IntroToML_takeHome

Rochan

2022-07-30

Book Problems

Version 1

Chapter 2 #10

This exercise involves the Boston housing data set.

Part a

```
library (MASS)
# ?Boston
summary(Boston)
```

```
##       crim                zn              indus            chas        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
##  1st Qu.: 0.08205   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
```

```
rows_indata=nrow(Boston)
print('Number of rows in Boston: ')
```

```
## [1] "Number of rows in Boston: "
```

```
print(rows_indata)
```

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

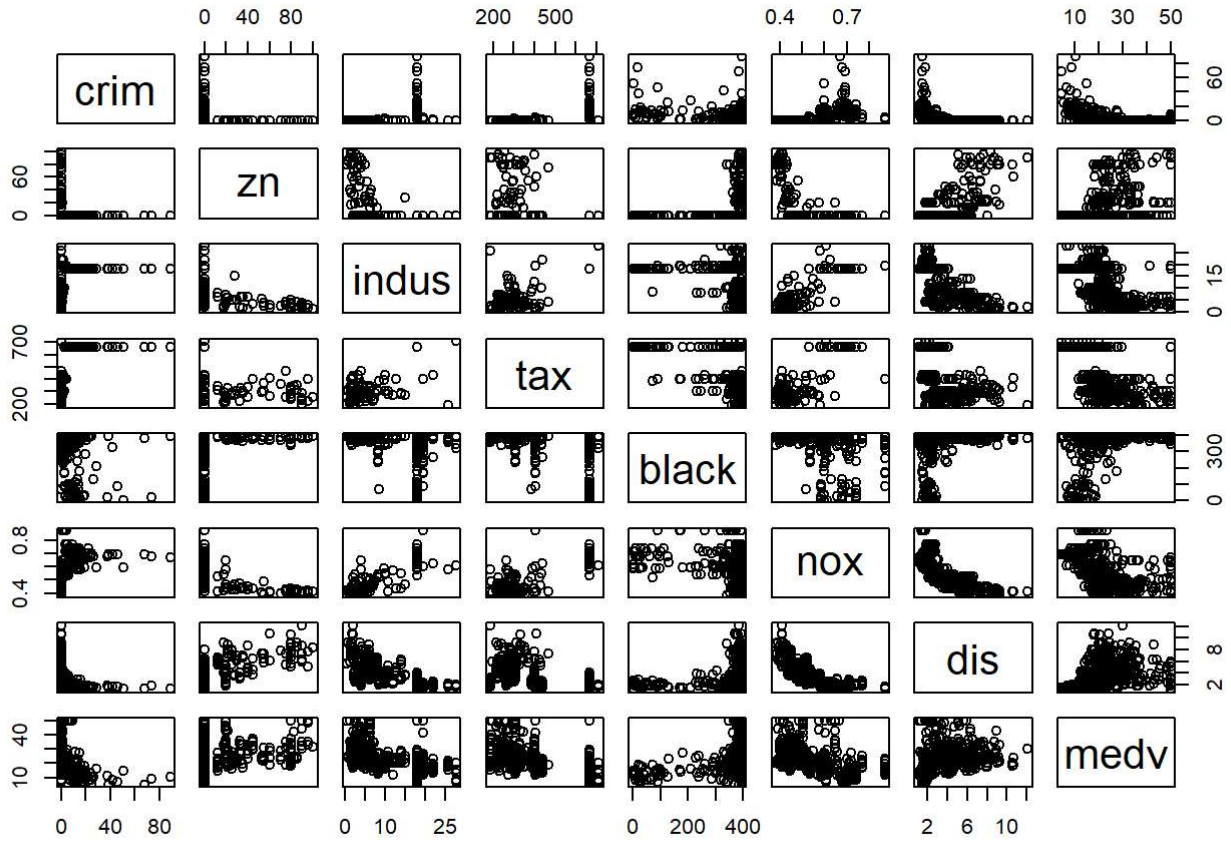
```
#?Boston #help shows number of rows and columns and their description
dim(Boston)
```

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

output shows 506 rows and 14 columns

Part B

Make some pairwise scatterplots of the predictors (columns) in this data set. Describe your findings



Observations-

- crim seems to have a negative trend with dis and medv as we can plot a negative exponential kind of curve scatter plot
- nox seems to have a negative trend with dis as we can plot a negative exponential kind of curve in their scatter plot
- dis also seems to have a negative trend with nox and a positive trend with medv
- unable to see a trend with tax
- medv has a positive trend with dis and negative trend with nox
- medv has a positive trend with black
- indus and a negative trend with dis and medv
- zn has a positive trend with dis and medv maybe

Part C

to find predictors we have to find the correlation coefficient

```
cor(Boston$crim,Boston$zn)
```

```
## [1] -0.2004692
```

```
cor(Boston$crim,Boston$indus)
```

```
## [1] 0.4065834
```

```
cor(Boston$crim,Boston$chas)
```

```
## [1] -0.05589158
```

```
cor(Boston$crim,Boston$nox)
```

```
## [1] 0.4209717
```

```
cor(Boston$crim,Boston$rm)
```

```
## [1] -0.2192467
```

```
cor(Boston$crim,Boston$age)
```

```
## [1] 0.3527343
```

```
cor(Boston$crim,Boston$dis)
```

```
## [1] -0.3796701
```

```
cor(Boston$crim,Boston$rad)
```

```
## [1] 0.6255051
```

```
cor(Boston$crim,Boston$tax)
```

```
## [1] 0.5827643
```

```
cor(Boston$crim,Boston$ptratio)
```

```
## [1] 0.2899456
```

```
cor(Boston$crim,Boston$black)
```

```
## [1] -0.3850639
```

```
cor(Boston$crim,Boston$lstat)
```

```
## [1] 0.4556215
```

```
cor(Boston$crim,Boston$medv)
```

```
## [1] -0.3883046
```

crim has a negative relationship with zn, rm, medv, dis, black. crim has a positive relationship with indus, nox, rad, tax, ptratio, lstat, age.

Part D

High crime rate= crime rate >95% of suburbs which is 2 std dev from the mean

```
nrow(Boston[which(Boston$crim > mean(Boston$crim) + 2*sd(Boston$crim)),])
```

```
## [1] 16
```

```
range(Boston$crim)
```

```
## [1]  0.00632 88.97620
```

```
mean(Boston$crim)
```

```
## [1] 3.613524
```

```
sd(Boston$crim)
```

```
## [1] 8.601545
```

```
nrow(Boston[which(Boston$tax > mean(Boston$tax) + 2*sd(Boston$tax)),])
```

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

```
nrow(Boston[which(Boston$tax > mean(Boston$tax) + sd(Boston$tax)),])
```

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

```
range(Boston$tax)
```

```
## [1] 187 711
```

```
mean(Boston$tax)
```

```
## [1] 408.2372
```

```
sd(Boston$tax)
```

```
## [1] 168.5371
```

```
nrow(Boston[which(Boston$ptratio > mean(Boston$ptratio) + 2*sd(Boston$ptratio)),])
```

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

```
nrow(Boston[which(Boston$ptratio > mean(Boston$ptratio) + sd(Boston$ptratio)),])
```

```
## [1] 56
```

```
range(Boston$ptratio)
```

```
## [1] 12.6 22.0
```

```
mean(Boston$ptratio)
```

```
## [1] 18.45553
```

```
sd(Boston$ptratio)
```

```
## [1] 2.164946
```

Observations-

- there are 16 suburbs with particularly high crime rates
- Crime ranges from as close to 0 to 89, a very wide range
- mean crime rate is 3.613
- std dev of crime rate is 8.6, that means some suburbs have extremely high crime rates as the range goes upto 89!
- there are 0 suburbs with particularly high tax rates
- there are 137 suburbs with tax rates higher than 1 std dev
- tax ranges from as close to 187 to 711, this is also a very wide range
- mean tax rate is 408.23
- std dev of tax rate is 168, that means some suburbs have extremely high tax rates.
- there are 0 suburbs with particularly high ptratio
- 56 suburbs have mean ptratio> 1 std dev.
- ptratio ranges from as close to 12.6 to 22, the range is very narrow
- mean ptratio rate is 18
- std dev of ptratio rate is 2.16

Part e

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

35 suburbs bound the Charles river

Part f

```
## [1] 19.05
```

the median pupil to teacher ratio is 19.05

Part g

```
which(Boston$medv == min(Boston$medv))
```

```
## [1] 399 406
```

there are 2 suburbs 399 and 406 that have the lowest median property values Now to calculate range, mean, sd of all columns

```
range(Boston$crim)
```

```
## [1]  0.00632 88.97620
```

```
range(Boston$zn)
```

```
## [1]   0 100
```

```
range(Boston$indus)
```

```
## [1]  0.46 27.74
```

```
range(Boston$chas)
```

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

```
range(Boston$nox)
```

```
## [1] 0.385 0.871
```

```
range(Boston$rm)
```

```
## [1] 3.561 8.780
```

```
range(Boston$age)
```

```
## [1]   2.9 100.0
```

```
range(Boston$dis) #
```

```
## [1]  1.1296 12.1265
```

```
range(Boston$rad) #
```

```
## [1]  1 24
```

```
range(Boston$tax) #
```

```
## [1] 187 711
```

```
range(Boston$ptratio) #
```

```
## [1] 12.6 22.0
```

```
range(Boston$lstat) #
```

```
## [1]  1.73 37.97
```

```
range(Boston$medv) #
```

```
## [1]  5 50
```

```
mean(Boston$crim)
```

```
## [1] 3.613524
```

```
mean(Boston$zn)
```

```
## [1] 11.36364
```

```
mean(Boston$indus)
```

```
## [1] 11.13678
```

```
mean(Boston$chas)
```

```
## [1] 0.06916996
```

```
mean(Boston$nox)
```

```
## [1] 0.5546951
```

```
mean(Boston$rm)
```

```
## [1] 6.284634
```

```
mean(Boston$age)
```

```
## [1] 68.5749
```

```
mean(Boston$dis)
```

```
## [1] 3.795043
```

```
mean(Boston$rad)
```

```
## [1] 9.549407
```

```
mean(Boston$tax)
```

```
## [1] 408.2372
```

```
mean(Boston$ptratio)
```

```
## [1] 18.45553
```

```
mean(Boston$lstat)
```

```
## [1] 12.65306
```

```
mean(Boston$medv)
```

```
## [1] 22.53281
```

```
sd(Boston$crim)
```

```
## [1] 8.601545
```

```
sd(Boston$zn)
```

```
## [1] 23.32245
```

```
sd(Boston$indus)
```

```
## [1] 6.860353
```

```
sd(Boston$chas)
```

```
## [1] 0.253994
```

```
sd(Boston$nox)
```

```
## [1] 0.1158777
```

```
sd(Boston$rm)
```

```
## [1] 0.7026171
```

```
sd(Boston$age)
```

```
## [1] 28.14886
```

```
sd(Boston$dis)
```

```
## [1] 2.10571
```

```
sd(Boston$rad)
```

```
## [1] 8.707259
```

```
sd(Boston$tax)
```

```
## [1] 168.5371
```

```
sd(Boston$ptratio)
```

```
## [1] 2.164946
```

```
sd(Boston$lstat)
```

```
## [1] 7.141062
```

```
sd(Boston$medv)
```

```
## [1] 9.197104
```

```
Boston[399,]
```

```
##        crim zn indus chas   nox    rm age    dis rad tax ptratio black lstat
## 399 38.3518  0  18.1    0 0.693 5.453 100 1.4896  24 666    20.2 396.9 30.59
##     medv
## 399    5
```

Observations for suburb 399-

- high crime rate
- lowest zn
- proportion of non-retail business acres per tow is average
- not bound to charles river
- nox very close to average
- a little less than mean rm
- proportion of owner-occupied units built prior to 1940 higher than mean age
- higher lstat(percentage of lower status of population) than average of all
- lowest median value of owner-occupied homes in $1000s.

```
##        crim zn indus chas   nox    rm age    dis rad tax ptratio  black lstat
## 406 67.9208  0  18.1    0 0.693 5.683 100 1.4254  24 666    20.2 384.97 22.98
##     medv
## 406    5
```

Observations for suburb 406-

- crim rate higher than the other suburb.
- lowest zn
- proportion of non-retail business acres per tow is average
- not bound to charles river
- nox very close to average
- a little less than mean rm

- proportion of owner-occupied units built prior to 1940 higher than mean age
- higher lstat(percentage of lower status of population) than average of all
- lowest median value of owner-occupied homes in $1000
- parameters for both these suburbs are more or less similar

Part h

```
length(which(Boston$rm > 7))
```

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

there are 64 suburbs with average more than 7 number of rooms per dwelling.

```
length(which(Boston$rm > 8))
```

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

there are 13 suburbs with average more than 7 number of rooms per dwelling.

```
summary(Boston)
```

```
##       crim                zn              indus            chas        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
##  1st Qu.: 0.08205   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
```

```
summary(subset(Boston,Boston$rm > 8))
```

```
##       crim                zn             indus              chas        
##  Min.   :0.02009   Min.   : 0.00   Min.   : 2.680   Min.   :0.0000  
##  1st Qu.:0.33147   1st Qu.: 0.00   1st Qu.: 3.970   1st Qu.:0.0000  
##  Median :0.52014   Median : 0.00   Median : 6.200   Median :0.0000  
##  Mean   :0.71879   Mean   :13.62   Mean   : 7.078   Mean   :0.1538  
##  3rd Qu.:0.57834   3rd Qu.:20.00   3rd Qu.: 6.200   3rd Qu.:0.0000  
##  Max.   :3.47428   Max.   :95.00   Max.   :19.580   Max.   :1.0000  
##       nox               rm              age              dis        
##  Min.   :0.4161   Min.   :8.034   Min.   : 8.40   Min.   :1.801  
##  1st Qu.:0.5040   1st Qu.:8.247   1st Qu.:70.40   1st Qu.:2.288  
##  Median :0.5070   Median :8.297   Median :78.30   Median :2.894  
##  Mean   :0.5392   Mean   :8.349   Mean   :71.54   Mean   :3.430  
##  3rd Qu.:0.6050   3rd Qu.:8.398   3rd Qu.:86.50   3rd Qu.:3.652  
##  Max.   :0.7180   Max.   :8.780   Max.   :93.90   Max.   :8.907  
##       rad              tax           ptratio          black       
##  Min.   : 2.000   Min.   :224.0   Min.   :13.00   Min.   :354.6  
##  1st Qu.: 5.000   1st Qu.:264.0   1st Qu.:14.70   1st Qu.:384.5  
##  Median : 7.000   Median :307.0   Median :17.40   Median :386.9  
##  Mean   : 7.462   Mean   :325.1   Mean   :16.36   Mean   :385.2  
##  3rd Qu.: 8.000   3rd Qu.:307.0   3rd Qu.:17.40   3rd Qu.:389.7  
##  Max.   :24.000   Max.   :666.0   Max.   :20.20   Max.   :396.9  
##      lstat            medv      
##  Min.   :2.47   Min.   :21.9  
##  1st Qu.:3.32   1st Qu.:41.7  
##  Median :4.14   Median :48.3  
##  Mean   :4.31   Mean   :44.2  
##  3rd Qu.:5.12   3rd Qu.:50.0  
##  Max.   :7.44   Max.   :50.0
```

For rm>8, we observe very less crime rate crim, lstat and higher medv than average.

