STA 380: Take Home Exam

Troy Richard

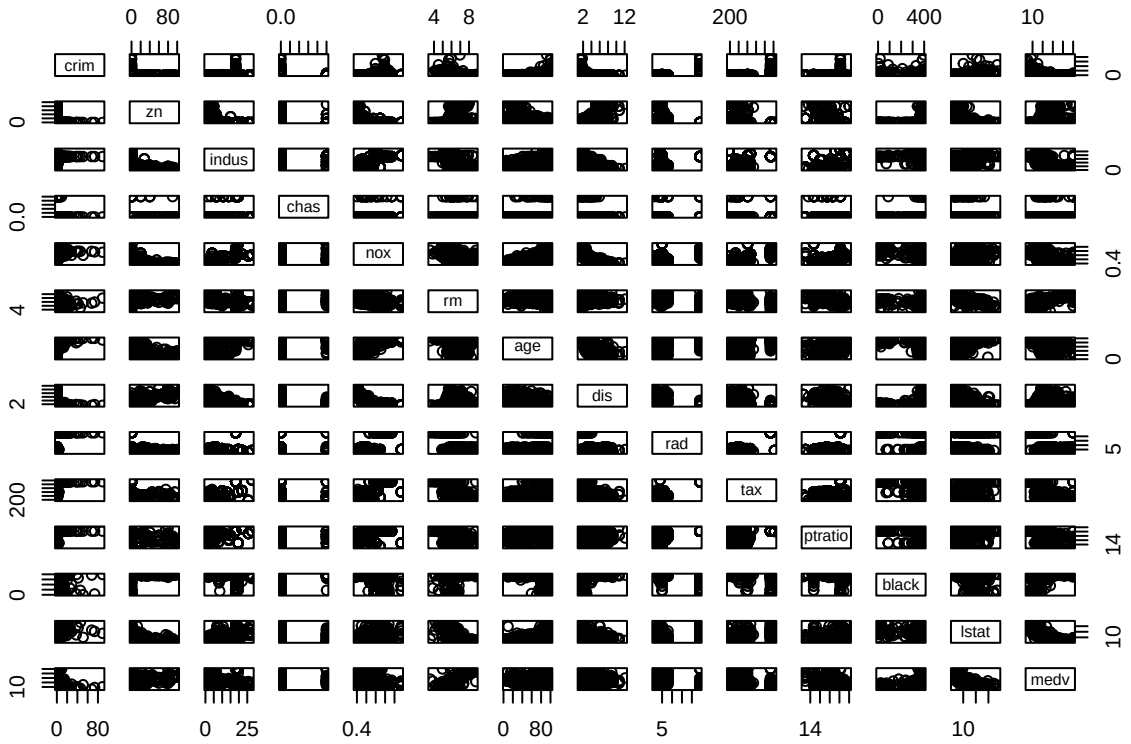
7/31/2021

Chapter 2, Question 10

Part A

506 rows which represent the 506 suburbs taken into account in this study, 14 columns represent the different attributes that can impact housing values.

Part B



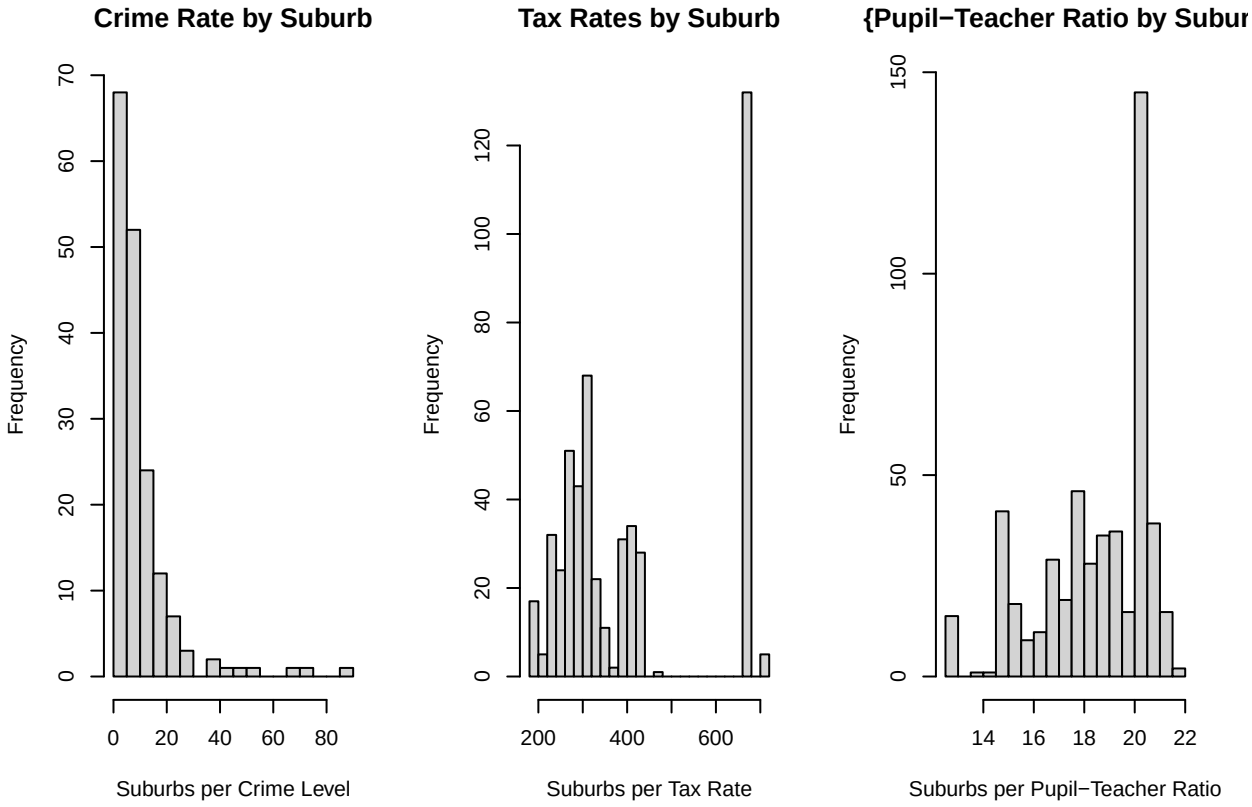
Higher tax for homes closer to rad, negative correlation between dis and nox, higher ptratio indicates higher medv, high crim indicates a high lstat, high rm indicates a higher medv, negative correlation between black and dis, no strong correlations with zn

Part C

```
##                crim          zn       indus         chas         nox
## crim     1.00000000 -0.20046922  0.40658341 -0.055891582  0.42097171
## zn      -0.20046922  1.00000000 -0.53382819 -0.042696719 -0.51660371
## indus    0.40658341 -0.53382819  1.00000000  0.062938027  0.76365145
## chas    -0.05589158 -0.04269672  0.06293803  1.000000000  0.09120281
## nox      0.42097171 -0.51660371  0.76365145  0.091202807  1.00000000
## rm      -0.21924670  0.31199059 -0.39167585  0.091251225 -0.30218819
## age      0.35273425 -0.56953734  0.64477851  0.086517774  0.73147010
## dis     -0.37967009  0.66440822 -0.70802699 -0.099175780 -0.76923011
## rad      0.62550515 -0.31194783  0.59512927 -0.007368241  0.61144056
## tax      0.58276431 -0.31456332  0.72076018 -0.035586518  0.66802320
## ptratio  0.28994558 -0.39167855  0.38324756 -0.121515174  0.18893268
## black   -0.38506394  0.17552032 -0.35697654  0.048788485 -0.38005064
## lstat    0.45562148 -0.41299457  0.60379972 -0.053929298  0.59087892
## medv    -0.38830461  0.36044534 -0.48372516  0.175260177 -0.42732077
##                  rm         age         dis          rad         tax    ptratio
## crim    -0.21924670  0.35273425 -0.37967009  0.625505145  0.58276431  0.2899456
## zn       0.31199059 -0.56953734  0.66440822 -0.311947826 -0.31456332 -0.3916785
## indus   -0.39167585  0.64477851 -0.70802699  0.595129275  0.72076018  0.3832476
## chas     0.09125123  0.08651777 -0.09917578 -0.007368241 -0.03558652 -0.1215152
## nox     -0.30218819  0.73147010 -0.76923011  0.611440563  0.66802320  0.1889327
## rm       1.00000000 -0.24026493  0.20524621 -0.209846668 -0.29204783 -0.3555015
## age     -0.24026493  1.00000000 -0.74788054  0.456022452  0.50645559  0.2615150
## dis      0.20524621 -0.74788054  1.00000000 -0.494587930 -0.53443158 -0.2324705
## rad     -0.20984667  0.45602245 -0.49458793  1.000000000  0.91022819  0.4647412
## tax     -0.29204783  0.50645559 -0.53443158  0.910228189  1.00000000  0.4608530
## ptratio -0.35550149  0.26151501 -0.23247054  0.464741179  0.46085304  1.0000000
## black    0.12806864 -0.27353398  0.29151167 -0.444412816 -0.44180801 -0.1773833
## lstat   -0.61380827  0.60233853 -0.49699583  0.488676335  0.54399341  0.3740443
## medv     0.69535995 -0.37695457  0.24992873 -0.381626231 -0.46853593 -0.5077867
##               black      lstat       medv
## crim    -0.38506394  0.4556215 -0.3883046
## zn       0.17552032 -0.4129946  0.3604453
## indus   -0.35697654  0.6037997 -0.4837252
## chas     0.04878848 -0.0539293  0.1752602
## nox     -0.38005064  0.5908789 -0.4273208
## rm       0.12806864 -0.6138083  0.6953599
## age     -0.27353398  0.6023385 -0.3769546
## dis      0.29151167 -0.4969958  0.2499287
## rad     -0.44441282  0.4886763 -0.3816262
## tax     -0.44180801  0.5439934 -0.4685359
## ptratio -0.17738330  0.3740443 -0.5077867
## black    1.00000000 -0.3660869  0.3334608
## lstat   -0.36608690  1.0000000 -0.7376627
## medv     0.33346082 -0.7376627  1.0000000
```

Strong correlation between crim and rad, crim and tax, and crim and lstat

Part D



Crim: Largely safe areas, tail-end of histogram shows roughly 20 suburbs with crime rates > 20 Tax: Large separation between areas with lowest tax rate and areas with highest tax rate Ptratio: Slightly skewed towards a high ratio

Part E

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

35

Part F

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

19.05

Part G

```
##              399      406
## crim     38.3518  67.9208
## zn        0.0000   0.0000
## indus    18.1000  18.1000
```

```
## chas      0.0000   0.0000
## nox       0.6930   0.6930
## rm        5.4530   5.6830
## age     100.0000 100.0000
## dis       1.4896   1.4254
## rad      24.0000  24.0000
## tax     666.0000 666.0000
## ptratio  20.2000  20.2000
## black   396.9000 384.9700
## lstat    30.5900  22.9800
## medv      5.0000   5.0000
```

406 and 309: high crim, indus, nox, age, rad, tax, ptratio, black, and lstat; low zn, chas, rm, dis, and medv

Part H

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

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

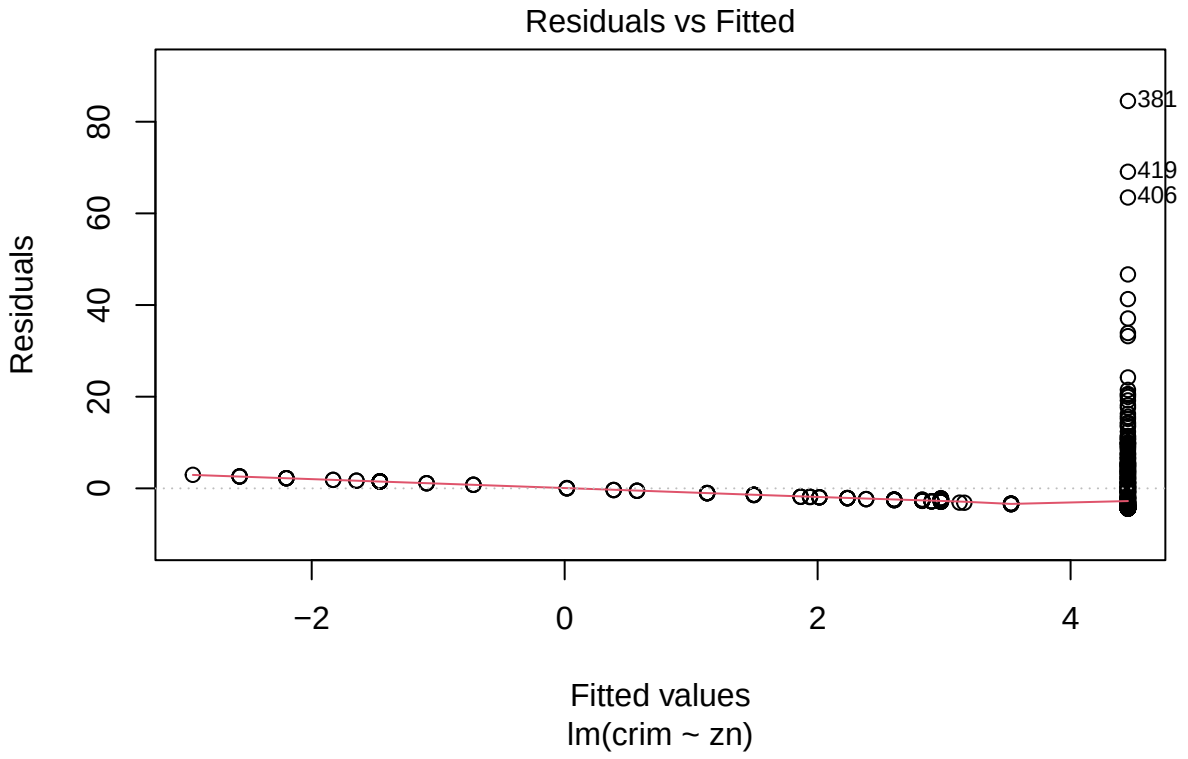
```
##       crim                zn             indus              chas        
##  Min.   :0.02009   Min.   : 0.00   Min.   : 2.680   Min.   :0.0000  
##  1st Qu.:0.33147   1st Qu.: 0.00   1st Qu.: 3.970   1st Qu.:0.0000  
##  Median :0.52014   Median : 0.00   Median : 6.200   Median :0.0000  
##  Mean   :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
```

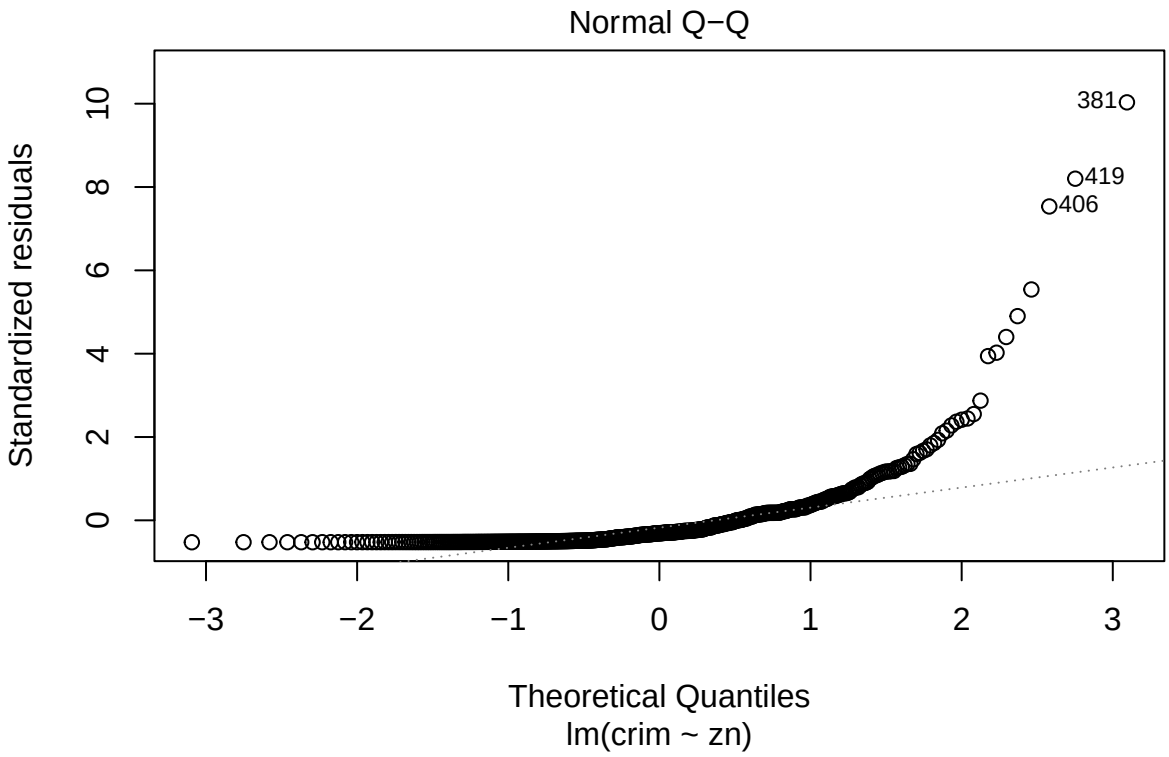
7: 64 8: 13 8 Summary: Low crim, ptratio, indus; high medv, age

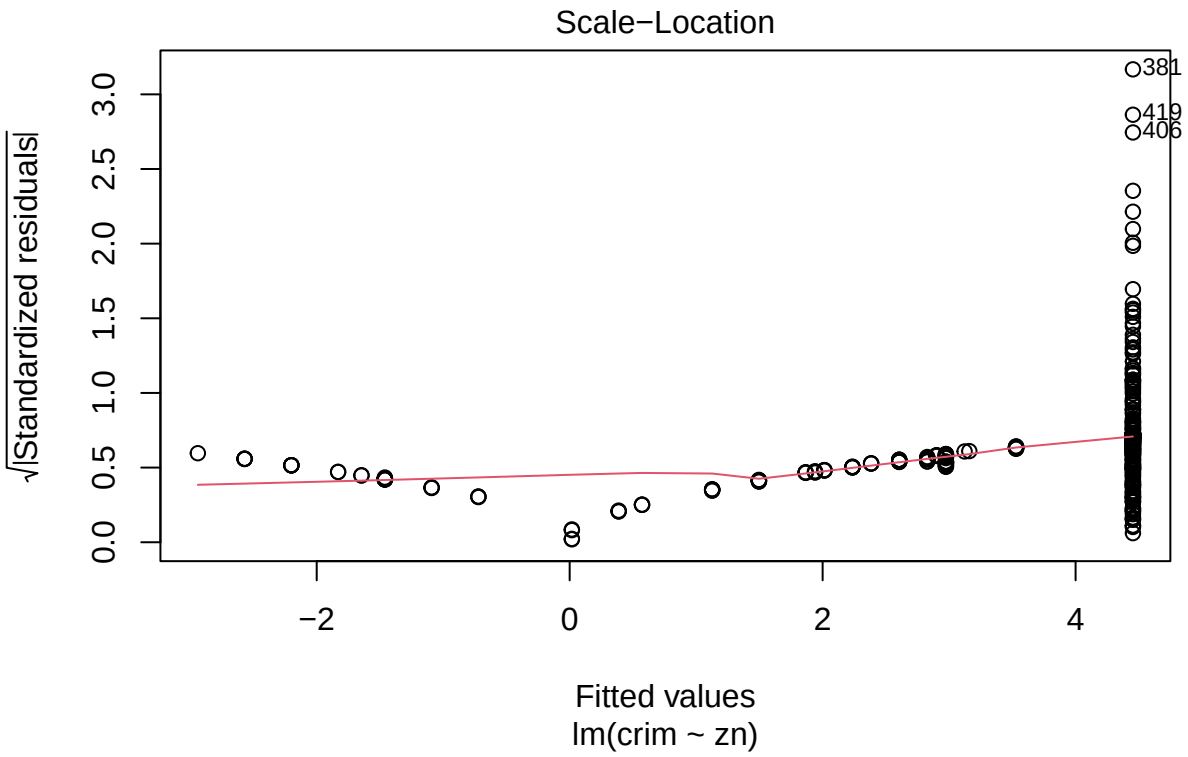
Chapter 3, Question 15

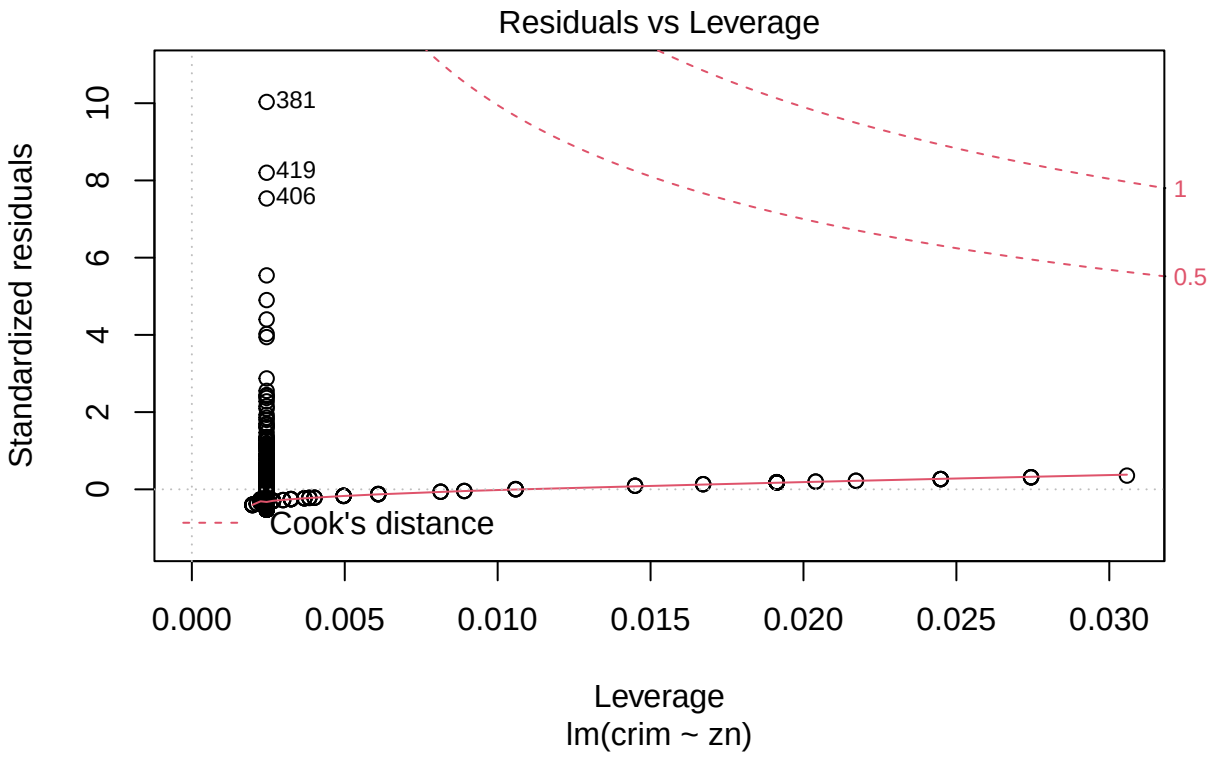
Part A

```
##
## 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
```



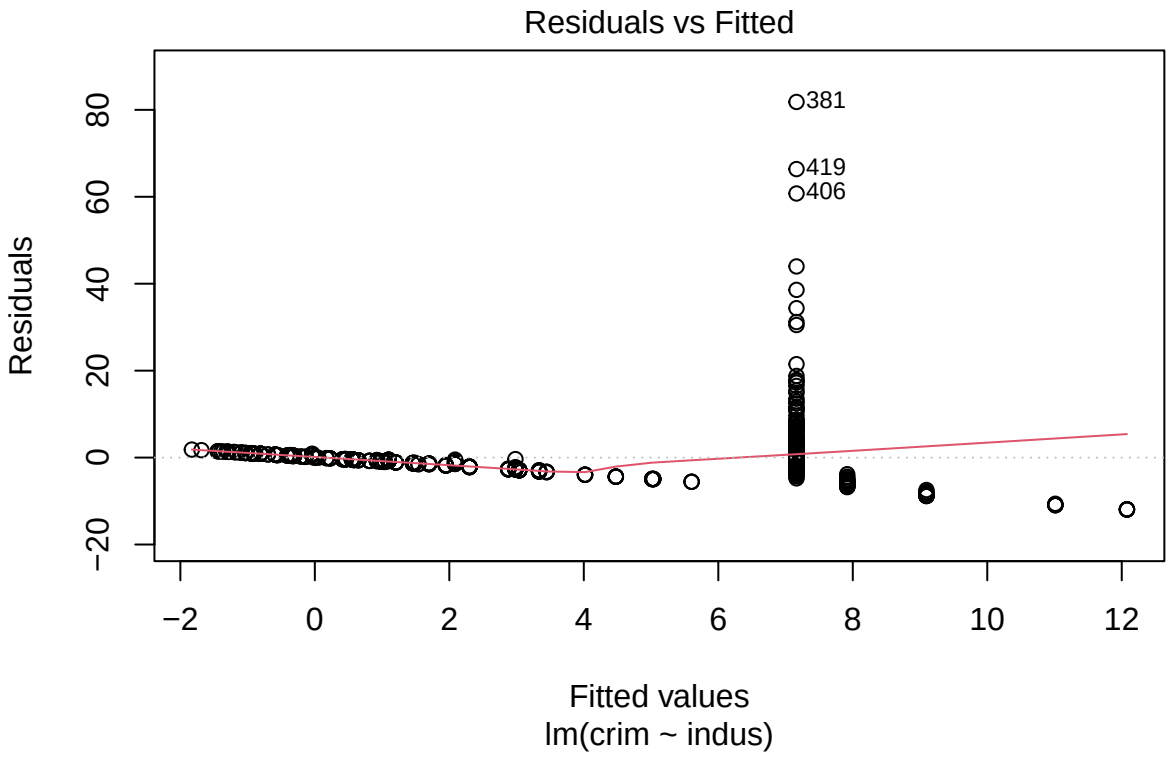

Normal Q−Q
Standardized residuals
0
2
4
6
8
10
381
419
406
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(crim ~ zn)

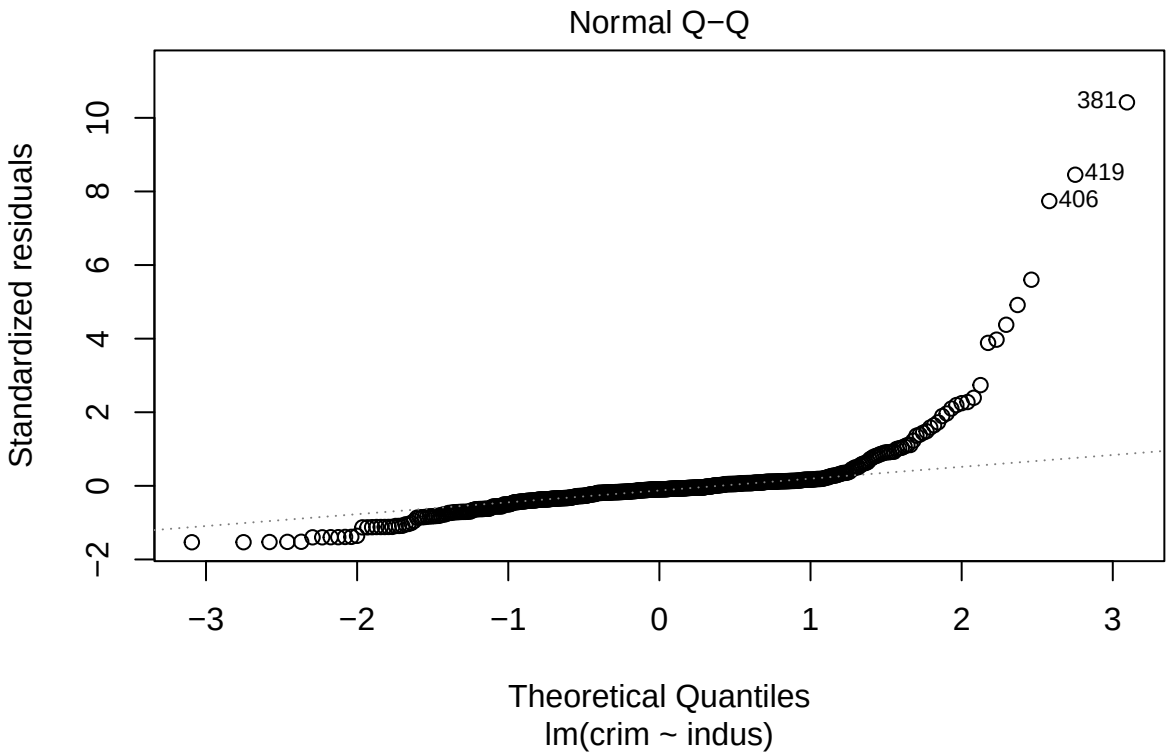

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
381
419
406
−2
0
2
4
Fitted values
lm(crim ~ zn)

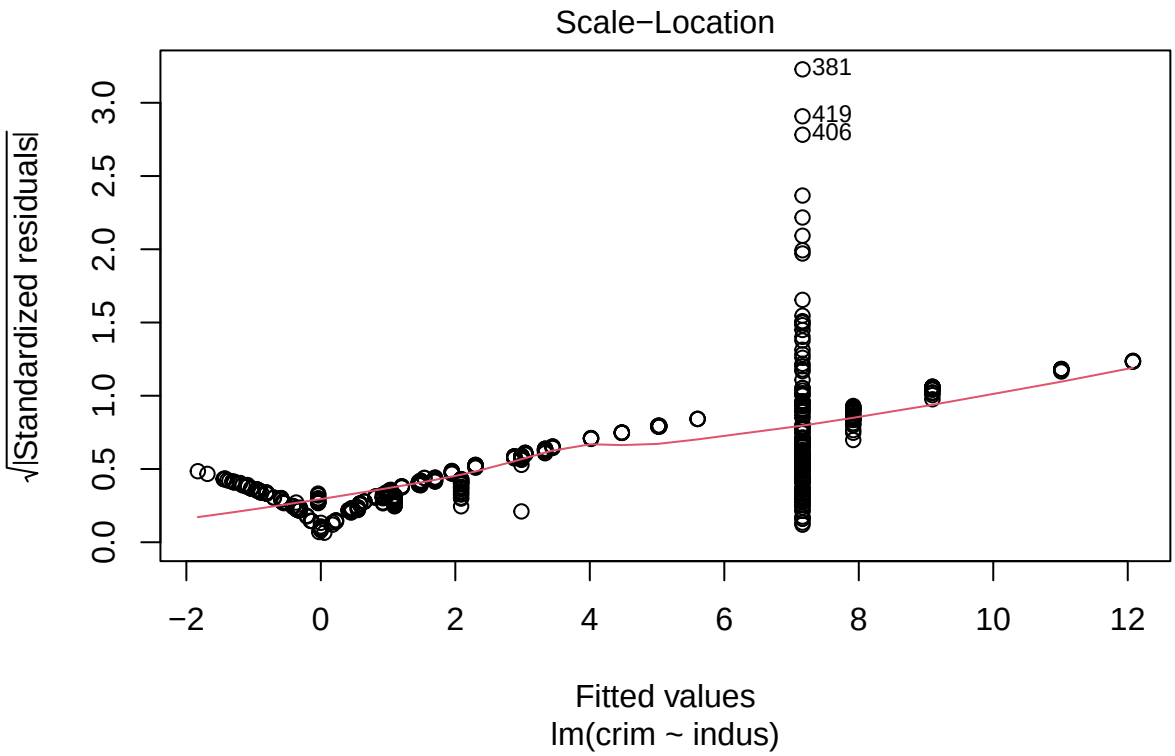


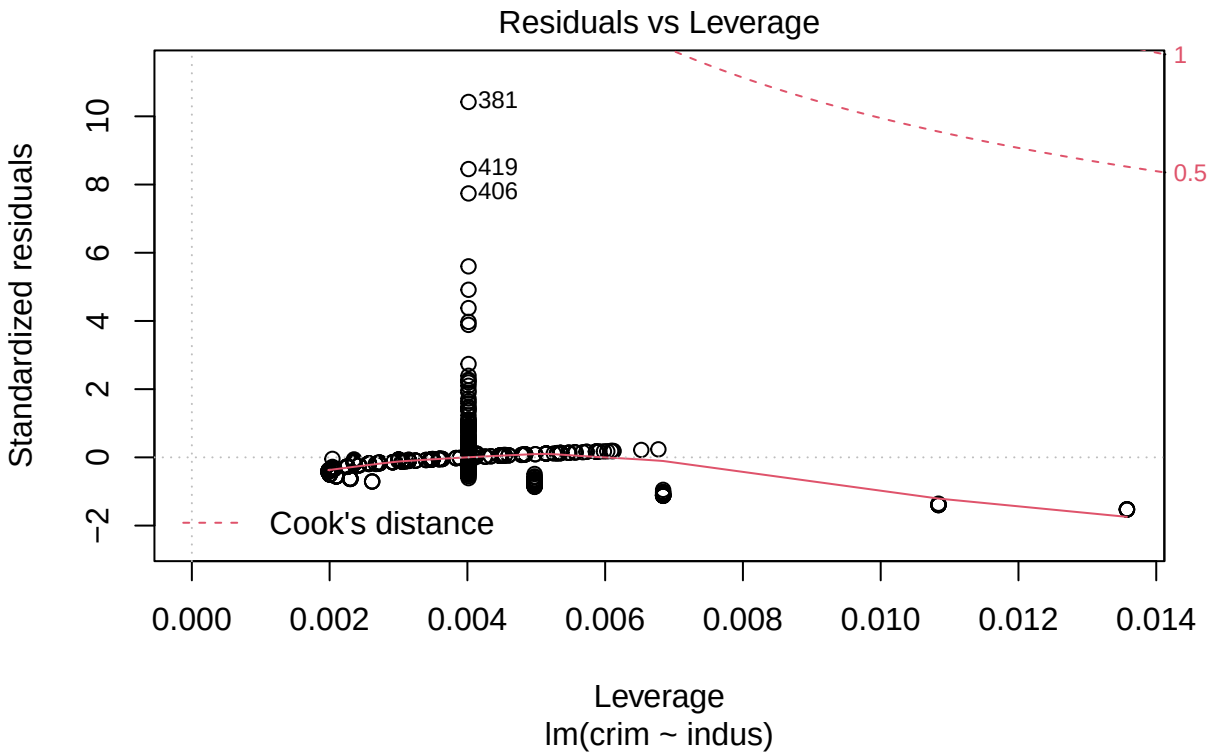
Part 1

```
##
## 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
```


