```
##       Home             Nbhd           Offers           SqFt       Brick
##  Min.   :  1.00   Min.   :1.000   Min.   :1.000   Min.   :1450   No :86
##  1st Qu.: 32.75   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:1880   Yes:42
##  Median : 64.50   Median :2.000   Median :3.000   Median :2000
##  Mean   : 64.50   Mean   :1.961   Mean   :2.578   Mean   :2001
##  3rd Qu.: 96.25   3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:2140
##  Max.   :128.00   Max.   :3.000   Max.   :6.000   Max.   :2590
##     Bedrooms       Bathrooms         Price
##  Min.   :2.000   Min.   :2.000   Min.   : 69100
##  1st Qu.:3.000   1st Qu.:2.000   1st Qu.:111325
##  Median :3.000   Median :2.000   Median :125950
##  Mean   :3.023   Mean   :2.445   Mean   :130427
##  3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:148250
##  Max.   :5.000   Max.   :4.000   Max.   :211200
```

```
houses$Nbhd = as.factor(houses$Nbhd)
nbhd.model = as.data.frame(model.matrix(Price~., data = houses))
lm.houses = lm(houses$Price ~., data = houses)
summary(lm.houses)
```

```
##
## Call:
## lm(formula = houses$Price ~ ., data = houses)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27897.8  -6074.8    -48.7   5551.8  27536.4
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2037.726   8911.501   0.229 0.819524
## Home          -11.456     25.387  -0.451 0.652616
## Nbhd2       -1729.613   2433.756  -0.711 0.478675
## Nbhd3       20534.706   3176.051   6.465 2.33e-09 ***
```

```
## Offers      -8350.128   1103.693  -7.566 8.96e-12 ***
## SqFt           53.634      5.926   9.051 3.30e-15 ***
## BrickYes    17313.540   1988.548   8.707 2.12e-14 ***
## Bedrooms     4136.461   1621.775   2.551 0.012023 *
## Bathrooms    7975.157   2133.831   3.737 0.000287 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10050 on 119 degrees of freedom
## Multiple R-squared:  0.8688, Adjusted R-squared:   0.86
## F-statistic: 98.54 on 8 and 119 DF,  p-value: < 2.2e-16
```

```r
# 1 - Is there a brick house premium? All others  constant.
  # Yes. Brick = $17,323.54

# 2 - Is there a premium for neighborhood 3?
  # Yes. Nbhd3 = $20,534.71
```

1. & 2. As is shown in the table above, both bring a brick house (BrickYes) and being in neighborhood 3 (Nbhd3) are siginicant value increases on the overall price of a house. Brick being a premium that increases the price of comparable house by an estimate of \$17,323.54 (+/- \$3,976 for a 95% confidence interval), and neighborhood 3 adding approximately \$20,534.71 (+/- \$6,352 for a 95% confidence interval). With t values of 8.707 and 6.465 respecifively, it can be said easily that each is significant as the chance of either lead coefficient being zero extremely small.

```r
# 3 - Extra premium for brick in neighborhood 3?

lm.nbhd3_brick = lm(houses$Price ~. + Nbhd3 * BrickYes, data = nbhd.model)
summary(lm.nbhd3_brick)
```

```
##
## Call:
## lm(formula = houses$Price ~ . + Nbhd3 * BrickYes, data = nbhd.model)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27515.9  -5681.0   -459.6   4451.0  26695.6
##
## Coefficients: (1 not defined because of singularities)
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     2885.512   8738.695   0.330  0.74183
## `(Intercept)`         NA         NA      NA       NA
## Home             -11.799     24.875  -0.474  0.63613
## Nbhd2           -846.146   2412.025  -0.351  0.72636
## Nbhd3          17086.915   3417.999   4.999 2.02e-06 ***
## Offers         -8486.348   1082.875  -7.837 2.26e-12 ***
## SqFt              54.726      5.823   9.397 5.36e-16 ***
## BrickYes       13839.320   2413.580   5.734 7.69e-08 ***
## Bedrooms        4605.046   1600.639   2.877  0.00477 **
## Bathrooms       6556.432   2170.200   3.021  0.00309 **
## Nbhd3:BrickYes 10192.783   4178.971   2.439  0.01621 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9850 on 118 degrees of freedom
## Multiple R-squared:  0.8751, Adjusted R-squared:  0.8656
```

```
## F-statistic:  91.9 on 9 and 118 DF,  p-value: < 2.2e-16
```

```
# Yes. Nbhd3 & Brick = $10,192.78
```

3. To analyze if there was a significant impact of an extra premium to have a brick house in neighborhood 3, an additional regression was done with all variables from above, but an addition of an interaction term between Nbhd3 and BrickYes. Once completed, the regression give a significant outcome of $10,192.78 (+/- $8,358 for a 95% confidence interval), telling that there is very little chance for this coefficient to be zero, and thus has significance.

```
# 4 - Can we combine neighborhood 1 and 2 into a single 'older' neighborhood?

# No. Explained below.
```

4. Utilizing the summary table from part 1 and 2, one can see that the premium of being in neighborhood 2 is possibly not significant. The 95% confidence interval includes zero as a possible value, and thus, could not have an impact on the house price. However, this does not necessarily mean that neighborhood 1 and 2 should be combined together as a single 'older neighborhood' category. This is making the assumption that both neighborhoods are in comparable locations, have similar sized houses, and desirable neighbors. All these assumptions are not something that can be said based on the data set, and placing them together could incorporate bias into the question.

Problem 3

```
##########################################################
# What causes what??
##########################################################

# Questions 1 - 4 answered below.
```

1. The flaws in the idea of running a regression based on crime rate and the number of police on the street are numerous. As intuitively as it is than a higher crime rate would lead to more police, one has to consider that there is a significant chance that the converse would be true as well. That relationship would have to be explored as well with a series of test and control sets to find the measure. There are two many contributing factors here to simply take those two measures, run a regression, and be able to predict well.
2. The researchers at UPENN were able to isolate the effect of police presence by studying days when there is a high-alert of a crime; and thus, more police would be present without a tie to crimes. The reaction to this would be able measure the effect that a higher police presense relates tocrime rate. Table 2 was able to compare the High-Alert rate with the crime rate while keeping ridership of the METRO system constant. What this table shows is that a larger police presense has a negative impact on crime rate to a 99% significance level.
3. The reason to have METRO ridership set during this experiment is to verify that people are there for an opportunity for crime to happen. This was done to verify that all contributing factors remain constant in order to create a reliable experiment.
4. Table 4 looks to show the effects of a high-alert times, and thus a larger police presense, with the crime-rate in certain areas of Washington DC. In this table, it is shown at a 99% level of significance that only District 1 has a correlation

Problem 4

For the Car project, I was in charge of testing and researching how to maximize with trees, random forest, and boosting. I also tested cross-validations with these methods to find the best depth, number of trees and what variables had the most variance.

After the data cleaning and predicting method were finished, I assisted in the creation of our group presentation. I was also in charge of the write-up for exploratory data analysis, tree method, and random forest regression.