Chapter 3 #15

Part a

statistical significance is denoted by the p-value. P value disproves the null hypothesis, that means for a p value higher than 0.05, it indicates that the variable does not have statistical significance

```
lm.zn = lm(crim~zn)
summary(lm.zn) # statistically significant
```

```
##
## 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
```

```
lm.indus = lm(crim~indus)
summary(lm.indus) # statistically significant
```

```
##
## 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
```

```
lm.chas = lm(crim~chas)
summary(lm.chas) # Very high p-value,0.2, which is greater than 0.05.
```

```
##
## 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
```

```
lm.nox = lm(crim~nox)
summary(lm.nox) # statistically significant
```

```
##
## 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
```

```
lm.rm = lm(crim~rm)
summary(lm.rm) # statistically significant
```

```
##
## 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
```

```
lm.age = lm(crim~age)
summary(lm.age) # statistically significant
```

```
##
## 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
```

```
lm.dis = lm(crim~dis)
summary(lm.dis) # statistically significant
```

```
##
## 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
```

```
lm.rad = lm(crim~rad)
summary(lm.rad) # statistically significant
```

```
##
## 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
```

```
lm.tax = lm(crim~tax)
summary(lm.tax) # statistically significant
```

```
##
## 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
```

```
lm.ptratio = lm(crim~ptratio)
summary(lm.ptratio) # statistically significant
```

```
##
## 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
```

```
lm.black = lm(crim~black)
summary(lm.black) # statistically significant
```

```
##
## 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
```

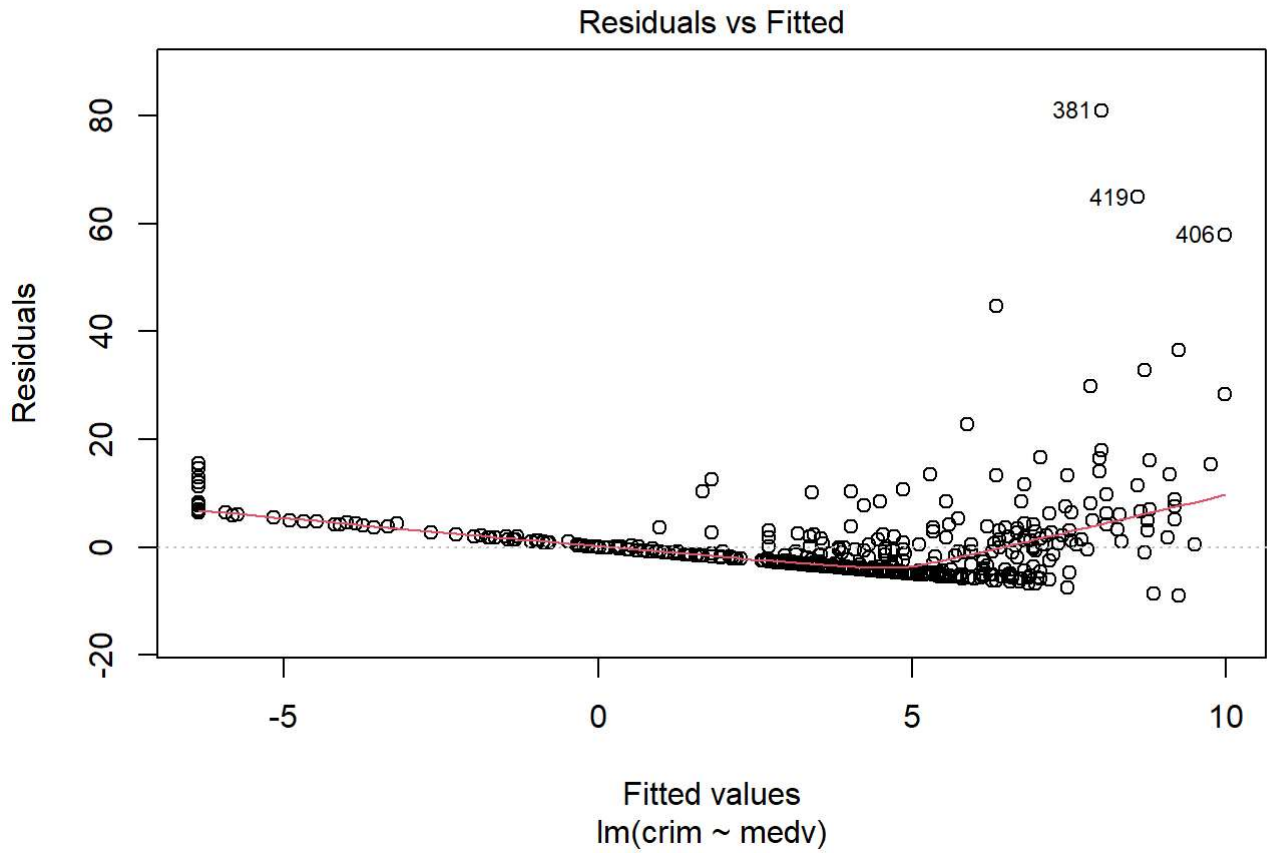
```
lm.lstat = lm(crim~lstat)
summary(lm.lstat) # statistically significant
```

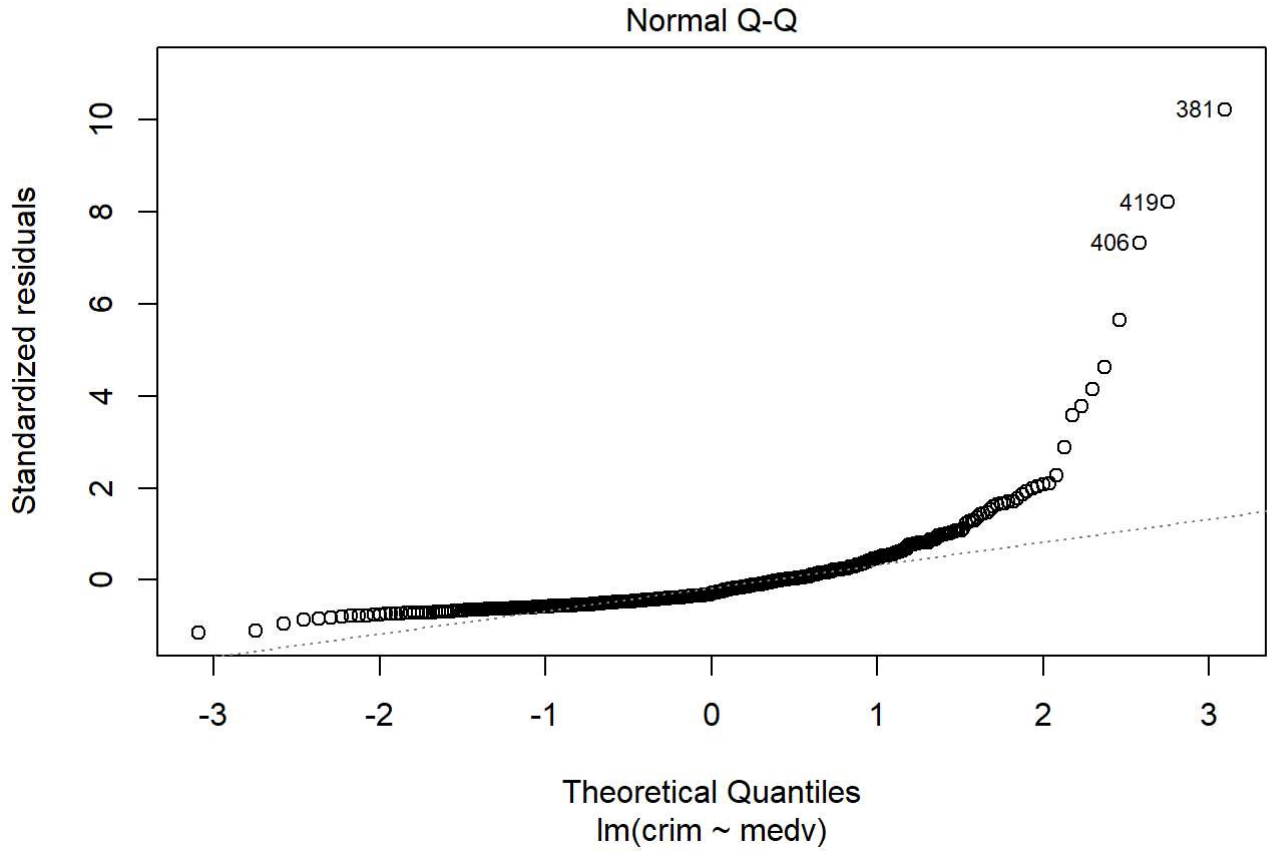
```
##
## Call:
## lm(formula = crim ~ lstat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.925  -2.822  -0.664   1.079  82.862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.33054    0.69376  -4.801 2.09e-06 ***
## lstat        0.54880    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
```

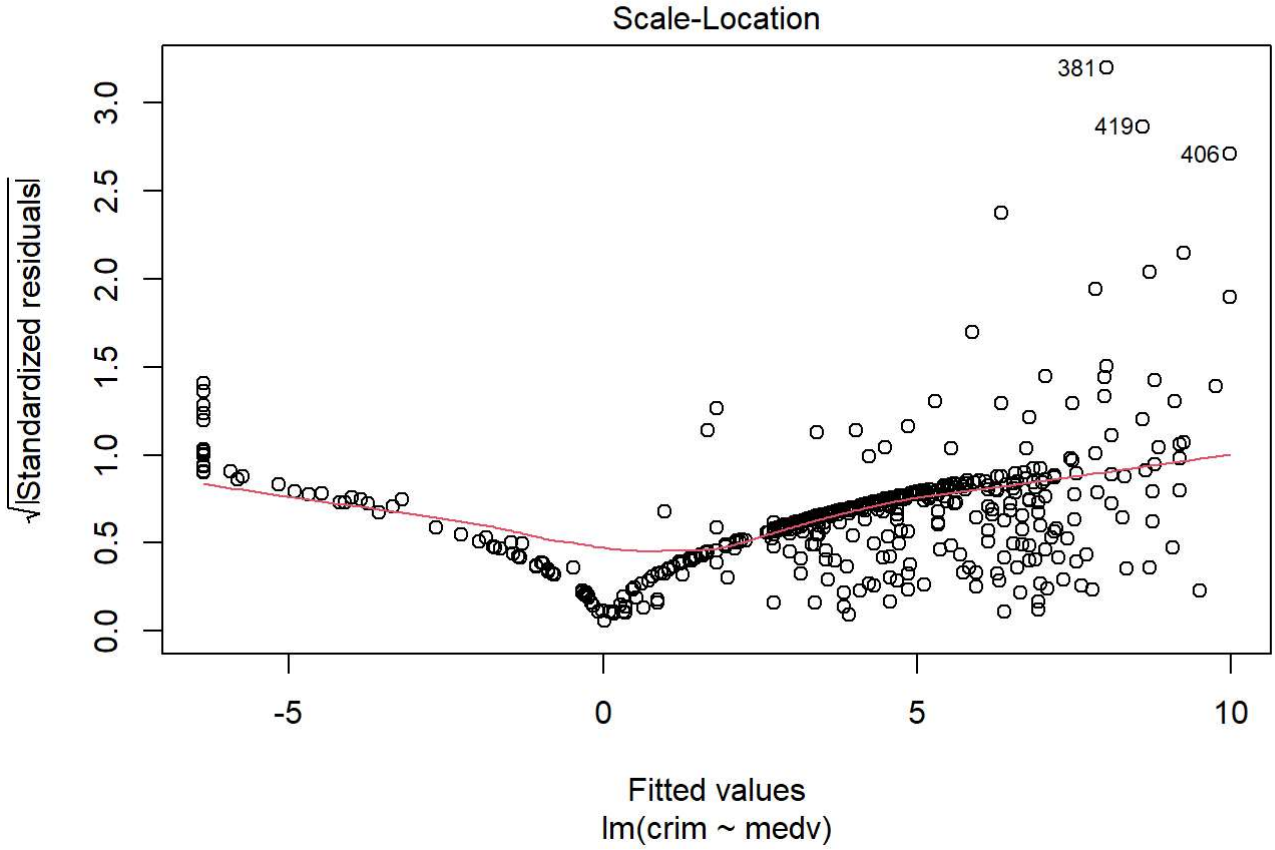
```
lm.medv = lm(crim~medv)
summary(lm.medv) # statistically significant
```

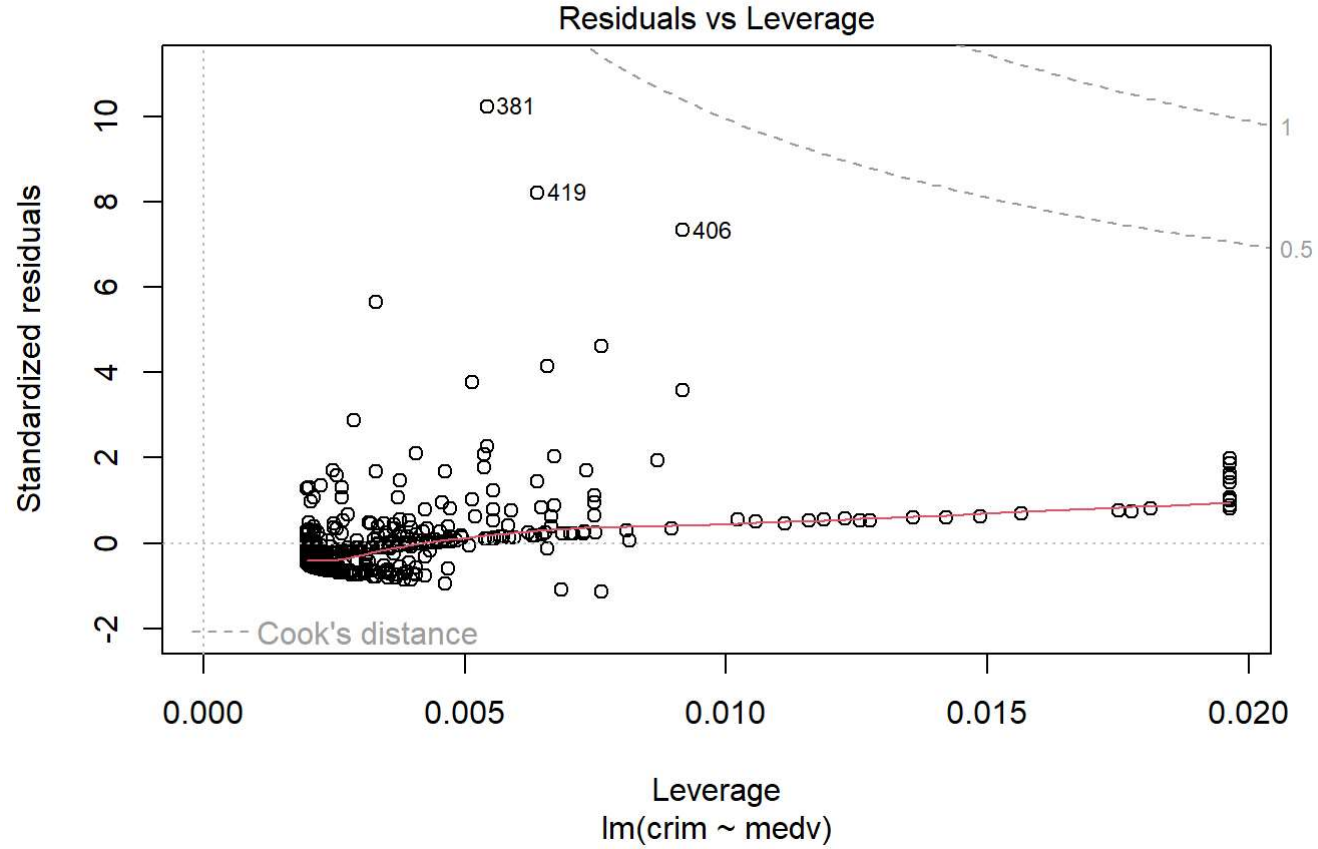
```
##
## Call:
## lm(formula = crim ~ medv)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.071 -4.022 -2.343  1.298 80.957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.79654    0.93419   12.63   <2e-16 ***
## medv        -0.36316    0.03839   -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
```

```
#plot(lm.chas)
#plot(lm.zn)
#plot(lm.nox)
#plot(lm.rm)
#plot(lm.dis)
#plot(lm.rad)
#plot(lm.tax)
#plot(lm.ptratio)
#plot(lm.lstat)
plot(lm.medv)
```


Residuals vs Fitted
Residuals
Fitted values
lm(crim ~ medv)
381
419
406
-20
0
20
40
60
80
-5
0
5
10


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(crim ~ medv)
381
419
406
0
2
4
6
8
10
-3
-2
-1
0
1
2
3