Residuals vs Fitted
Residuals
Fitted values
lm(crim ~ indus)
381
419
406
−20
0
20
40
60
80
−2
0
2
4
6
8
10
12


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

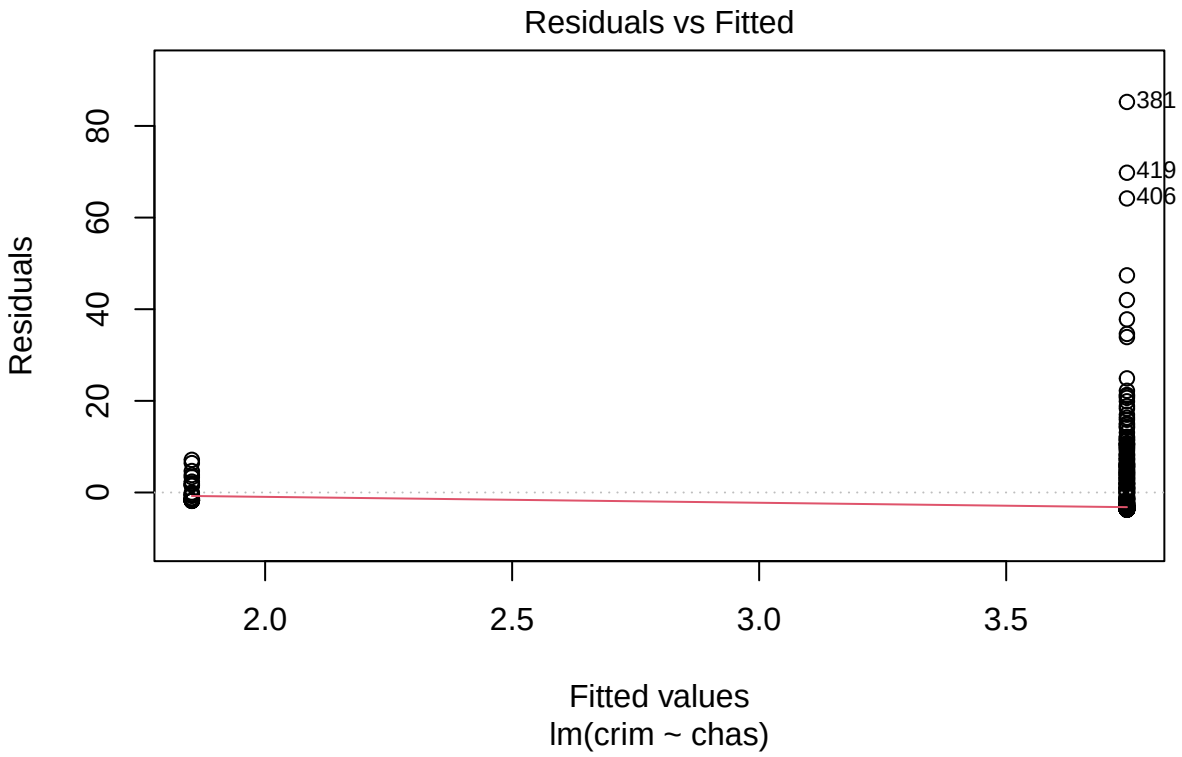

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
381
419
406
−2
0
2
4
6
8
10
12
Fitted values
lm(crim ~ indus)

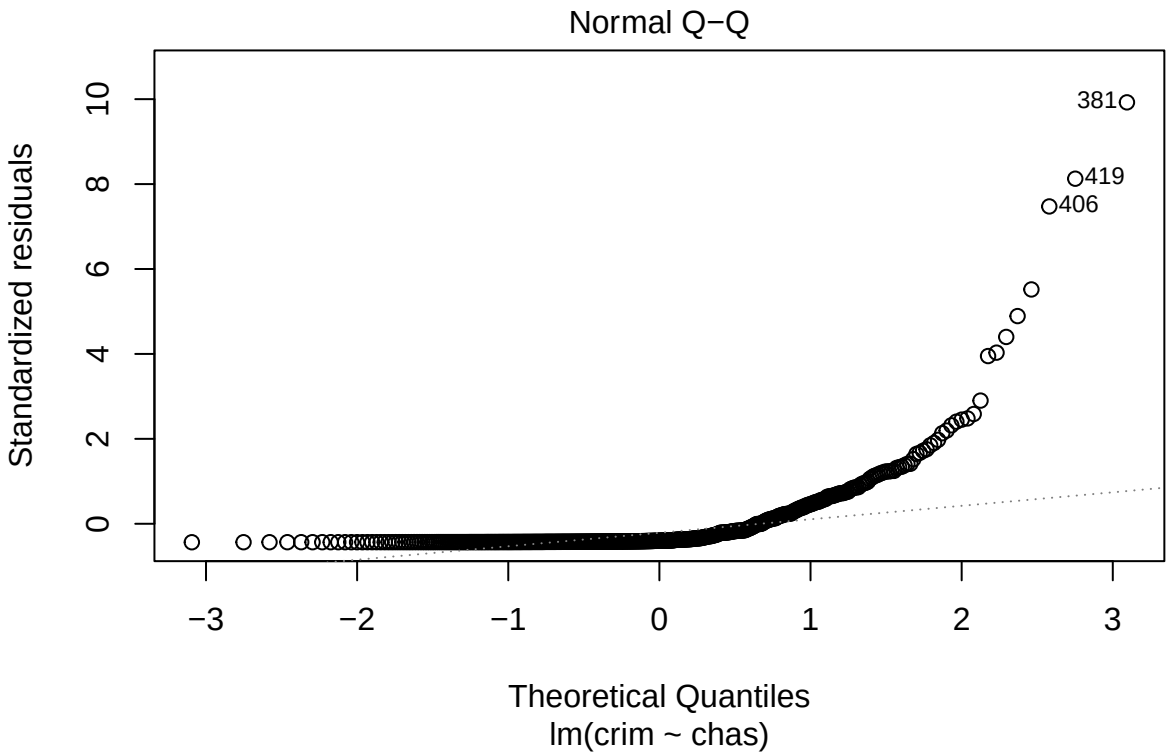


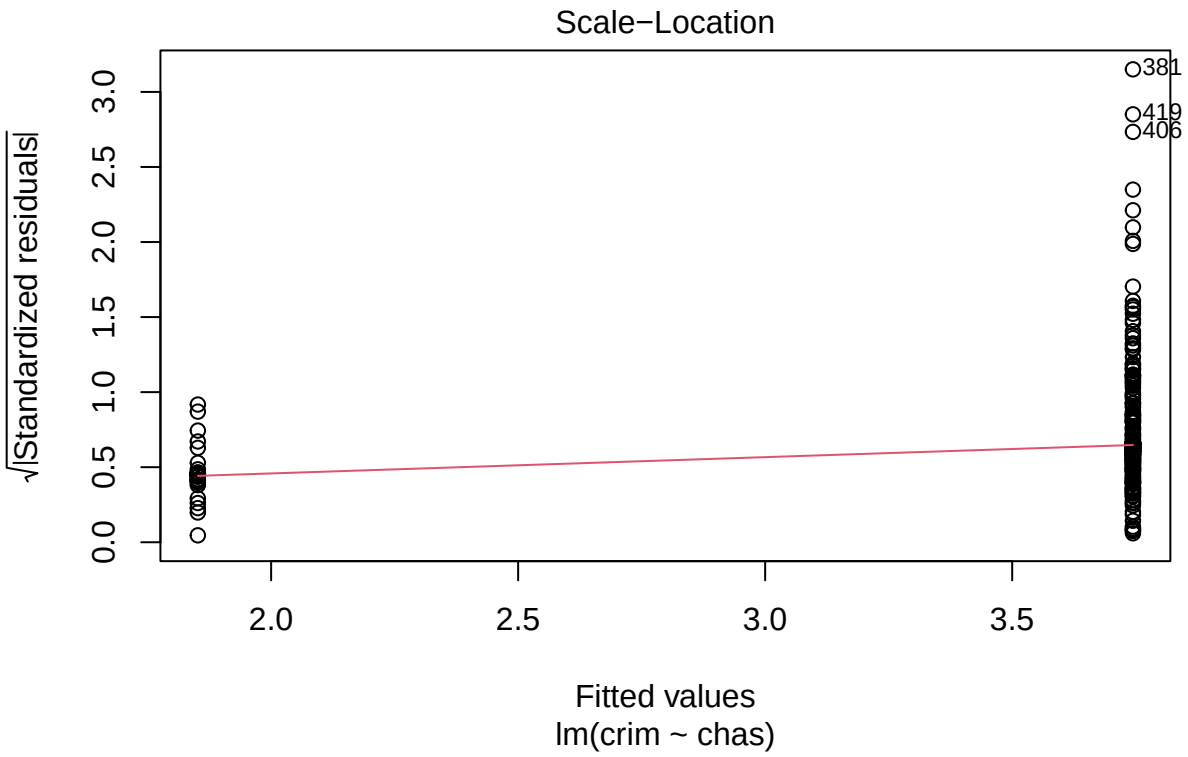
1.) p-value is < 0.05 so there is statistically significant association between crim and zn this means that changes in zn are related to changes in crim

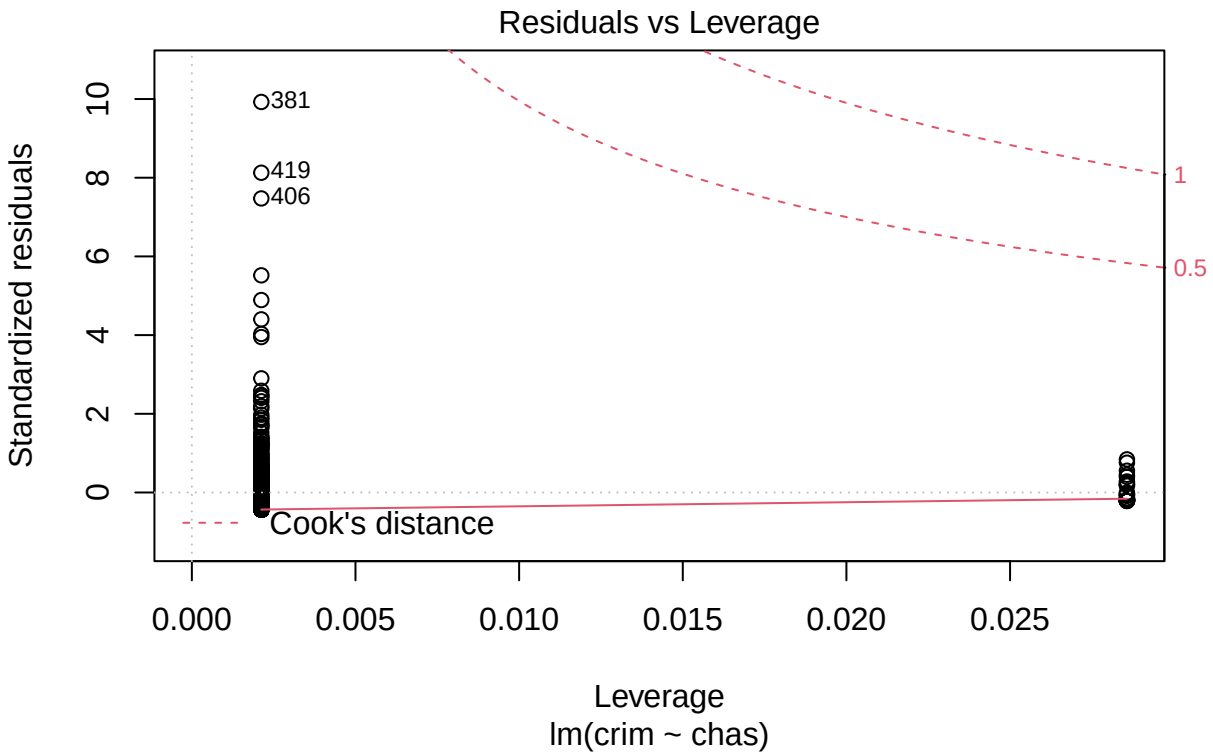
Part 2

```
##
## 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 ***
## chas1        -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
```


Residuals vs Fitted
Residuals
0
20
40
60
80
381
419
406
2.0
2.5
3.0
3.5
Fitted values
lm(crim ~ chas)


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

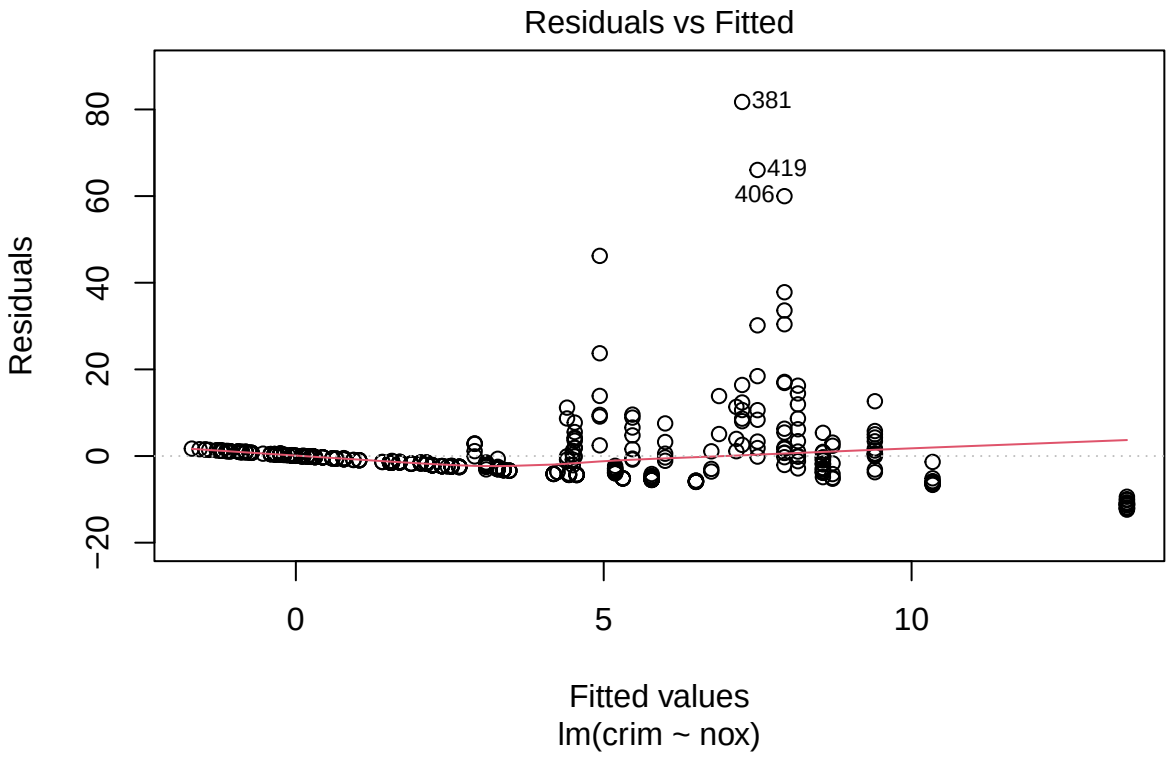

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
381
419
406
2.0
2.5
3.0
3.5
Fitted values
lm(crim ~ chas)

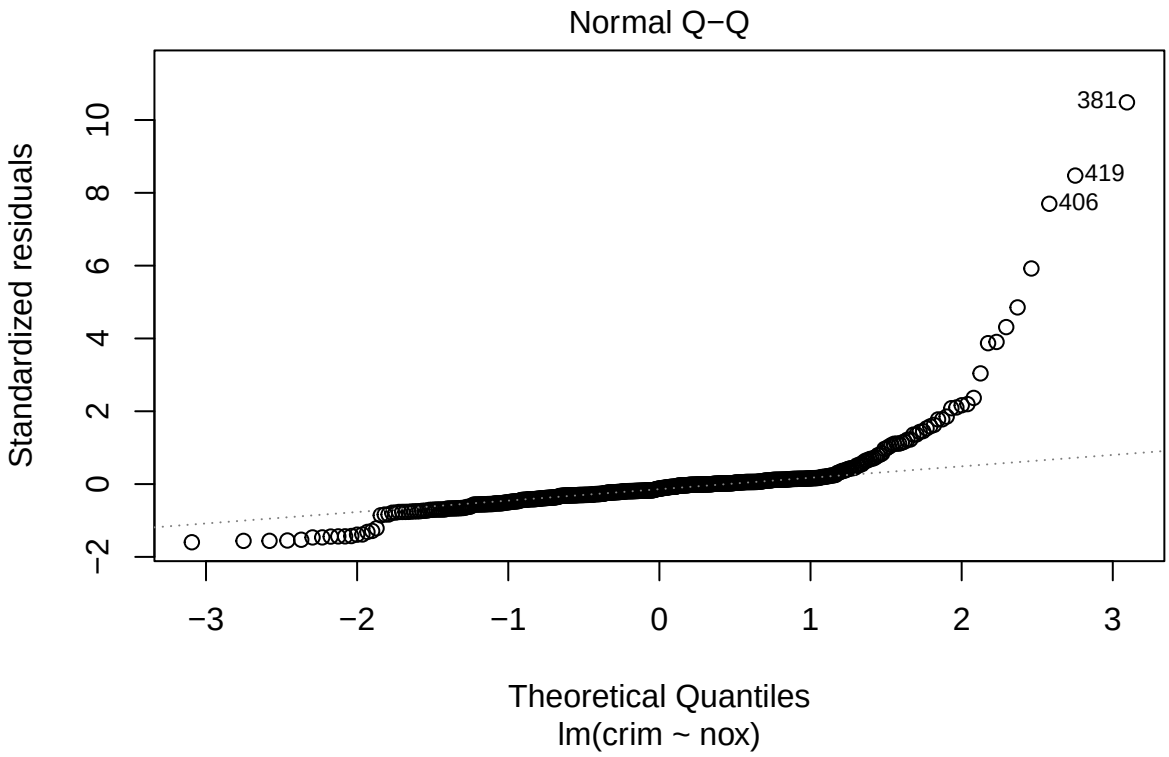


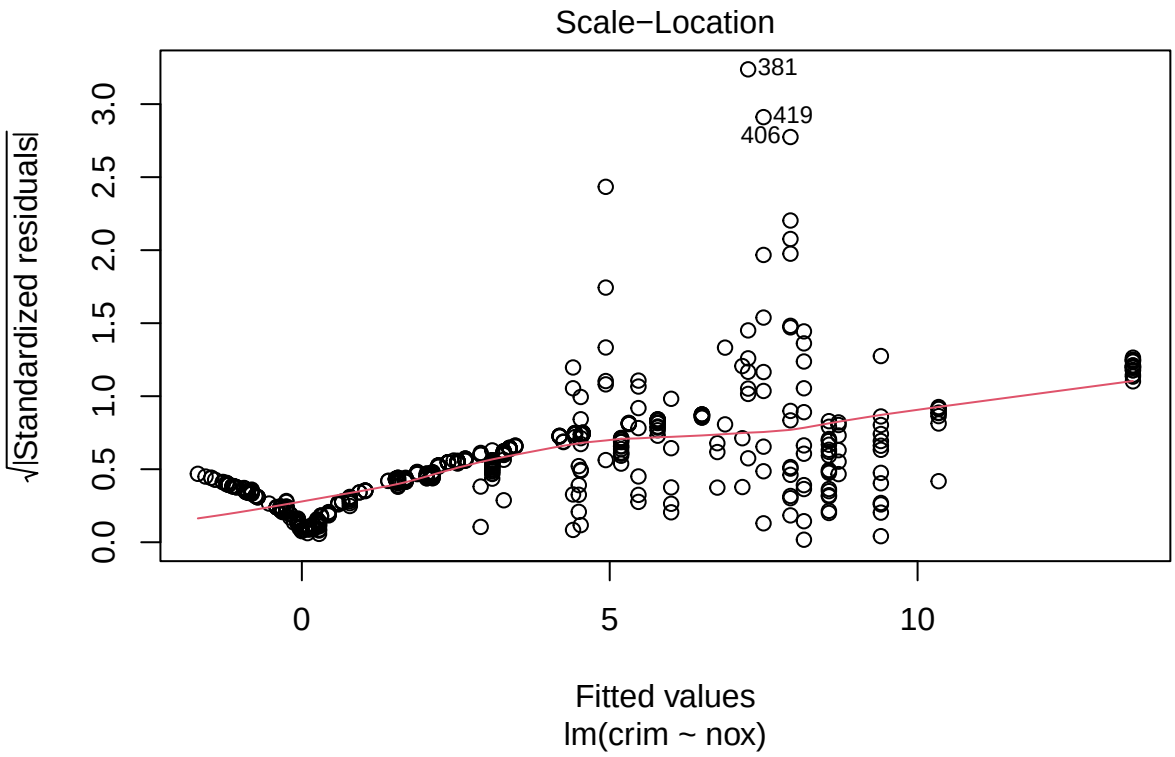
2.) p-value is > 0.05 so there is not a statistically significant association between crim and chas this means that changes in chas are likely not related to changes in crim

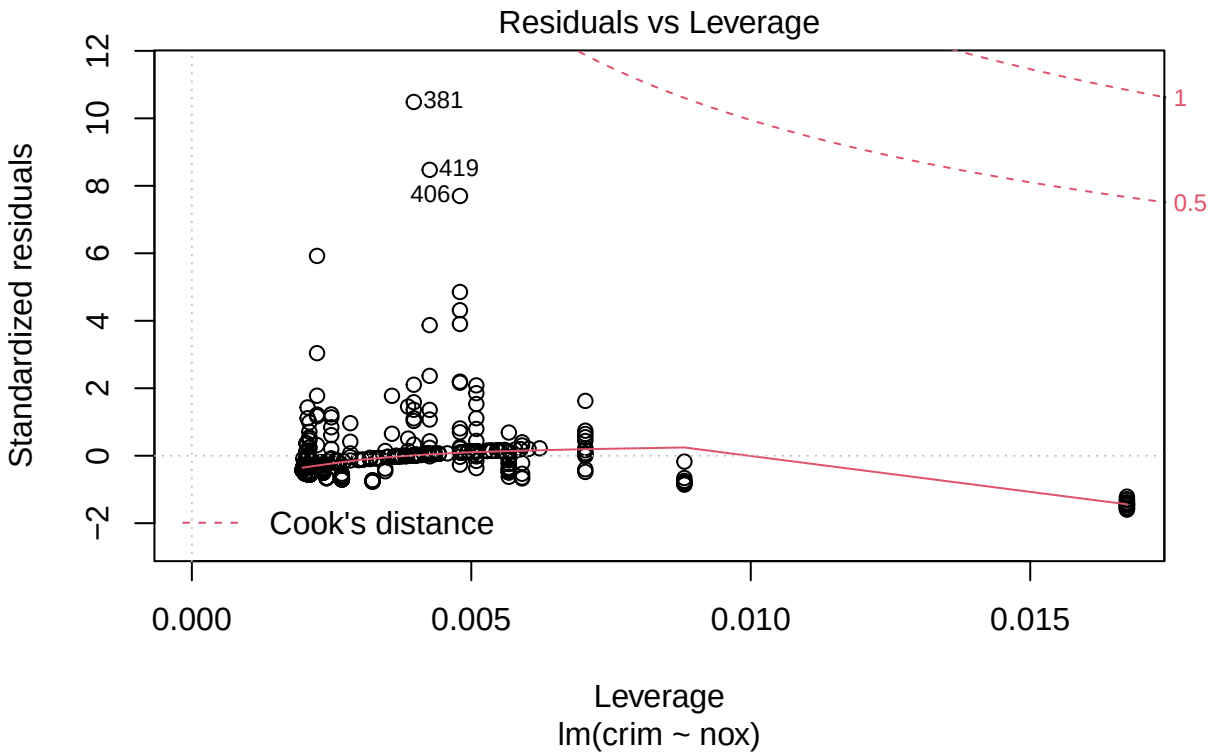
Part 3

```
##
## 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
```


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


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

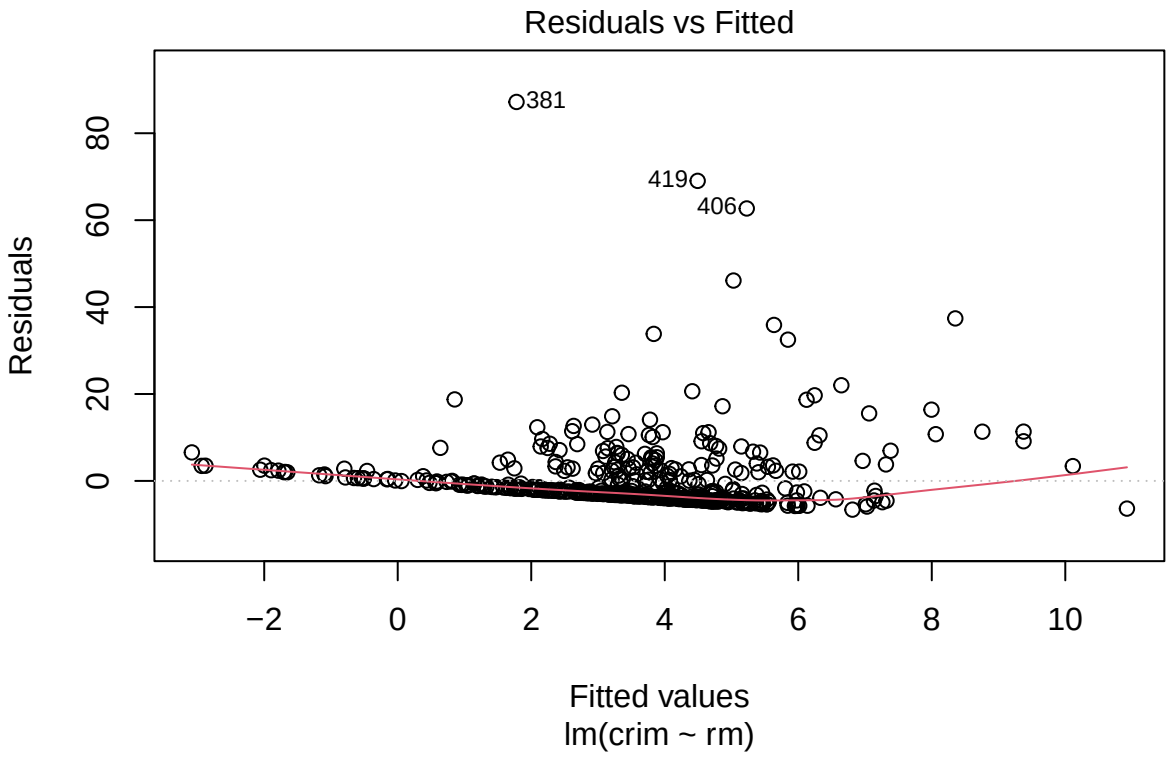

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
381
419
406
0
5
10
Fitted values
lm(crim ~ nox)

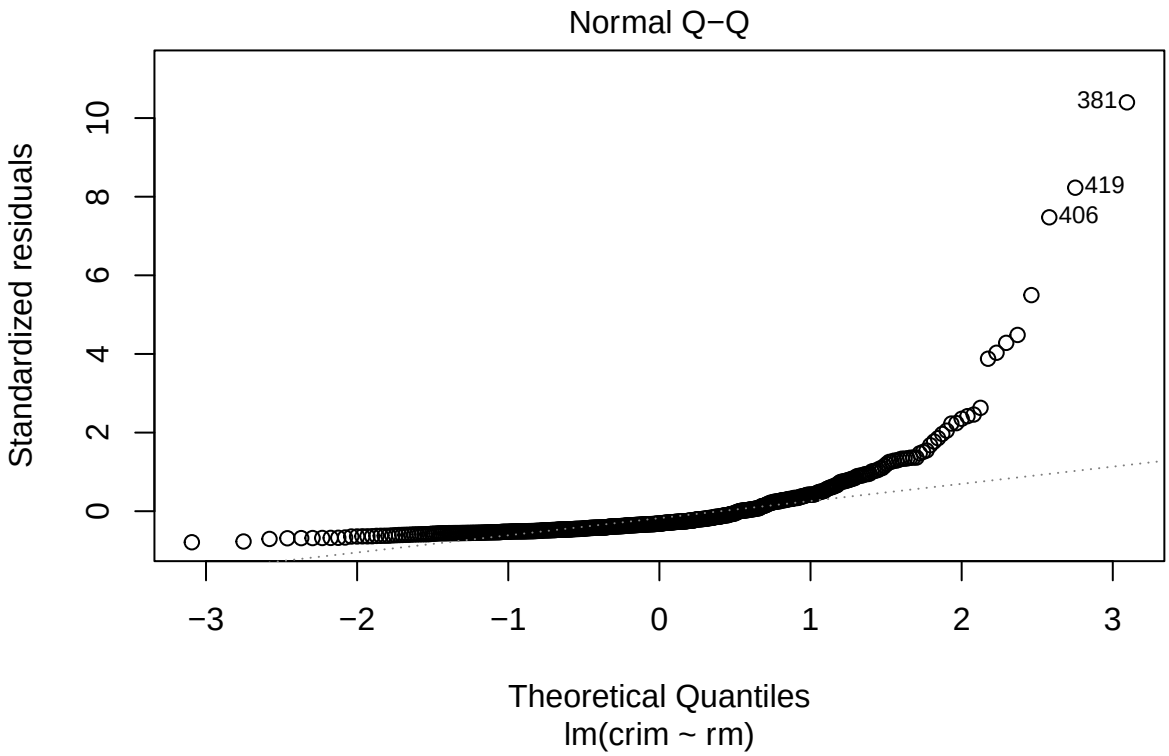


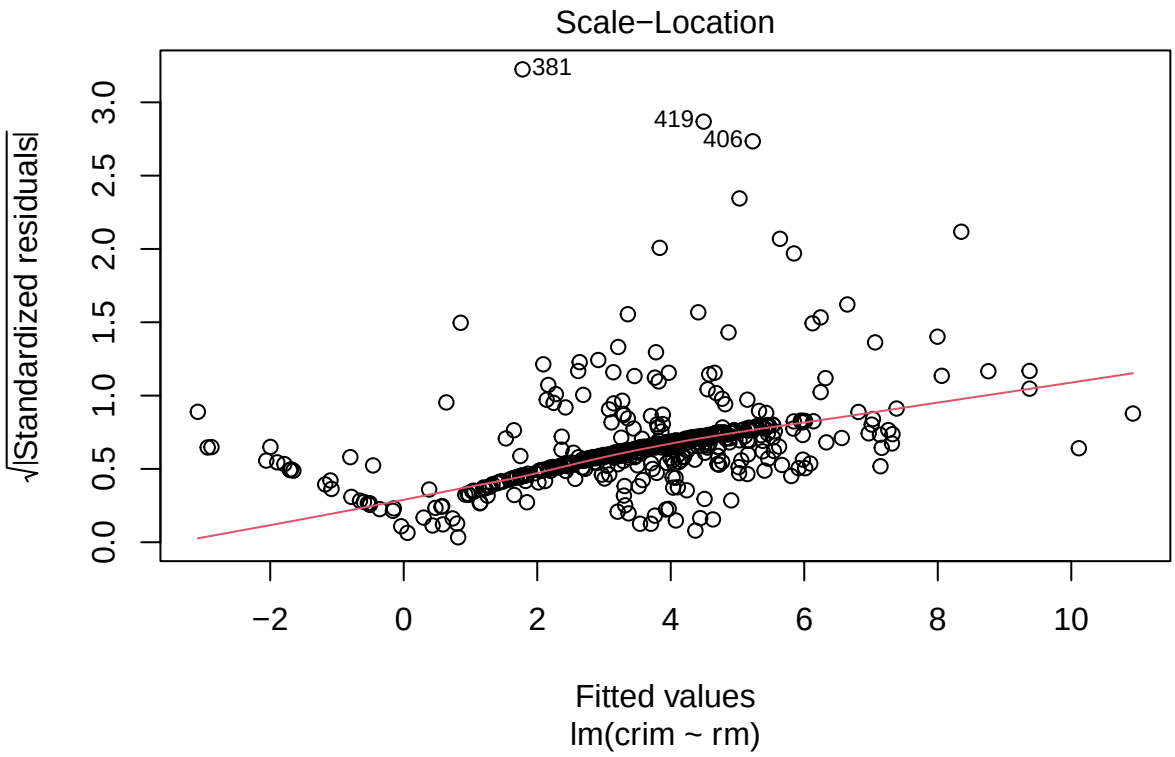
3.) p-value is < 0.05 so there is statistically significant association between crim and nox this means that changes in nox are related to changes in crim

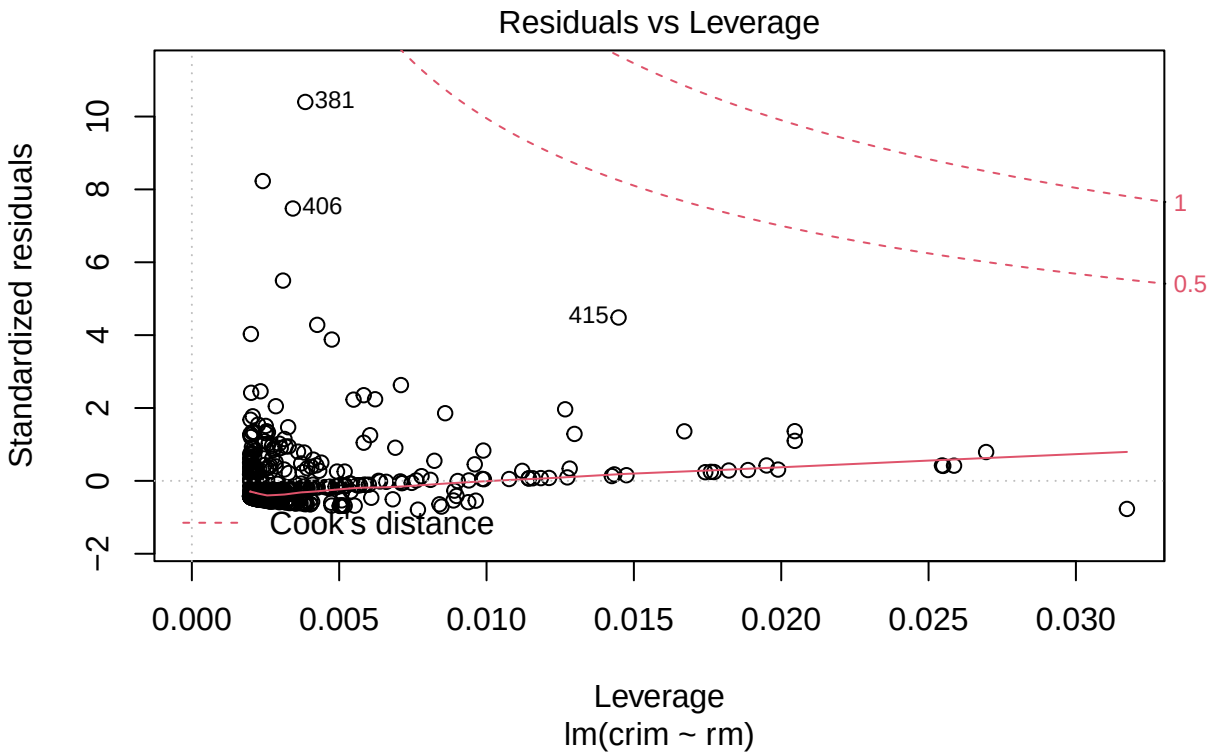
Part 4

```
##
## 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
```