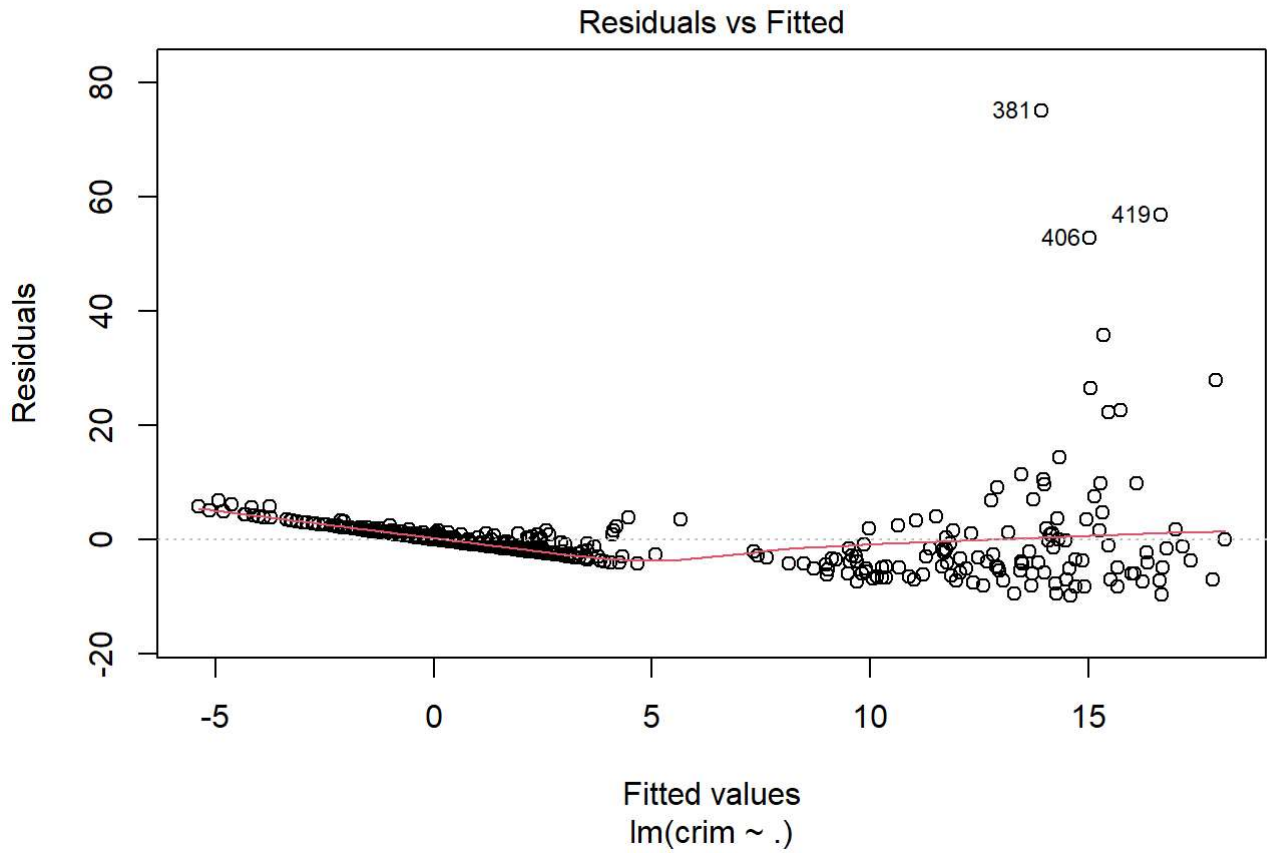


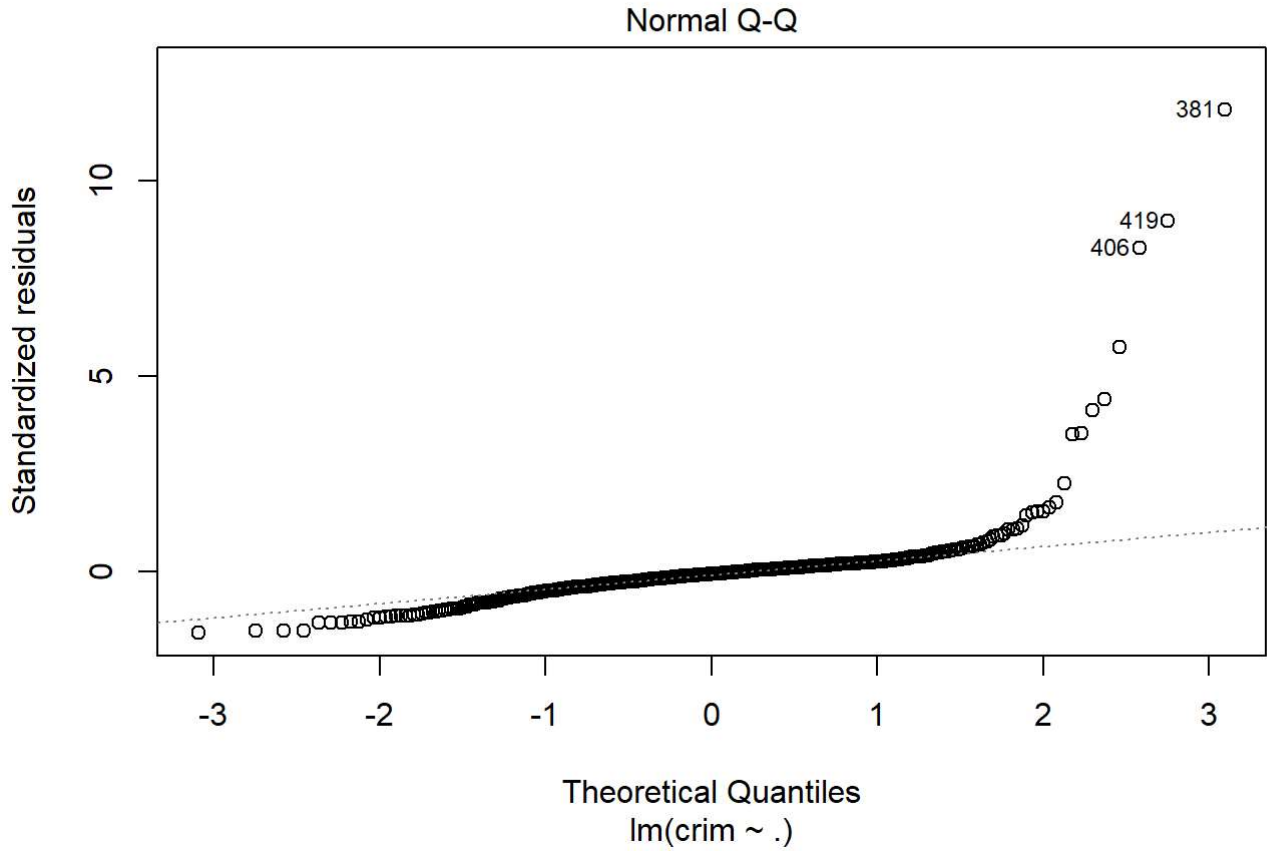
* all predictors except chas are statistically significant * The coefficients show a positive linear relationship between crim and ptratio and a negative relationship between crim and dis

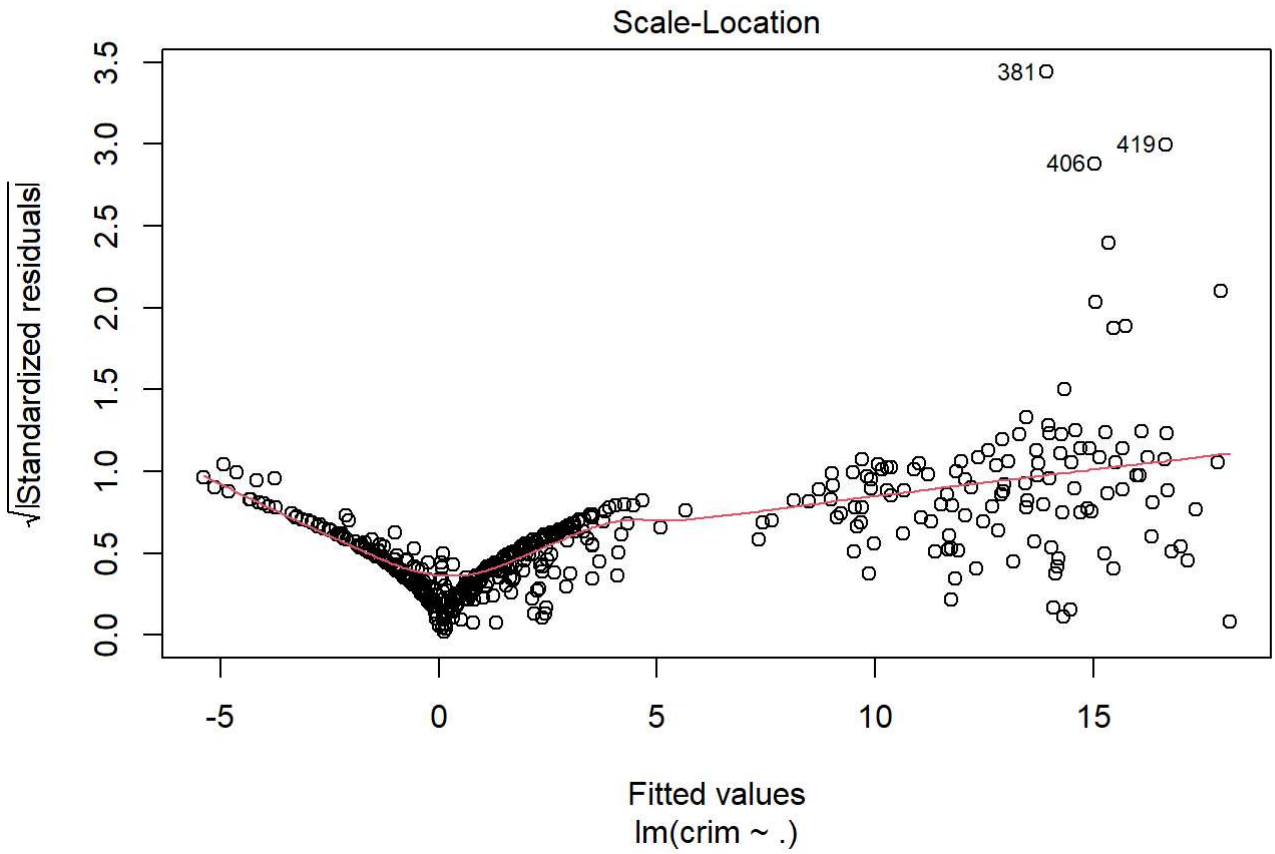
Part b

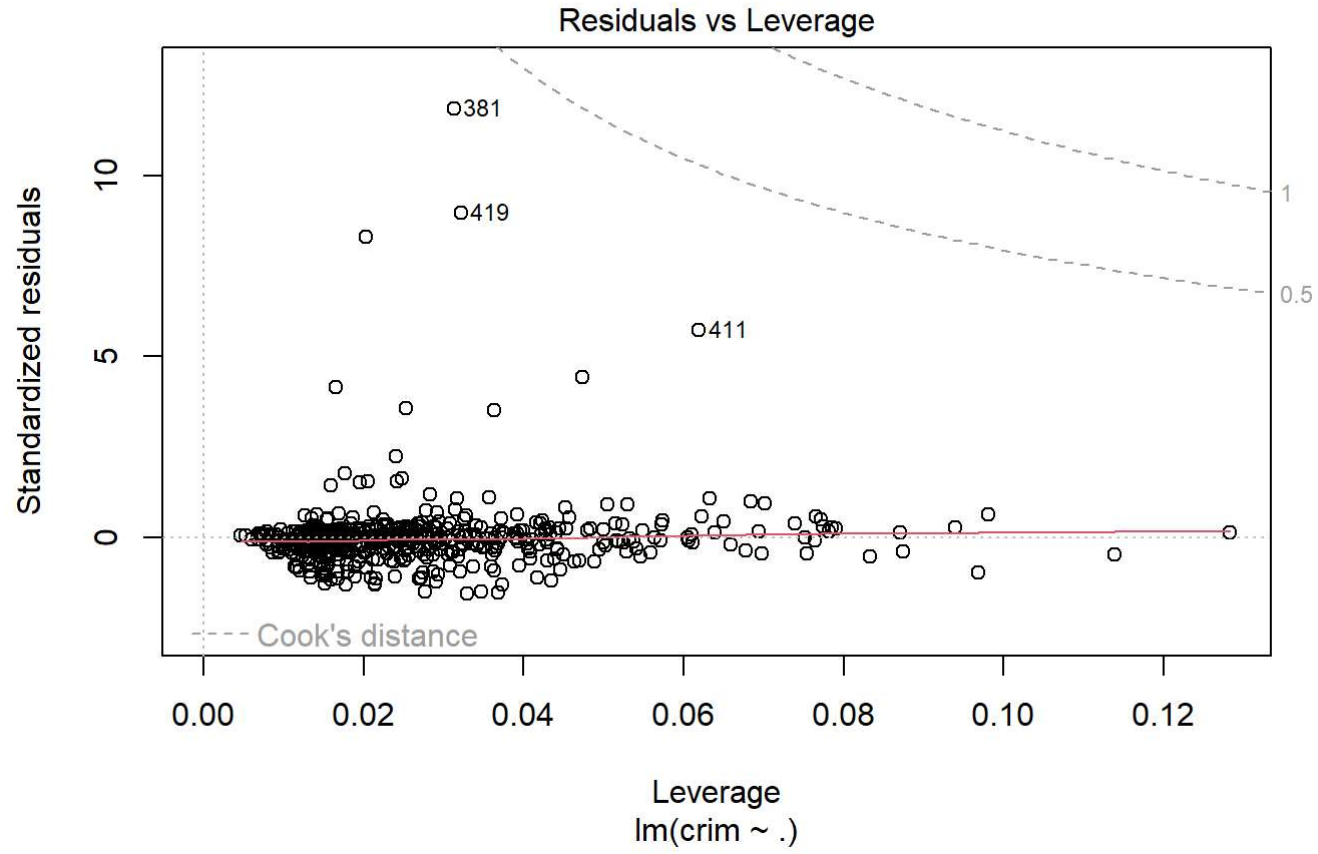
Fit a multiple regression model to predict the response using all of the predictors. Describe your results. For which predictors can we reject the null hypothesis

```
##
## 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
```


Residuals vs Fitted
Residuals
Fitted values
lm(crim ~ .)
381
419
406
-20
0
20
40
60
80
-5
0
5
10
15


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(crim ~ .)
381
419
406
0
5
10
-3
-2
-1
0
1
2
3

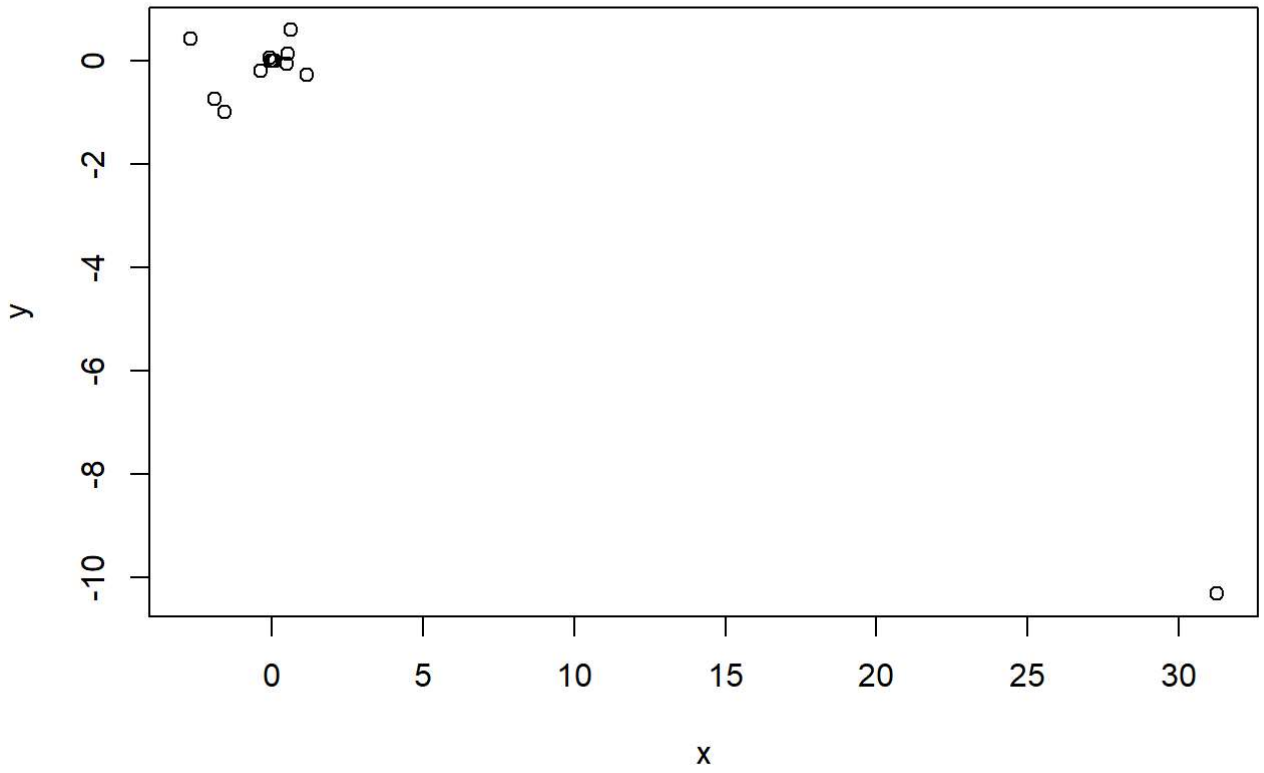

Scale-Location
√|Standardized residuals|
381
419
406
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
-5
0
5
10
15
Fitted values
lm(crim ~ .)