Residuals vs Fitted
381
419
406
Residuals
0
20
40
60
80
−2
0
2
4
6
8
10
Fitted values
lm(crim ~ rm)


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

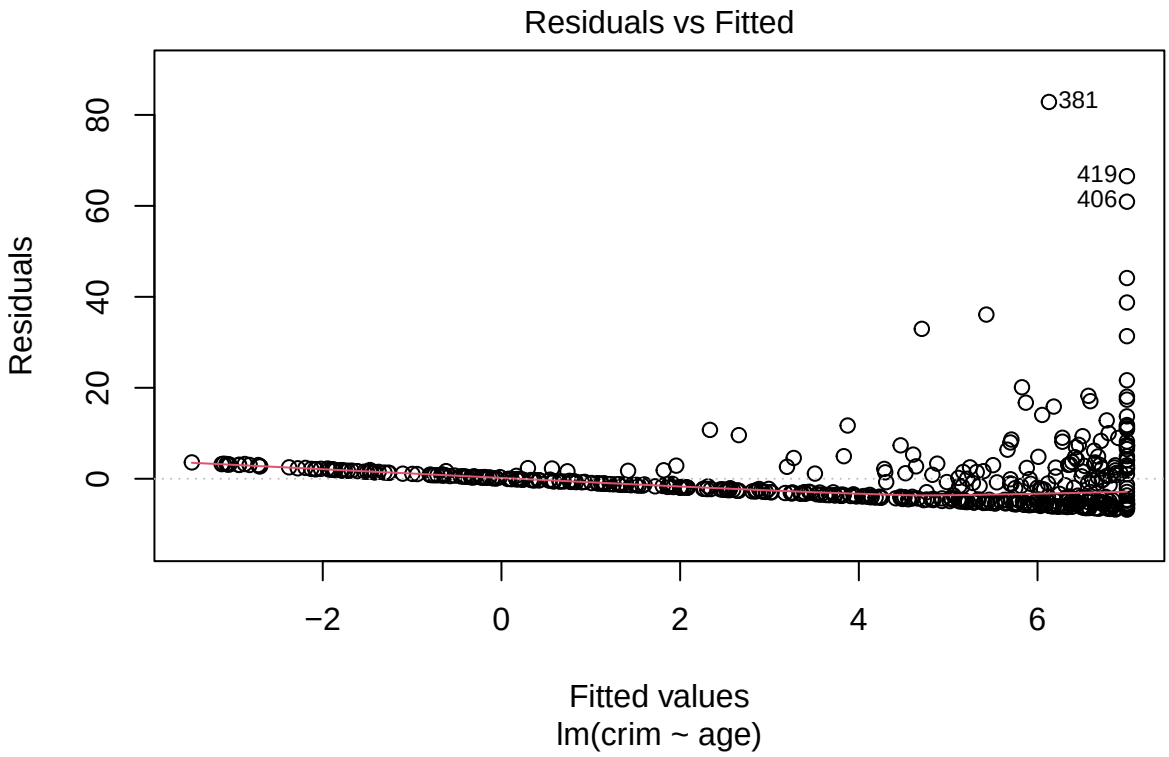

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
381
419
406
−2
0
2
4
6
8
10
Fitted values
lm(crim ~ rm)

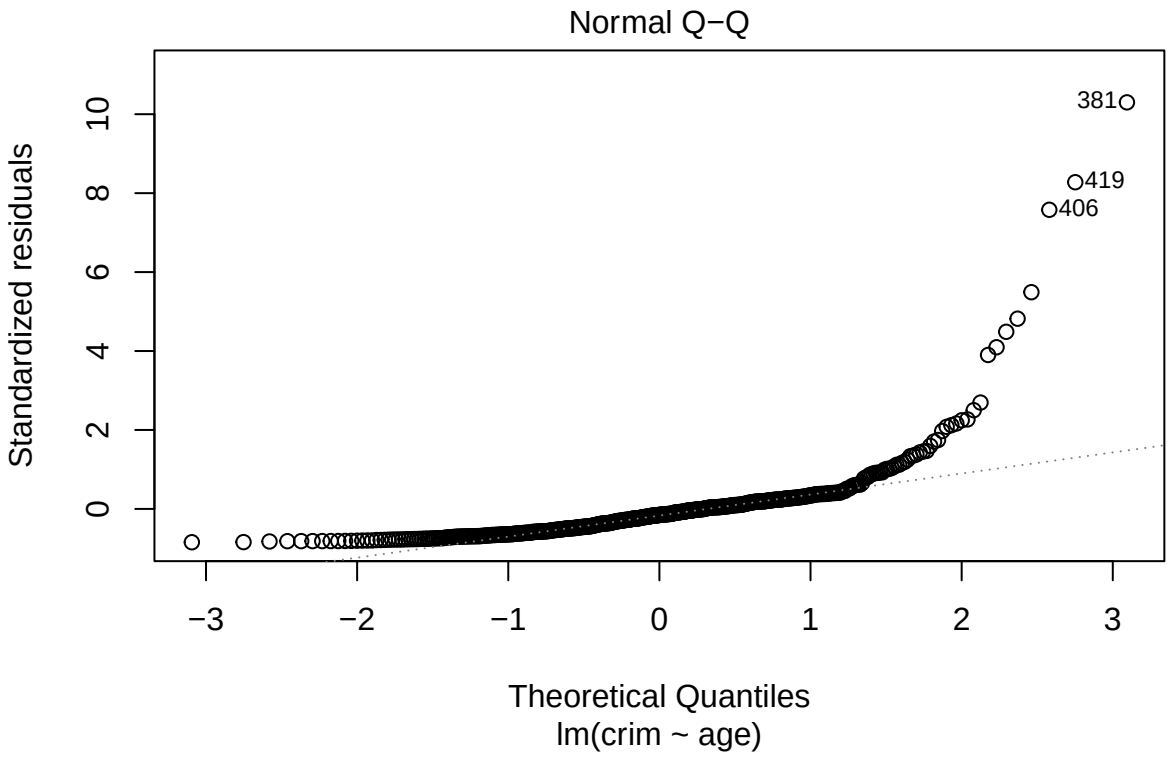


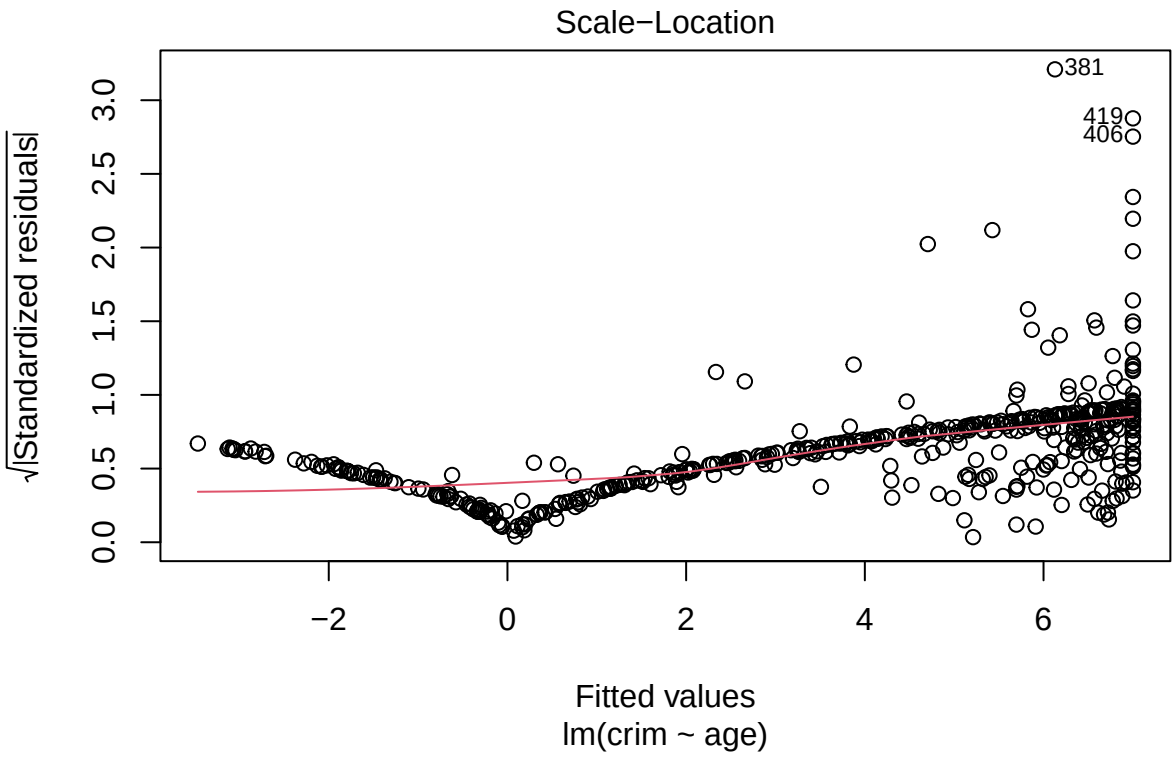
4.) p-value is < 0.05 so there is statistically significant association between crim and rm this means that changes in rm are related to changes in crim

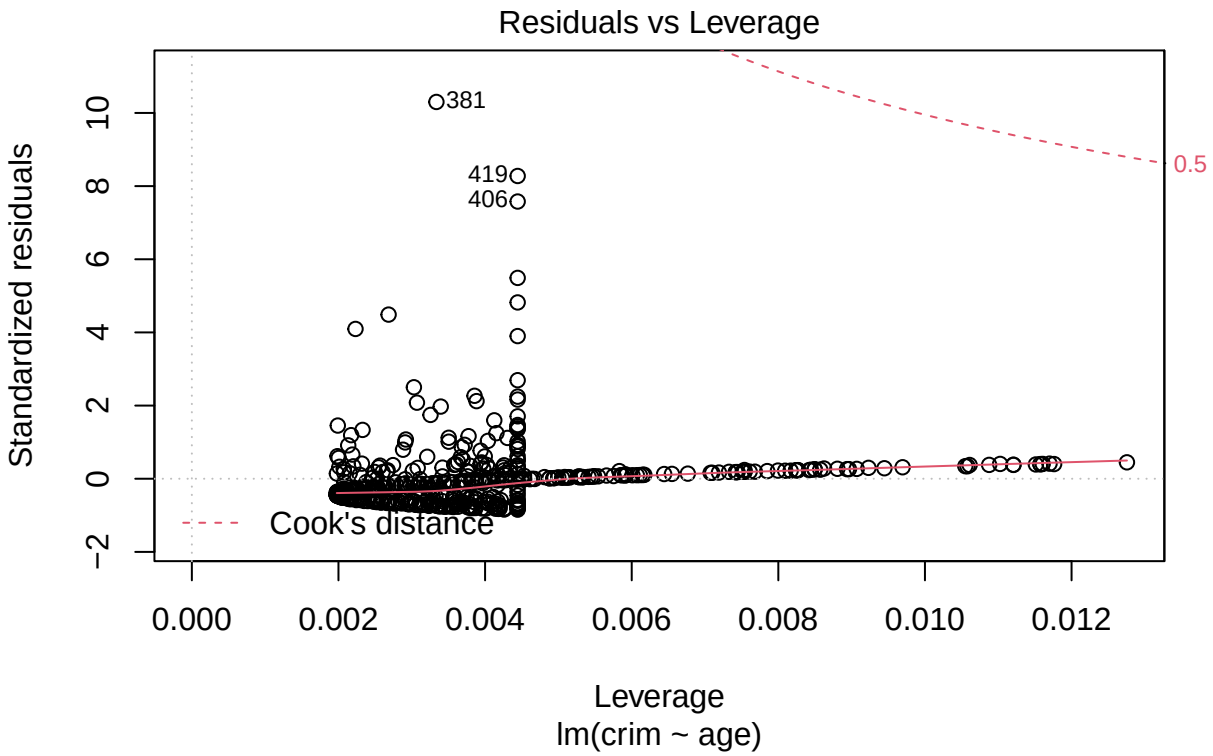
Part 5

```
##
## 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
```


Residuals vs Fitted
Residuals
0
20
40
60
80
381
419
406
−2
0
2
4
6
Fitted values
lm(crim ~ age)


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

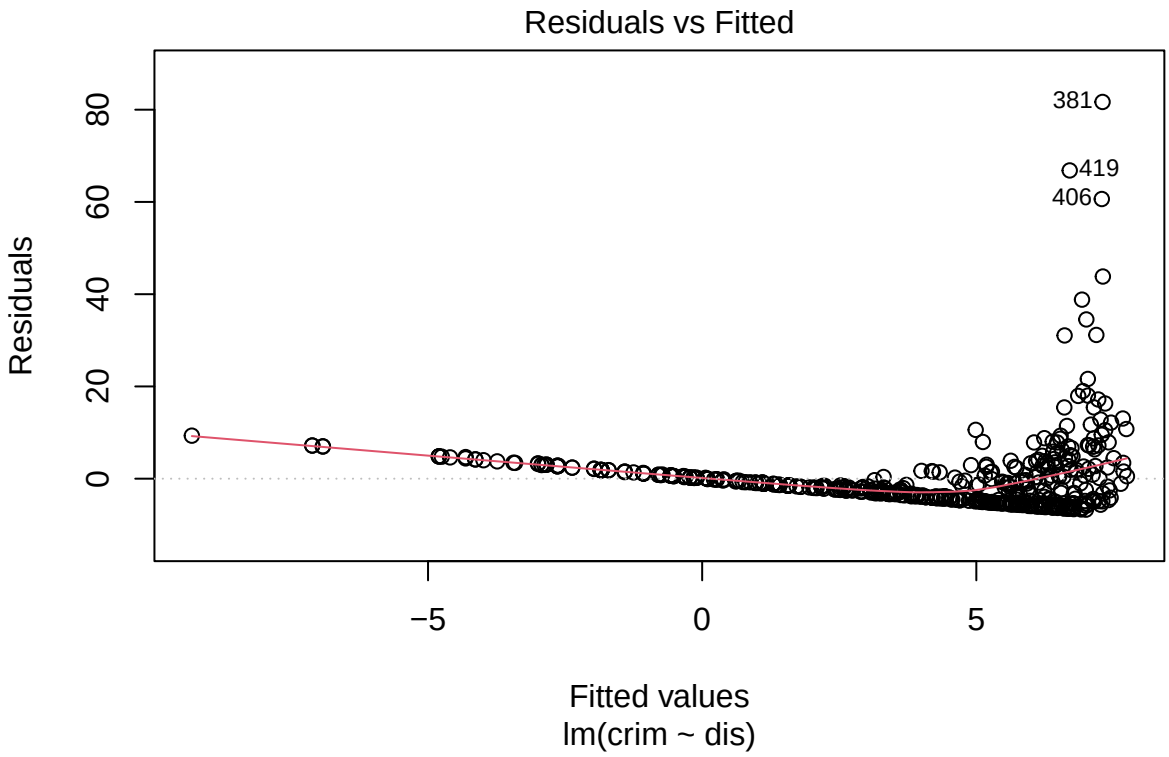

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
381
419
406
−2
0
2
4
6
Fitted values
lm(crim ~ age)

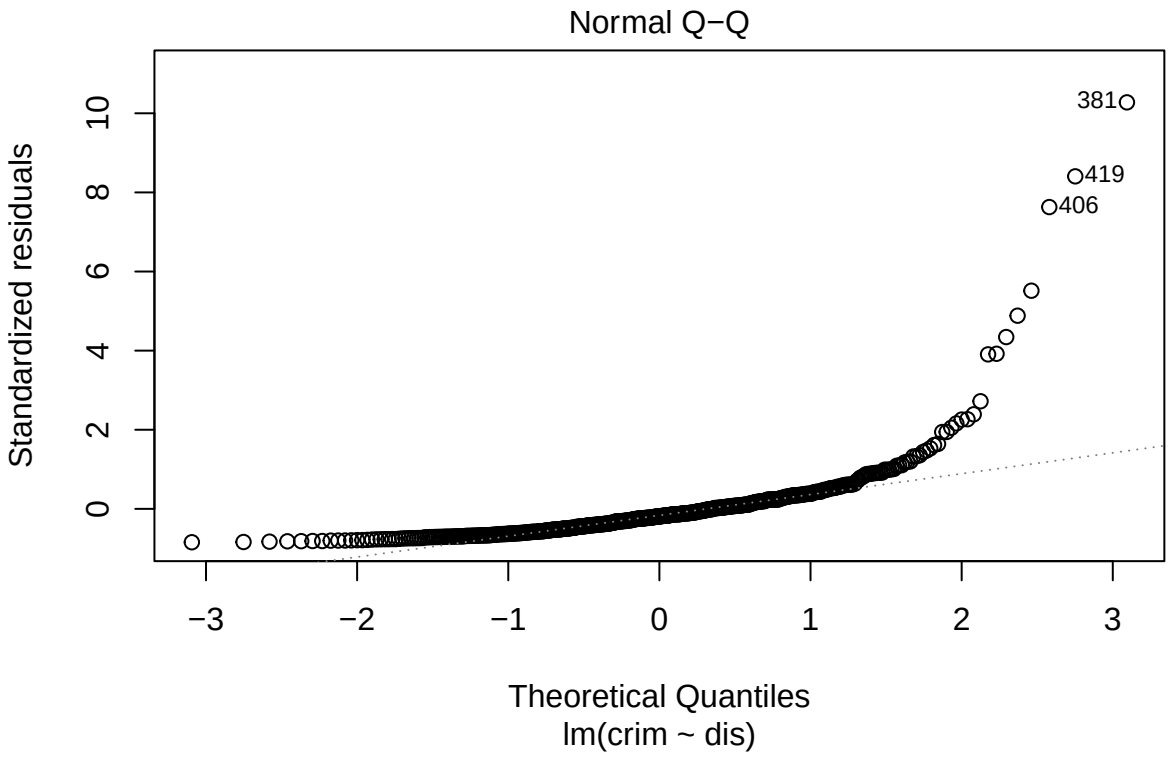


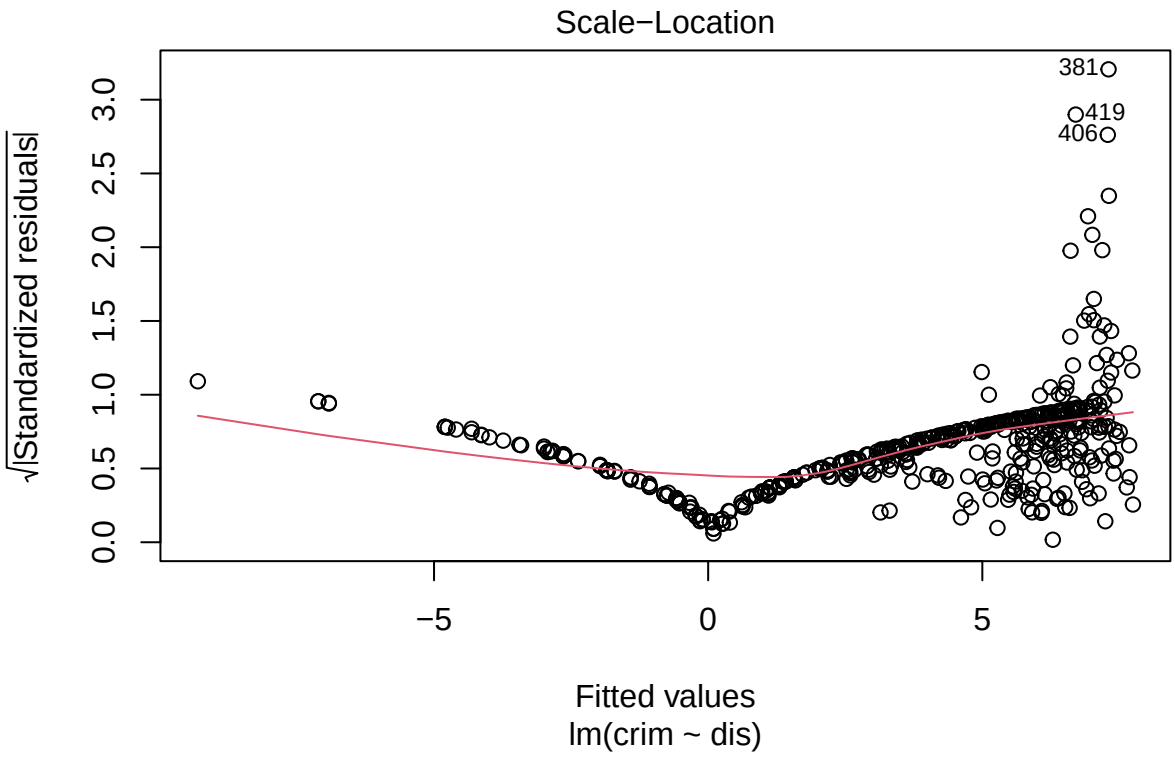
5.) p-value is < 0.05 so there is statistically significant association between crim and age this means that changes in age are related to changes in crim

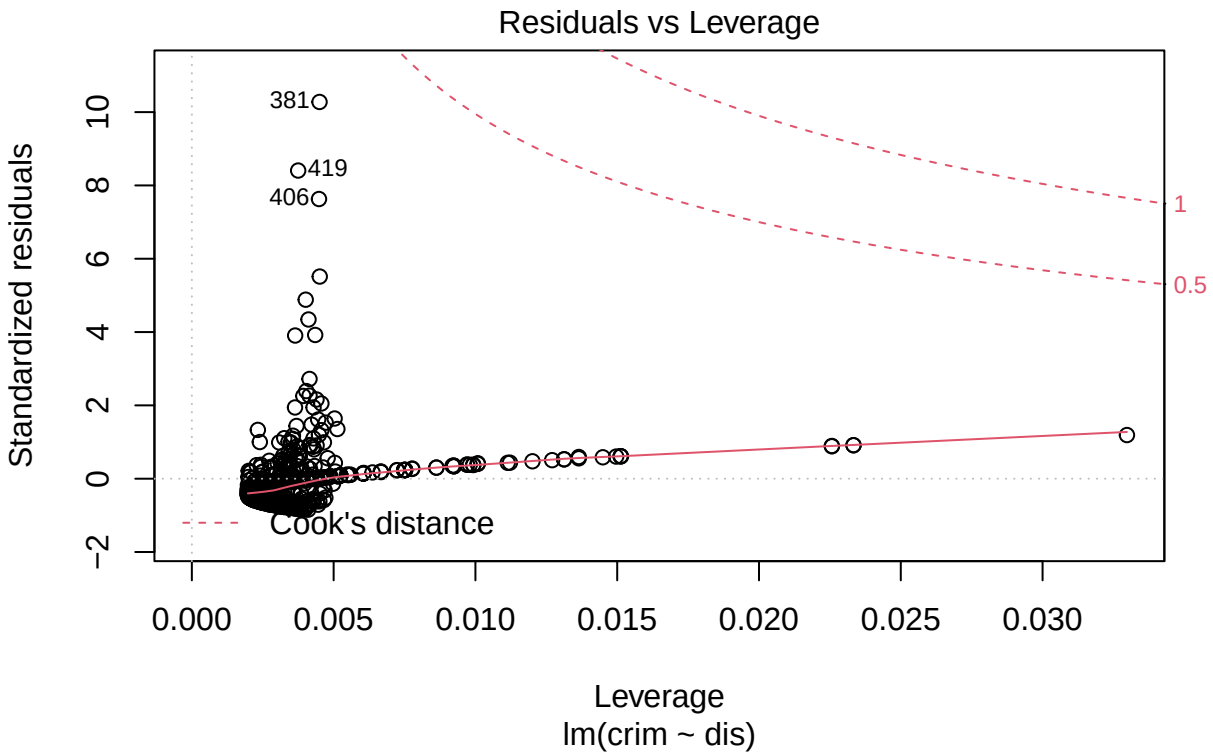
Part 6

```
##
## 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
```


Residuals vs Fitted
Residuals
0
20
40
60
80
381
419
406
−5
0
5
Fitted values
lm(crim ~ dis)


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

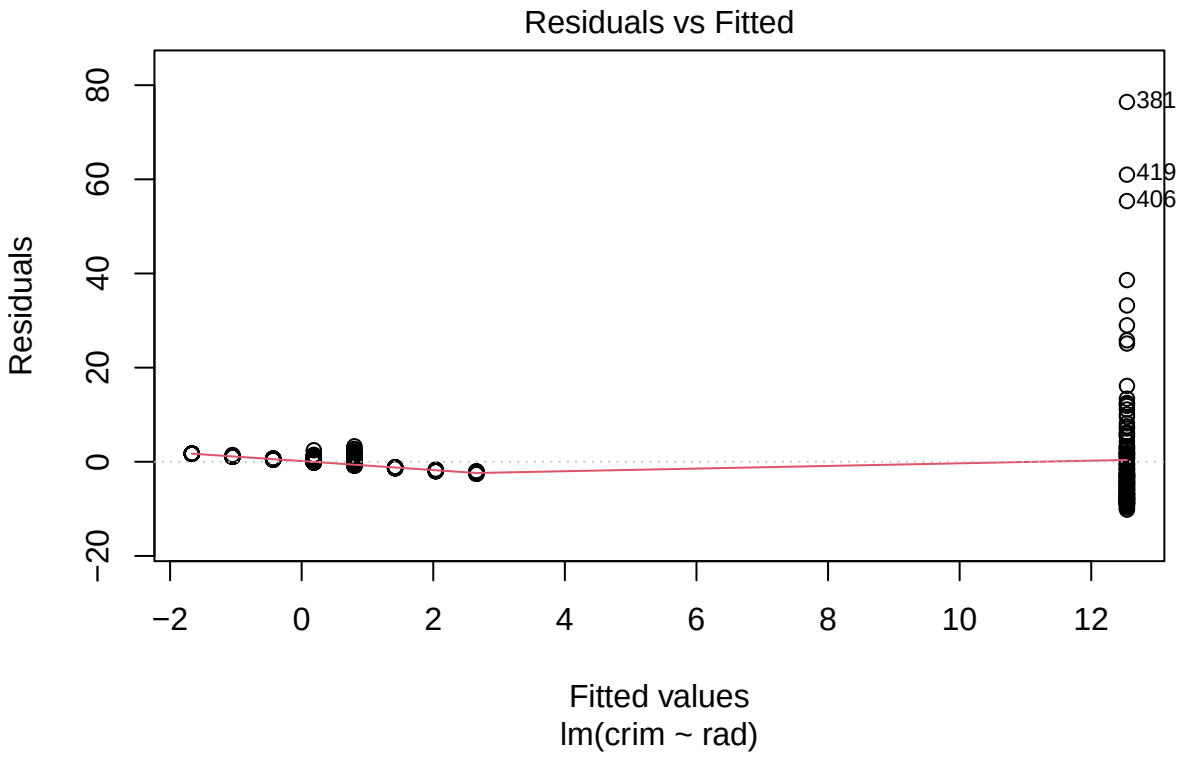

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
381
419
406
−5
0
5
Fitted values
lm(crim ~ dis)

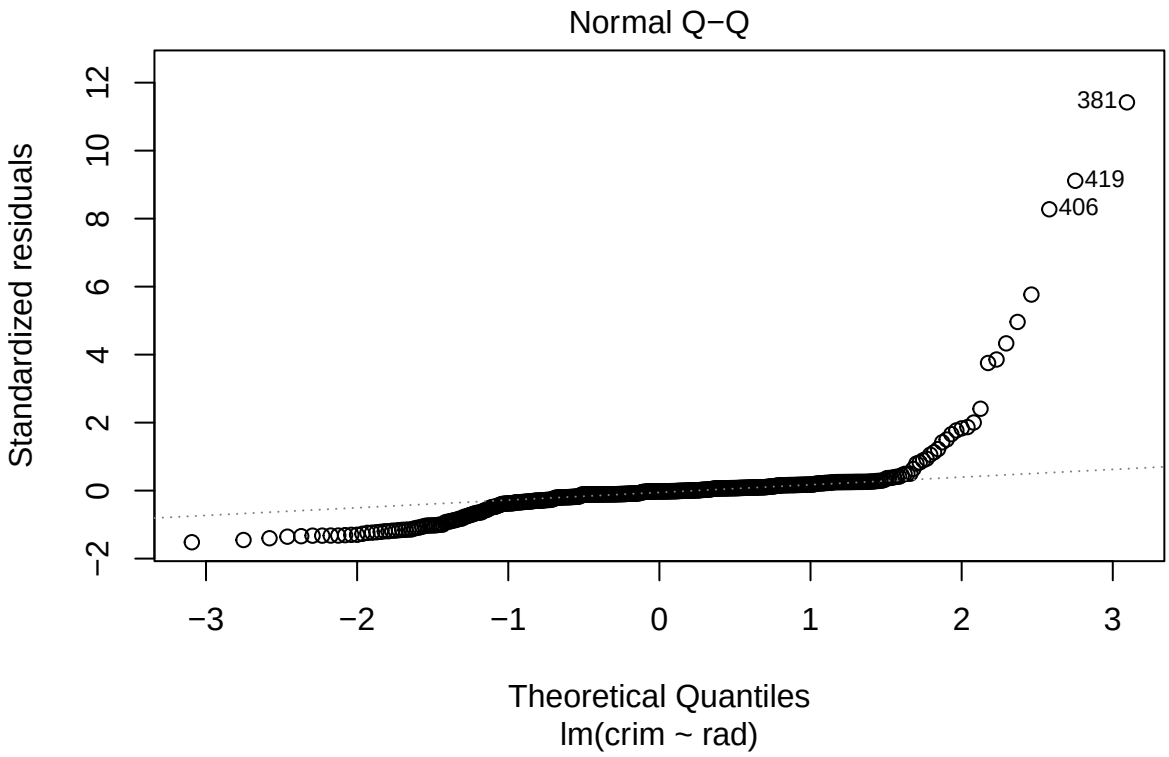


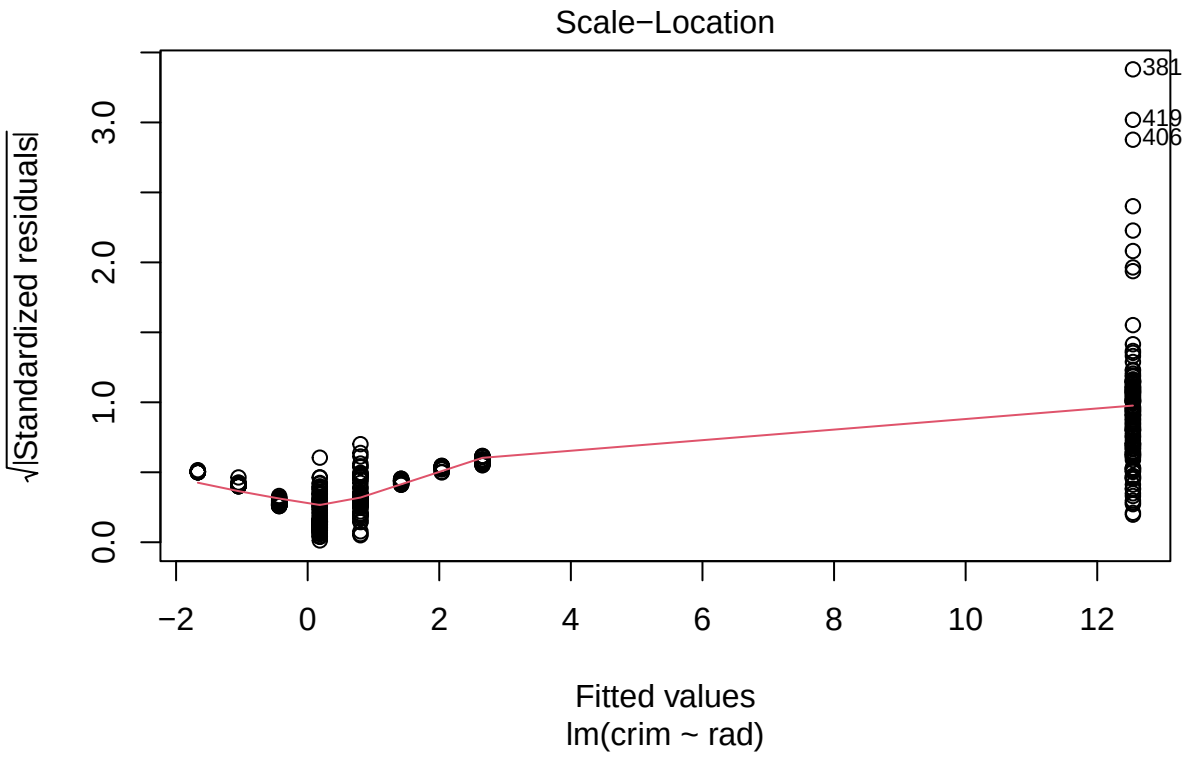
6.) p-value is < 0.05 so there is statistically significant association between crim and dis this means that changes in dis are related to changes in crim