Residuals vs Leverage
Standardized residuals
381
419
411
1
0.5
Cook's distance
0
5
10
0.00
0.02
0.04
0.06
0.08
0.10
0.12
Leverage
lm(crim ~ .)

- zn, dis, rad, black and medv have Pr(>|t|) values less than 0.05, hence we can reject null hypothesis for these predictors

Part c

how do your results from a compare to b? One is uni variate regression and the other is Multivariate regression.



- the coefficient estimate of nox is -10 in multivariate regression and 31 in uni variate regression. If we remove nox from set of variables in multivariate analysis, the effect of other variables would be the same for their respective univariate regressions.

Part d

Is there evidence of non-linear association between any of the predictors and the response?

```
##
## 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
```

Pr(>|t|) value of coeff of cubic variable is greater than 0.05, therefore this is not statistically significant ,hence no non-linear effect

```
##
## 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
```

all Pr(>|t|) value of all coefficients is below 0.05, this indicates adequacy of cubic fit,therefore non-linear relationship is observed

```
##
## 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
```

all Pr(>|t|) value of all coefficients is below 0.05, this indicates adequacy of cubic fit,therefore non-linear relationship is observed

```
##
## 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
```

Pr(>|t|) value of coeff of cubic variable is greater than 0.05, therefore this is not statistically significant,hence no non-linear effect

```
##
## 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
```

all Pr(>|t|) value of all coefficients is below 0.05, this indicates adequacy of cubic fit,therefore non-linear relationship is observed

```
##
## 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
```

all Pr(>|t|) value of all coefficients is below 0.05, this indicates adequacy of cubic fit, therefore non-linear relationship is observed

```
##
## 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
```

Pr(>|t|) value of coeff of cubic variable is greater than 0.05, therefore this is not statistically significant,hence no non-linear effect

```
##
## 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
```

Pr(>|t|) value of coeff of cubic variable is greater than 0.05, therefore this is not statistically significant ,hence no non-linear effect

```
##
## 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
```

all Pr(>|t|) value of all coefficients is below 0.05, this indicates adequacy of cubic fit, therefore non-linear relationship is observed

```
##
## 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
```

2 Pr(>|t|) values of quadratic and cubic coefficients is above 0.05, therefore no linear relationship is observed

```
##
## 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
```

Pr(>|t|) value of coeff of cubic variable is greater than 0.05, therefore this is not statistically significant ,hence no non-linear effect

```
##
## 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
```

all Pr(>|t|) value of all coefficients is below 0.05, this indicates adequacy of cubic fit, therefore non-linear relationship is observed

Chapter 6 #9

Part a

Split the data into train and test

Part b

Linear model using least squares (that is ridge regression with lambda=0) and get the test MSE

```
##           Length Class     Mode
## a0         100   -none-    numeric
## beta      1700   dgCMatrix S4
## df         100   -none-    numeric
## dim          2   -none-    numeric
## lambda     100   -none-    numeric
## dev.ratio  100   -none-    numeric
## nulldev      1   -none-    numeric
## npasses      1   -none-    numeric
## jerr         1   -none-    numeric
## offset       1   -none-    logical
## call         5   -none-    call
## nobs         1   -none-    numeric
```

```
## [1] 1734097
```

The MSE obtained is 1734097

Part c

ridge regression model with lambda from cross validation

```
## [1] 376.9825
```

```
##           Length Class     Mode
## a0         100   -none-    numeric
## beta      1700   dgCMatrix S4
## df         100   -none-    numeric
## dim          2   -none-    numeric
## lambda     100   -none-    numeric
## dev.ratio  100   -none-    numeric
## nulldev      1   -none-    numeric
## npasses      1   -none-    numeric
## jerr         1   -none-    numeric
## offset       1   -none-    logical
## call         5   -none-    call
## nobs         1   -none-    numeric
```

```
## [1] 1711214
```

Best Lambda is 376.98 The MSE obtained is 1711214

Part d

Lasso regression

```
## [1] 0.01
```

```
## [1] 1734040
```

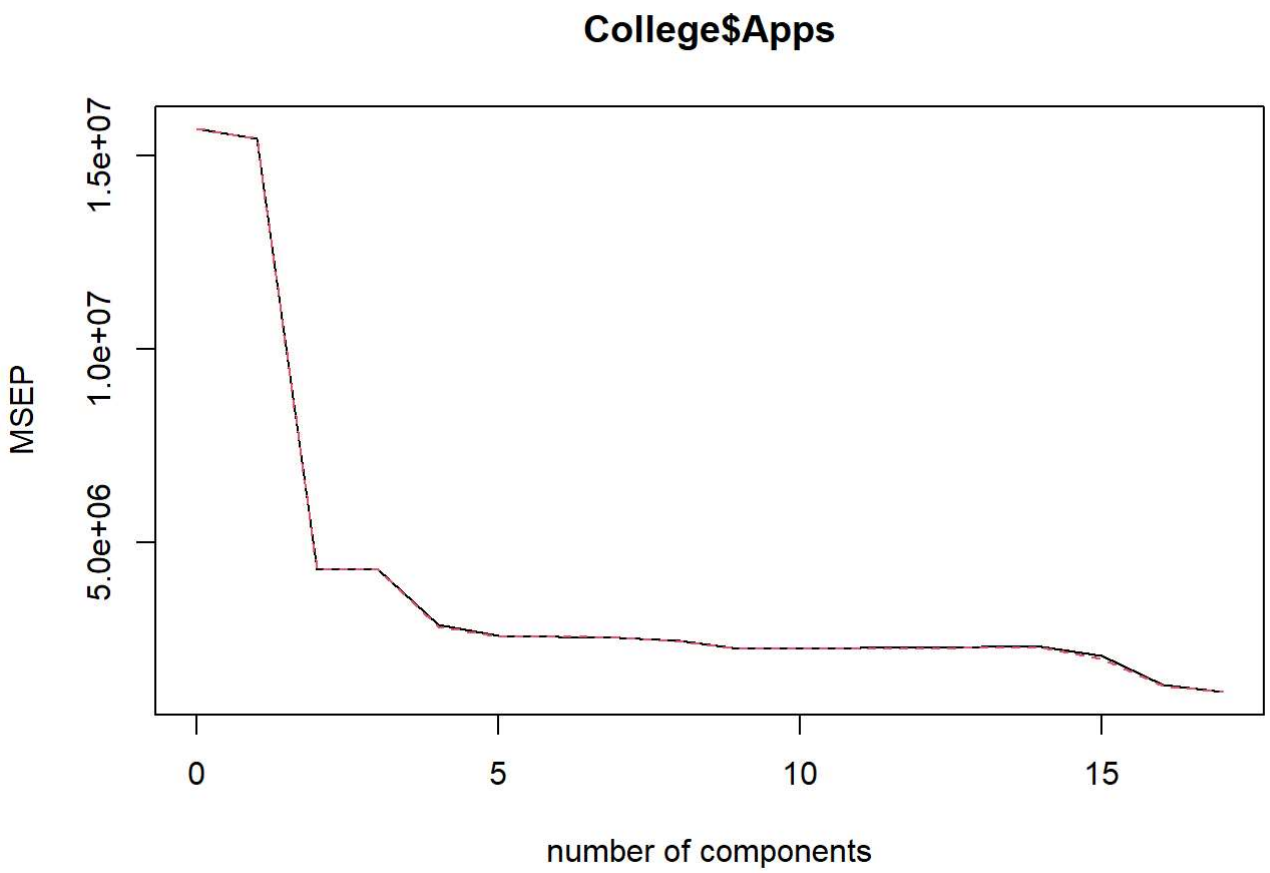
```
##           Length Class     Mode
## a0         100   -none-    numeric
## beta      1700   dgCMatrix S4
## df         100   -none-    numeric
## dim          2   -none-    numeric
## lambda     100   -none-    numeric
## dev.ratio  100   -none-    numeric
## nulldev      1   -none-    numeric
## npasses      1   -none-    numeric
## jerr         1   -none-    numeric
## offset       1   -none-    logical
## call         5   -none-    call
## nobs         1   -none-    numeric
```

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

The Best lambda is 0.01 The MSE obtained is 1734040 all 18 coefficients of the lasso regression are non-zero

Part e

PCR model on the training set, with M chosen by cross validation



```
## Data:    X dimension: 647 17
##  Y dimension: 647 1
## Fit method: svdpc
## Number of components considered: 17
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            3960     3928     2076     2075     1694     1609     1604
## adjCV         3960     3927     2073     2075     1683     1599     1600
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        1593     1564     1504      1500      1507      1507      1514
## adjCV     1593     1558     1501      1497      1505      1504      1511
##        14 comps  15 comps  16 comps  17 comps
## CV         1515      1439      1146      1070
## adjCV      1512      1416      1138      1064
##
## TRAINING: % variance explained
##               1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X              31.595    56.89    64.10    69.95    75.24    80.18    83.97
## College$Apps    1.938    73.21    73.33    82.60    84.29    84.29    84.59
##               8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## X               87.47    90.67     92.98     95.02     96.80     97.90
## College$Apps    85.38    86.12     86.31     86.31     86.38     86.38
##               14 comps  15 comps  16 comps  17 comps
## X                98.72     99.34     99.83    100.00
## College$Apps     86.39     91.73     93.30     93.94
```

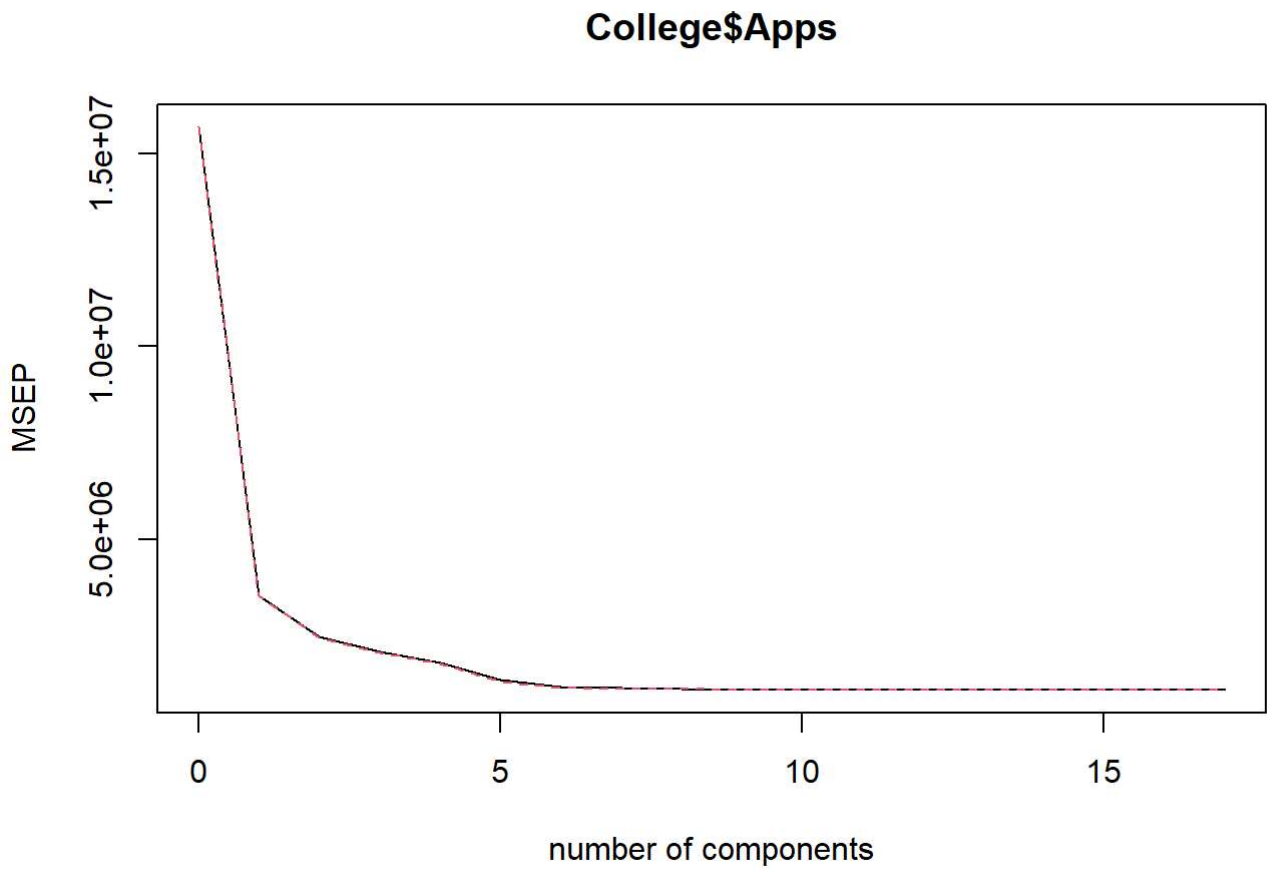
```
## [1] 1734462
```

best MSEP observed for components greater than 15 selected value of M is 17 for which MSE is lowest