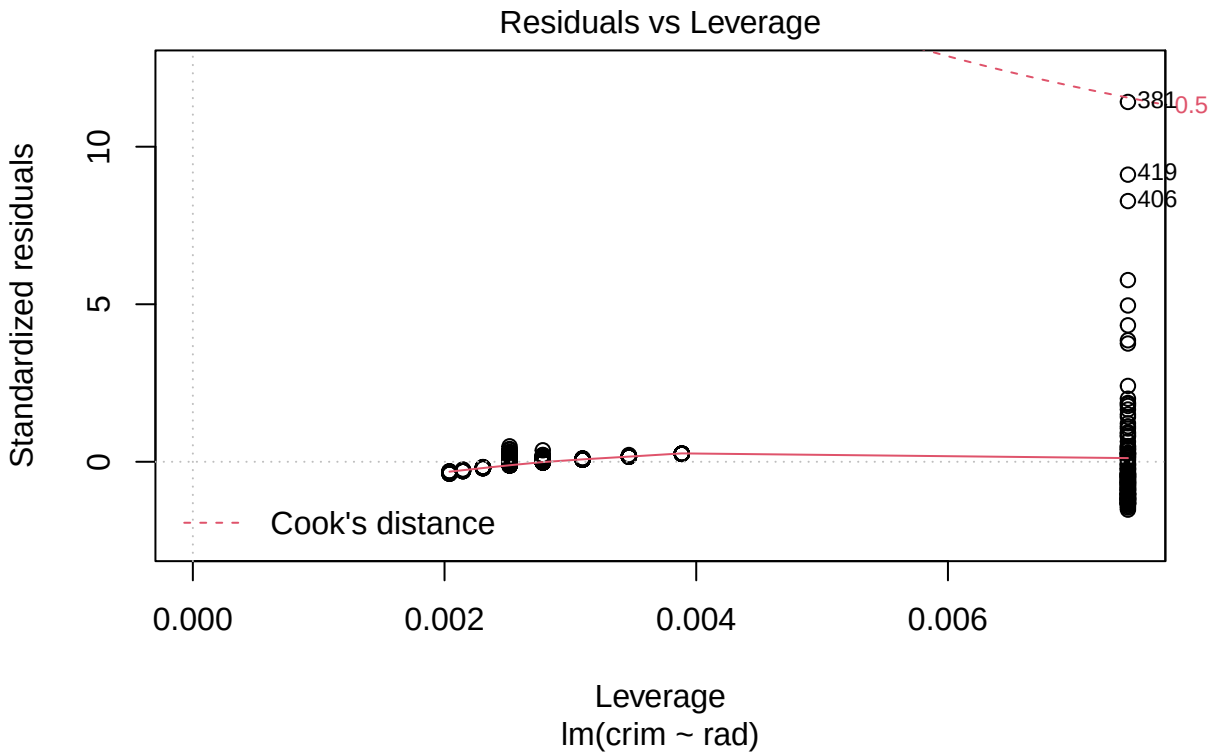
Part 7

```
##
## 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
```


Residuals vs Fitted
Residuals
Fitted values
lm(crim ~ rad)
381
419
406
−20
0
20
40
60
80
−2
0
2
4
6
8
10
12


Normal Q−Q
Standardized residuals
−2
0
2
4
6
8
10
12
381
419
406
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(crim ~ rad)

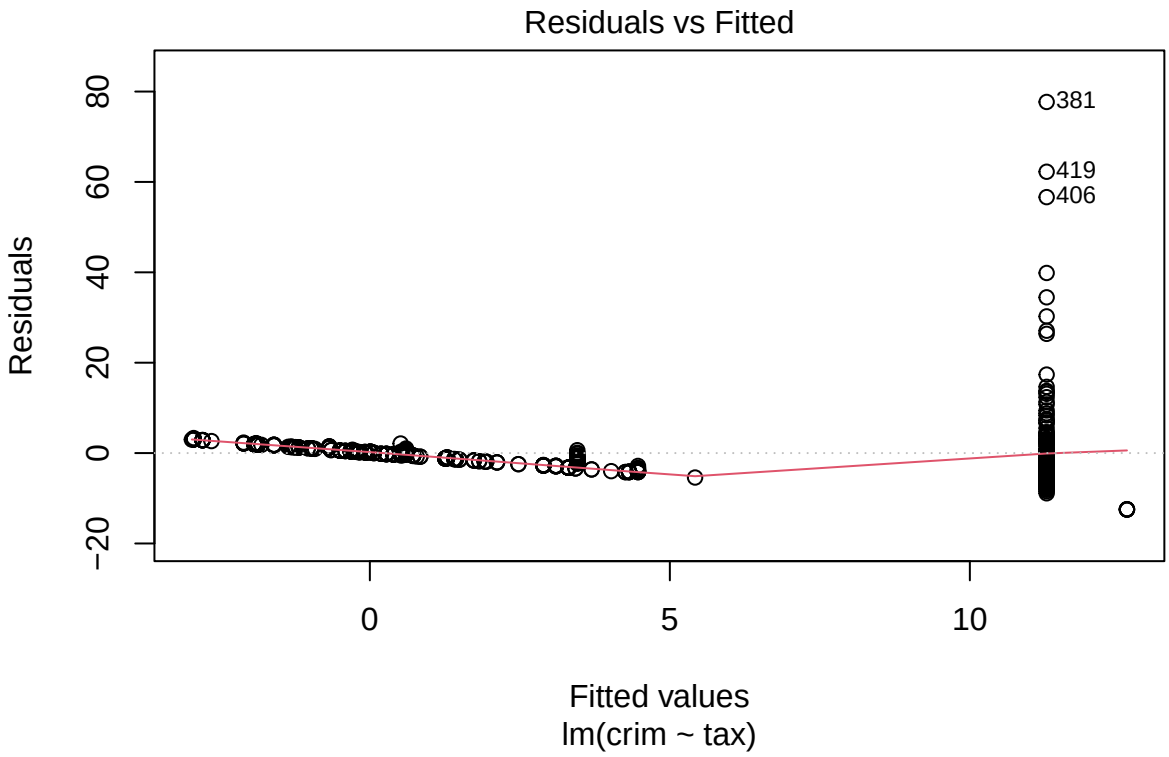

Scale−Location
√|Standardized residuals|
0.0
1.0
2.0
3.0
381
419
406
−2
0
2
4
6
8
10
12
Fitted values
lm(crim ~ rad)

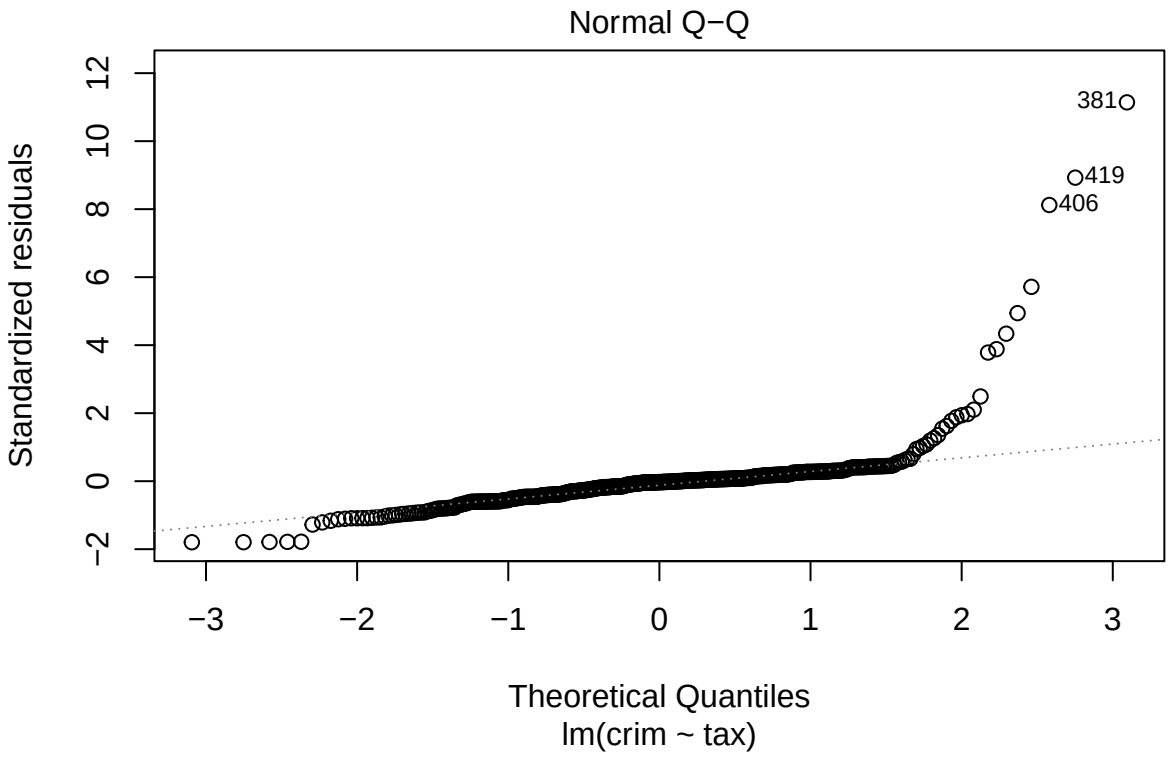


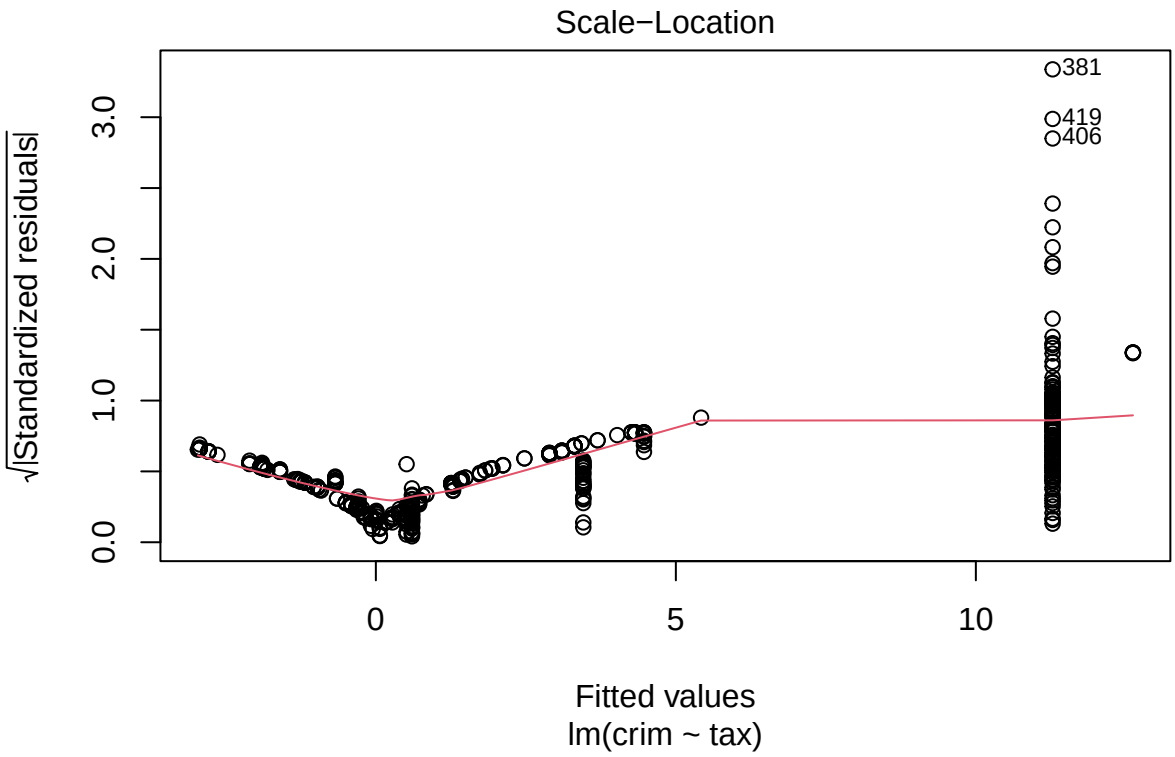
7.) p-value is < 0.05 so there is statistically significant association between crim and rad this means that changes in rad are related to changes in crim

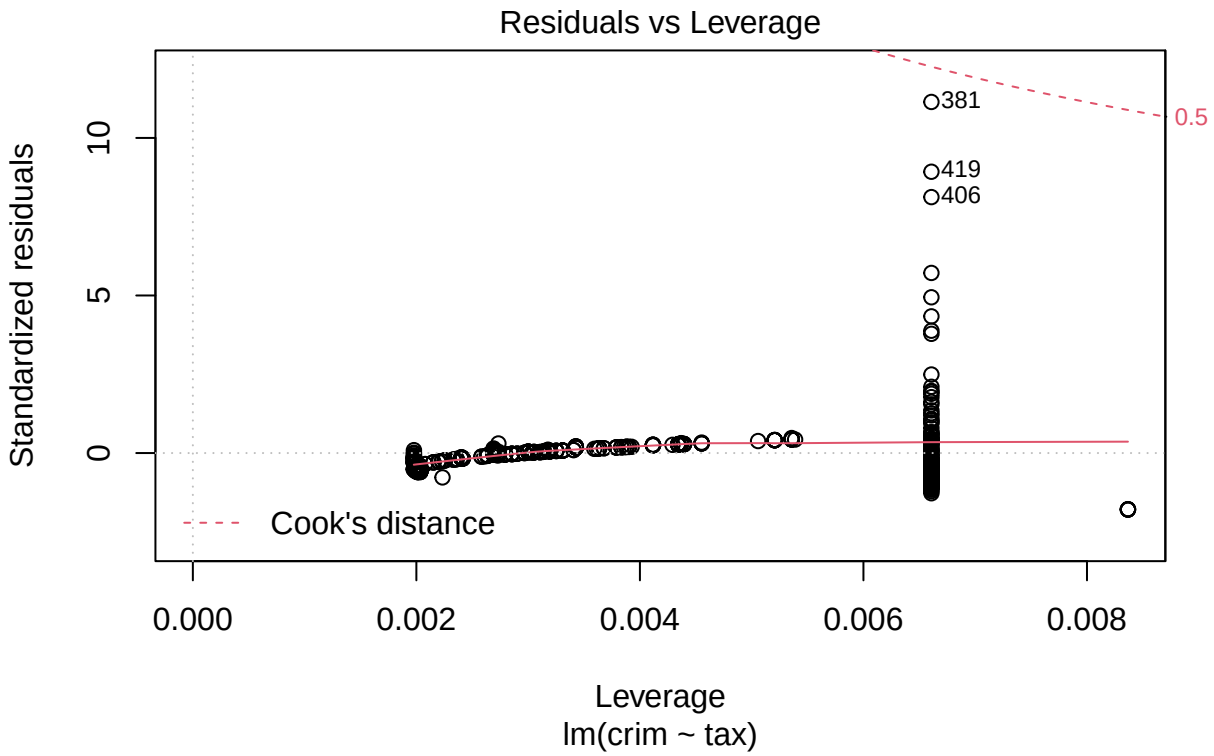
Part 8

```
##
## 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
```


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


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(crim ~ tax)
381
419
406
−2
0
2
4
6
8
10
12
−3
−2
−1
0
1
2
3

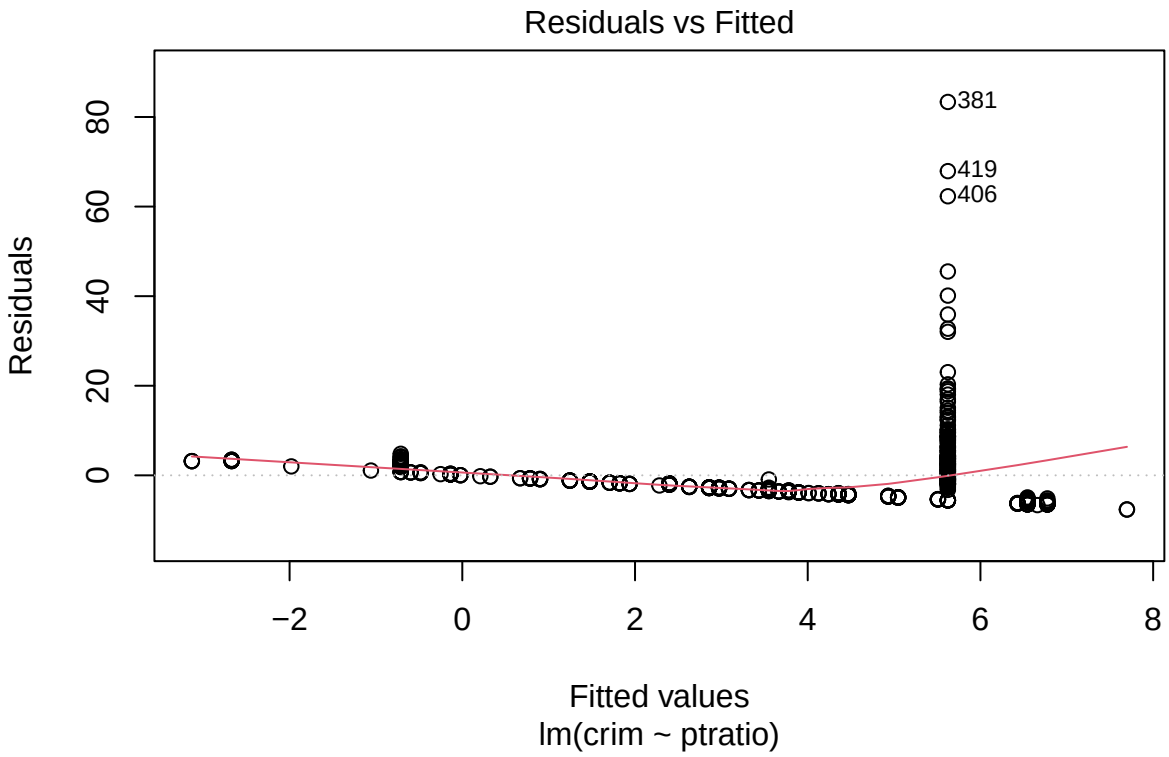

Scale−Location
√|Standardized residuals|
0.0
1.0
2.0
3.0
0
5
10
381
419
406
Fitted values
lm(crim ~ tax)

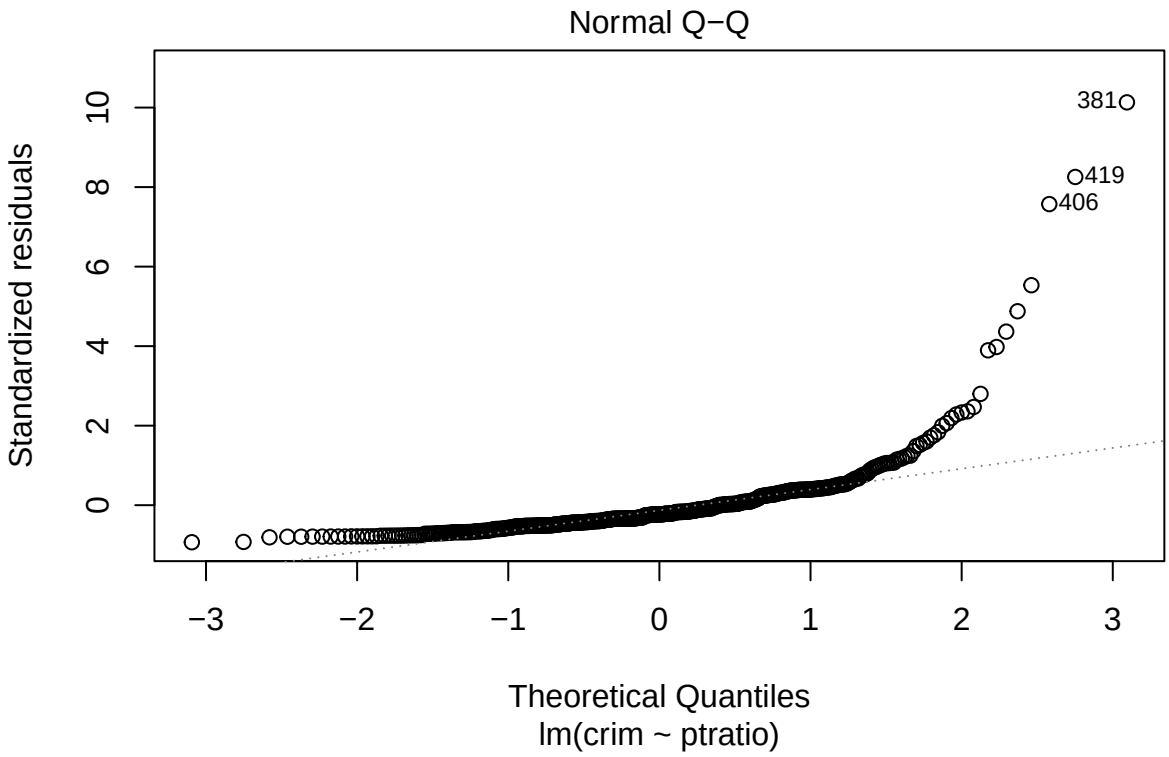


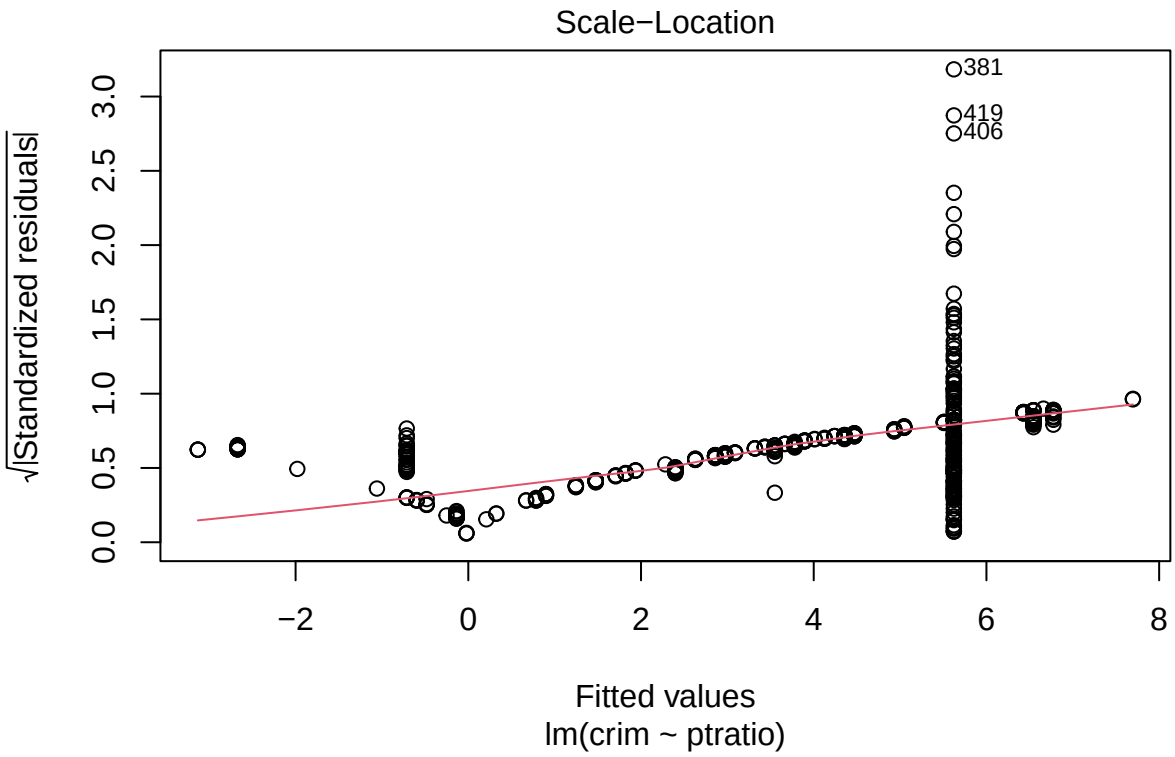
8.) p-value is < 0.05 so there is statistically significant association between crim and tax this means that changes in tax are related to changes in crim

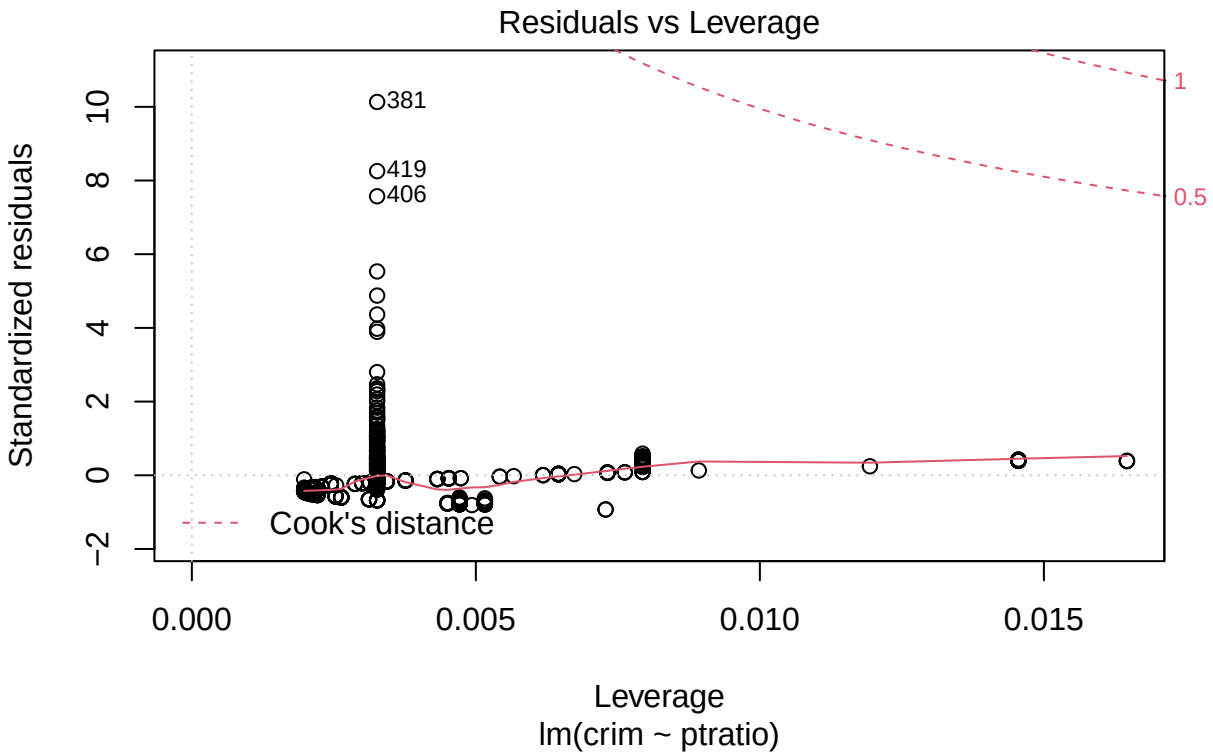
Part 9

```
##
## 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
```


Residuals vs Fitted
Residuals
0
20
40
60
80
381
419
406
−2
0
2
4
6
8
Fitted values
lm(crim ~ ptratio)


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

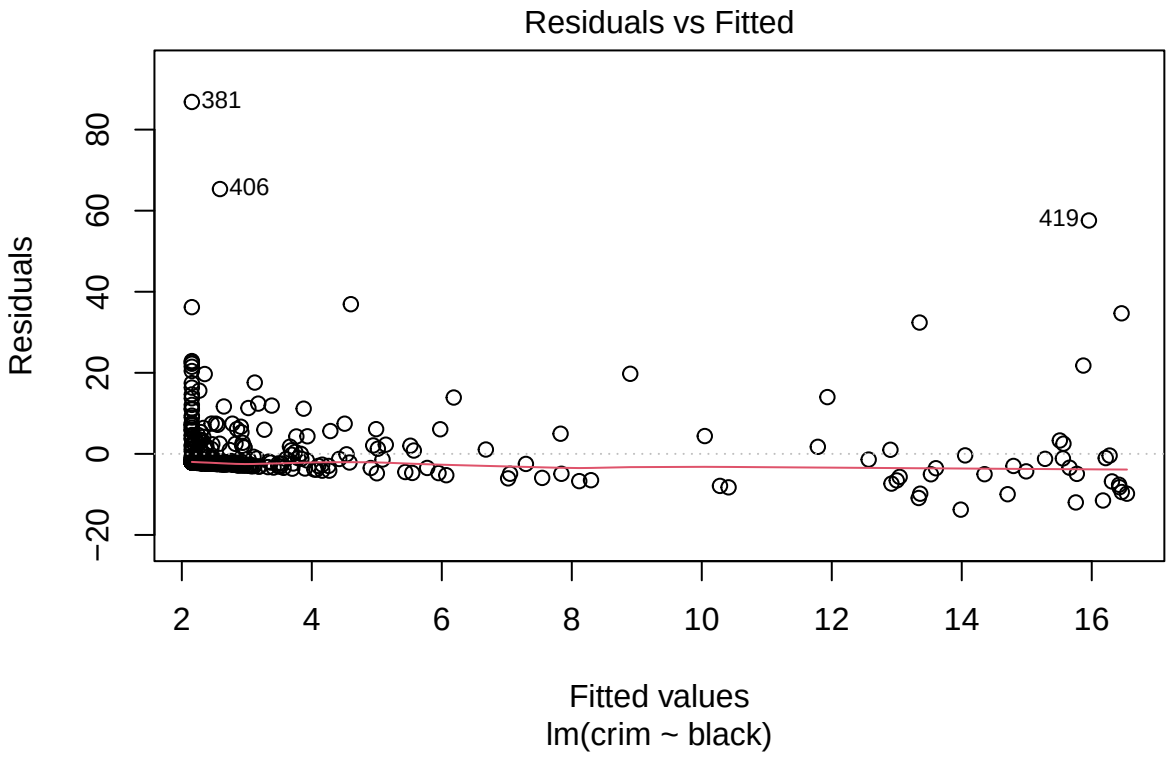

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
381
419
406
−2
0
2
4
6
8
Fitted values
lm(crim ~ ptratio)

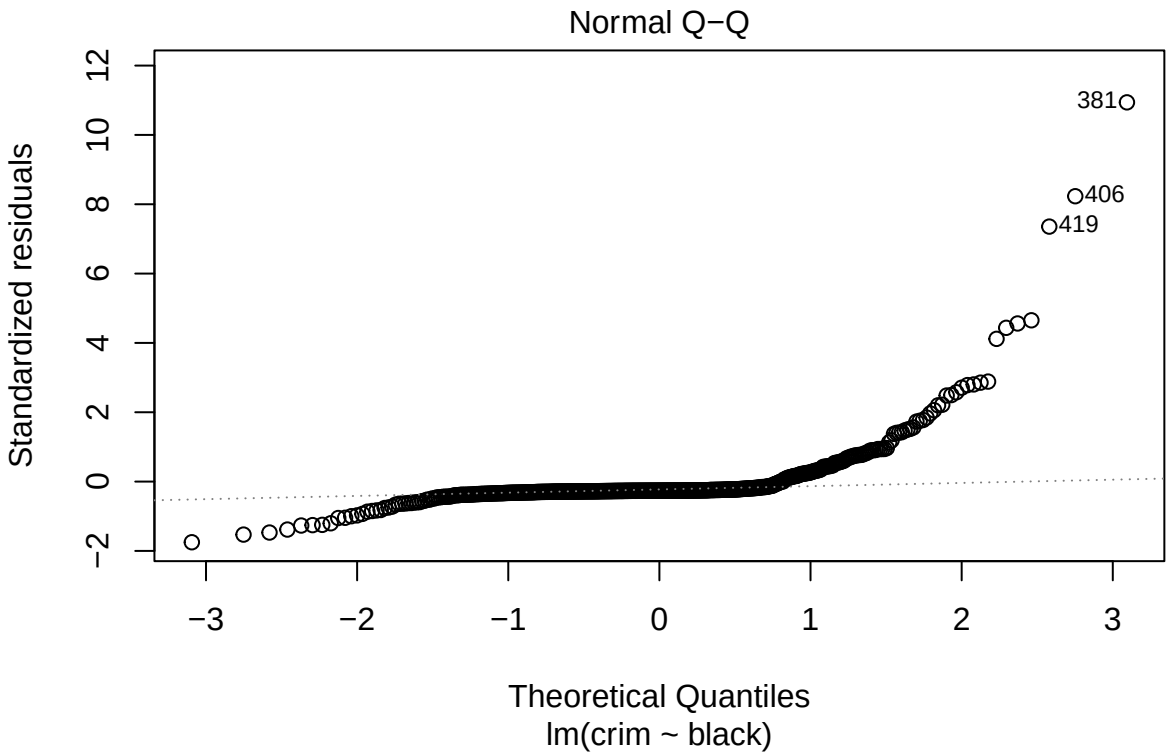


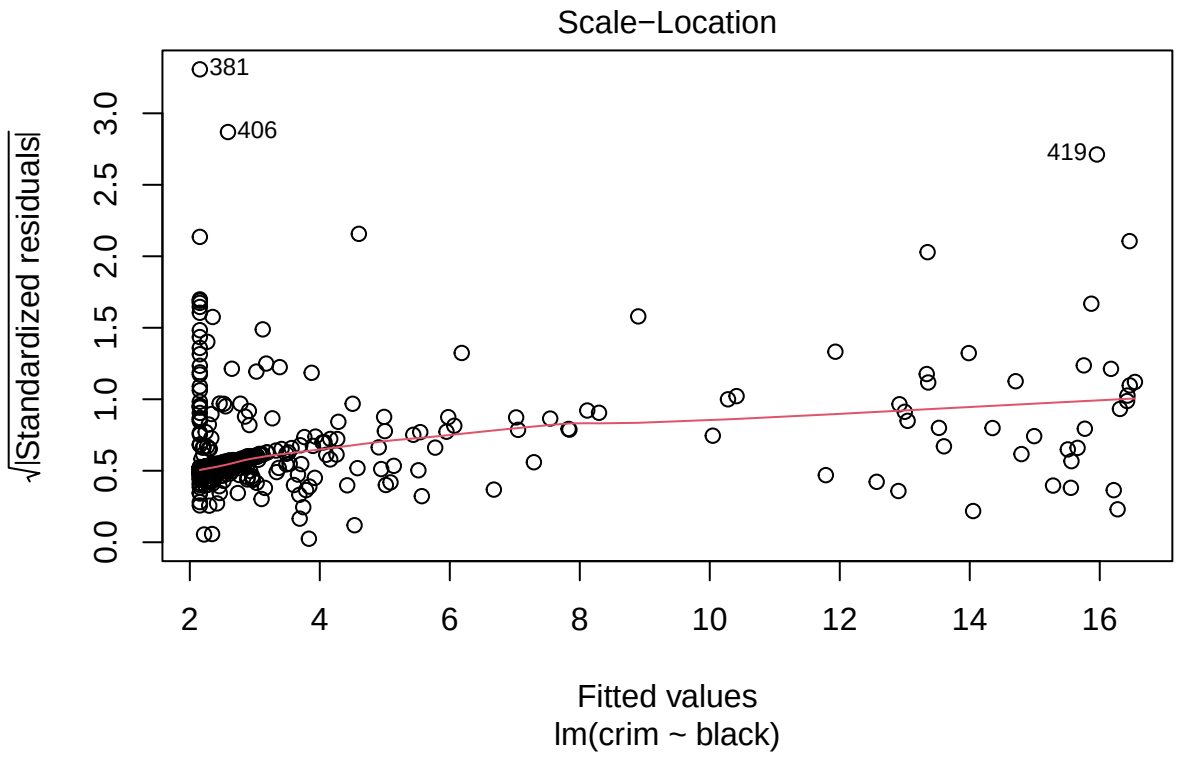
9.) p-value is < 0.05 so there is statistically significant association between crim and ptratio this means that changes in ptratio are related to changes in crim