MSE obtained from PCR is 1734462

Part f

PLS Regression

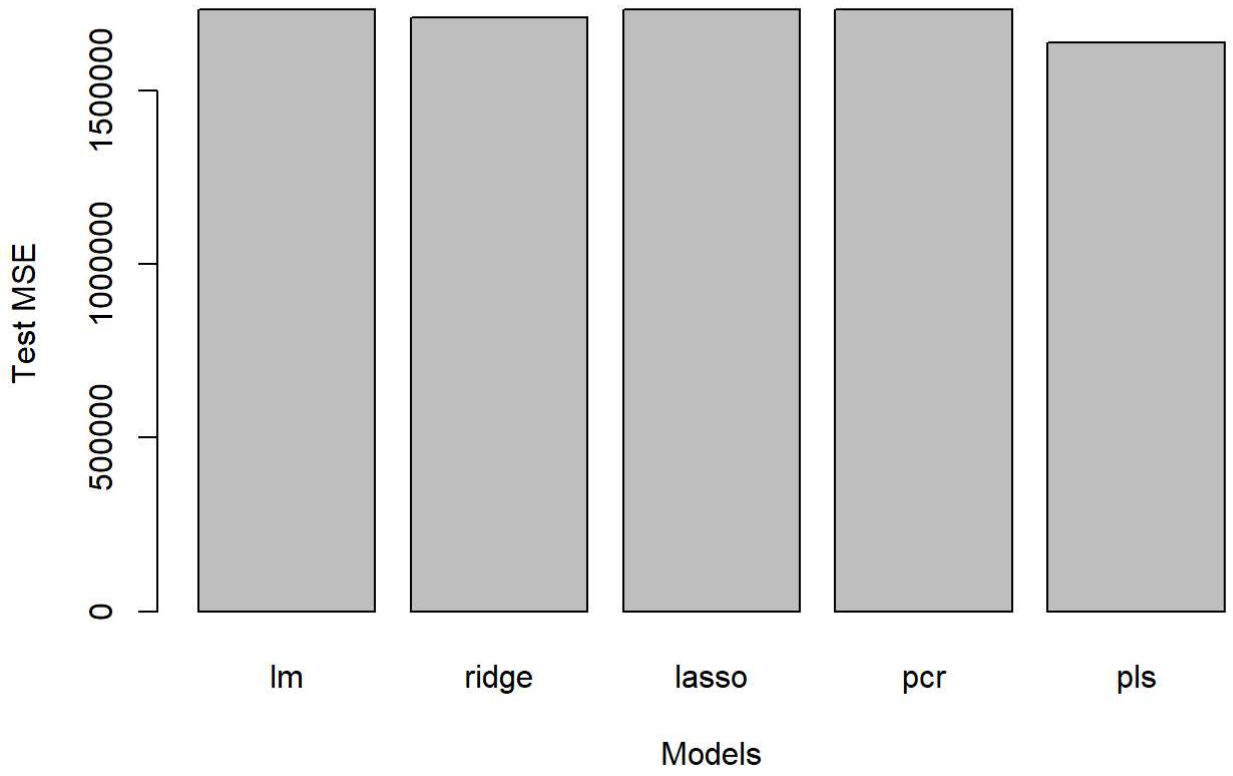


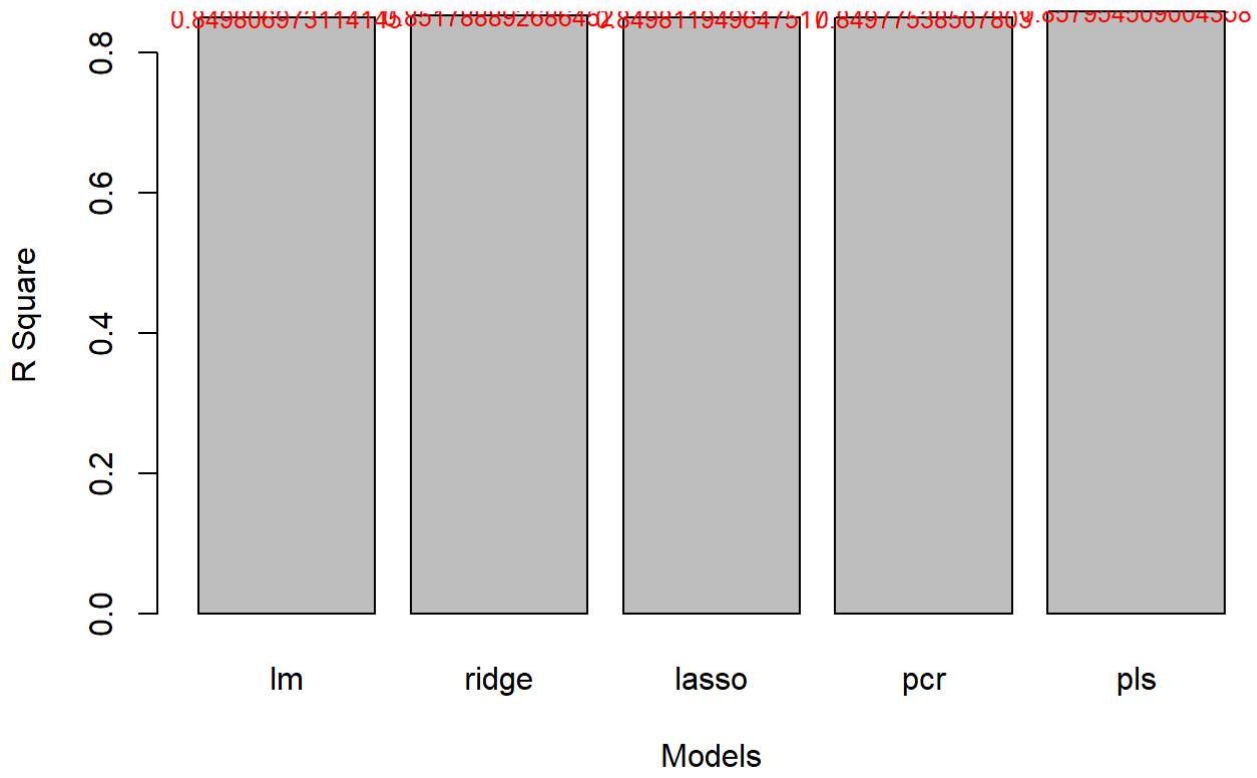

```
## [1] 1640027
```

lowest MSE observed for M=8 The MSE obtained is 1640027

Part g

Comment on the results obtained. How accurately can we predict the number of college applications received? Is there much difference among the test errors resulting from these five approaches?


Test MSE
0
500000
1000000
1500000
lm
ridge
lasso
pcr
pls
Models


R Square
0.0
0.2
0.4
0.6
0.8
lm
ridge
lasso
pcr
pls
Models

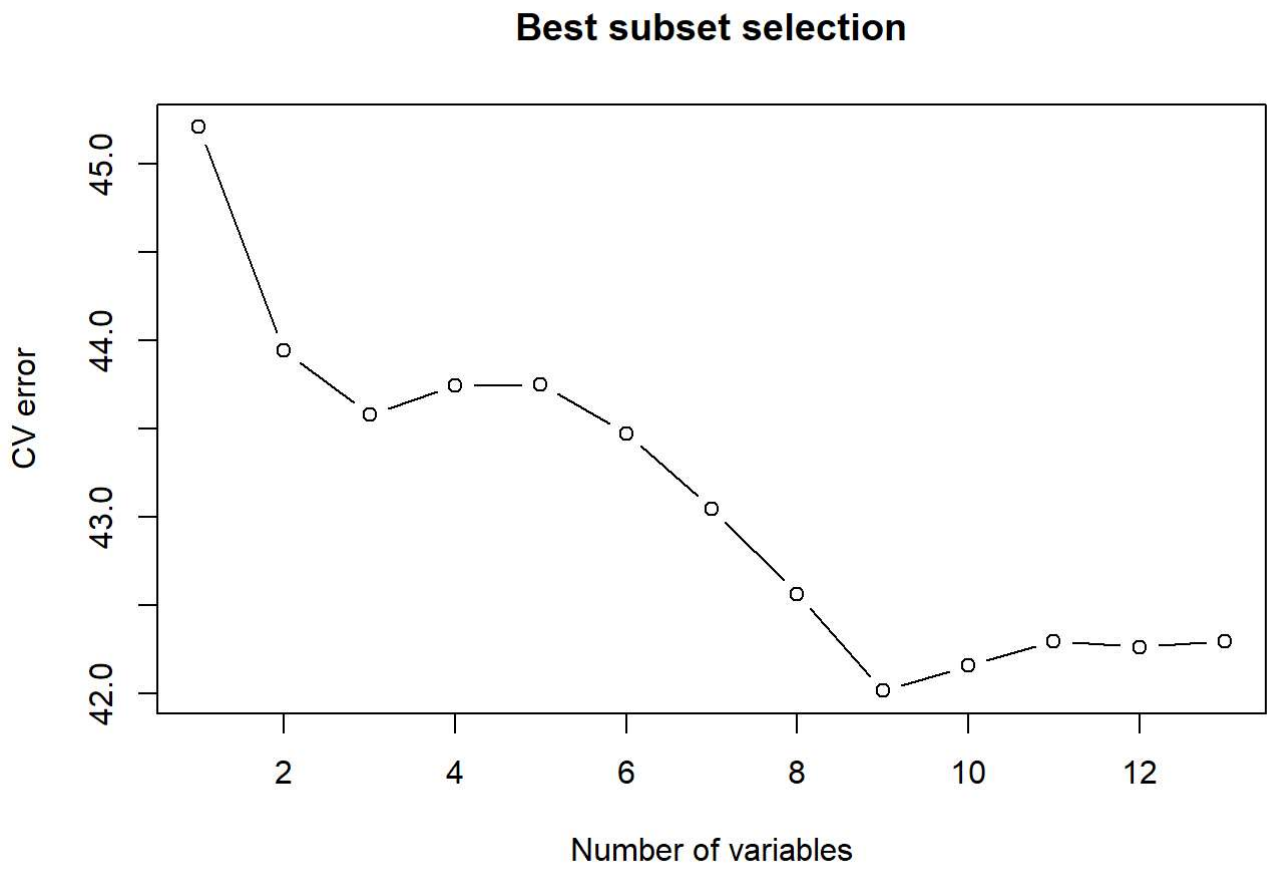
Observations

- pls found to have lowest MSE relatively.
- almost all models have r2 slightly below or above 0.85. Ridge and PLS have a R2 above 0.85 accurately and others have a R2 very slightly below 0.85
- SO we can be reasonably confident about the accuracy of the predictions of Ridge and PLS.

Chapter 6 Problem 11

Part a

Best subset selection using cross validation with 10 folds.



Observation:

- CV error is lowest for model with 9 variables. CV Error = 42.01511

Lasso Regression

```
## [1] 34.69733
```

```
##           Length Class     Mode
## a0         100   -none-    numeric
## beta      1300   dgCMatrix S4
## df         100   -none-    numeric
## dim          2   -none-    numeric
## lambda     100   -none-    numeric
## dev.ratio  100   -none-    numeric
## nulldev      1   -none-    numeric
## npasses      1   -none-    numeric
## jerr         1   -none-    numeric
## offset       1   -none-    logical
## call         5   -none-    call
## nobs         1   -none-    numeric
```

```
##   (Intercept)            zn         indus          chas           nox
## 10.7608042269  0.0343973842 -0.0860050389 -0.1885423574 -7.8970993472
##            rm           age           dis           rad           tax
##  0.5791113777 -0.0004465565 -0.8015605001  0.5014943709  0.0000000000
##       ptratio         black         lstat
## -0.1925606486 -0.0059761389  0.1572149314
```

Lasso regression gives MSE of 34.697

Ridge Regression

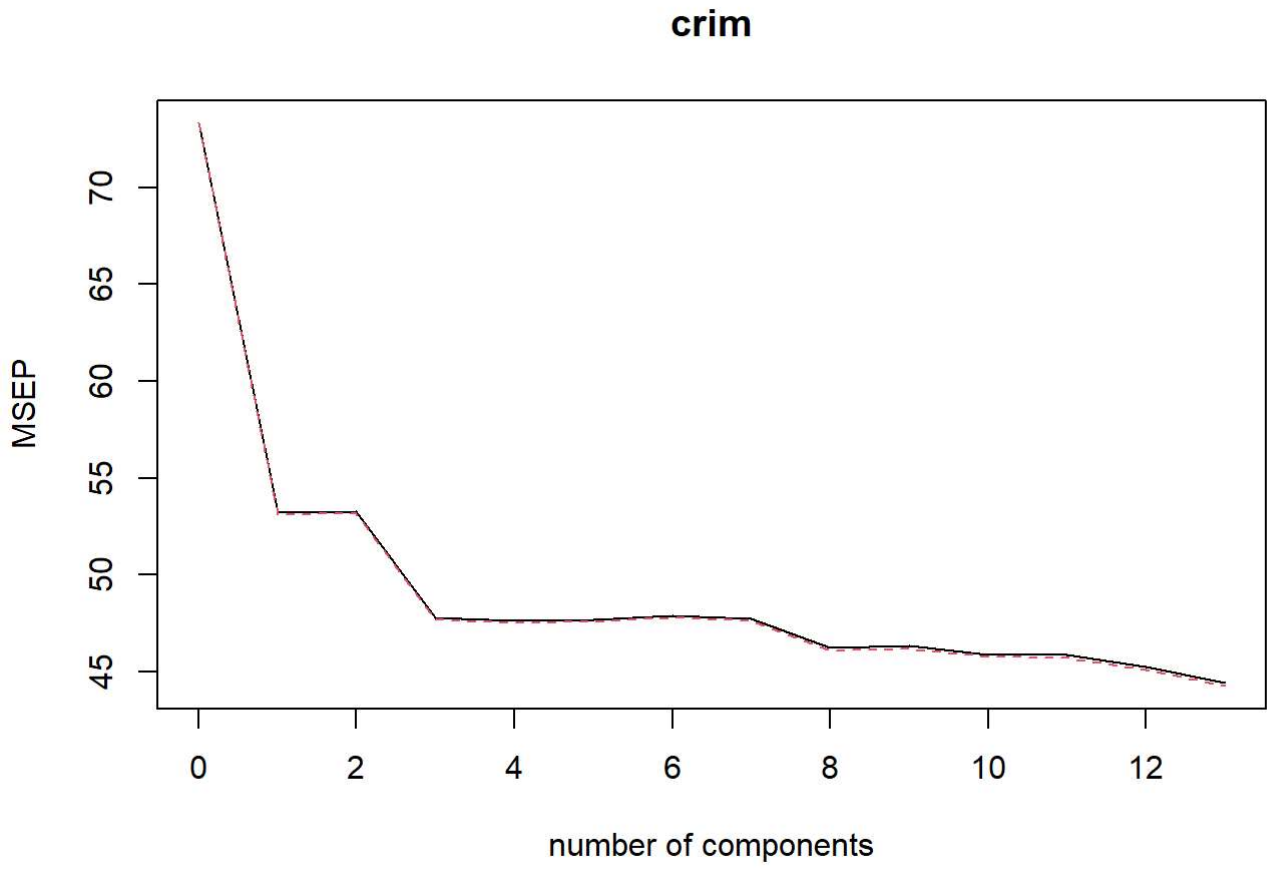
```
## [1] 35.27351
```

```
##           Length Class     Mode
## a0         100   -none-    numeric
## beta      1300   dgCMatrix S4
## df         100   -none-    numeric
## dim          2   -none-    numeric
## lambda     100   -none-    numeric
## dev.ratio  100   -none-    numeric
## nulldev      1   -none-    numeric
## npasses      1   -none-    numeric
## jerr         1   -none-    numeric
## offset       1   -none-    logical
## call         6   -none-    call
## nobs         1   -none-    numeric
```

```
##  (Intercept)           zn        indus         chas          nox           rm
##  7.003424412  0.030583027 -0.092446190 -0.383738264 -5.572135241  0.633468800
##          age          dis          rad          tax      ptratio        black
## -0.002993471 -0.713604592  0.409216418  0.003359205 -0.124666152 -0.007001757
##        lstat
##  0.171188357
```

We get MSE to be around 35.27

PCR




```
## Data:    X dimension: 421 13
##  Y dimension: 421 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           8.563    7.294    7.297    6.914    6.900    6.905    6.920
## adjCV        8.563    7.289    7.292    6.907    6.893    6.898    6.912
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       6.909    6.798    6.807     6.774     6.772     6.727     6.664
## adjCV    6.901    6.787    6.797     6.765     6.762     6.716     6.652
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X       48.08    60.42    69.54    76.67    83.15    88.07    91.20    93.50
## crim    28.98    29.08    36.72    37.10    37.14    37.30    37.65    39.73
##       9 comps  10 comps  11 comps  12 comps  13 comps
## X       95.50      97.1     98.50     99.52    100.00
## crim    39.77      40.4     40.83     41.92     43.04
```

```
## [1] 34.75129
```

MSE is around 34.75 ### Observations: * I would choose the Lasso model because it gives me the least MSE compared to other models * Lasso models are generally more interpret able. * It results in a sparse model with 10 variables. Two variables whose effect on the response were below the required threshold were removed.

Chapter 8 #8

A simulated data set containing sales of child car seats at 400 different stores.