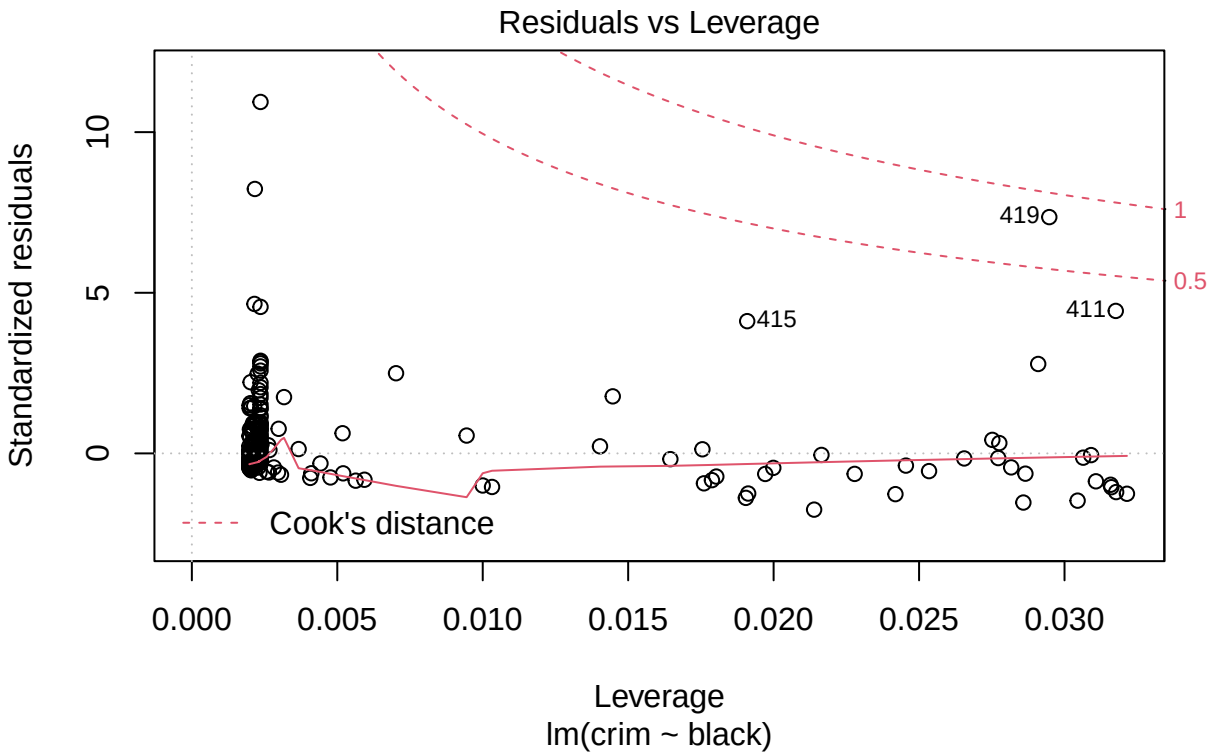
Part 10

```
##
## 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
```


Residuals vs Fitted
381
406
419
Residuals
−20
0
20
40
60
80
2
4
6
8
10
12
14
16
Fitted values
lm(crim ~ black)


Normal Q−Q
Standardized residuals
12
10
8
6
4
2
0
−2
381
406
419
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(crim ~ black)

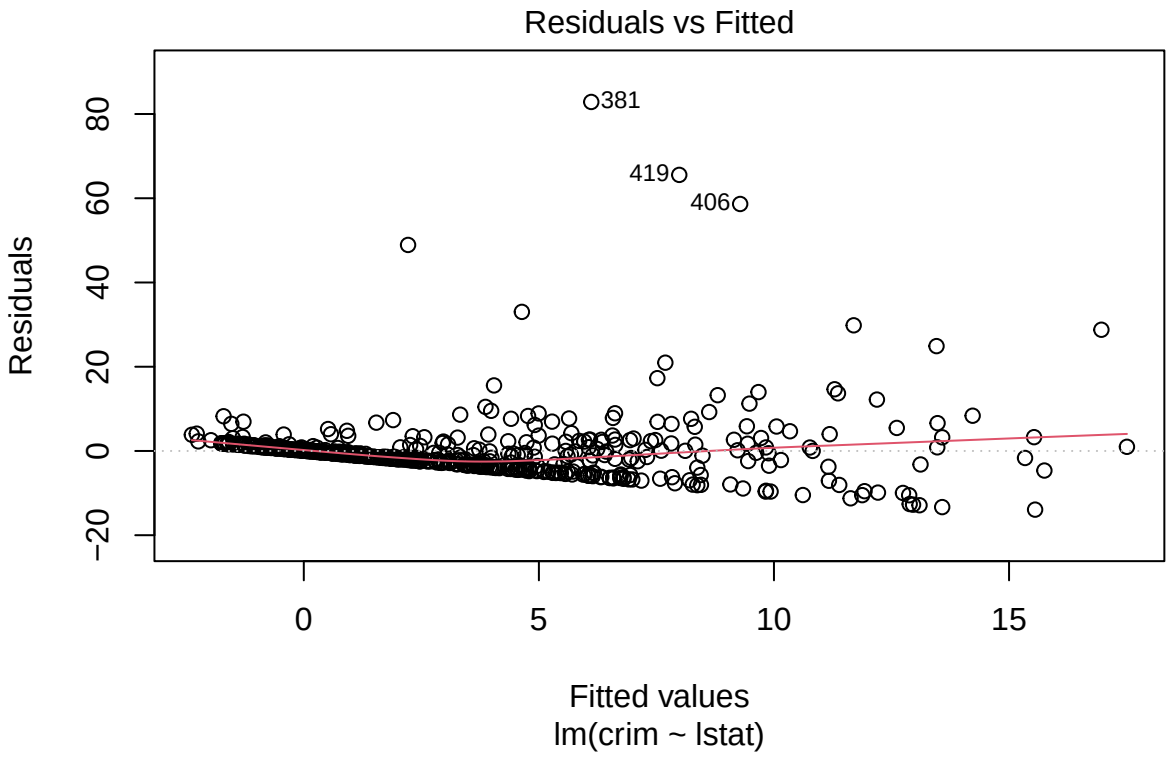

Scale−Location
381
406
419
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
2
4
6
8
10
12
14
16
Fitted values
lm(crim ~ black)

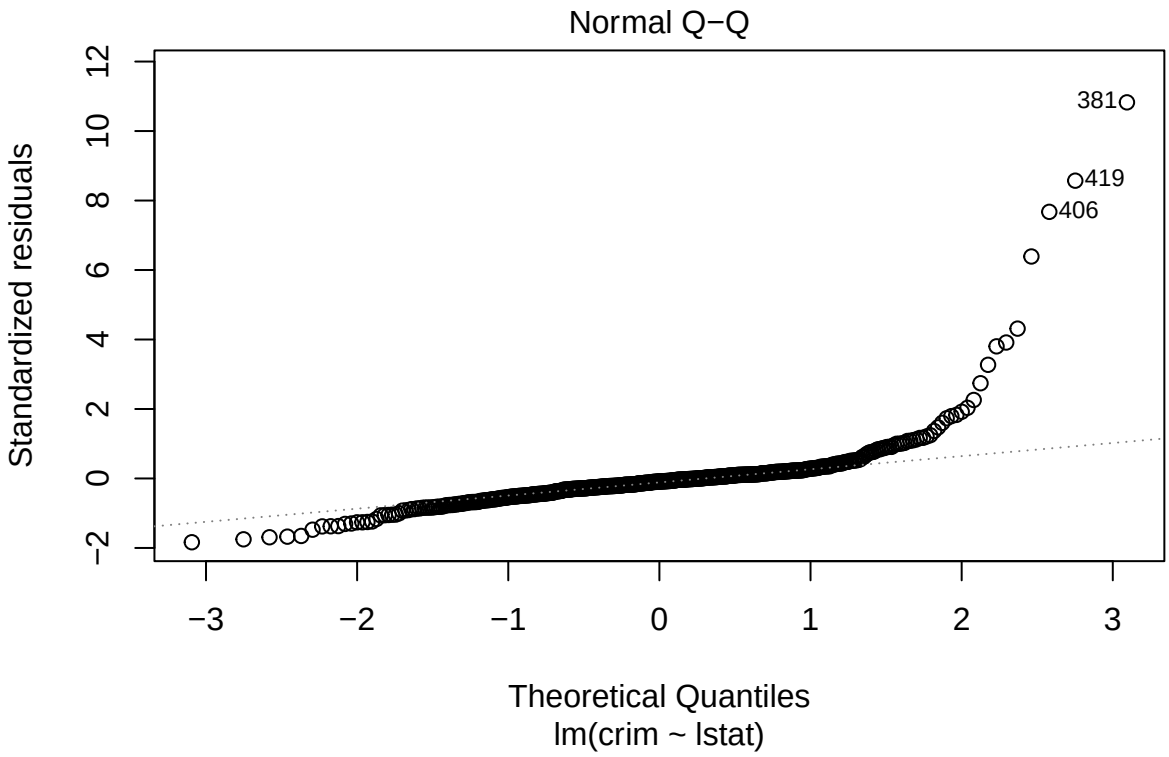


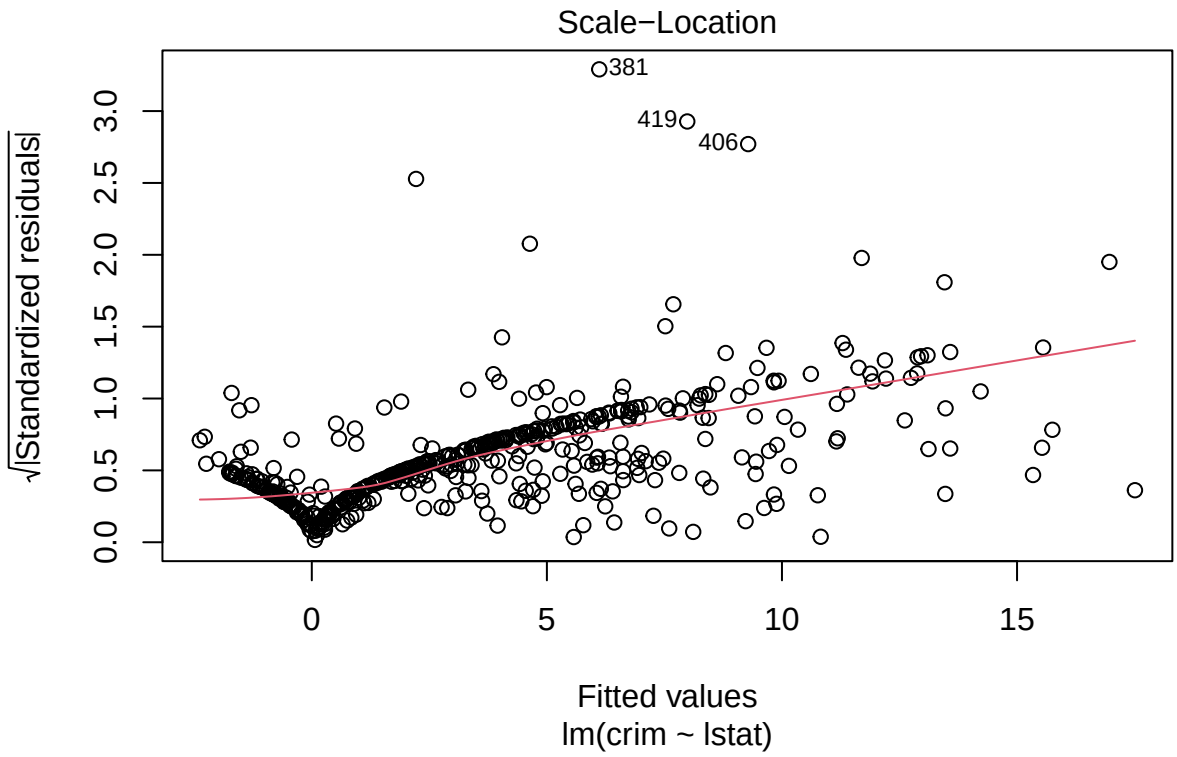
10.) p-value is < 0.05 so there is statistically significant association between crim and black this means that changes in black are related to changes in crim

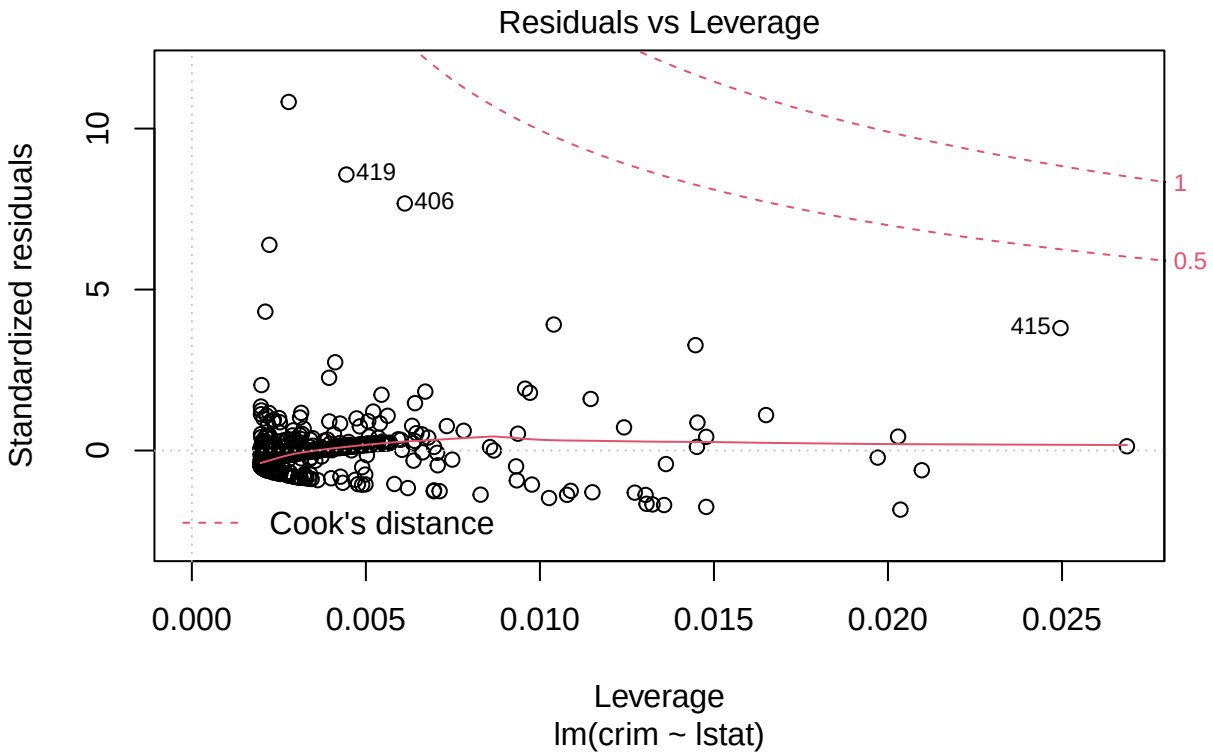
Part 11

```
##
## 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
```


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


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(crim ~ lstat)
381
419
406
−2
0
2
4
6
8
10
12
−3
−2
−1
0
1
2
3

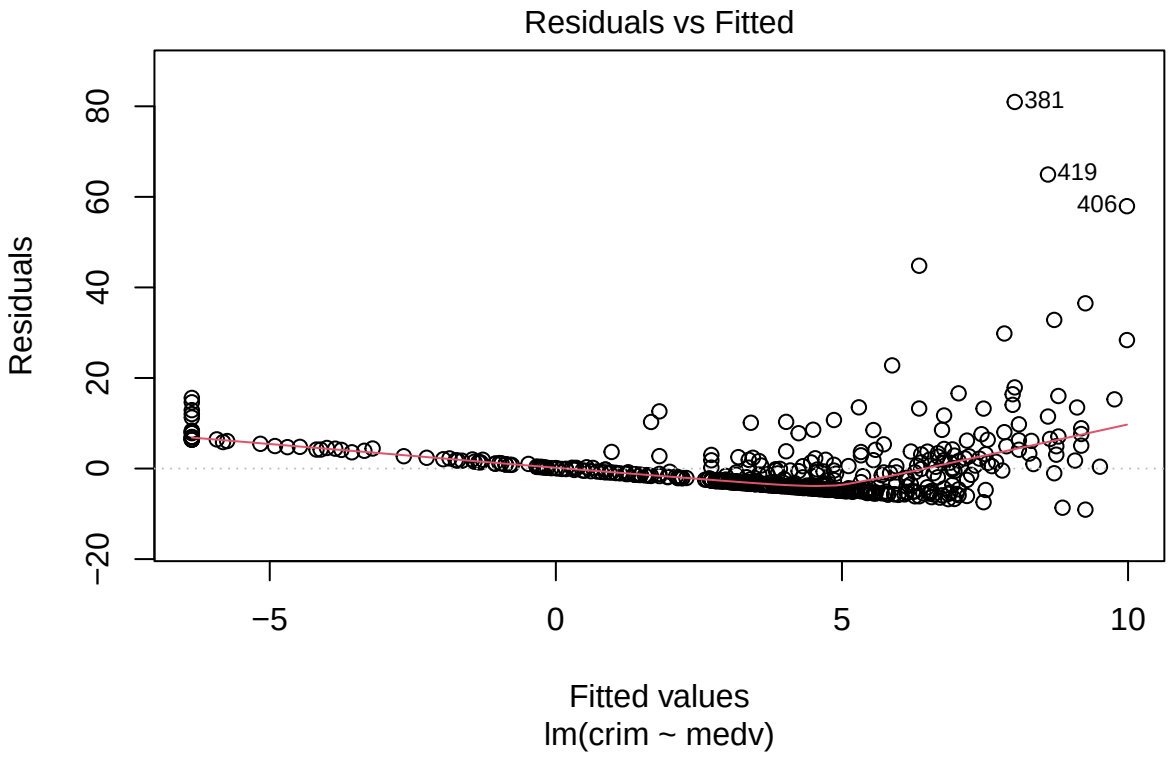

Scale−Location
381
419
406
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
5
10
15
Fitted values
lm(crim ~ lstat)

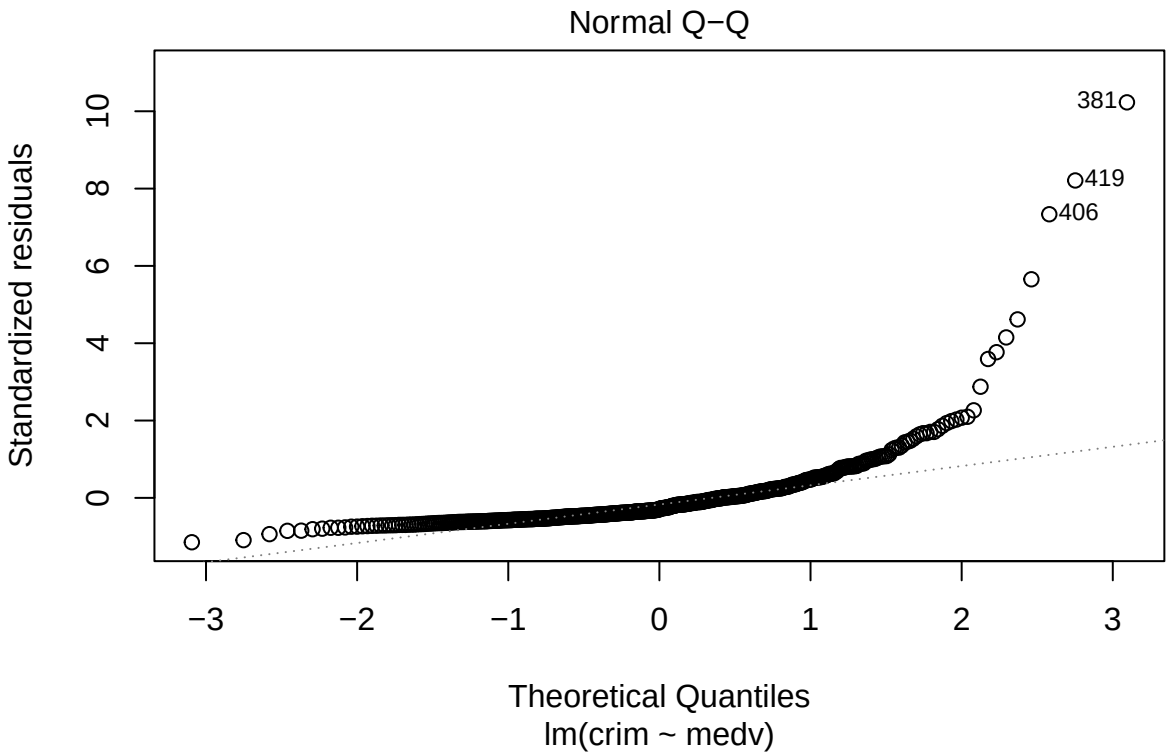


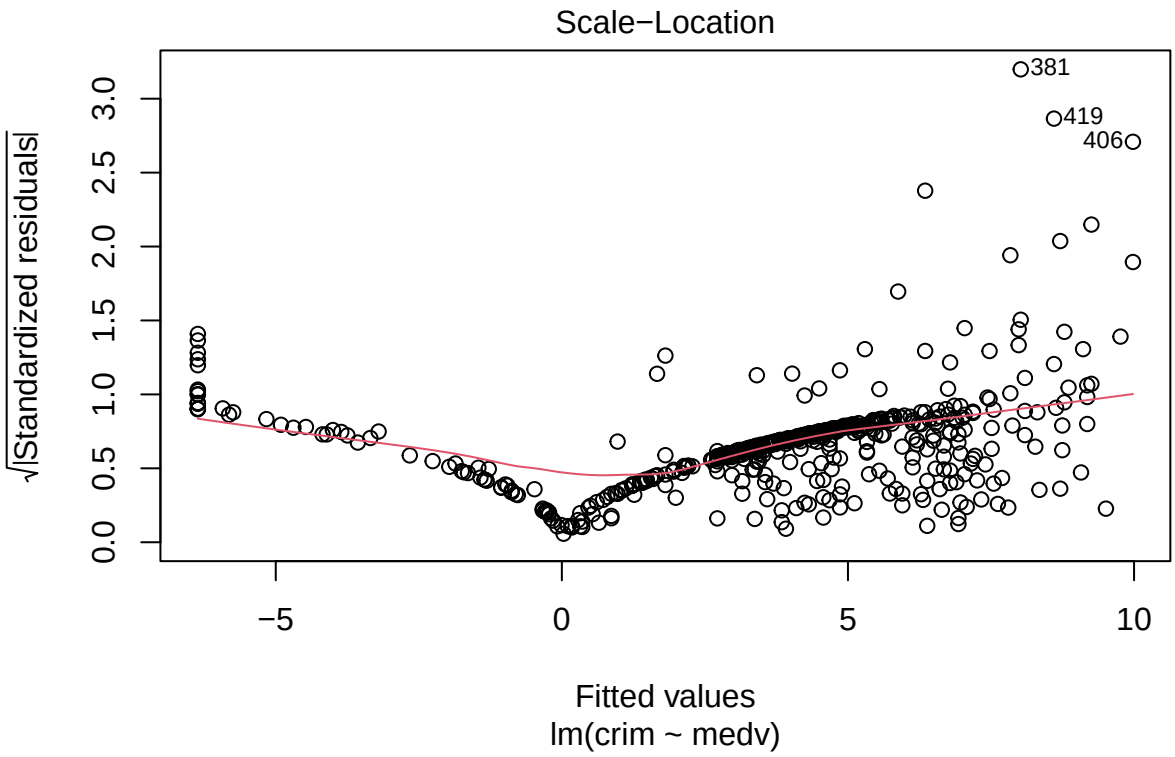
11.) p-value is < 0.05 so there is statistically significant association between crim and lstat this means that changes in lstat are related to changes in crim

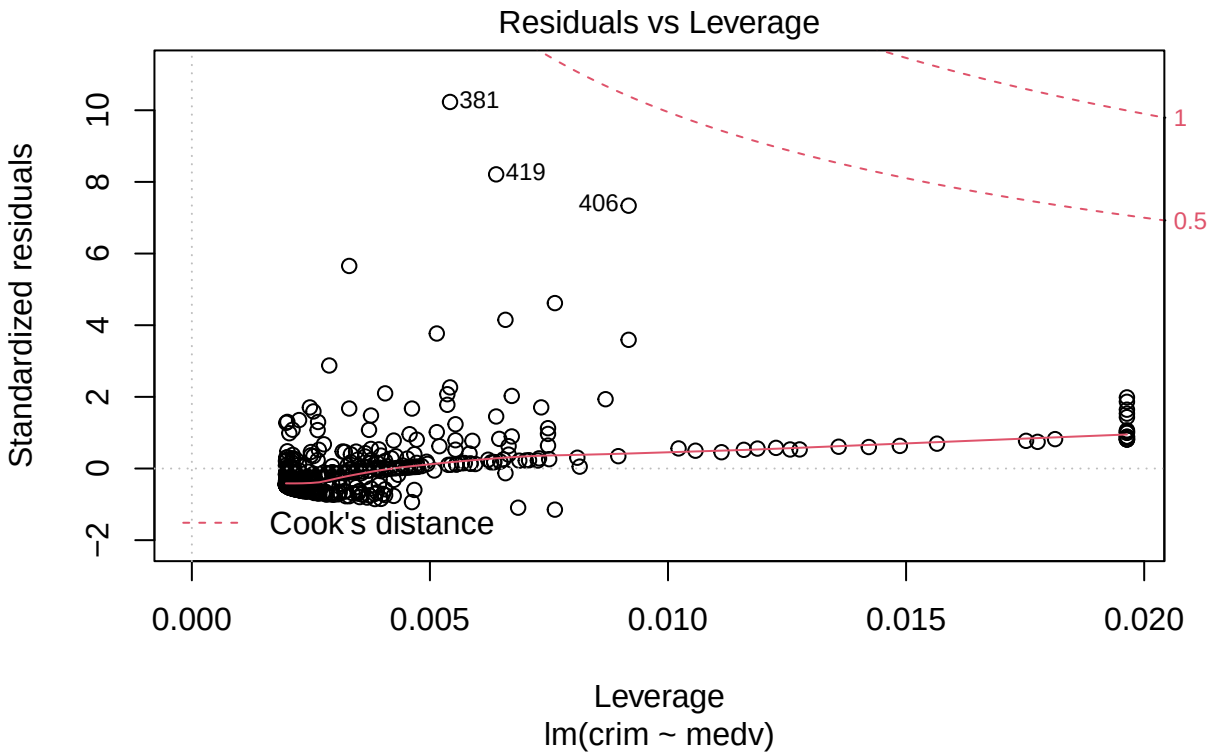
Part 12

```
##
## 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
```


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
0
2
4
6
8
10
381
419
406
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(crim ~ medv)


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



12.) p-value is < 0.05 so there is statistically significant association between crim and medv this means that changes in medv are related to changes in crim

Part A Conclusion:

Chas has a p-value greater than 0.05, so it cannot be determined to have a statistically significant association with crim. All other variables do have a statistically significant association.

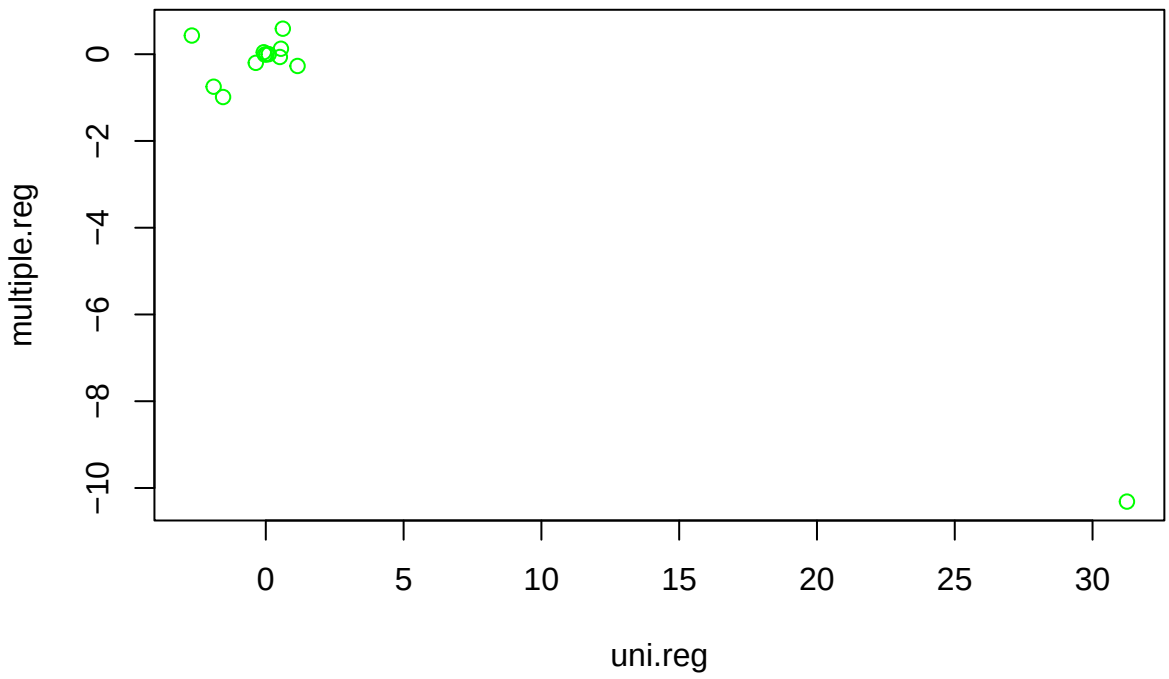
Part B

```
##
## 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
```

Reject H0 for zn (p<.1), dis (p<.001), rad (p<.001), black (p<.1), medv (p<.01)

Part C



The difference between univariate and multiple regression comes from univariate regression's slope showing how an increase in a predictor can impact the model. Multiple regression shows the impact of an increase of a predictor, all things constant, which is quite different from univariate. This means that the multiple regression model would show no connection between response and predictors, which is not the case in univariate.

Part D

```
##
## 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

##
## 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

##
## 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
```

```
##
## 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
```

```
##
## 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
```

```
##
```

```
## 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

##
## Call:
## lm(formula = rad ~ poly(rad, 3))
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -9.295e-13  1.000e-17  1.800e-15  2.100e-15  4.924e-13
##
## Coefficients:
##                 Estimate Std. Error    t value Pr(>|t|)
## (Intercept)    9.549e+00  2.088e-15  4.572e+15   <2e-16 ***
## poly(rad, 3)1  1.957e+02  4.698e-14  4.165e+15   <2e-16 ***
## poly(rad, 3)2 -1.046e-14  4.698e-14 -2.230e-01    0.824
## poly(rad, 3)3 -2.479e-14  4.698e-14 -5.280e-01    0.598
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.698e-14 on 502 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 5.783e+30 on 3 and 502 DF,  p-value: < 2.2e-16

##
## 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
```

```
##
## 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
```

```
##
## 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
```

```
##
## 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
```

```
##
## 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
```

zn, rm, rad, tax, and lstat predictors' cubic coefficients are not statistically significant as shown by their p-values indus, nox, age, dis, ptratio, and medv predictors' cubic coefficients are startically significant as shown by their p-values

Chapter 6, Quesiton 9

Part A

```
## [1] 388  18
```

```
## [1] 389  18
```

Part B:

```
## [1] 0.0612919
```

Test error obtained is 0.0612919

Part C:

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-2
```

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

```
## [1] 0.07042784
```

best = 0.01 mean = 0.07042784

Part D:

```
## [1] 0.0231013
```

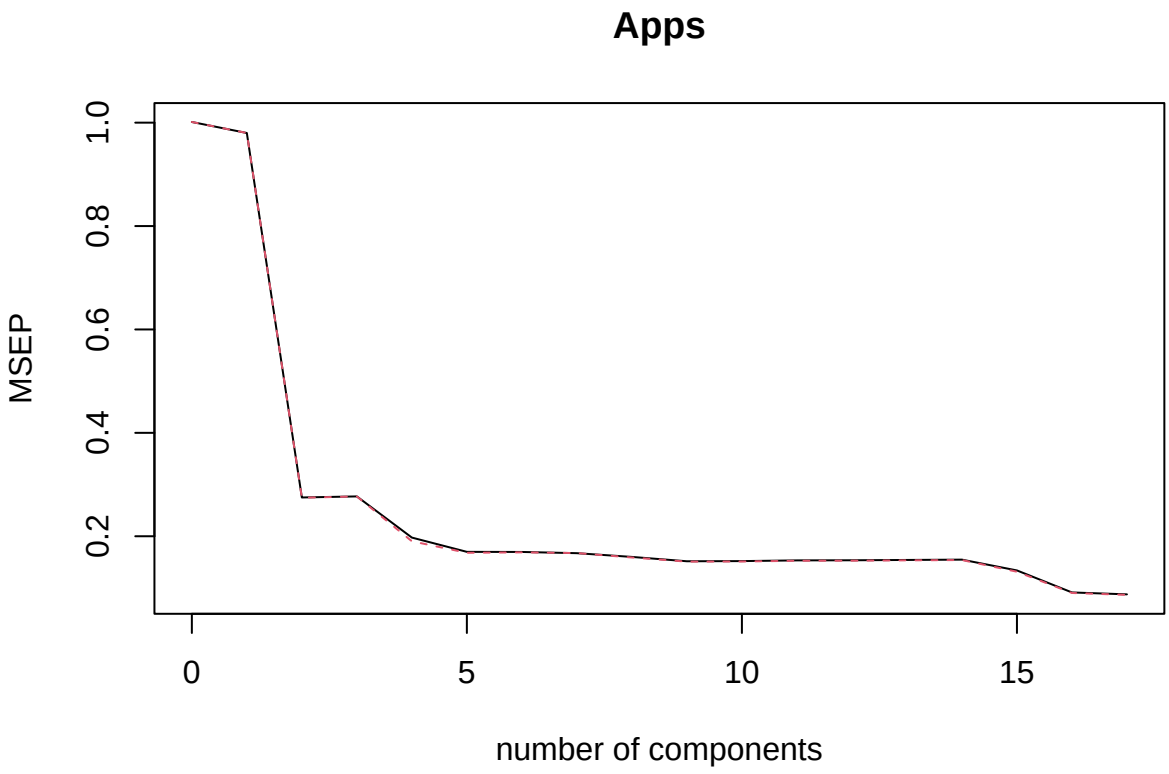
```
## [1] 0.07616939
```

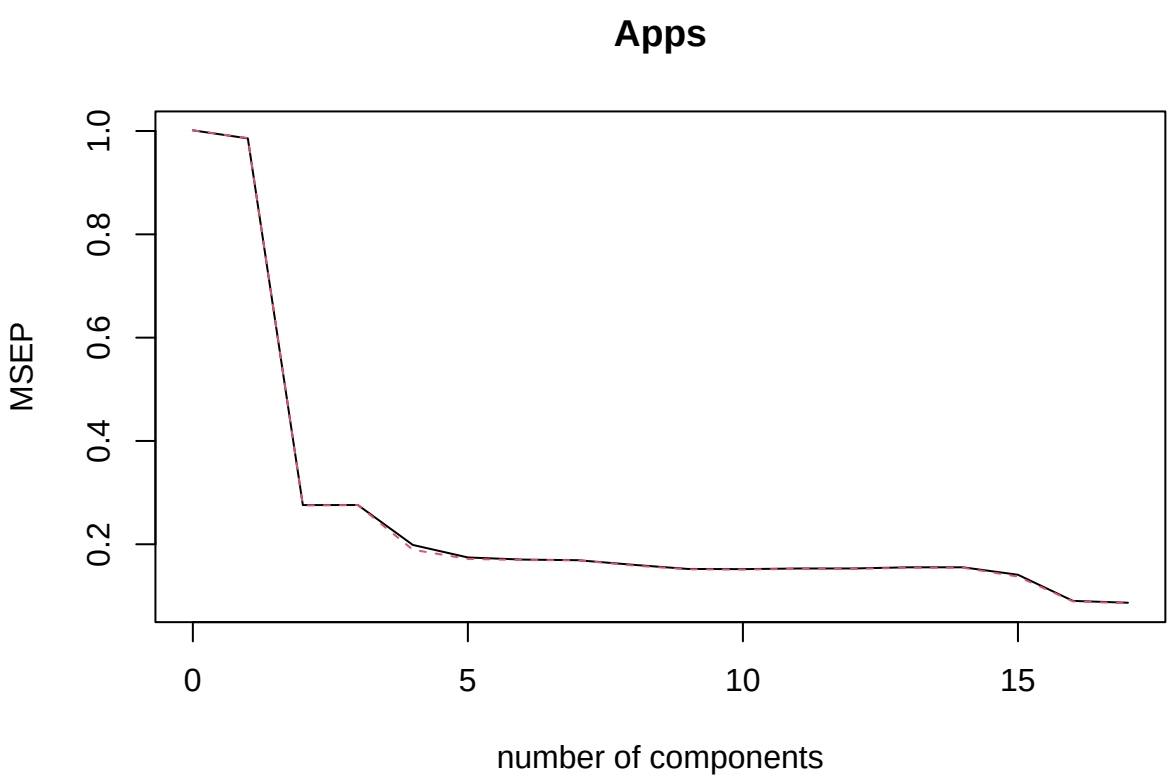
```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                         s1
## (Intercept) -1.494915e-02
## PrivateNo    5.479006e-02
## PrivateYes  -4.293745e-14
## Accept       8.799119e-01
## Enroll       .
## Top10perc    1.110172e-01
## Top25perc    .
## F.Undergrad  .
## P.Undergrad  .
## Outstate     .
## Room.Board   .
## Books        .
## Personal     .
## PhD          .
## Terminal     .
## S.F.Ratio    .
## perc.alumni  .
## Expend       5.967475e-02
## Grad.Rate    .
```

best = 0.01 mean = 0.06997521 There are 11 non-zero coefficients

Part E:

```
##
## Attaching package: 'pls'

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

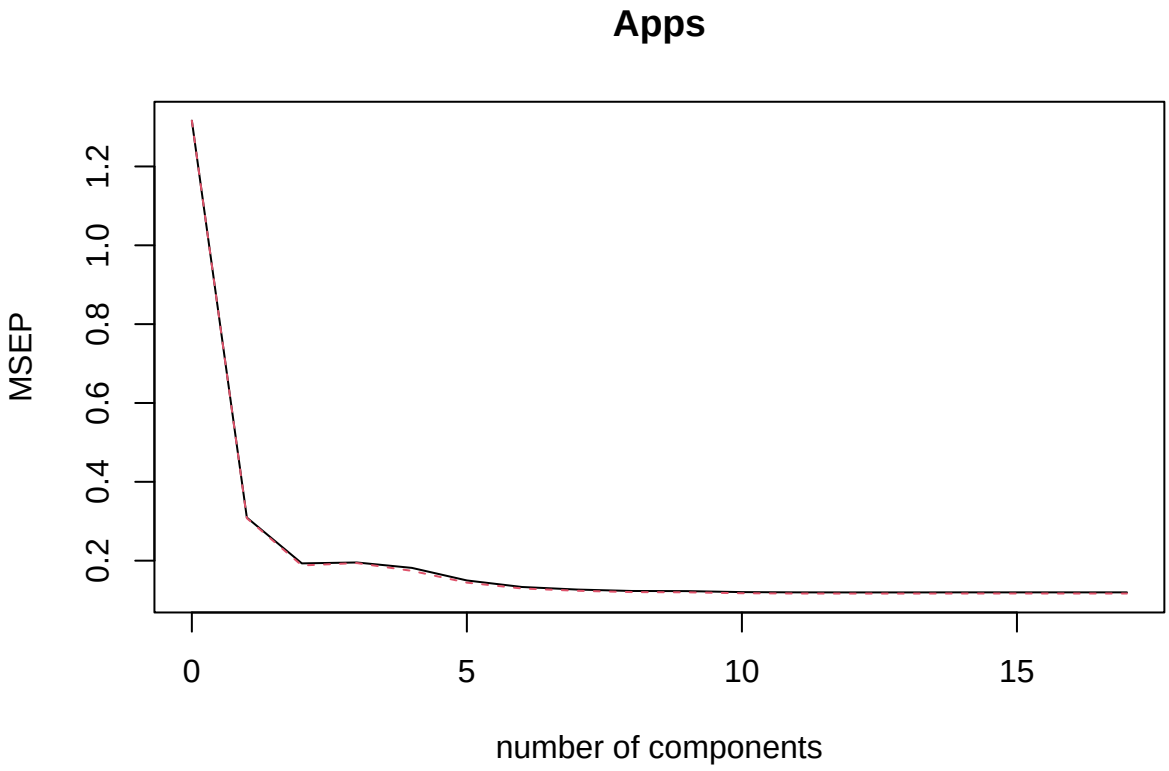





```
## [1] 0.06348735
```

0.06348735

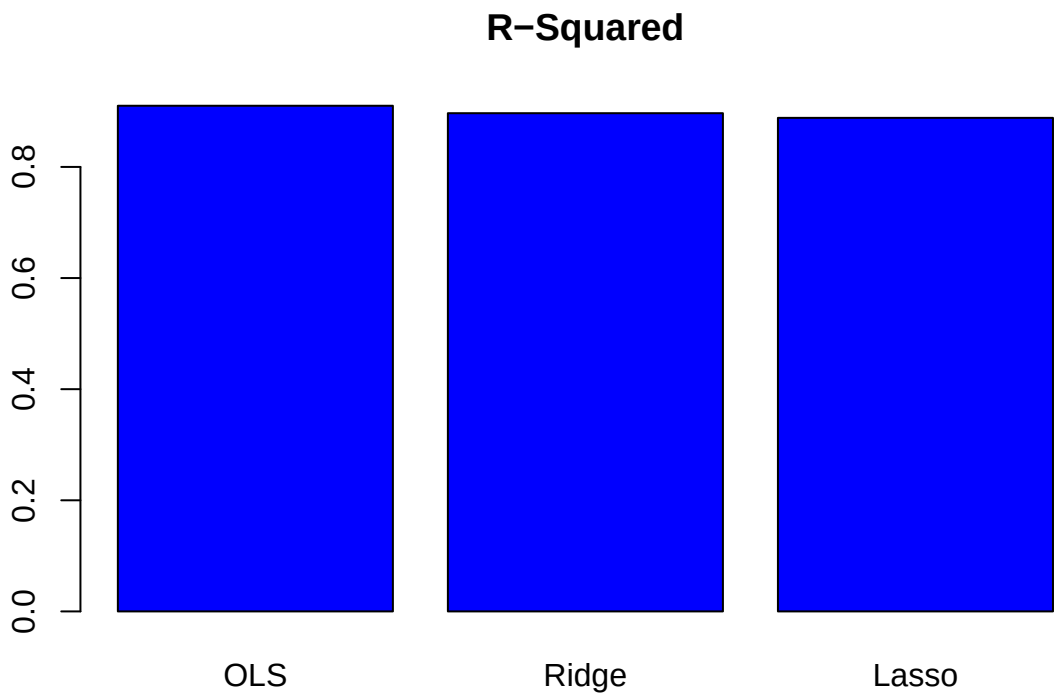
Part F:




```
## [1] 0.07142963
```

.07142963

Part G:



All of the models have high R^2 and are relatively similar, meaning they are likely not too different.

Chapter 6, Question 11

Part A:

```
## The following object is masked _by_ .GlobalEnv:
##
##     chas

## The following objects are masked from Boston (pos = 8):
##
##     age, black, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad,
##     rm, tax, zn

## [1] 506  14

## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
```

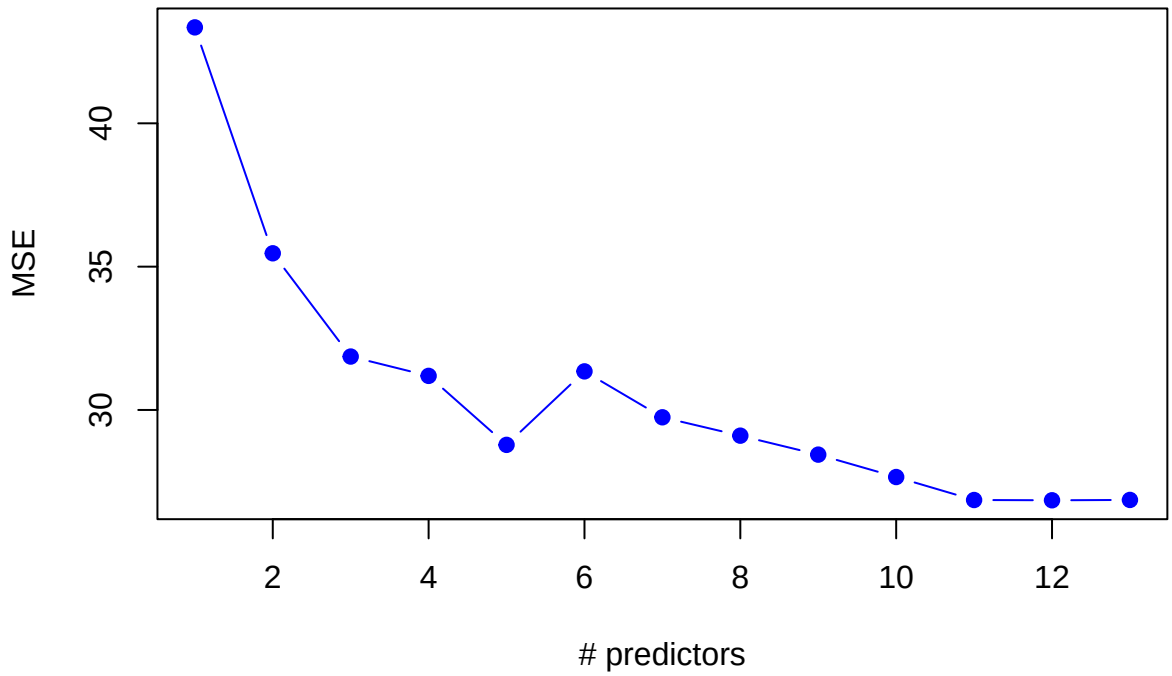
```
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ black  : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

```
##       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
```

Subset Selection:

```
## Subset selection object
## Call: regsubsets.formula(medv ~ ., data = train, nbest = 1, nvmax = 13)
## 13 Variables  (and intercept)
##         Forced in Forced out
## crim        FALSE      FALSE
## zn          FALSE      FALSE
## indus       FALSE      FALSE
## chas        FALSE      FALSE
## nox         FALSE      FALSE
## rm          FALSE      FALSE
## age         FALSE      FALSE
```

```
## dis         FALSE      FALSE
## rad         FALSE      FALSE
## tax         FALSE      FALSE
## ptratio     FALSE      FALSE
## black       FALSE      FALSE
## lstat       FALSE      FALSE
## 1 subsets of each size up to 13
## Selection Algorithm: exhaustive
##           crim zn  indus chas nox rm  age dis rad tax ptratio black lstat
## 1  ( 1 )  " "  " " " " " "   " "  " " " " " " " " " " " "     " "   "*"
## 2  ( 1 )  " "  " " " " " "   " "  " " "*" " " " " " " " " "     " "   "*"
## 3  ( 1 )  " "  " " " " " "   " "  " " "*" " " " " " " " " "*"     " "   "*"
## 4  ( 1 )  " "  " " " " " "   "*"  " " "*" " " " " " " " " "*"     " "   "*"
## 5  ( 1 )  " "  " " " " " "   " "  "*" "*" " " " " "*" " " " " "*"     " "   "*"
## 6  ( 1 )  " "  " " " " " "   " "  " " "*" " " " " "*" "*" "*" "*"     " "   "*"
## 7  ( 1 )  " "  " " " " " "   " "  "*" "*" " " " " "*" "*" "*" "*"     " "   "*"
## 8  ( 1 )  "*"  " " " " " "   " "  "*" "*" " " " " "*" "*" "*" "*"     " "   "*"
## 9  ( 1 )  "*"  " " " " " "   "*"  "*" "*" " " " " "*" "*" "*" "*"     " "   "*"
## 10  ( 1 ) "*"  " " " " " "   "*"  "*" "*" " " " " "*" "*" "*" "*"     "*"   "*"
## 11  ( 1 ) "*"  "*" " " " "   "*"  "*" "*" " " " " "*" "*" "*" "*"     "*"   "*"
## 12  ( 1 ) "*"  "*" "*" "*"   "*"  "*" "*" " " " " "*" "*" "*" "*"     "*"   "*"
## 13  ( 1 ) "*"  "*" "*" "*"   "*"  "*" "*" "*" "*" "*" "*" "*" "*"     "*"   "*"
```



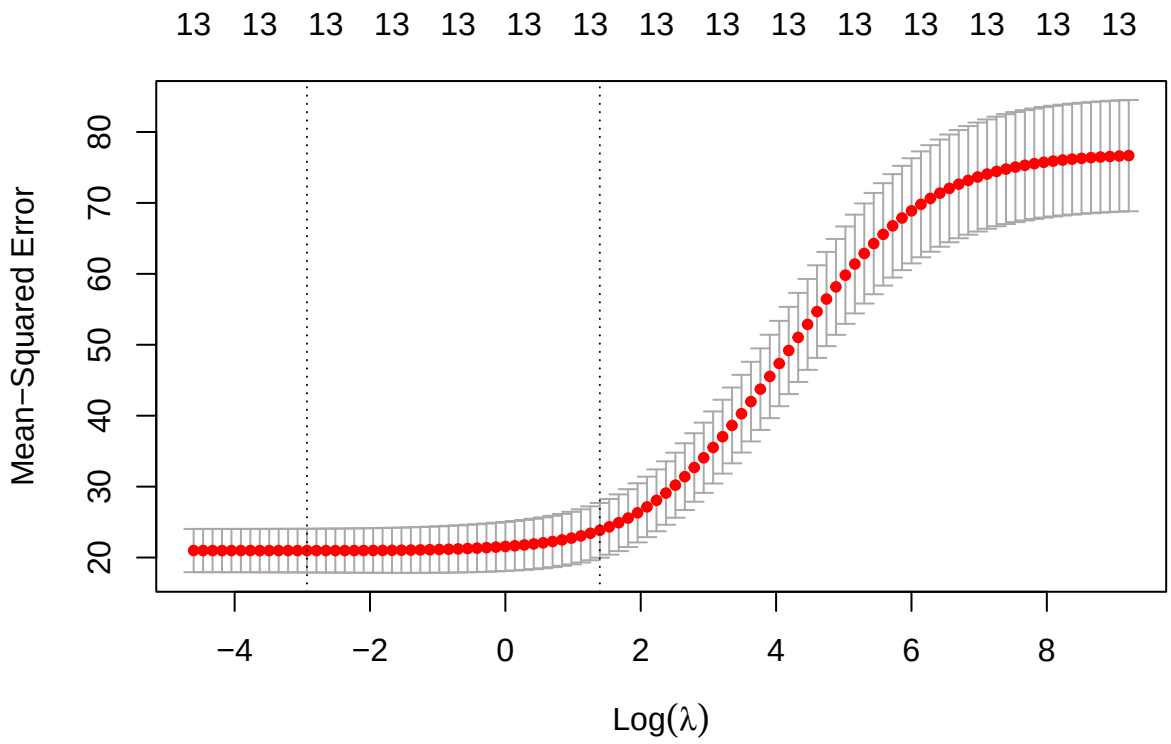

```
## [1] 12
```

```
##   (Intercept)          crim            zn         indus          chas
```

```
##  34.831120307  -0.091918101   0.030237800   0.033374023   2.295987544
##           nox            rm           dis           rad           tax
## -13.811099542   4.079128964  -1.244699757   0.382084606  -0.019216474
##       ptratio         black         lstat
##  -0.985922901   0.006773852  -0.492341180
```

```
## [1] 26.84946
```

which.min(errors) = 12 best_subset_mse = 26.84946 To determine the optimal model, test set MSE is determined for each. The model with the lowest MSE is selected, which in this case is the predictor value of 11



Ridge:

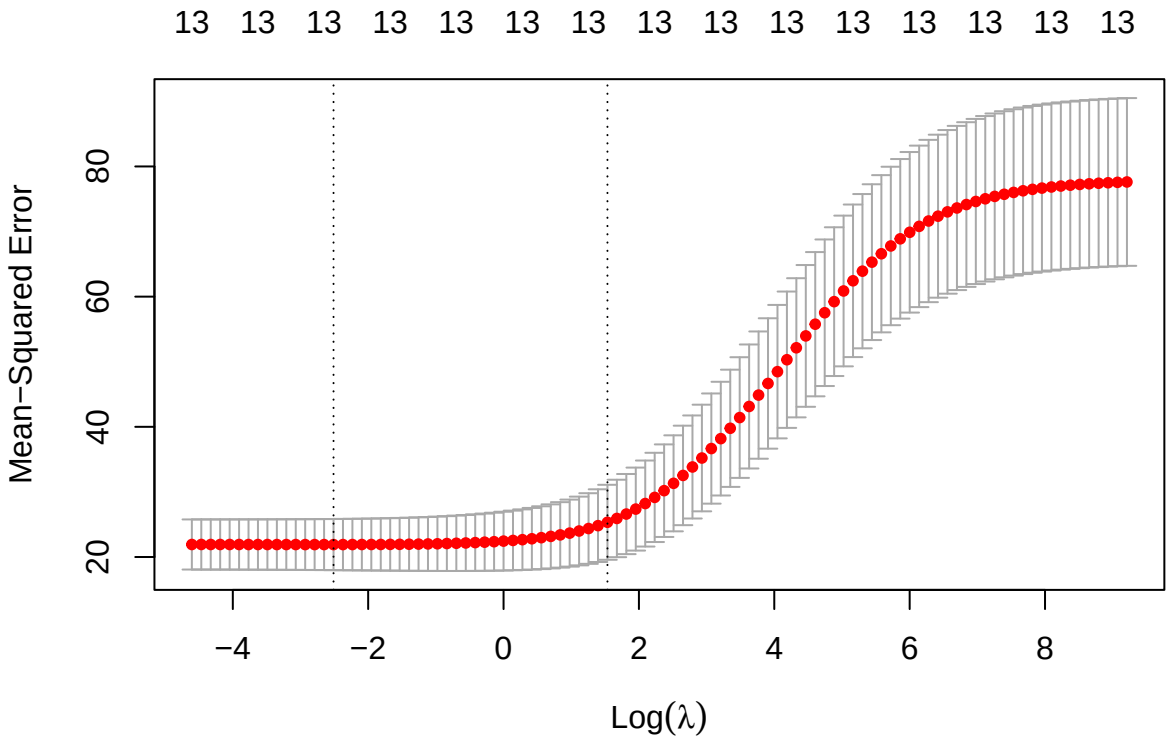
```
## [1] 0.05336699
```

```
## 15 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept)  33.549938944
## (Intercept)   .
## crim         -0.089333575
## zn            0.028630555
## indus         0.022489614
## chas          2.372039925
## nox         -13.053954757
## rm            4.138827718
## age          -0.004384557
## dis          -1.236522044
## rad           0.347615984
```

```
## tax          -0.017518196
## ptratio      -0.965923891
## black         0.006772119
## lstat        -0.481050642
```

```
## [1] 26.79522
```

best.ridge = 0.05336699 MSE.ridge = 26.79522 Ridge regression with minimum MSE is what the final model and prediction is built on. The lambda that had the lowest MSE is 0.05336699 ### Lasso:




```
## [1] 0.08111308
```

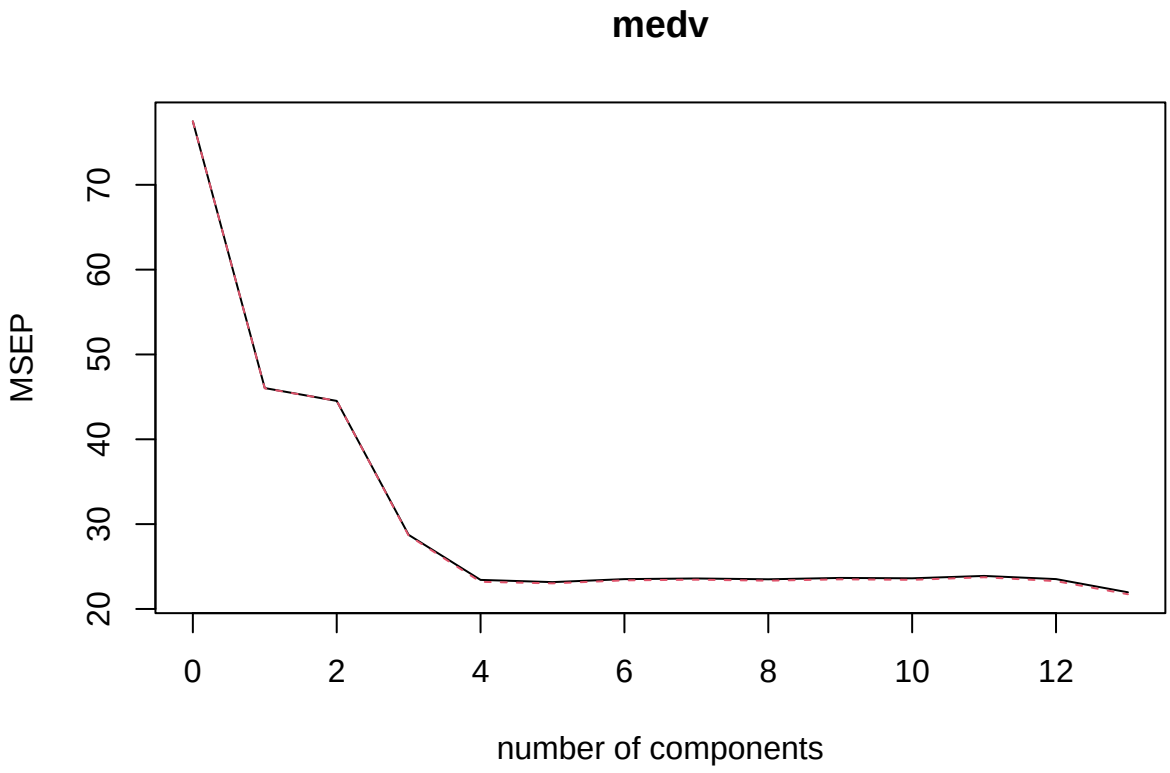
```
## 15 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept)  33.024094010
## (Intercept)   .
## crim         -0.088159911
## zn            0.027992223
## indus         0.018467094
## chas          2.400885843
## nox         -12.807002261
## rm            4.155104180
## age          -0.004805140
## dis          -1.221950655
## rad           0.333681186
## tax          -0.016820543
```

```
## ptratio      -0.959305356
## black         0.006760755
## lstat        -0.478261284
```

```
## [1] 26.77717
```

best.lasso = 0.08111308 MSE.lasso = 26.77717 Lasso regression with minimum MSE is what the final model and prediction is built on. The lambda that had the lowest MSE is 0.08111308 ### PCR:

```
## Data:    X dimension: 253 13
##  Y dimension: 253 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           8.802    6.785    6.673    5.361    4.840    4.813    4.849
## adjCV        8.802    6.783    6.674    5.352    4.817    4.797    4.834
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       4.857    4.847    4.864     4.860     4.888     4.850     4.686
## adjCV    4.842    4.830    4.847     4.841     4.872     4.825     4.662
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X       48.02    58.80    67.99    75.11    80.85    85.73    89.55    92.97
## medv    40.83    43.75    64.55    72.78    72.81    72.85    72.88    73.28
##       9 comps  10 comps  11 comps  12 comps  13 comps
## X       95.09     96.92     98.29     99.50    100.00
## medv    73.43     73.77     73.78     75.04     76.83
```




```
## Data:    X dimension: 253 13
##  Y dimension: 253 1
## Fit method: svdpc
## Number of components considered: 5
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps
## X       48.02    58.80    67.99    75.11    80.85
## medv    40.83    43.75    64.55    72.78    72.81
```

```
## , , 1 comps
##
##                medv
## crim    -0.58901583
## zn       0.55290872
## indus   -0.75658758
## chas     0.04169713
## nox     -0.77186413
## rm       0.42824335
## age     -0.69998703
## dis      0.72258349
## rad     -0.71757547
## tax     -0.75690381
## ptratio -0.48362619
## black    0.48103143
## lstat   -0.68991293
```

```
##
## , , 2 comps
##
##               medv
## crim    -1.0327137
## zn       0.1183425
## indus   -0.5461255
## chas     0.5908909
## nox     -0.5331747
## rm       0.4320955
## age     -0.3325371
## dis      0.3178432
## rad     -1.0960500
## tax     -1.0634332
## ptratio -0.7103027
## black    0.9657038
## lstat   -0.6915826
##
## , , 3 comps
##
##                medv
## crim    -0.41021087
## zn       1.06227200
## indus   -0.44361778
## chas     2.25645994
## nox      0.04055328
## rm       2.48038051
## age     -0.31224947
## dis     -0.15731819
## rad     -0.18301087
## tax     -0.34805561
## ptratio -2.08747479
## black    0.25720650
## lstat   -1.63011220
##
## , , 4 comps
##
##                medv
## crim    -0.94766845
## zn       0.14082804
## indus   -0.37620155
## chas     0.97649527
## nox      0.02713317
## rm       3.94426169
## age     -0.14133317
## dis     -0.64803172
## rad      0.03854496
## tax     -0.18770194
## ptratio -1.34190432
## black    0.43537465
## lstat   -2.62947540
##
## , , 5 comps
##
```

```
##                medv
## crim    -0.936993259
## zn       0.161504585
## indus   -0.364860941
## chas     0.864967231
## nox      0.063304433
## rm       3.952029786
## age     -0.105517898
## dis     -0.675079851
## rad     -0.005581025
## tax     -0.214616117
## ptratio -1.471135730
## black    0.405716567
## lstat   -2.596174401
```

```
## [1] 30.28912
```

MSE.pcr = 30.28912 5 as number of components since 5 components account for 80% of the variation

Part B:

The MSEs I obtained in these models were: Subset - 26.84946 Ridge - 26.79522 Lasso - 26.77717 PCR - 30.28912 Therefore, I would recommend using the lasso model as it has the lowest MSE of all the models

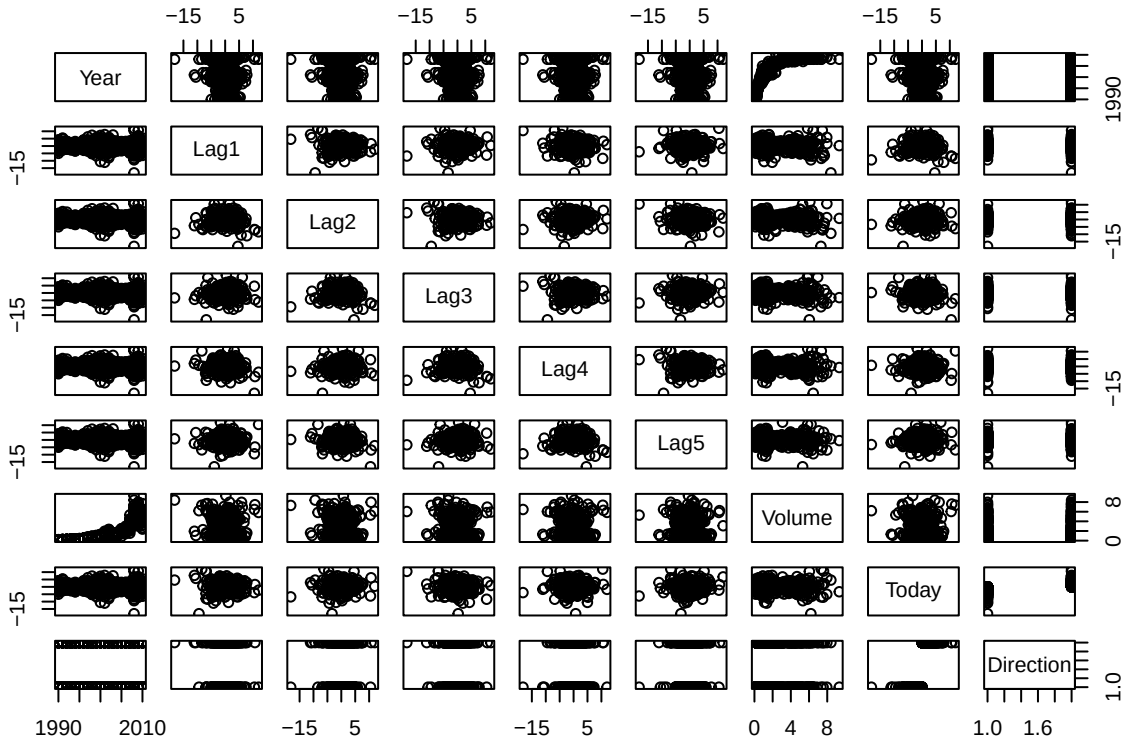
Part C:

The lasso models has all predictor variables Therefore, all of the predictor variables seem to play a role in the prediction of the response variable

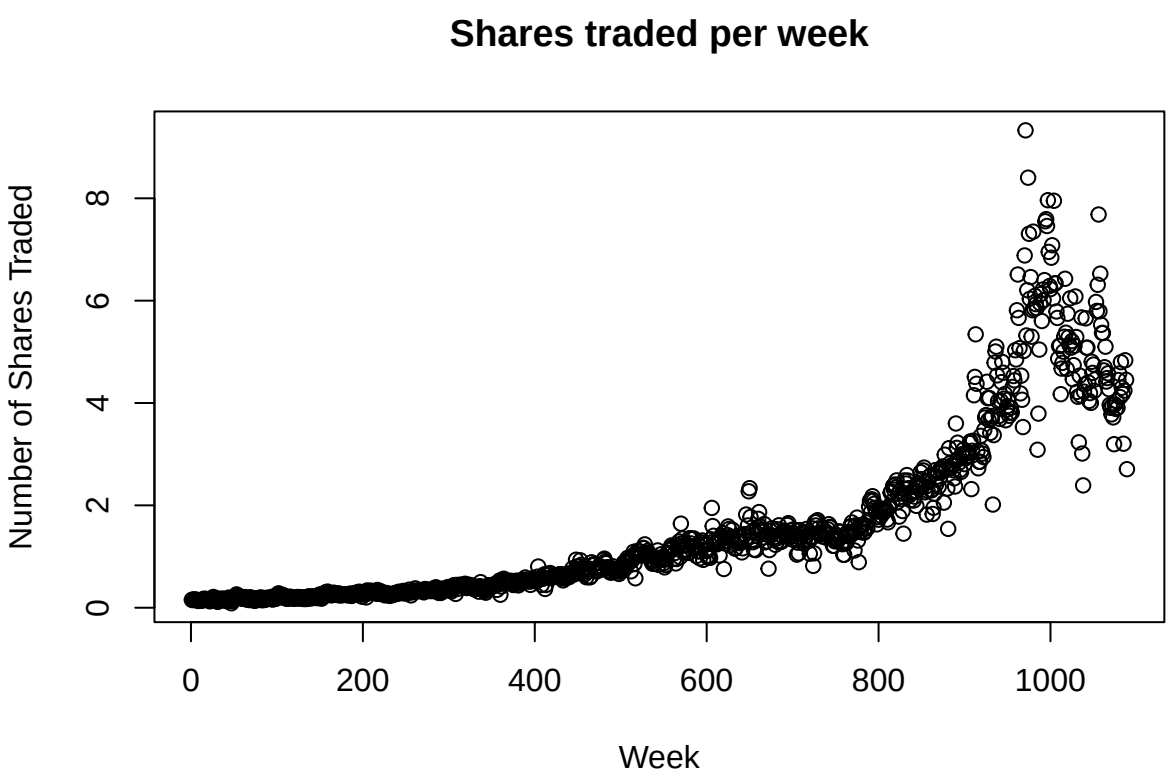
Chapter 4, Question 10:

Part A:

```
##               Year         Lag1        Lag2        Lag3         Lag4
## Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923
## Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876
## Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535
## Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865
## Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000
## Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.075675027
## Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617
## Today  -0.03245989 -0.075031842  0.05916672 -0.07124364 -0.007825873
##                Lag5      Volume        Today
## Year   -0.030519101  0.84194162 -0.032459894
## Lag1   -0.008183096 -0.06495131 -0.075031842
## Lag2   -0.072499482 -0.08551314  0.059166717
## Lag3    0.060657175 -0.06928771 -0.071243639
## Lag4   -0.075675027 -0.06107462 -0.007825873
## Lag5    1.000000000 -0.05851741  0.011012698
## Volume -0.058517414  1.00000000 -0.033077783
## Today   0.011012698 -0.03307778  1.000000000
```




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



Little correlation between the Lag variables and returns, somewhat strong correlation between Volume and Year Up to this point there does not seem to be patterns except for a high correlation between trading volume and years Up until 2008 there was a steady growth for the amount of shares traded each week, but this then decreased in 2008

Part B:

```
##
## Call:
## glm(formula = Direction ~ +Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

Lag 2 has a 0.0296 p-value, which suggests that at a significance level of 5% there is enough evidence to reject H0 This means there is no relation between Lag 2 and Direction

Part C:

```
##
## glm.predict Down  Up
##        Down   54  48
##        Up    430 557
```

```
## [1] 0.5610652
```

```
## [1] 0.4389348
```

Predictions were correct 54/484 down weeks and 557/605 up weeks In total the predictions were correct 56.11% of the times Down predictions were accurate only 54/484 weeks, which is relatively low in accuracy Whereas Up predictions were accurate 557/605 weeks, which is relatively high in accuracy There is a training error rate of 43.9%

Part D:

```
##
## Call:
## glm(formula = Direction ~ Lag2, family = "binomial", data = Weekly,
##     subset = ptd.train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.536  -1.264   1.021   1.091   1.368
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.20326    0.06428   3.162  0.00157 **
## Lag2         0.05810    0.02870   2.024  0.04298 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1354.7  on 984  degrees of freedom
## Residual deviance: 1350.5  on 983  degrees of freedom
```

```
## AIC: 1354.5
##
## Number of Fisher Scoring iterations: 4
```

```
##
## glm.predict Down Up
##        Down    9  5
##        Up     34 56
```

```
## [1] 0.625
```

```
## [1] 0.5865385
```

This model accurately predicted 62.5% of the weeks tested

Part G:

```
##
## knn.pred Down Up
##     Down   21 30
##     Up     22 31
```

```
## [1] 0.5
```

The prediction accuracy using KNN method is 50%

Part H:

Logistic Regression had a mean success rate of 56.1% whereas KNN had a mean success rate of 50%, so Logistic Regression is slightly superior.