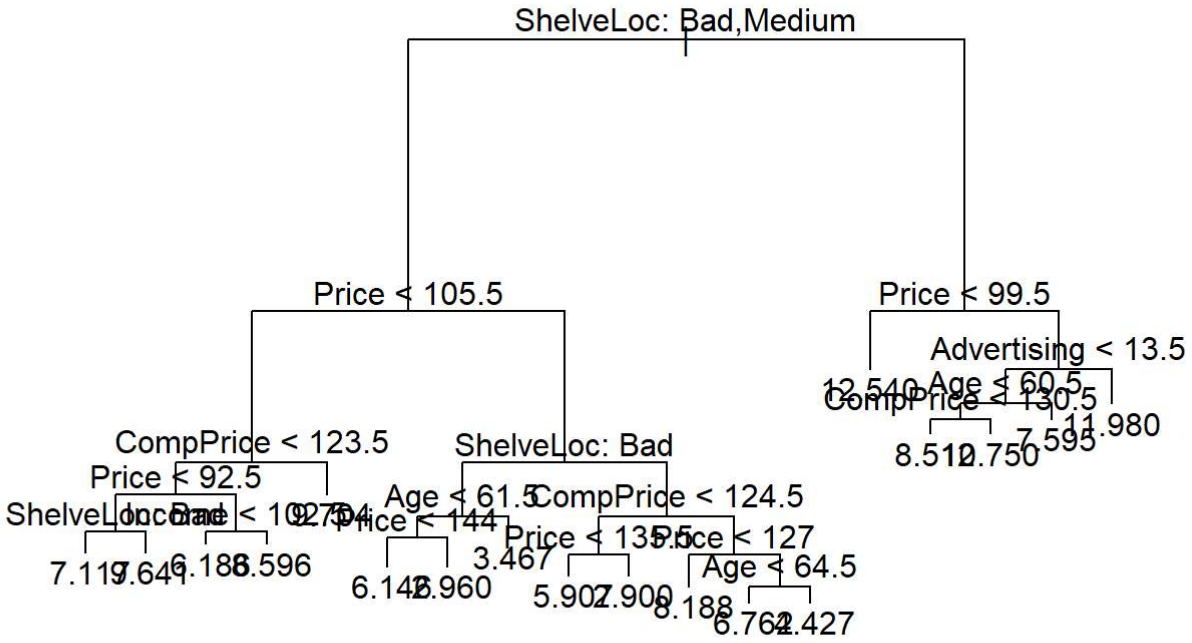
Part a

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

Part b

Fit a regression tree to the training set. Plot the tree, and interpret the results. What test error rate do you obtain

```
##
## Regression tree:
## tree(formula = Sales ~ ., data = Carseats.train)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "CompPrice"   "Income"      "Age"
## [6] "Advertising"
## Number of terminal nodes:  18
## Residual mean deviance:  2.45 = 690.9 / 282
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -4.42700 -1.03300 -0.06031  0.00000  0.89350  4.28300
```



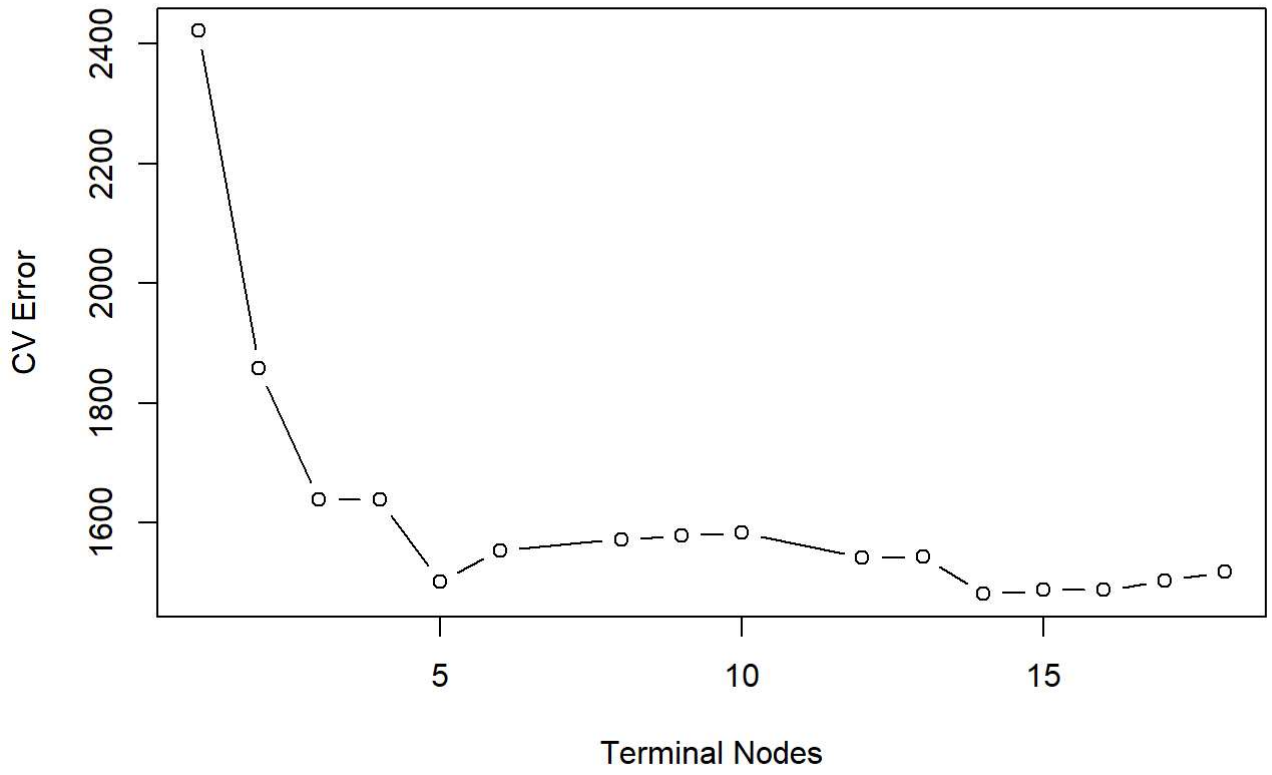

```
## [1] 4.223801
```

above is how the tree looks like. ShevLoc being first comparison node and Price being the second node, they are the most important predictors

The test MSE is about $4.223801

Part c

Use cross-validation in order to determine the optimal level of tree complexity. Does pruning the tree improve the test error? We will find out for what number of terminal nodes we get least error



CV error is the lowest for 14 terminal nodes, tree can be pruned to 14 terminal nodes

```
## [1] 4.067209
```

Test MSE is reduced after pruning the tree

Part d

Bagging using Random Forests mtry=10

```
## randomForest 4.7-1.1
```

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

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
##                %IncMSE IncNodePurity
## CompPrice   34.9816412     287.17190
## Income       5.9985965     112.75680
## Advertising 22.5019577     179.28467
## Population  -0.4400871      80.72751
## Price       73.4424794     733.14939
## ShelveLoc   71.0090207     622.78234
## Age         20.3948844     206.28776
## Education    1.8908535      58.85880
## Urban        0.8333364      10.20234
## US           4.9488667      11.32860
```

ShelveLoc and Price are the 2 main important predictors, similar to the tree we plotted earlier, followed by Compprice and Age

```
## [1] 2.156419
```

We get an MSE of 2.156419, lower than pruning the tree.

Part e

RF for m=10/2, sqrt(10), 20

```
##                %IncMSE IncNodePurity
## CompPrice   23.118120     251.59649
## Income       5.427294     144.54516
## Advertising 18.452762     186.50087
## Population  -1.434953     108.21869
## Price       54.656991     670.58250
## ShelveLoc   58.571077     556.50115
## Age         17.450332     232.22395
## Education    1.529511      80.83309
## Urban        1.037256      14.54158
## US           2.669245      22.07004
```

RF shows that Price and ShelveLoc are most imp predictors, but here Price is more important than ShelveLoc RF for m=10/2, sqrt(10), 20 yields the following MSE

```
## [1] 2.150195
```

```
## [1] 2.539206
```

```
## [1] 2.134395
```

For mtry/2 we are getting an MSE of 2.15019, slightly lower MSE than that of original mtry.

Chapter 8 #11

Caravan dataset

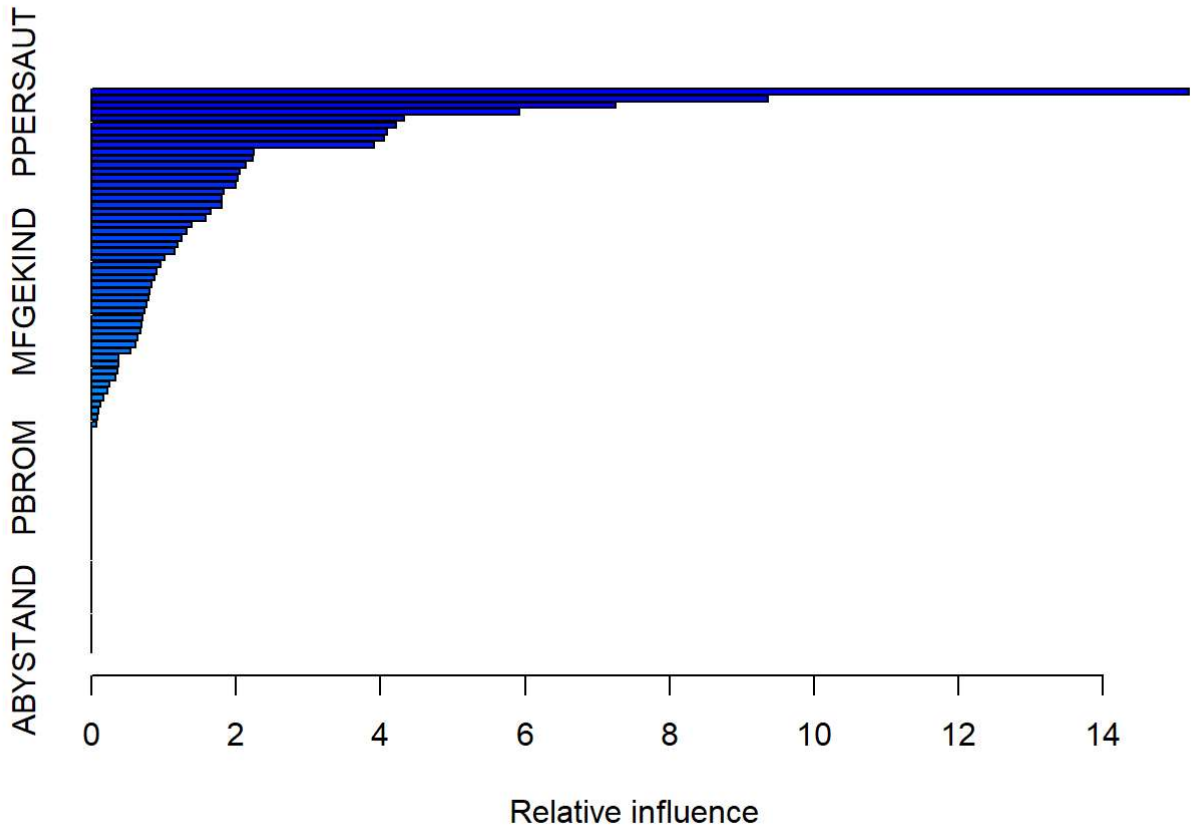
Part a

first create a training set of 1000 and test set of remaining 4822

Part B

perform boosting with n.trees=1000 and shrinkage=0.01, Get important predictors

```
## Loaded gbm 2.1.8
```



```
##               var     rel.inf
## PPERSAUT PPERSAUT 15.18578551
## MKOOPKLA MKOOPKLA  9.35767827
## MOPLHOOG MOPLHOOG  7.25905854
## MBERMIDD MBERMIDD  5.92619837
## ABRAND     ABRAND  4.32711269
## MINK3045 MINK3045  4.22291856
## MOSTYPE   MOSTYPE  4.09655115
## PBRAND     PBRAND  4.05019047
## MGODGE     MGODGE  3.90724958
## MGODOV     MGODOV  2.25413216
## MAUT1       MAUT1  2.23401257
## MSKC         MSKC  2.13918428
## PWAPART   PWAPART  2.04730058
## MBERARBG MBERARBG  2.02849506
## MSKA         MSKA  1.99371484
## MBERHOOG MBERHOOG  1.83880280
## MGODPR     MGODPR  1.80822724
## MAUT2       MAUT2  1.80333735
## MINKGEM   MINKGEM  1.64695612
## PBYSTAND PBYSTAND  1.57718278
## MFWEKIND MFWEKIND  1.38339400
## MINKM30   MINKM30  1.31225379
## MSKB1       MSKB1  1.24533400
## MOPLMIDD MOPLMIDD  1.20102690
## MINK7512 MINK7512  1.15606542
## MINK4575 MINK4575  1.01726205
## MGODRK     MGODRK  0.96042862
## MRELGE     MRELGE  0.89608376
## MAUT0       MAUT0  0.87880042
## MRELOV     MRELOV  0.83717981
## MFGEKIND MFGEKIND  0.80376217
## MOSHOOFD MOSHOOFD  0.79820189
## MBERBOER MBERBOER  0.76043923
## APERSAUT APERSAUT  0.73632389
## MGEMLEEF MGEMLEEF  0.71246127
## MGEMOMV   MGEMOMV  0.69655499
## MZFONDS   MZFONDS  0.68699883
## MHHUUR     MHHUUR  0.63993881
## MHKOOP     MHKOOP  0.60626344
## MINK123M MINK123M  0.53681606
## MSKD         MSKD  0.37731499
## PLEVEN     PLEVEN  0.37206671
## MSKB2       MSKB2  0.35701821
## MZPART     MZPART  0.32856499
## PMOTSCO   PMOTSCO  0.24654480
## MBERARBO MBERARBO  0.21998300
## MRELSA     MRELSA  0.16421095
## MOPLLAAG MOPLLAAG  0.12365031
## MFALLEEN MFALLEEN  0.09540910
## MBERZELF MBERZELF  0.08141022
## MAANTHUI MAANTHUI  0.06414848
```

```
## 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
```

PPERSAUT is the most important predictor in the dataset, followed by MKOOPKLA and MOPLHOOG

Part C

Predict that a person will make a purchase if the estimated probability of purchase is greater than 20 %, get confusion matrix and compare it with logistic regresssion

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

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

21.15% of people predicted to make purchase actually end up making one.

We will compare this with logistic regression using glm in R

```
##    lm.pred
##        0    1
##   0 4183  350
##   1  231   58
```

```
## [1] 0.1421569
```

14.21% people who were predicted to make purchase using logistic regression actually end up making one, this percentage is lower than that of boosting

Chapter 10#7

In the chapter, we mentioned the use of correlation-based distance and Euclidean distance as dissimilarity measures for hierarchical clustering. It turns out that these two measures are almost equivalent: if each observation has been centered to have mean zero and standard deviation one, and if we let rij denote the correlation between the ith and jth observations, then the quantity 1 − rij is proportional to the squared Euclidean distance between the ith and jth observations.

USArrests dataset

scale function-normalizing of a dataset using the mean value and standard deviation is known as scaling. dist-Euclidean distance cor- correlation

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000086 0.069135 0.133943 0.234193 0.262589 4.887686
```