Part I:

Part 1:

```
##
## Call:
## glm(formula = Direction ~ +Lag1 + Lag2 + Lag3 + Volume, family = binomial,
##     data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6247  -1.2623   0.9975   1.0844   1.4673
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.25420    0.08509   2.987  0.00281 **
## Lag1        -0.03898    0.02632  -1.481  0.13867
## Lag2         0.05795    0.02671   2.170  0.03003 *
## Lag3        -0.01431    0.02638  -0.542  0.58763
```

```
## Volume      -0.01928    0.03673  -0.525  0.59956
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1487.7  on 1084  degrees of freedom
## AIC: 1497.7
##
## Number of Fisher Scoring iterations: 4
```

```
##
## glm.predict1 Down  Up
##         Down  484 605
```

```
## [1] 0.4444444
```

```
## [1] 0.5555556
```

Mean success rate of 44.44%, which is lower than the original calculation including all Lags

Part 2:

```
##
## knn.pred Down Up
##     Down   16 20
##     Up     27 41
```

```
## [1] 0.5480769
```

Mean success rate 54.81%

Part 3:

```
##
## knn.pred Down Up
##     Down   16 21
##     Up     27 40
```

```
## [1] 0.5384615
```

Mean success rate 53.85%

Part 4:

```
##
## knn.pred Down Up
##     Down   17 20
##     Up     26 41
```

```
## [1] 0.5576923
```

Mean success rate 55.77%

Part I Conclusion:

Lag 2 had had the highest accuracy among the variables. The logistic regression method had the highest level of accuracy, as did its confusion matrix.

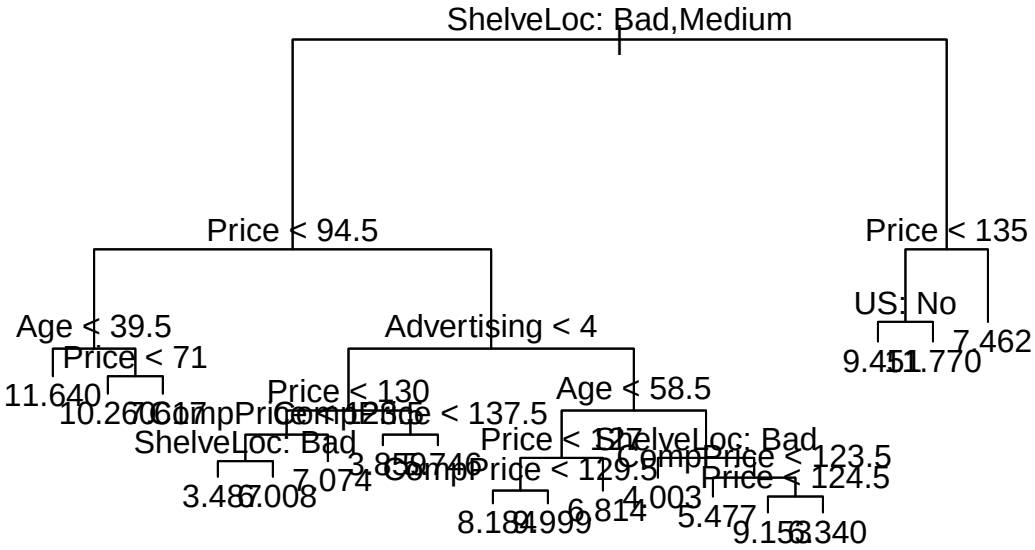
Chapter 8, Question 8:

Part A:

```
## [1] 200  11
```

```
## [1] 200  11
```

Part B:




```
##
## Regression tree:
## tree(formula = Sales ~ ., data = carseats_train)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "Age"         "Advertising" "CompPrice"
## [6] "US"
## Number of terminal nodes:  18
## Residual mean deviance:  2.167 = 394.3 / 182
```

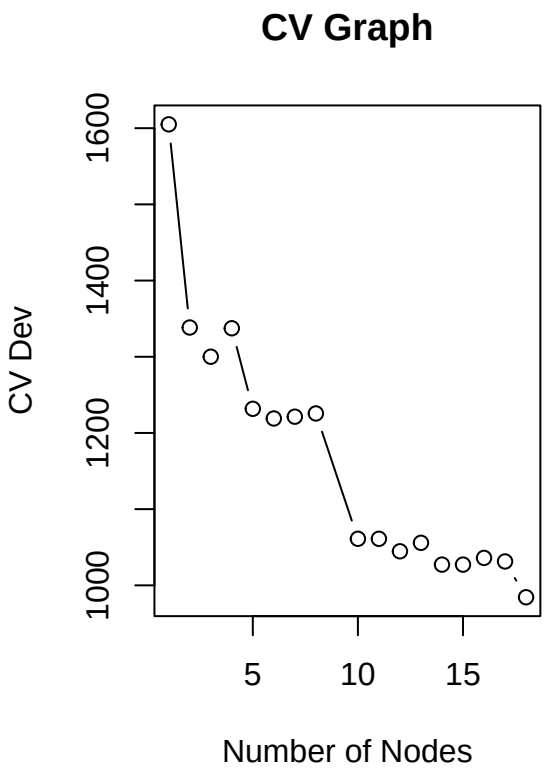
```
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -3.88200 -0.88200 -0.08712  0.00000  0.89590  4.09900
```

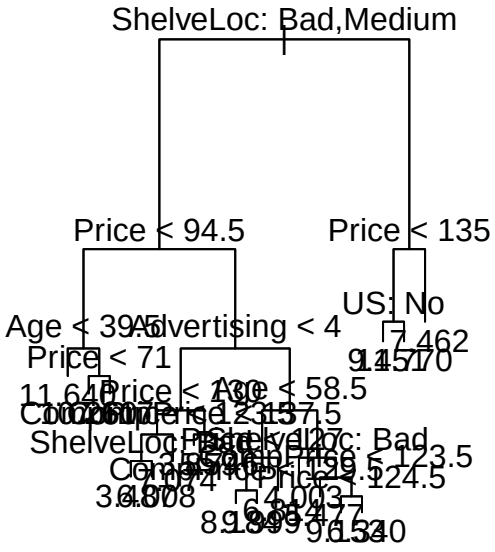
```
## [1] 4.922039
```

ShelveLoc is the first varibale used to sub-divide the data, followed by price. This makes sense considering the better the location, the higher likelihood of a customer purchasing the item. MSE = 4.922039

Part C:

```
## $size
##  [1] 18 17 16 15 14 13 12 11 10  8  7  6  5  4  3  2  1
##
## $dev
##  [1]  984.3936 1031.3372 1036.0021 1027.2166 1027.2166 1055.8168 1044.6955
##  [8] 1061.0899 1061.0899 1225.5973 1221.3487 1219.0219 1231.6886 1337.3952
## [15] 1300.0524 1338.3702 1605.0221
##
## $k
##  [1]      -Inf  16.99544  20.56322  25.01730  25.57104  28.01938  30.36962
##  [8]  31.56747  31.80816  40.75445  44.44673  52.57126  76.21881  99.59459
## [15] 116.69889 159.79501 337.60153
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```






```
## [1] 4.922039
```

Optimal level of tree complexity is 18 since lowest dev MSE = 4.922039 (the exact same as before pruning), so CV indicates that pruning does not improve MSE

Part D:

```
## randomForest 4.6-14
```

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

```
##
## Call:
##  randomForest(formula = Sales ~ ., data = Carseats, mtry = 10,      importance = TRUE, subset = train
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 10
##
##           Mean of squared residuals: 2.889221
##                     % Var explained: 63.26
```

```
## [1] 2.605253
```

```
##                %IncMSE IncNodePurity
## CompPrice   24.8888481    170.182937
## Income       4.7121131     91.264880
## Advertising 12.7692401     97.164338
## Population  -1.8074075     58.244596
## Price       56.3326252    502.903407
## ShelveLoc   48.8886689    380.032715
## Age         17.7275460    157.846774
## Education    0.5962186     44.598731
## Urban        0.1728373      9.822082
## US           4.2172102     18.073863
```

Bagging improves the test, returning an MSE of 2.605%, which is better than the pruned and original tree Price and ShelveLoc are the most important variables by a high margin

Part E:

```
## [1] 4.691554
```

```
## [1] 3.502607
```

```
## [1] 2.960559
```

```
##                %IncMSE IncNodePurity
## CompPrice   14.8840765     158.82956
## Income       4.3293950     125.64850
## Advertising  8.2215192     107.51700
## Population  -0.9488134      97.06024
## Price       34.9793386     385.93142
## ShelveLoc   34.9248499     298.54210
## Age         14.3055912     178.42061
## Education    1.3117842      70.49202
## Urban       -1.2680807      17.39986
## US           6.1139696      33.98963
```

Part 1: MSE under random forest increased to 4.692% Part 2: MSE under random forest increased to 3.503% Part 3: MSE under random forest increased to 2.961% Importance: Price and ShelveLoc are still the most important variables, but both decreased in importance significantly under RF

Chapter 8, Question 11:

Part A:

```
## Loaded gbm 2.1.8
```

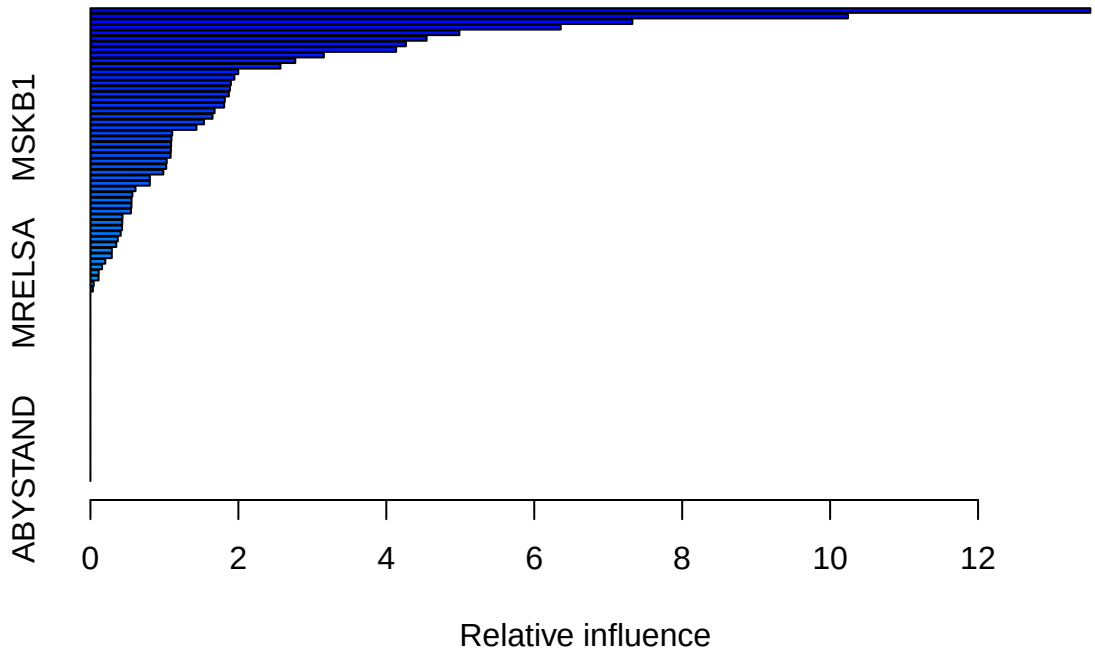
```
## [1] 1000   86
```

```
## [1] 4822   86
```

Part B:

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

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




```
##                var     rel.inf
## PPERSAUT PPERSAUT 13.51824557
## MKOOPKLA MKOOPKLA 10.24062778
## MOPLHOOG MOPLHOOG  7.32689780
## MBERMIDD MBERMIDD  6.35820558
## PBRAND     PBRAND  4.98826360
## ABRAND     ABRAND  4.54504653
## MGODGE     MGODGE  4.26496875
## MINK3045 MINK3045  4.13253907
## PWAPART   PWAPART  3.15612877
## MAUT1       MAUT1  2.76929763
## MOSTYPE   MOSTYPE  2.56937935
## MAUT2       MAUT2  1.99879666
## MSKA         MSKA  1.94618539
## MBERARBG MBERARBG  1.89917331
## PBYSTAND PBYSTAND  1.88591514
## MINKGEM   MINKGEM  1.87131472
```

```
## MGODOV     MGODOV  1.81673309
## MGODPR     MGODPR  1.80814745
## MFWEKIND MFWEKIND  1.67884570
## MSKC         MSKC  1.65075962
## MBERHOOG MBERHOOG  1.53559951
## MSKB1       MSKB1  1.43339514
## MOPLMIDD MOPLMIDD  1.10617074
## MHHUUR     MHHUUR  1.09608784
## MRELGE     MRELGE  1.09039794
## MINK7512 MINK7512  1.08772012
## MZFONDS   MZFONDS  1.08427551
## MGODRK     MGODRK  1.03126657
## MINK4575 MINK4575  1.02492795
## MZPART     MZPART  0.98536712
## MRELOV     MRELOV  0.80356854
## MFGEKIND MFGEKIND  0.80335689
## MBERARBO MBERARBO  0.60909852
## APERSAUT APERSAUT  0.56707821
## MGEMOMV   MGEMOMV  0.55589456
## MOSHOOFD MOSHOOFD  0.55498375
## MAUT0       MAUT0  0.54748481
## PMOTSCO   PMOTSCO  0.43362597
## MSKB2       MSKB2  0.43075446
## MSKD         MSKD  0.42751490
## MINK123M MINK123M  0.40920707
## MINKM30   MINKM30  0.36996576
## MHKOOP     MHKOOP  0.34941518
## MBERBOER MBERBOER  0.28967068
## MFALLEEN MFALLEEN  0.28877552
## MGEMLEEF MGEMLEEF  0.20084195
## MOPLLAAG MOPLLAAG  0.15750616
## MBERZELF MBERZELF  0.11203381
## PLEVEN     PLEVEN  0.11030994
## MRELSA     MRELSA  0.04500507
## MAANTHUI MAANTHUI  0.03322830
## 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
```

```
##    pred.test
##        0    1
##   0 4493   40
##   1  278   11
```

```
## [1] 0.2156863
```

21.57% of customers predicted to make the purchase actually purchased Most important variables are PPERSAUT and MKOOPKLA

Part C:

```
##    predict.test
##        0    1
##   0 4493   40
##   1  278   11
```

```
## [1] 0.2156863
```

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

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

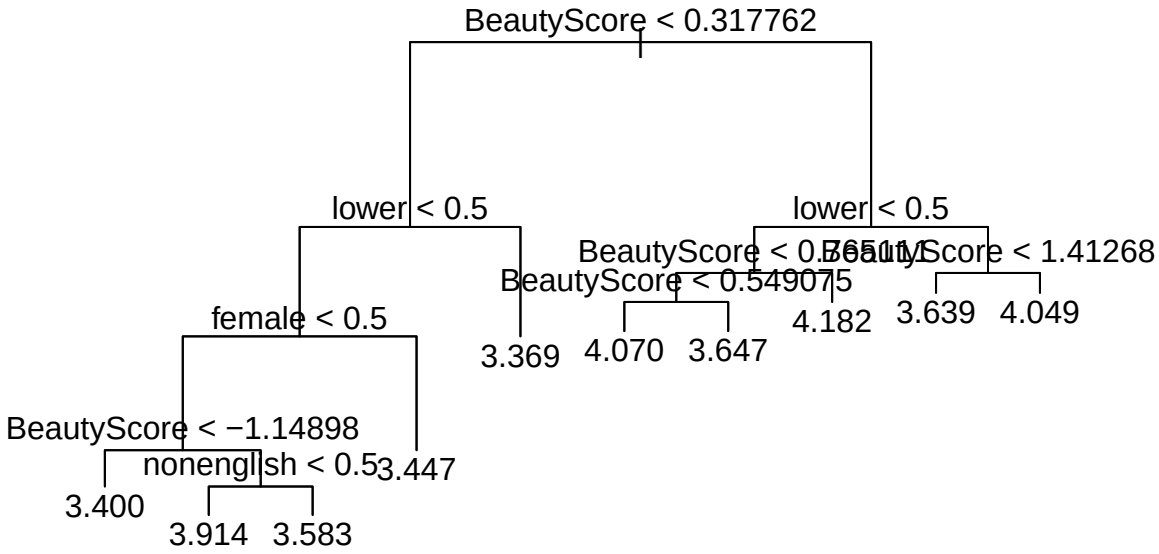
```
##    predict.test.c
##        0    1
##   0 4493   40
##   1  278   11
```

```
## [1] 0.2156863
```

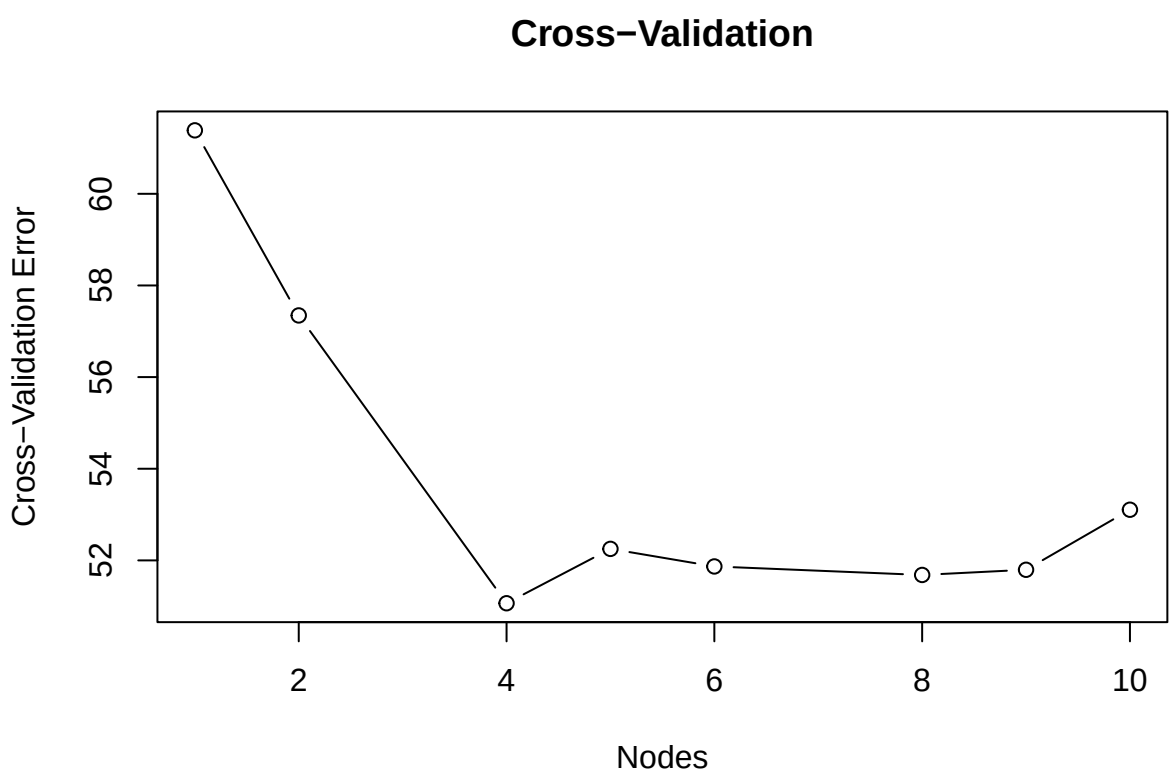
21.57% of people actually make a purchase

Problem 1:

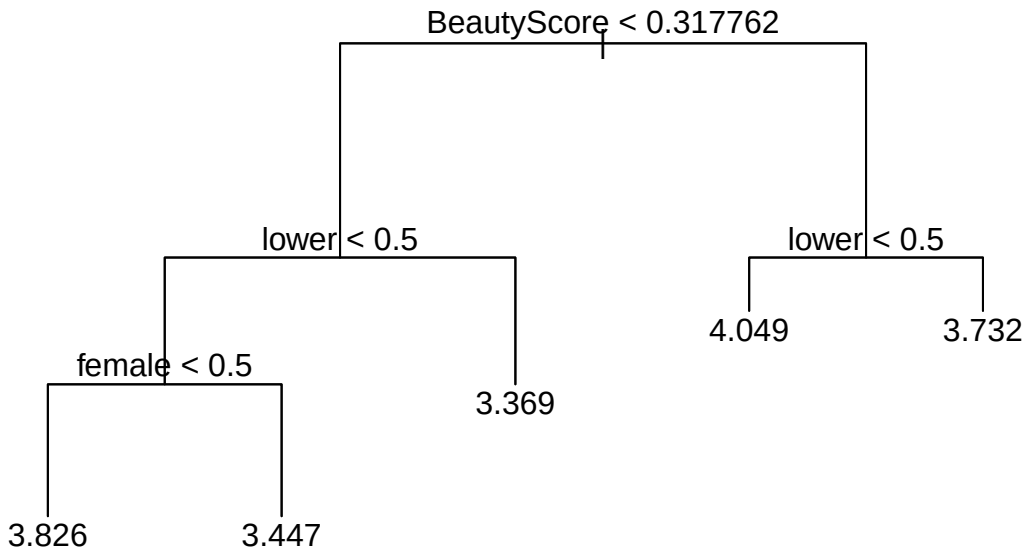
Part 1:



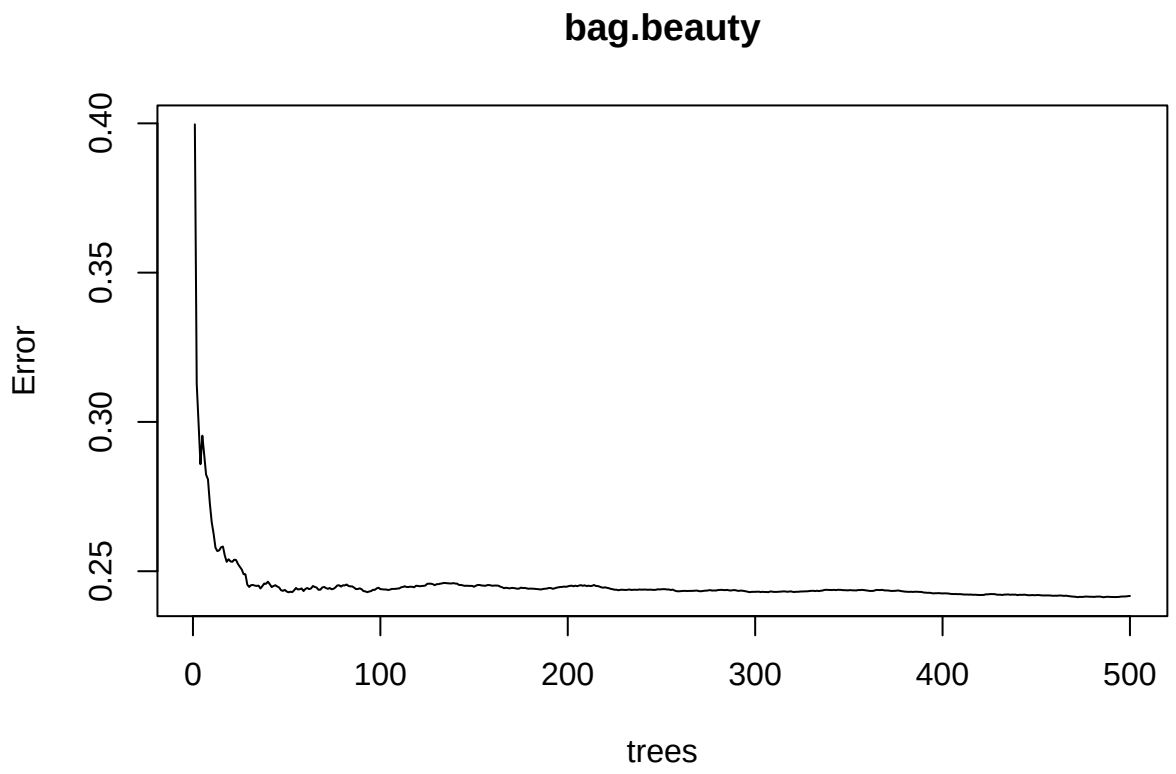

```
##
## Regression tree:
## tree(formula = CourseEvals ~ ., data = train.beauty)
## Variables actually used in tree construction:
## [1] "BeautyScore" "lower"       "female"      "nonenglish"
## Number of terminal nodes:  10
## Residual mean deviance:  0.1791 = 39.58 / 221
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -1.344000 -0.269700  0.001307  0.000000  0.262500  1.353000

## [1] 0.2365938
```



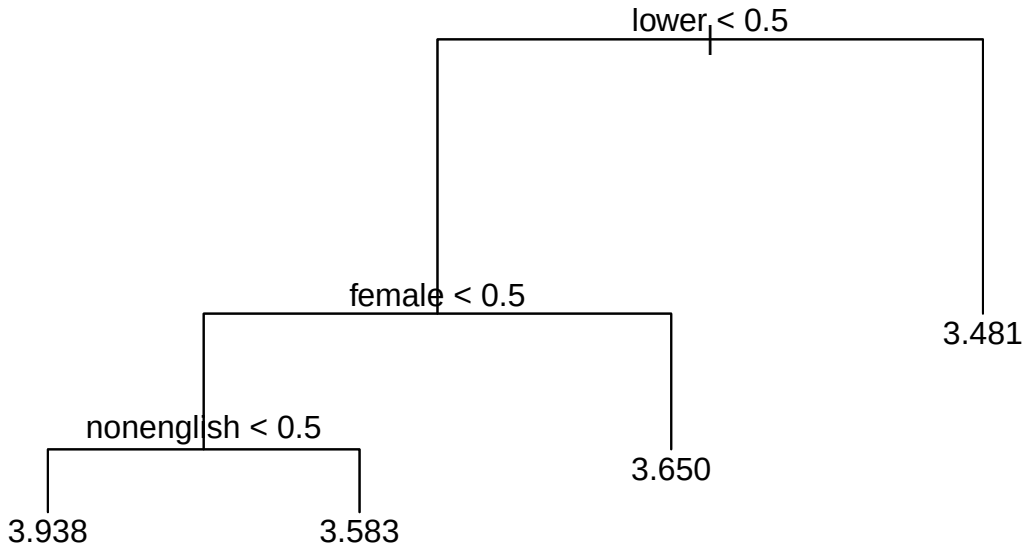

```
## $size
## [1] 10  9  8  6  5  4  2  1
##
## $dev
## [1] 53.10543 51.79520 51.68201 51.86791 52.25104 51.06476 57.34591 61.38483
##
## $k
## [1]      -Inf 0.6490978 0.9311412 0.9503400 1.1864647 1.5033592 3.6479934
## [8] 6.0402616
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"

## [1] 0.2361772
```


BeautyScore < 0.317762
lower < 0.5
lower < 0.5
4.049
3.732
female < 0.5
3.369
3.826
3.447

```
## [1] 0.2306021
```

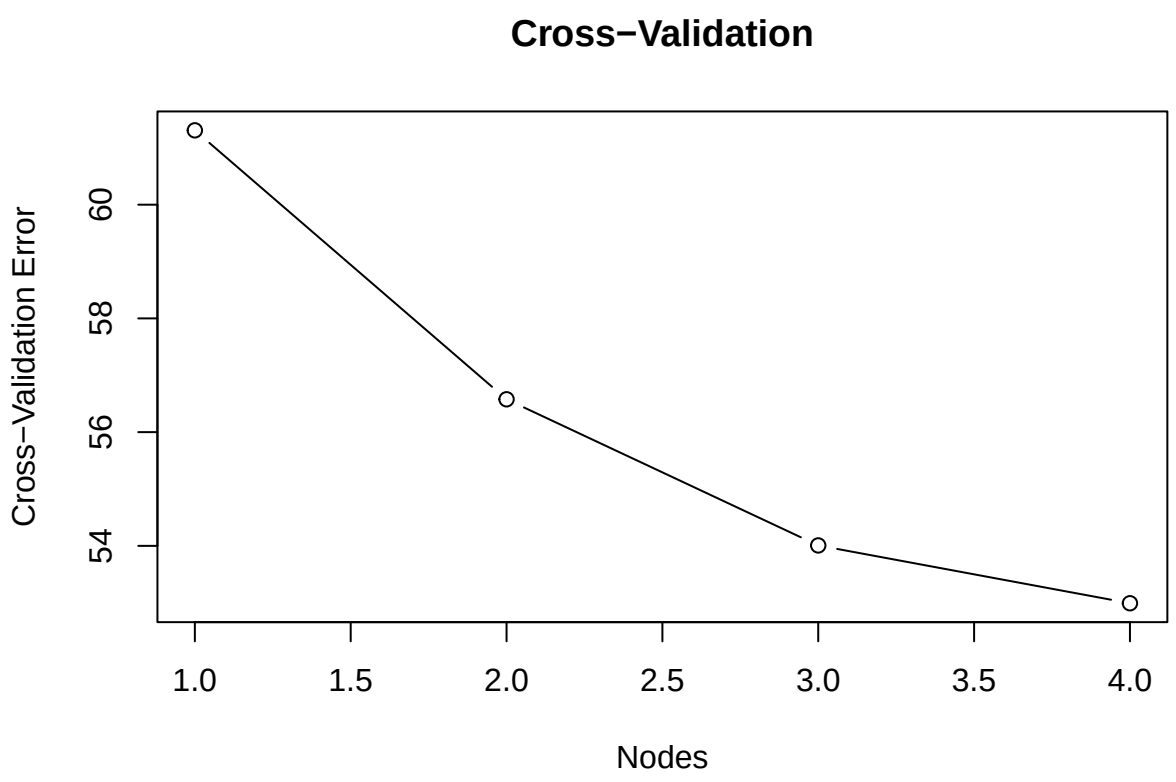



```
##                %IncMSE IncNodePurity
## BeautyScore 26.282685     27.125598
## female      22.693881      4.734619
## lower       27.951427      5.829280
## nonenglish  15.173485      1.199290
## tenuretrack -1.609842      1.242593
```




```
##
## Regression tree:
## tree(formula = CourseEvals ~ . - BeautyScore, data = train.beauty)
## Variables actually used in tree construction:
## [1] "lower"      "female"     "nonenglish"
## Number of terminal nodes:  4
## Residual mean deviance:  0.2231 = 50.64 / 227
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -1.36800 -0.30620 -0.04101  0.00000  0.30660  1.31000
```

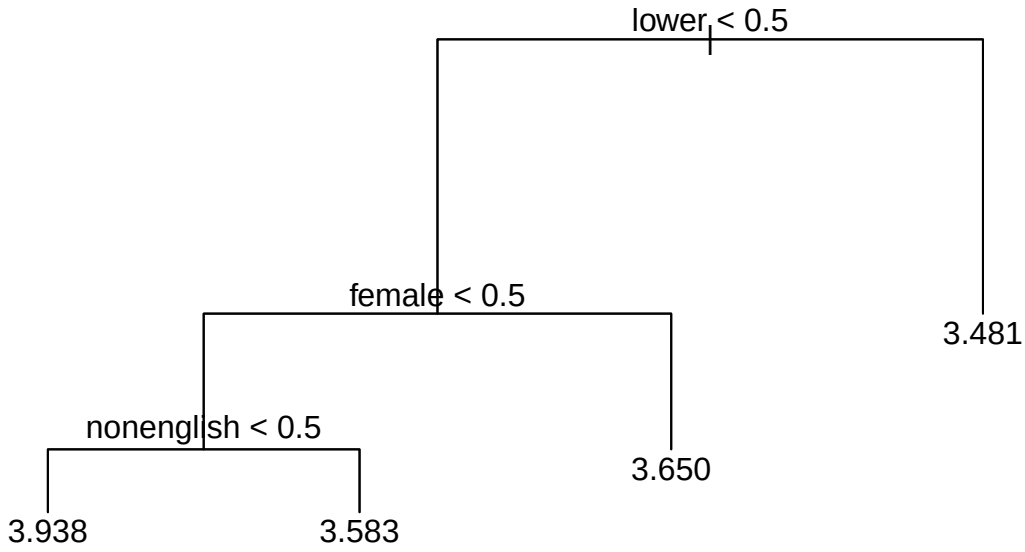
```
## [1] 0.268033
```




```
## $size
## [1] 4 3 2 1
##
## $dev
## [1] 52.99103 54.00746 56.57754 61.30708
##
## $k
## [1]     -Inf 1.123451 2.424533 4.898956
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
## Warning in prune.tree(tree.beauty1, best = 5): best is bigger than tree size
```