As Min. is very close to zero, we have proved that the proportionality holds

Problem 1: Beauty Pays!

```
##     CourseEvals BeautyScore female lower nonenglish tenuretrack
## 1      3.235245   0.2015666      1     0          0           1
## 2      3.226328  -0.8260813      0     0          0           1
## 3      3.647712  -0.6603327      0     0          0           1
## 4      3.372062  -0.7663125      1     0          0           1
## 5      4.292705   1.4214450      1     0          0           1
## 6      4.239140   0.5002196      0     0          0           1
## 7      3.005187  -0.2143501      1     0          0           1
## 8      3.842654  -0.3465390      1     0          0           1
## 9      3.547257   0.0613435      1     0          0           1
## 10     4.448234   0.4525679      0     0          0           0
## 11     3.785277   0.1432643      0     0          1           1
## 12     3.510400  -0.1550228      0     0          0           0
## 13     4.044416   0.1285433      0     0          0           1
## 14     3.398674  -0.3470453      0     0          1           1
## 15     4.247372   0.4619388      1     0          0           1
## 16     3.727991  -0.1503849      0     0          0           1
## 17     2.778554  -1.0707340      1     0          0           0
## 18     3.381657  -0.1426931      0     1          0           0
## 19     3.807480  -0.1563634      1     0          0           1
## 20     3.723180  -0.0589354      1     0          0           0
## 21     3.620108   0.1511368      1     0          0           0
## 22     2.340655  -0.9370760      1     0          1           1
## 23     2.569861  -0.8214405      0     0          0           1
## 24     3.761175  -0.1994712      0     0          0           1
## 25     3.873202   1.6871670      1     0          0           1
## 26     3.285414  -0.4101976      0     0          0           1
## 27     3.265633   0.1233448      0     0          1           1
## 28     3.951155  -0.8059942      0     0          0           1
## 29     3.233127  -0.8210418      0     0          0           1
## 30     4.191457   0.3198185      1     0          0           1
## 31     4.168352  -0.0985568      0     0          0           0
## 32     3.729772  -1.0123060      0     0          0           1
## 33     4.244241   0.4883314      0     1          0           1
## 34     2.831165  -0.1719505      1     1          0           1
## 35     3.559648  -1.1679070      0     0          0           1
## 36     3.572859   0.9525534      1     0          0           1
## 37     3.987697  -1.1404570      0     0          0           1
## 38     3.641837  -0.8161172      0     0          0           1
## 39     3.295457  -0.4837456      0     0          0           1
## 40     4.584049   0.8450468      0     0          0           1
## 41     3.252351  -0.3261277      0     0          0           1
## 42     3.947180   0.1865589      0     0          0           1
## 43     4.501650   1.8816740      1     1          0           0
## 44     3.692983   0.9626006      0     0          0           1
## 45     3.830549   0.1534694      0     1          0           1
## 46     4.439508   1.2610460      0     0          0           1
## 47     4.114474   0.0232144      1     0          0           1
## 48     3.673407  -0.3460746      1     0          0           1
## 49     2.951502  -0.7446180      1     1          0           0
## 50     3.719092  -0.7447439      0     0          0           1
## 51     3.796821   0.6361284      1     0          0           1
```

```
## 52     4.070508   0.5434852      0     0           0           1
## 53     3.921111   0.9825948      1     0           0           1
## 54     3.294075  -1.5388430      1     1           0           1
## 55     4.269250   1.0675310      0     0           0           1
## 56     3.335991  -0.4143639      0     1           0           1
## 57     3.379242   0.7962944      1     1           0           1
## 58     3.289998  -0.6466943      1     1           0           1
## 59     3.810146  -0.4299034      0     0           1           1
## 60     4.121907  -0.5924066      1     0           0           1
## 61     3.888851  -0.9889295      0     0           0           1
## 62     2.820813   0.5546651      0     0           0           1
## 63     3.214753   0.6851749      1     0           0           1
## 64     3.527765   0.9755885      1     0           0           1
## 65     2.987767  -1.1787380      1     0           0           0
## 66     3.411730  -1.1575110      0     0           0           1
## 67     4.675235   1.3273460      0     0           0           1
## 68     4.213340  -1.5112680      0     0           0           1
## 69     3.523058   0.3034731      1     0           0           1
## 70     3.163561  -0.5783117      0     1           0           1
## 71     3.911674  -0.6719360      0     1           0           0
## 72     4.444582   0.6683338      0     0           0           1
## 73     3.573231   0.7924695      0     1           0           1
## 74     3.678528  -0.6548505      0     1           0           1
## 75     3.476903   0.2510546      1     1           0           1
## 76     3.820326  -0.2397178      1     1           0           1
## 77     2.864969  -1.1133460      1     1           0           0
## 78     4.497694  -0.1135935      0     1           0           1
## 79     3.879921  -0.8351366      0     0           0           1
## 80     4.220469   0.9005994      1     0           0           1
## 81     3.614302  -0.6207690      0     1           0           0
## 82     2.902700  -0.8552412      0     1           0           1
## 83     3.399885   1.0659060      1     1           0           0
## 84     3.440172   1.1442530      1     1           0           1
## 85     3.487115   1.6859850      0     1           0           0
## 86     4.347249   1.6848020      0     1           0           1
## 87     3.857517   0.6362435      0     1           0           1
## 88     3.570822  -1.2699750      0     1           0           1
## 89     4.280377   1.6811030      1     1           0           1
## 90     3.472247  -0.6756103      0     1           0           1
## 91     3.323662  -0.0901815      1     1           0           1
## 92     3.640567  -0.1450257      1     0           1           1
## 93     3.300622   1.1430450      0     1           0           1
## 94     3.805578   0.3320507      1     0           1           1
## 95     3.344075   0.2015666      1     0           0           1
## 96     4.136361   0.2015666      1     0           0           1
## 97     3.578567   0.2015666      1     0           0           1
## 98     3.734100  -0.8260813      0     0           0           1
## 99     3.978873  -0.8260813      0     0           0           1
## 100    4.361488  -0.6603327      0     0           0           1
## 101    3.534745  -0.7663125      1     0           0           1
## 102    3.634222  -0.7663125      1     0           0           1
## 103    2.900904  -0.7663125      1     0           0           1
```

```
## 104    3.308927  -0.7663125      1     0           0            1
## 105    2.752542  -0.7663125      1     0           0            1
## 106    3.178273  -0.7663125      1     0           0            1
## 107    3.651181  -0.7663125      1     0           0            1
## 108    3.230186   1.4214450      1     0           0            1
## 109    3.570194   1.4214450      1     0           0            1
## 110    4.959779   1.4214450      1     0           0            1
## 111    3.985090   1.4214450      1     0           0            1
## 112    4.397228   1.4214450      1     0           0            1
## 113    4.546311   0.5002196      0     0           0            1
## 114    4.298962   0.5002196      0     0           0            1
## 115    3.873059   0.5002196      0     0           0            1
## 116    3.898825   0.5002196      0     0           0            1
## 117    3.608001   0.5002196      0     0           0            1
## 118    4.227334   0.5002196      0     0           0            1
## 119    3.382149  -0.2143501      1     0           0            1
## 120    4.430401  -0.2143501      1     0           0            1
## 121    3.347793  -0.2143501      1     0           0            1
## 122    4.285852  -0.2143501      1     0           0            1
## 123    3.301356  -0.3465390      1     0           0            1
## 124    3.446049  -0.3465390      1     0           0            1
## 125    3.229863  -0.3465390      1     0           0            1
## 126    3.335312  -0.3465390      1     1           0            1
## 127    3.109384  -0.3465390      1     0           0            1
## 128    3.154574  -0.3465390      1     0           0            1
## 129    3.179857   0.0613435      1     0           0            1
## 130    2.556547   0.0613435      1     0           0            1
## 131    4.213071   0.0613435      1     0           0            1
## 132    3.813925   0.0613435      1     0           0            1
## 133    3.580280   0.0613435      1     0           0            1
## 134    3.544532   0.0613435      1     0           0            1
## 135    3.848226   0.4525679      0     0           0            0
## 136    3.547114   0.4525679      0     0           0            0
## 137    4.591037   0.4525679      0     0           0            0
## 138    3.188196   0.4525679      0     1           0            0
## 139    4.254096   0.4525679      0     1           0            0
## 140    4.176637   0.4525679      0     0           0            0
## 141    3.783156   0.4525679      0     1           0            0
## 142    3.806184   0.4525679      0     0           0            0
## 143    4.417182   0.4525679      0     1           0            0
## 144    3.246565   0.1432643      0     0           1            1
## 145    2.946635   0.1432643      0     0           1            1
## 146    4.421489  -0.1550228      0     0           0            0
## 147    3.647188  -0.1550228      0     0           0            0
## 148    3.721048  -0.1550228      0     0           0            0
## 149    4.125384  -0.1550228      0     0           0            0
## 150    4.262076   0.1285433      0     0           0            1
## 151    3.640012   0.1285433      0     0           0            1
## 152    4.960730   0.1285433      0     0           0            1
## 153    4.410735   0.1285433      0     0           0            1
## 154    3.383422   0.1285433      0     0           0            1
## 155    3.995402   0.1285433      0     0           0            1
```

```
## 156    3.806578  -0.3470453      0     0          1             1
## 157    2.926010  -0.3470453      0     0          1             1
## 158    3.244946  -0.3470453      0     0          1             1
## 159    3.927866   0.4619388      1     0          0             1
## 160    3.298322   0.4619388      1     0          0             1
## 161    3.213052   0.4619388      1     0          0             1
## 162    3.409057  -0.1503849      0     1          0             1
## 163    3.690855  -0.1503849      0     1          0             1
## 164    3.722308  -0.1503849      0     0          0             1
## 165    4.354871  -0.1503849      0     0          0             1
## 166    3.586000  -0.1503849      0     1          0             1
## 167    4.200080  -1.0707340      1     0          0             0
## 168    3.412579  -1.0707340      1     0          0             0
## 169    3.829444  -1.0707340      1     0          0             0
## 170    3.345034  -1.0707340      1     0          0             0
## 171    3.893311  -0.1426931      0     0          0             0
## 172    4.723378  -0.1426931      0     0          0             0
## 173    3.736814  -0.1426931      0     1          0             0
## 174    3.792077  -0.1426931      0     1          0             0
## 175    3.788314  -0.1426931      0     0          0             0
## 176    3.309631  -0.1426931      0     1          0             0
## 177    3.459618  -0.1426931      0     1          0             0
## 178    3.343926  -0.1563634      1     1          0             1
## 179    4.215102  -0.1563634      1     0          0             1
## 180    4.635707  -0.1563634      1     0          0             1
## 181    2.790811  -0.1563634      1     0          0             1
## 182    4.135203  -0.1563634      1     0          0             1
## 183    3.800293  -0.1563634      1     1          0             1
## 184    3.643434  -0.1563634      1     0          0             1
## 185    3.447930  -0.1563634      1     1          0             1
## 186    3.797817  -0.0589354      1     0          0             0
## 187    3.529184  -0.0589354      1     0          0             0
## 188    3.925275  -0.0589354      1     0          0             0
## 189    4.651210  -0.0589354      1     0          0             0
## 190    3.384576  -0.0589354      1     0          0             0
## 191    3.937139  -0.0589354      1     0          0             0
## 192    4.116656  -0.0589354      1     0          0             0
## 193    3.894128  -0.0589354      1     0          0             0
## 194    3.355303  -0.0589354      1     0          0             0
## 195    3.675017   0.1511368      1     0          0             0
## 196    4.188179   0.1511368      1     0          0             0
## 197    3.916365   0.1511368      1     0          0             0
## 198    3.395640   0.1511368      1     0          0             0
## 199    3.566903   0.1511368      1     0          0             0
## 200    3.167407  -0.8214405      0     1          0             1
## 201    4.329310  -0.8214405      0     0          0             1
## 202    3.316138  -0.8214405      0     1          0             1
## 203    2.761513  -0.8214405      0     1          0             1
## 204    4.377066  -0.1994712      0     0          0             1
## 205    3.199457  -0.1994712      0     0          0             1
## 206    4.046399  -0.1994712      0     0          0             1
## 207    4.170626  -0.1994712      0     0          0             1
```

```
## 208    3.339951  -0.1994712      0     0           0            1
## 209    3.668727  -0.1994712      0     0           0            1
## 210    3.759193   1.6871670      1     0           0            1
## 211    4.334691  -0.4101976      0     0           0            1
## 212    3.794903  -0.4101976      0     1           0            1
## 213    3.231782  -0.4101976      0     0           0            1
## 214    3.535794  -0.4101976      0     1           0            1
## 215    4.124055   0.1233448      0     0           1            1
## 216    3.740584   0.1233448      0     0           1            1
## 217    4.339190   0.1233448      0     0           1            1
## 218    3.891285   0.1233448      0     0           1            1
## 219    3.688362   0.1233448      0     0           1            1
## 220    3.847355   0.1233448      0     0           1            1
## 221    3.750015  -0.8059942      0     0           0            1
## 222    4.589163  -0.8059942      0     0           0            1
## 223    3.338356  -0.8059942      0     0           0            1
## 224    3.742066  -0.8210418      0     0           0            1
## 225    3.408857  -0.8210418      0     0           0            1
## 226    4.720617  -0.8210418      0     0           0            1
## 227    4.156698  -0.0985568      0     0           0            0
## 228    3.962801  -0.0985568      0     0           0            0
## 229    4.696007  -0.0985568      0     0           0            0
## 230    4.176934  -0.0985568      0     0           0            0
## 231    4.036244  -0.0985568      0     0           0            0
## 232    4.229053  -0.0985568      0     0           0            0
## 233    4.482093  -1.0123060      0     0           0            1
## 234    3.848287  -1.0123060      0     0           0            1
## 235    4.247241   0.4883314      0     1           0            1
## 236    4.058249   0.4883314      0     0           0            1
## 237    4.282606   0.4883314      0     0           0            1
## 238    4.788563   0.4883314      0     0           0            1
## 239    3.025418  -0.1719505      1     0           0            1
## 240    2.428782  -0.1719505      1     1           0            1
## 241    3.347392  -0.1719505      1     1           0            1
## 242    3.808244  -0.1719505      1     0           0            1
## 243    3.286227  -0.1719505      1     1           0            1
## 244    2.743567  -0.1719505      1     1           0            1
## 245    3.305207  -0.1719505      1     1           0            1
## 246    3.295048  -0.1719505      1     1           0            1
## 247    3.378420  -0.1719505      1     1           0            1
## 248    3.111543  -0.1719505      1     1           0            1
## 249    2.912535  -0.1719505      1     1           0            1
## 250    2.778702  -0.1719505      1     1           0            1
## 251    3.681537  -1.1679070      0     0           0            1
## 252    3.379590  -1.1679070      0     0           0            1
## 253    4.163090   0.9525534      1     0           0            1
## 254    3.320500   0.9525534      1     0           0            1
## 255    4.379856   0.9525534      1     0           0            1
## 256    3.584711  -1.1404570      0     0           0            1
## 257    3.426319  -1.1404570      0     0           0            1
## 258    3.917635  -1.1404570      0     0           0            1
## 259    4.540712  -1.1404570      0     0           0            1
```

```
## 260    3.125360  -1.1404570      0     0           0           1
## 261    3.950637  -1.1404570      0     0           0           1
## 262    3.593758  -1.1404570      0     0           0           1
## 263    3.434629  -0.8161172      0     0           0           1
## 264    3.441312  -0.8161172      0     0           0           1
## 265    3.845103  -0.4837456      0     0           0           1
## 266    4.221293  -0.4837456      0     0           0           1
## 267    3.752941  -0.4837456      0     0           0           1
## 268    4.204142  -0.4837456      0     0           0           1
## 269    4.124062  -0.4837456      0     0           0           1
## 270    3.867031  -0.4837456      0     0           0           1
## 271    3.711877  -0.4837456      0     1           0           1
## 272    4.113999  -0.3261277      0     0           0           1
## 273    4.856975  -0.3261277      0     0           0           1
## 274    3.816781  -0.3261277      0     0           0           1
## 275    4.344058  -0.3261277      0     0           0           1
## 276    3.369957   0.1865589      0     1           0           1
## 277    3.438816   0.1865589      0     0           0           1
## 278    3.628980   0.1865589      0     1           0           1
## 279    4.327729   1.8816740      1     0           0           0
## 280    4.672759   1.8816740      1     0           0           0
## 281    3.916479   1.8816740      1     1           0           0
## 282    4.205980   0.9626006      0     0           0           1
## 283    4.485994   0.9626006      0     0           0           1
## 284    4.239315   0.1534694      0     0           0           1
## 285    3.169362   0.1534694      0     1           0           1
## 286    3.631672   0.1534694      0     0           0           1
## 287    4.449169   1.2610460      0     0           0           1
## 288    4.467667   1.2610460      0     0           0           1
## 289    3.220097  -0.3460746      1     0           0           1
## 290    3.560472  -0.3460746      1     0           0           1
## 291    3.920946  -0.7446180      1     1           0           0
## 292    4.009781  -0.7446180      1     1           0           0
## 293    2.798896  -0.7446180      1     0           0           0
## 294    2.844491  -0.7446180      1     0           0           0
## 295    3.148540  -0.7446180      1     0           0           0
## 296    3.328789  -0.7446180      1     0           0           0
## 297    4.322717  -0.7447439      0     0           0           1
## 298    3.622868  -0.7447439      0     0           0           1
## 299    4.037217  -0.7447439      0     0           0           1
## 300    3.709301  -0.7447439      0     0           0           1
## 301    3.652536  -0.7447439      0     0           0           1
## 302    3.638145  -0.7447439      0     0           0           1
## 303    3.896012  -0.7447439      0     0           0           1
## 304    4.101540  -0.7447439      0     0           0           1
## 305    3.962840  -0.7447439      0     0           0           1
## 306    3.242964  -0.7447439      0     0           0           1
## 307    3.744274  -0.7447439      0     0           0           1
## 308    4.146144  -0.7447439      0     0           0           1
## 309    3.612215   0.6361284      1     0           0           1
## 310    3.989669   0.6361284      1     0           0           1
## 311    4.152150   0.6361284      1     0           0           1
```

```
## 312    3.615051   0.6361284      1     0           0            1
## 313    2.734052   0.6361284      1     0           0            1
## 314    4.095480   0.5434852      0     0           0            1
## 315    3.119450   0.5434852      0     0           0            1
## 316    4.584502   0.5434852      0     0           0            1
## 317    4.102741   0.9825948      1     1           0            1
## 318    4.295607   0.9825948      1     0           0            1
## 319    3.657459   0.9825948      1     0           0            1
## 320    3.588537   0.9825948      1     0           0            1
## 321    3.489054   0.9825948      1     1           0            1
## 322    4.030481   0.9825948      1     0           0            1
## 323    2.264917  -1.5388430      1     1           0            1
## 324    2.719367  -1.5388430      1     1           0            1
## 325    3.548390  -1.5388430      1     0           0            1
## 326    2.869927  -1.5388430      1     0           0            1
## 327    2.630010  -1.5388430      1     0           0            1
## 328    4.850540   1.0675310      0     0           0            1
## 329    3.958514   1.0675310      0     1           0            1
## 330    4.212623  -0.4143639      0     1           0            1
## 331    3.425340  -0.4143639      0     1           0            1
## 332    3.490320  -0.4143639      0     0           0            1
## 333    3.870452  -0.4143639      0     0           0            1
## 334    3.695950   0.7962944      1     0           0            1
## 335    3.220623  -0.6466943      1     0           0            1
## 336    2.467698  -0.6466943      1     1           0            1
## 337    3.434277  -0.6466943      1     1           0            1
## 338    1.944243  -0.6466943      1     1           0            1
## 339    3.111029  -0.6466943      1     1           0            1
## 340    3.104837  -0.6466943      1     1           0            1
## 341    4.180427  -0.6466943      1     0           0            1
## 342    2.936440  -0.6466943      1     1           0            1
## 343    3.490122  -0.6466943      1     0           0            1
## 344    3.557594  -0.4299034      0     1           1            1
## 345    3.755875  -0.5924066      1     0           0            1
## 346    3.121156  -0.5924066      1     0           0            1
## 347    3.909109   0.6851749      1     0           0            1
## 348    4.856480   0.9755885      1     0           0            1
## 349    4.033145   0.9755885      1     0           0            1
## 350    3.310308  -1.1787380      1     0           0            0
## 351    3.528436  -1.1787380      1     0           0            0
## 352    3.019437  -1.1787380      1     0           0            0
## 353    2.501746  -1.1787380      1     0           0            0
## 354    3.722521  -1.1787380      1     0           0            0
## 355    2.816316  -1.1787380      1     0           0            0
## 356    2.984322  -1.1575110      0     0           0            1
## 357    3.495188  -1.1575110      0     0           0            1
## 358    4.037871  -1.1575110      0     0           0            1
## 359    2.825915  -1.1575110      0     0           0            1
## 360    3.422935  -1.1575110      0     0           0            1
## 361    4.441669   1.3273460      0     0           0            1
## 362    3.593346  -1.5112680      0     0           0            1
## 363    3.241932  -1.5112680      0     0           0            1
```

```
## 364    3.440158  -0.5783117      0     1           0            1
## 365    3.060216  -0.5783117      0     1           0            1
## 366    3.056650  -0.5783117      0     1           0            1
## 367    4.064338  -0.5783117      0     1           0            1
## 368    2.948672  -0.5783117      0     1           0            1
## 369    3.421832  -0.5783117      0     1           0            1
## 370    3.280265  -0.5783117      0     1           0            1
## 371    3.829392  -0.5783117      0     1           0            1
## 372    3.387053  -0.6719360      0     1           0            0
## 373    4.265154  -0.6719360      0     1           0            0
## 374    3.149728  -0.6719360      0     1           0            0
## 375    2.810285  -0.6719360      0     1           0            0
## 376    3.682914  -0.6719360      0     1           0            0
## 377    3.139237  -0.6719360      0     1           0            0
## 378    3.910100  -0.6719360      0     1           0            0
## 379    4.009281  -0.6719360      0     1           0            0
## 380    3.979895  -0.6719360      0     1           0            0
## 381    3.288346   0.6683338      0     0           0            1
## 382    4.274797   0.6683338      0     0           0            1
## 383    5.000000   0.6683338      0     0           0            1
## 384    4.276027   0.6683338      0     0           0            1
## 385    5.000000   0.6683338      0     0           0            1
## 386    3.542861   0.7924695      0     1           0            1
## 387    3.258540   0.7924695      0     1           0            1
## 388    2.961228   0.7924695      0     1           0            1
## 389    3.837807   0.7924695      0     1           0            1
## 390    3.922090  -0.6548505      0     1           0            1
## 391    3.921732  -0.6548505      0     1           0            1
## 392    3.710004  -0.6548505      0     1           0            1
## 393    3.024209   0.2510546      1     1           0            1
## 394    4.044561  -0.2397178      1     1           0            1
## 395    3.122263  -1.1133460      1     1           0            0
## 396    2.857190  -1.1133460      1     1           0            0
## 397    3.261738  -1.1133460      1     1           0            0
## 398    3.783343  -1.1133460      1     1           0            0
## 399    2.389286  -1.1133460      1     1           0            0
## 400    4.168160  -0.1135935      0     1           0            1
## 401    3.690853  -0.1135935      0     0           0            1
## 402    3.687427  -0.1135935      0     0           0            1
## 403    4.176725  -0.8351366      0     0           0            1
## 404    3.588007  -0.8351366      0     0           0            1
## 405    4.401393   0.9005994      1     0           0            1
## 406    3.932989   0.9005994      1     0           0            1
## 407    4.749722   0.9005994      1     0           0            1
## 408    3.526500  -0.6207690      0     1           0            0
## 409    3.719947  -0.6207690      0     1           0            0
## 410    3.257813  -0.6207690      0     1           0            0
## 411    3.057960  -0.8552412      0     1           0            1
## 412    3.054810  -0.8552412      0     1           0            1
## 413    3.035142  -0.8552412      0     1           0            1
## 414    2.819714  -0.8552412      0     1           0            1
## 415    3.791035  -0.8552412      0     1           0            1
```

```
## 416    3.287817  -0.8552412      0     1          0           1
## 417    3.346200  -0.8552412      0     1          0           1
## 418    3.856797  -0.8552412      0     0          0           1
## 419    3.480681  -0.8552412      0     1          0           1
## 420    2.889805  -0.8552412      0     1          0           1
## 421    4.244275   1.0659060      1     1          0           0
## 422    3.280905   1.0659060      1     1          0           0
## 423    4.027994   1.0659060      1     1          0           0
## 424    3.733446   1.0659060      1     1          0           0
## 425    3.734694   1.1442530      1     1          0           1
## 426    3.678469   1.1442530      1     1          0           1
## 427    4.026700   1.1442530      1     1          0           1
## 428    3.569580   1.1442530      1     1          0           1
## 429    3.392855   1.6859850      0     1          0           0
## 430    4.244513   1.6859850      0     1          0           0
## 431    4.644818   1.6859850      0     1          0           0
## 432    4.525024   1.6859850      0     1          0           0
## 433    4.428229   1.6859850      0     1          0           0
## 434    4.350663   1.6859850      0     1          0           0
## 435    4.162443   1.6859850      0     1          0           0
## 436    3.615677   1.6848020      0     1          0           1
## 437    4.128806   1.6848020      0     0          0           1
## 438    3.621035   0.6362435      0     1          0           1
## 439    3.650574  -1.2699750      0     1          0           1
## 440    3.400730  -1.2699750      0     1          0           1
## 441    3.602580  -1.2699750      0     1          0           1
## 442    3.128079  -1.2699750      0     0          0           1
## 443    2.795498  -1.2699750      0     1          0           1
## 444    3.784506  -1.2699750      0     1          0           1
## 445    3.234750   1.6811030      1     1          0           1
## 446    4.136941   1.6811030      1     1          0           1
## 447    2.746388  -0.6756103      0     1          0           1
## 448    3.852290  -0.0901815      1     1          0           1
## 449    3.491208  -0.0901815      1     1          0           1
## 450    3.315225  -0.1450257      1     0          1           1
## 451    3.990372  -0.1450257      1     0          1           1
## 452    3.369571  -0.1450257      1     0          1           1
## 453    2.891549  -0.1450257      1     0          1           1
## 454    3.403679  -0.1450257      1     0          1           1
## 455    3.595618  -0.1450257      1     0          1           1
## 456    3.096778   1.1430450      0     1          0           1
## 457    4.581000   1.1430450      0     0          0           1
## 458    3.708582   1.1430450      0     1          0           1
## 459    4.438559   1.1430450      0     1          0           1
## 460    4.860266   1.1430450      0     0          0           1
## 461    4.154587   0.3320507      1     0          1           1
## 462    3.111189   0.3320507      1     0          1           1
## 463    3.434022   0.3320507      1     1          1           1
```

Given the columnms into consideration from the Beauty Data, we can comfortably infer that the overall ratings could be interpreted as a linear function of BeautyScore, Female, lower, nonenglish and tenuretrack. In mathematical terms:

Ratings= B0 + B1*BeautyScore* + *B2*Female + B3*lower* + *B4*nonenglish + B5*tenuretrack + e

where B0,B1,…,B5 are coefficients of the linear function and e is the error in our prediction.

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

As we can see, only BeautyScore has a positive coefficient whereas other predictors have a negative coefficient. We can infer from this that more the BeautyScore, more the ratings. On the other hand, female and lower have an equally negative coefficient, that means a positive change in their values results in a decline in their ratings.

Also, BeautyScore, female, lower turn out to be the most statistically significant predictors followed by non-english and lastly, tenurerack.

2. Dr. Hamermesh, by stating this- “Disentangling whether this outcome represents productivity or discrimination is, as with the issue generally, probably impossible”, probably wants to impose the question of whether beauty is an indicator of someone being a better teacher or is having beauty perceived to be related to better teaching. This analysis cannot answer this question. In my honest opinion, this is just discrimination and I truly believe that beauty is not the indicator of one’s ability to teach. I believe teaching is an ability developed and mastered with experience, ability and passion.

Problem 2: Housing Price Structure

MidCity Data

To begin we create dummy variable Neb_2 and Neb_3 to indicate if a house came from neighborhood two and neighborhood three respectively. Using these dummy variables and the other covariates, we ran a regression for the model

We observe that BrickYes has a statistical significance on the dataset, along with the Neb_3 measure .

```
## The following object is masked from Carseats:
##
##     Price
```

```
##  Nbhd       Offers            SqFt.V1        Brick       Bedrooms
##  1:44   Min.   :1.000   Min.   :-2.6040137   No :86   Min.   :2.000
##  2:45   1st Qu.:2.000   1st Qu.:-0.5716128   Yes:42   1st Qu.:3.000
##  3:39   Median :3.000   Median :-0.0044311            Median :3.000
##         Mean   :2.578   Mean   : 0.0000000            Mean   :3.023
##         3rd Qu.:3.000   3rd Qu.: 0.6572808            3rd Qu.:3.000
##         Max.   :6.000   Max.   : 2.7842120            Max.   :5.000
##    Bathrooms         Price
##  Min.   :2.000   Min.   : 69100
##  1st Qu.:2.000   1st Qu.:111325
##  Median :2.000   Median :125950
##  Mean   :2.445   Mean   :130427
##  3rd Qu.:3.000   3rd Qu.:148250
##  Max.   :4.000   Max.   :211200
```

```
##
## Call:
## lm(formula = Price ~ ., data = MidCity)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27337.3  -6549.5    -41.7   5803.4  27359.3
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   108197       6689  16.175  < 2e-16 ***
## Nbhd2          -1561       2397  -0.651  0.51621
## Nbhd3          20681       3149   6.568 1.38e-09 ***
## Offers         -8268       1085  -7.621 6.47e-12 ***
## SqFt           11212       1213   9.242 1.10e-15 ***
## BrickYes       17297       1982   8.729 1.78e-14 ***
## Bedrooms        4247       1598   2.658  0.00894 **
## Bathrooms       7883       2117   3.724  0.00030 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10020 on 120 degrees of freedom
## Multiple R-squared:  0.8686, Adjusted R-squared:  0.861
## F-statistic: 113.3 on 7 and 120 DF,  p-value: < 2.2e-16
```

Yes there is a premium for Brick houses, keeping variables constant, one would give $17297 premium. ## Part B

We will introduce Neb_3 variable first and Neb_2 variable later

```
##
## Call:
## lm(formula = Price ~ MidCity$Nbhd, data = MidCity)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -42931 -12310  -1643  11251  51905
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     110155       2693  40.904  < 2e-16 ***
## MidCity$Nbhd2    15077       3787   3.981 0.000116 ***
## MidCity$Nbhd3    49140       3929  12.508  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17860 on 125 degrees of freedom
## Multiple R-squared:  0.565,  Adjusted R-squared:  0.558
## F-statistic: 81.16 on 2 and 125 DF,  p-value: < 2.2e-16
```

As there is statistical significance for Neighbourhood 3 with a positive coefficient, we can infer that there is premium for houses for neighbourhood 3 for $49140

```
##
## Call:
## lm(formula = Price ~ ., data = MidCity)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26939.1  -5428.7   -213.9   4519.3  26211.4
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      111190       6668  16.675  < 2e-16 ***
## Nbhd2              -673       2376  -0.283  0.77751
## Nbhd3             17241       3391   5.084 1.39e-06 ***
## Offers            -8401       1064  -7.893 1.62e-12 ***
## SqFt              11439       1192   9.593  < 2e-16 ***
## BrickYes          13826       2406   5.748 7.11e-08 ***
## Bedrooms           4718       1578   2.991  0.00338 **
## Bathrooms          6463       2154   3.000  0.00329 **
## Brick_Nbhd3Yes    10182       4165   2.444  0.01598 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9817 on 119 degrees of freedom
## Multiple R-squared:  0.8749, Adjusted R-squared:  0.8665
## F-statistic:   104 on 8 and 119 DF,  p-value: < 2.2e-16
```

there is an extra premium of $10182 for brick houses in neighborhood 3

Part d

Yes we can combine the two neighbourhoods 1,2 into one column by tagging a 1 for a match for the same. But we saw earlier that neighbourhood 2 has a non-zero coefficient, meaning it has an impact on house prices. Hence it is not a good idea to merge the two and perform a prediction, we might get an NA as the coefficient.

Problem 3: What causes what??

Answer 1: You can get data from different cities on Crime and Police, but it would be very difficult to establish a relationship between the two. Also datasets provided to a regression model should be free of a bias. If we perform the act of introducing more cops on the streets to reduce the crime rate and then use the data, we are already coming in with a bias.

Answer 2: the researchers in DC were able to go around the bias of increase in cops leading to reduced crime rate because the number of cops on the streets were increased by the mayor on high alert days. High alert days are determined by potential terrorist attacks and are not dependent on crime rate. They observed that the crime rate dropped. Maybe because People wouldn't venture out in the first place because there was a high alert?

In table 2, we see that controlling ridership in the Metro results in a lower crime as coefficient is negative. It means that keeping ridership constant, more police results in less crime.

Answer 3: The METRO ridership was controlled to ensure that it was not affecting the crime rate. UPENN researchers could hence establish clearer relationship between the crime rate and increase in the police deployed on the streets.

Answer 4: Table 4 further enforces our statement of reduction in crime rate due to increase in police on the streets, as the coefficients are negative.

using interaction between location and high alert, the table is able to differentiate the effect of the experiment for different location. We find that District 1 has an effect of high alert days on crime, unlike other districts.

Problem 4: Final Project

In my final group project, my group tried to predict Walmart Sales using Linear Regression, Random Forests and Boosting. We achieved this in both Python and R. I worked mainly on the Python implementation of the same along with a teammate, Manvi. I helped her in performing EDA of the dataset and removing the outliers from the data. We also broke down the date column into day, month and year and performed the aforementioned 3 methods for prediction using sklearn.

For the Random Forest implementation, as our problem statement involves a regression, I used the RandomForestRegressor() method for the same. I divided the data into a 80-20 train test split. We define the parameters for the Regressor while calling the methods, such as setting the max_depth of the tree and considering out-of-bag score for bootstrapping. I then fitted the model and calculated the metrics-accuracy, MSE and RMSE. I also plotted a sample tree out of the forest for visualization and the feature importances.

For boosting, I used the GradientBoostingRegressor() to fit a boosting model and get the aforementioned metrics. Upon comparison, we found that Random Forest gives a better prediction than boosting.