```
## [1] 0.268033
```




```
## [1] 0.2620418
```

```
##                %IncMSE IncNodePurity
## female      21.7259908      3.158258
## lower       50.0966280      5.157994
## nonenglish   8.2963724      1.400529
## tenuretrack  0.2961264      1.496597
```

I took the BeautyScore out of the model to determine if and how it would decrease the accuracy of my model, since it was the most important variable in my original model, followed by female and lower. This decreased the accuracy of the model by 47.61045%, and the MSE without the BeautyScore increased from the MSE with BeautyScore. Once BeautyScore was removed, I noticed that other variables maintained impacts on the model with various degrees of importance. For example, tenuretrack and nonenglish were not included in the tree when BeautyScore was present, however after it was taken out both variables became end nodes.

Part 2:

Considering human nature promotes biases, it would be impossible to fully differentiate between discrimination and productivity in the current method of data gathering for this model.It would be ideal to fully understand the impacts of beauty, and in fact all other variables measured, by creating a control group. This can be done by fully hiding the identity of the professor.Further, beauty is not quantifiable, and each individual has a different method of measuring attractiveness, making this a rather subjective variable.Finally, the model relied more on other variables after removing beauty. This further entanglement of variables shows that it is likely that even without beauty, the other factors are still impacted by personal biases of participants.

Problem 2:

Code for Parts 1, 2, and 3:

```
##
## Call:
## lm(formula = Price ~ Nbhd + Offers + SqFt + Brick + Bathrooms +
##     Nbhd:Brick, data = test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.00904 -0.24346  0.00281  0.18435  1.13781
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.4559025  0.0704279  -6.473 2.24e-09 ***
## Nbhd2           0.0001016  0.1016753   0.001 0.999204
## Nbhd3           0.8006322  0.1188654   6.736 6.13e-10 ***
## Offers         -0.3087324  0.0430966  -7.164 7.09e-11 ***
## SqFt            0.4498625  0.0453502   9.920  < 2e-16 ***
## BrickYes        0.4817117  0.1566201   3.076 0.002606 **
## Bathrooms       0.1446066  0.0421474   3.431 0.000828 ***
## Nbhd2:BrickYes  0.0591436  0.1944563   0.304 0.761546
## Nbhd3:BrickYes  0.3606671  0.2034039   1.773 0.078761 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3787 on 119 degrees of freedom
## Multiple R-squared:  0.8656, Adjusted R-squared:  0.8566
## F-statistic: 95.81 on 8 and 119 DF,  p-value: < 2.2e-16
```

```
##                      2.5 %     97.5 %
## (Intercept)    -0.59535676 -0.3164482
## Nbhd2          -0.20122556  0.2014288
## Nbhd3           0.56526689  1.0359975
## Offers         -0.39406790 -0.2233969
## SqFt            0.36006456  0.5396605
## BrickYes        0.17158814  0.7918352
## Bathrooms       0.06115050  0.2280627
## Nbhd2:BrickYes -0.32589936  0.4441865
## Nbhd3:BrickYes -0.04209288  0.7634271
```

```
## [1] 0.03361621
```

Part 1:

There is a premium for brick houses everything being equal. Positive .482 slope for BrickYes versus the negative .456 slope for Intercept, there is a positive net value so this indicates a premium. Therefore, if a house is made of bricks, there will be an increase in price by .482 standard deviations.

Part 2:

There is a premium for houses in neighborhood 3. Positive .801 slope for Nbhd3 versus the negative .456 slope for Intercept, there is a positive net value so this indicates a premium. Therefore, if a house is in Nbhd3, there will be an increase in price by .801 standard deviations.

Part 3:

There is no premium for brick houses in neighborhood 3. Positive .361 slope for Nbhd3:BrickYes versus the negative .456 slope for Intercept, negative net so this indicates there is no premium. The confidence interval also includes zero, so this further justifies that there is no premium associated with brick houses in neighborhood 3.

Part 4:

```
##
## Call:
## lm(formula = Price ~ Nb_new + Offers + SqFt + Brick + Bathrooms +
##     Nb_new:Brick, data = test2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.00761 -0.24246 -0.00014  0.19055  1.13602
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.34225    0.08360   4.094 7.70e-05 ***
## Nb_newold          -0.79648    0.10294  -7.737 3.41e-12 ***
## Offers             -0.31152    0.04046  -7.699 4.16e-12 ***
## SqFt                0.45290    0.04341  10.433  < 2e-16 ***
## BrickYes            0.84155    0.12739   6.606 1.11e-09 ***
## Bathrooms           0.14530    0.04152   3.500 0.000653 ***
## Nb_newold:BrickYes -0.31795    0.15581  -2.041 0.043469 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3758 on 121 degrees of freedom
## Multiple R-squared:  0.8655, Adjusted R-squared:  0.8588
## F-statistic: 129.7 on 6 and 121 DF,  p-value: < 2.2e-16

##                          2.5 %       97.5 %
## (Intercept)         0.17673411  0.507766849
## Nb_newold          -1.00028439 -0.592671745
## Offers             -0.39162585 -0.231409323
## SqFt                0.36696087  0.538843011
## BrickYes            0.58934435  1.093759656
## Bathrooms           0.06310442  0.227500213
## Nb_newold:BrickYes -0.62642465 -0.009475166

## [1] 0.03363269
```

Yes, neighborhood 1 and neighborhood 2 can be combined into a single old neighborhood

Problem 3:

Part 1:

Each city has different cause and effect for crime rates that cannot be compared against each other. Simply taking crime and police data does not allow the study of if high police levels creates high crime levels and vice versa. Looking at DC for example, crime was impacted by higher police presence due to increased terrorist threats leading to increased police presence. However, on the same day the police presence in Austin might not be increased and crime rates would react differently than those in DC.

Part 2:

The researchers at UPENN isolated this by using Washington, DC as a test city for their study due to the increase in police presence on days when terrorism threats are increased. This means that the city had more police presence at times, unrelated to the level of crime actually being committed in the city. This allowed the study to recognize that days with higher police presence did indeed experience decreases in crime. The table shows that as you hold ridership rates of the DC Metro at a fixed rate on days with high terrorism alerts, an increase in police does have a negative effect on crime levels, but at a lower rate than when one does not hold the ridership rate at a fixed rate. This might not be fool-proof though, given that criminals might not be as willing to commit crimes on days that are dangerous to be out, such as those with high terrorism rates.

Part 3:

The study controlled METRO ridership in DC to see if the amount of civilians outside on days during high terrorism alerts was similar to those without terrorism alerts. This ensures that the number of potential victims of crime did not decrease, which in turn would decrease the crime rate, rather that the variable which impacted this decrease was the higher police presence.

Part 4:

Table 4 amended the study to see if the crime rate was similarly impacted in different areas of DC. The table shows that District 1's crime rate was the only one that was impacted, taking into account location and days with a terrorism threat.This is a reasonable conclusion given that District 1 contains the political center of the country, meaning much of the increased law enforcement presence is likely focused on District 1. The impact in Districts other than District 1, is indeed negative, but far smaller especially given the high standard error rate.

Problem 4:

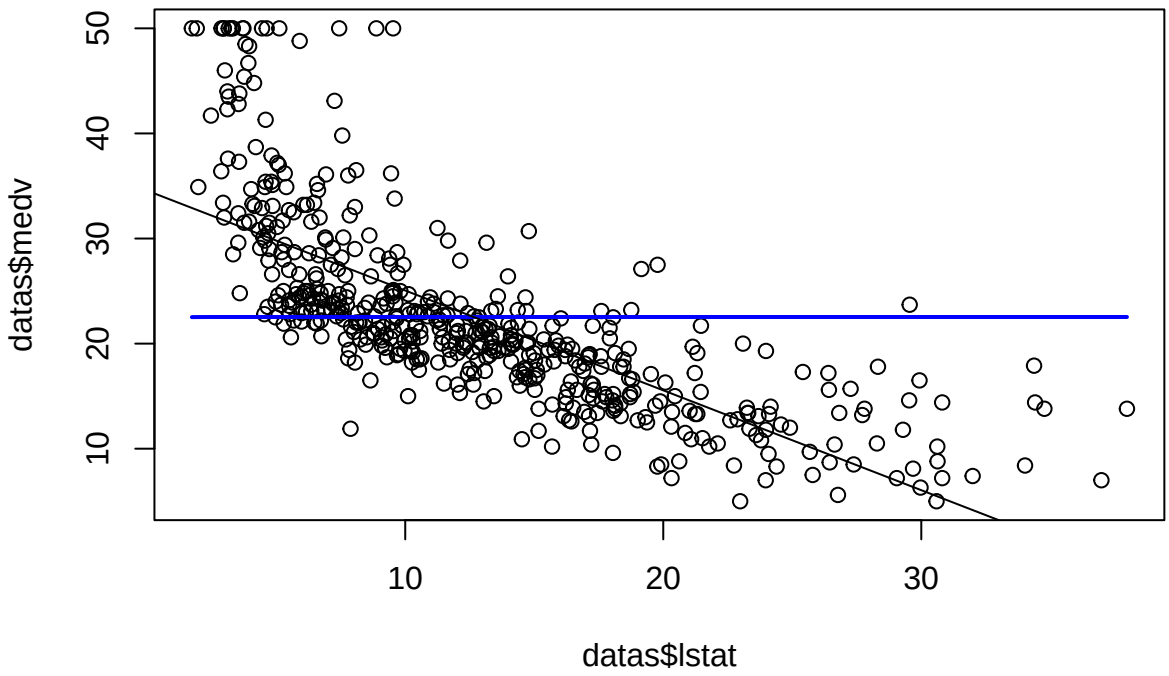
```
## The following object is masked _by_ .GlobalEnv:
##
##     chas

## The following objects are masked from Boston (pos = 11):
##
##     age, black, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad,
##     rm, tax, zn
```

```
## The following objects are masked from Boston (pos = 17):
##
##     age, black, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad,
##     rm, tax, zn
```

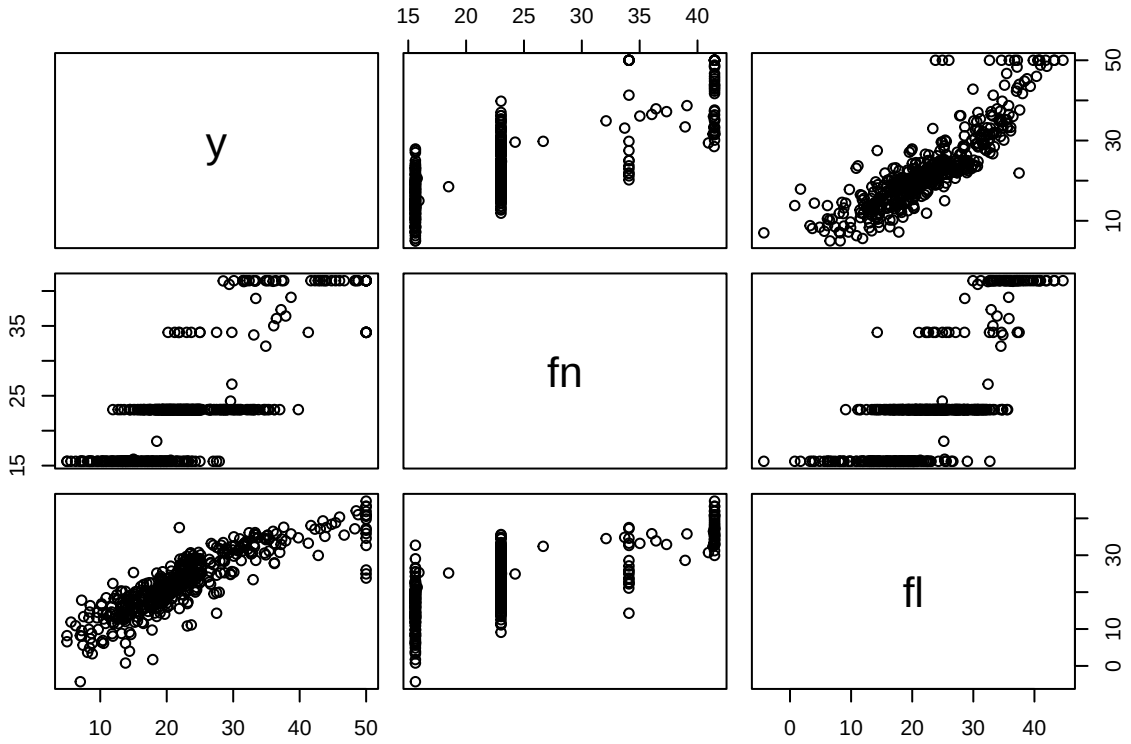
```
##       crim                zn              indus             chas        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
##  1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000  
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000  
##  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
```

```
## # weights:  10
## initial  value 300845.802131
## iter  10 value 42746.556919
## iter  20 value 42740.130992
## iter  30 value 42734.229089
## final  value 42734.208503
## converged
```

```
## a 1-3-1 network with 10 weights
## options were - linear output units  decay=0.1
##  b->h1 i1->h1
##   1.71   3.32
##  b->h2 i1->h2
##   1.69   3.33
##  b->h3 i1->h3
##   1.71   3.32
##  b->o h1->o h2->o h3->o
##  5.87  5.56  5.55  5.56
```




```
## # weights:  76
## initial  value 323953.665339
## iter  10 value 35150.873848
## iter  20 value 34599.755598
## iter  30 value 27637.708240
## iter  40 value 23298.894217
## iter  50 value 22518.752386
## iter  60 value 21597.132374
## iter  70 value 18131.721766
## iter  80 value 17057.701146
## iter  90 value 16097.998080
## iter 100 value 15433.565906
## final  value 15433.565906
## stopped after 100 iterations
```

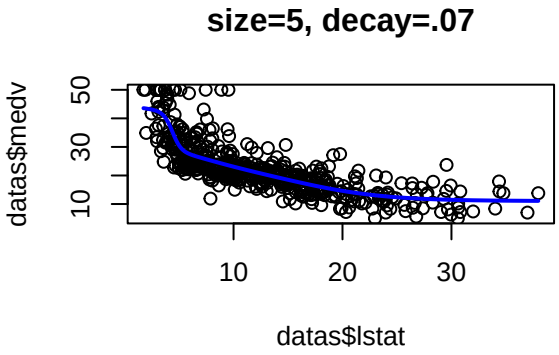


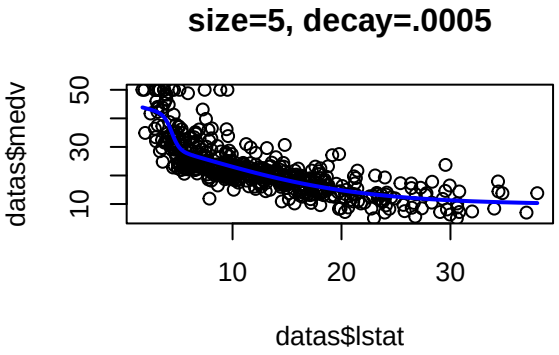

```
##            y        fn        fl
## y  1.0000000 0.8020297 0.8606060
## fn 0.8020297 1.0000000 0.7410829
## fl 0.8606060 0.7410829 1.0000000
```

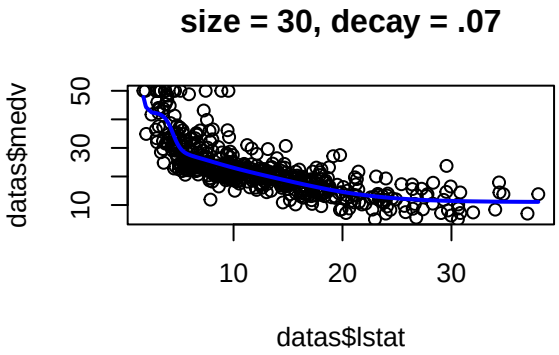
```
## # weights:  16
## initial  value 293622.446774
## iter  10 value 34105.427273
## iter  20 value 16947.072400
## iter  30 value 14496.523374
## iter  40 value 13309.106097
## iter  50 value 13271.175344
## iter  60 value 13261.858850
## iter  70 value 13258.123400
## iter  80 value 13250.451732
## iter  90 value 13241.302553
## iter 100 value 13238.722454
## final  value 13238.722454
## stopped after 100 iterations
```

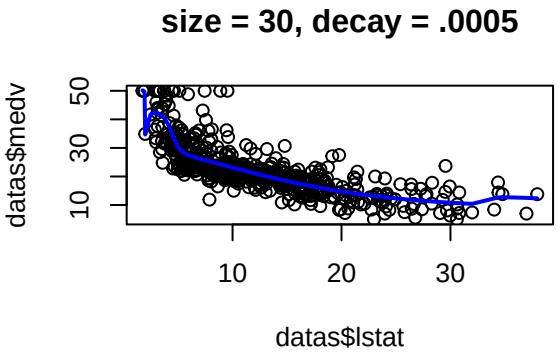
```
## # weights:  16
## initial  value 296603.373593
## iter  10 value 24630.877715
## iter  20 value 17393.698952
## iter  30 value 14939.757489
```

```
## iter  40 value 13562.613695
## iter  50 value 13231.680085
## iter  60 value 13229.975278
## iter  70 value 13223.122304
## iter  80 value 13215.918988
## iter  90 value 13212.090891
## iter 100 value 13204.924529
## final  value 13204.924529
## stopped after 100 iterations

## # weights:  91
## initial  value 301024.719165
## iter  10 value 21340.444598
## iter  20 value 14406.166170
## iter  30 value 13748.864309
## iter  40 value 13263.103277
## iter  50 value 13221.790571
## iter  60 value 13218.435679
## iter  70 value 13215.807404
## iter  80 value 13212.697521
## iter  90 value 13209.671221
## iter 100 value 13206.979768
## final  value 13206.979768
## stopped after 100 iterations

## # weights:  91
## initial  value 266538.324045
## iter  10 value 36850.091426
## iter  20 value 18100.941719
## iter  30 value 13799.212661
## iter  40 value 13326.514792
## iter  50 value 13273.722889
## iter  60 value 13129.840292
## iter  70 value 13047.395338
## iter  80 value 13039.476793
## iter  90 value 13031.621486
## iter 100 value 13026.943018
## final  value 13026.943018
## stopped after 100 iterations
```



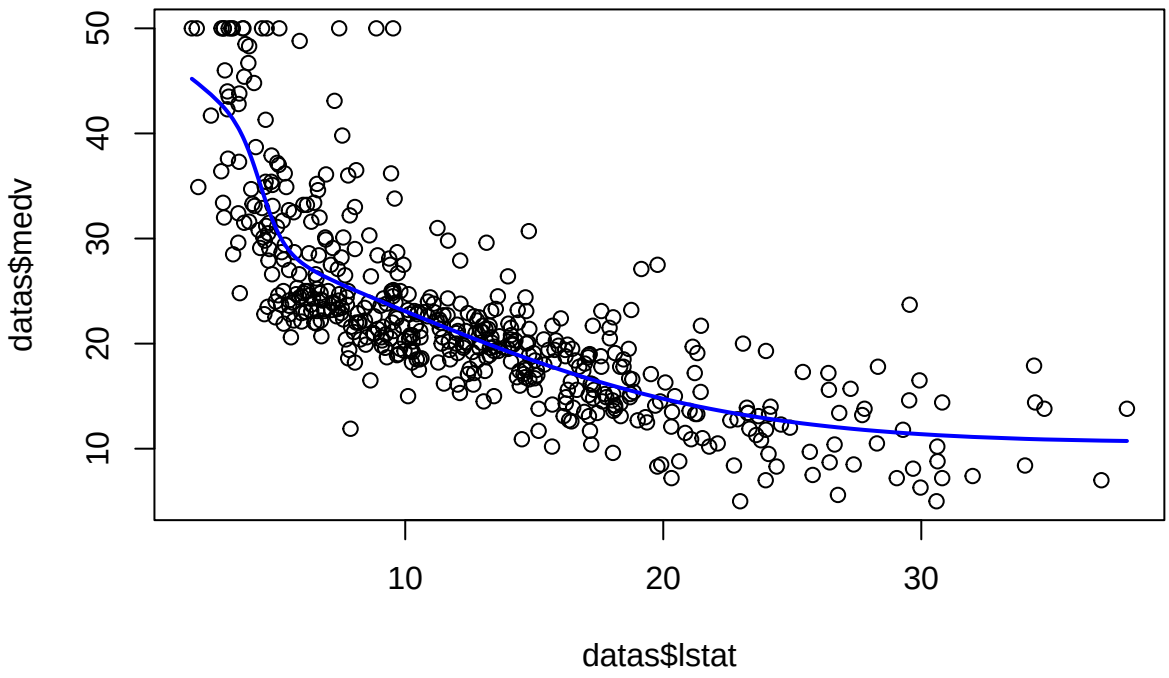




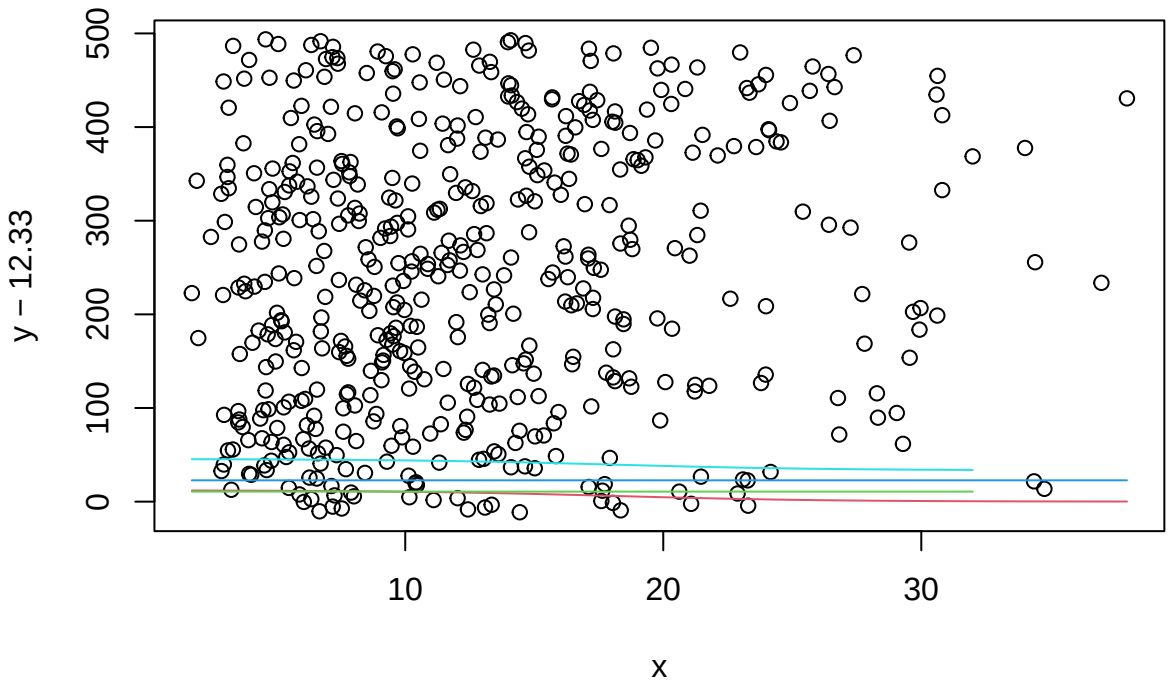


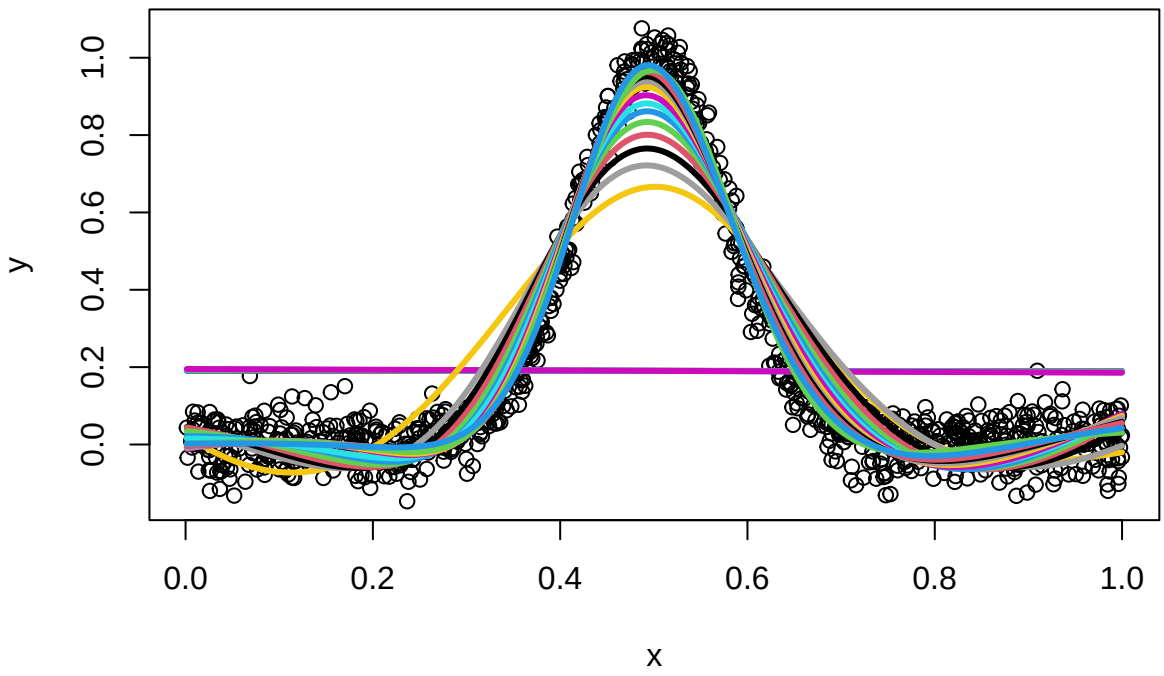

```
## # weights:  121
## initial  value 328251.010338
## iter  10 value 32795.828871
## iter  20 value 15061.181161
## iter  30 value 14703.727829
## iter  40 value 14325.587496
## iter  50 value 13924.220503
## iter  60 value 13552.877649
## iter  70 value 13514.345068
## iter  80 value 13504.678743
## iter  90 value 13487.730170
## iter 100 value 13478.154121
## final  value 13478.154121
## stopped after 100 iterations

## # weights:  121
## initial  value 305936.573157
## iter  10 value 18634.706076
## iter  20 value 16523.364877
## final  value 16195.258241
## stopped after 25 iterations

## # weights:  121
## initial  value 284981.141752
## iter  10 value 29870.311607
```

```
## iter  20 value 14488.441832
## iter  30 value 13981.390463
## iter  40 value 13548.149062
## iter  50 value 13493.723803
## iter  60 value 13480.299528
## iter  70 value 13476.796645
## iter  80 value 13472.912004
## iter  90 value 13469.128131
## iter 100 value 13465.811507
## iter 110 value 13462.270482
## iter 120 value 13459.287231
## iter 130 value 13456.707289
## iter 140 value 13455.429671
## iter 150 value 13454.446443
## iter 160 value 13453.814630
## iter 170 value 13453.003352
## iter 180 value 13452.478525
## iter 190 value 13452.020511
## iter 200 value 13451.658659
## iter 210 value 13451.372099
## iter 220 value 13451.021242
## iter 230 value 13450.772993
## iter 240 value 13450.680014
## iter 250 value 13450.609703
## iter 260 value 13450.572701
## iter 270 value 13450.516752
## iter 280 value 13450.434701
## iter 290 value 13450.353816
## iter 300 value 13450.194804
## iter 310 value 13450.073361
## iter 320 value 13450.049598
## iter 330 value 13450.003707
## iter 340 value 13449.948610
## iter 350 value 13449.929280
## final  value 13449.928822
## converged
```




```
## a 13-5-1 network with 76 weights
## options were - linear output units  decay=0.1
##   b->h1  i1->h1  i2->h1  i3->h1  i4->h1  i5->h1  i6->h1  i7->h1  i8->h1  i9->h1
##    0.01   -0.01    0.02   -0.15    0.00   -0.02    0.48   -4.18    0.63    0.29
## i10->h1 i11->h1 i12->h1 i13->h1
##   -2.12   -1.06    3.37   -3.10
##   b->h2  i1->h2  i2->h2  i3->h2  i4->h2  i5->h2  i6->h2  i7->h2  i8->h2  i9->h2
##   -1.78    2.14    1.83   -6.72    5.44    7.99  -19.33    0.21    7.26  -10.36
## i10->h2 i11->h2 i12->h2 i13->h2
##    0.16    8.45   -0.34   16.18
##   b->h3  i1->h3  i2->h3  i3->h3  i4->h3  i5->h3  i6->h3  i7->h3  i8->h3  i9->h3
##    0.00    0.00    0.00    0.00    0.00    0.00    0.01    0.06    0.01    0.00
## i10->h3 i11->h3 i12->h3 i13->h3
##    0.30    0.02    0.50    0.00
##   b->h4  i1->h4  i2->h4  i3->h4  i4->h4  i5->h4  i6->h4  i7->h4  i8->h4  i9->h4
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.03    0.00    0.01
## i10->h4 i11->h4 i12->h4 i13->h4
##    0.19    0.01    0.09    0.00
##   b->h5  i1->h5  i2->h5  i3->h5  i4->h5  i5->h5  i6->h5  i7->h5  i8->h5  i9->h5
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.02    0.00    0.00
## i10->h5 i11->h5 i12->h5 i13->h5
##    0.07    0.00    0.01    0.00
##   b->o  h1->o  h2->o  h3->o  h4->o  h5->o
##   8.52   7.40 -18.46   8.51   8.52   8.52
```


y − 12.33
0
100
200
300
400
500
10
20
30
x




```
## # weights:  10
## initial  value 667.209484
## iter  10 value 101.026184
## iter  20 value 100.820012
## final  value 100.810813
## converged
## [1] 1
##
## # weights:  10
## initial  value 102.440980
## iter  10 value 100.826954
## iter  20 value 100.807261
## final  value 100.806135
## converged
## [1] 2
##
## # weights:  10
## initial  value 541.209225
## iter  10 value 100.852025
## iter  20 value 100.816299
## iter  30 value 100.802004
## final  value 100.801949
## converged
## [1] 3
##
## # weights:  10
```

```
## initial  value 788.096197
## iter  10 value 101.194838
## iter  20 value 100.915328
## iter  30 value 100.806993
## iter  40 value 100.797891
## final  value 100.797834
## converged
## [1] 4
##
## # weights:  10
## initial  value 478.117642
## iter  10 value 101.685651
## iter  20 value 100.832005
## iter  30 value 100.794559
## iter  40 value 100.793583
## final  value 100.793575
## converged
## [1] 5
##
## # weights:  10
## initial  value 146.845885
## iter  10 value 100.818033
## iter  20 value 100.796759
## iter  30 value 100.790239
## iter  40 value 100.789554
## iter  50 value 100.789452
## iter  50 value 100.789451
## iter  50 value 100.789451
## final  value 100.789451
## converged
## [1] 6
##
## # weights:  10
## initial  value 329.771585
## iter  10 value 100.844858
## iter  20 value 97.533692
## iter  30 value 75.549915
## iter  40 value 65.922405
## iter  50 value 59.084944
## iter  60 value 56.115796
## iter  70 value 55.842128
## iter  80 value 55.635677
## iter  90 value 55.605246
## iter 100 value 55.600017
## iter 110 value 55.596450
## iter 120 value 55.584171
## iter 130 value 55.551805
## iter 140 value 54.874792
## iter 150 value 54.185943
## iter 160 value 53.228593
## final  value 53.226884
## converged
## [1] 7
##
```

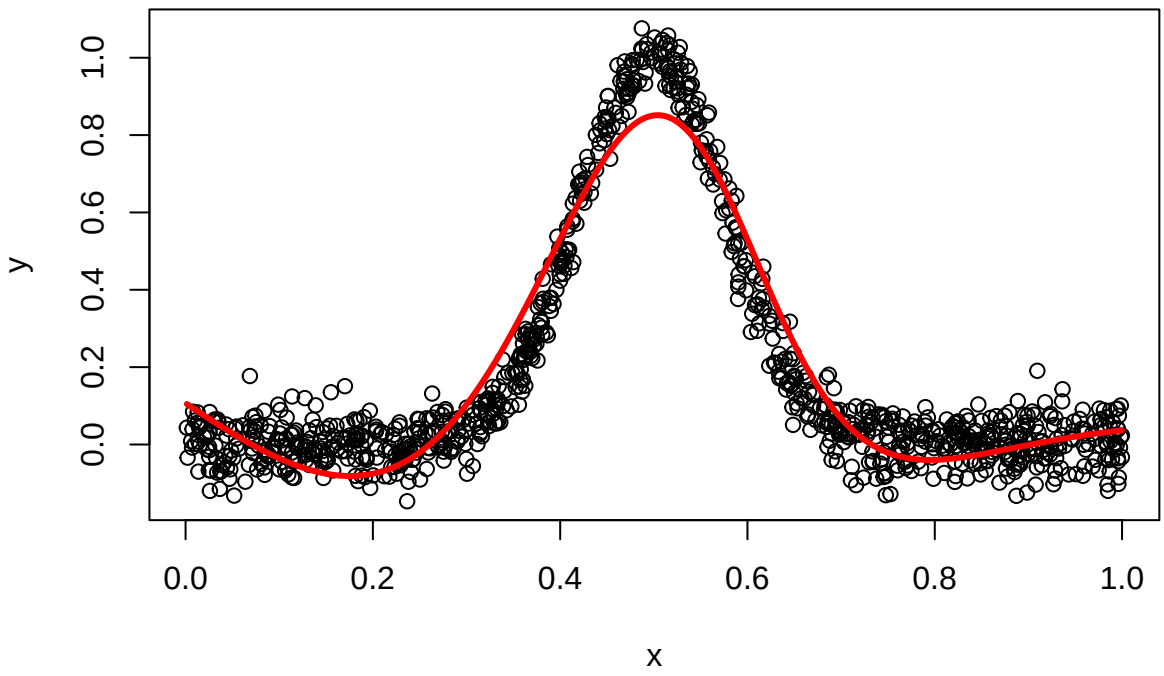
```
## # weights:  10
## initial  value 3068.426044
## iter  10 value 100.371708
## iter  20 value 68.246061
## iter  30 value 48.625330
## iter  40 value 45.276644
## iter  50 value 44.401786
## iter  60 value 43.643690
## iter  70 value 43.471221
## iter  80 value 43.345321
## iter  90 value 43.337162
## iter  90 value 43.337162
## iter  90 value 43.337162
## final  value 43.337162
## converged
## [1] 8
##
## # weights:  10
## initial  value 475.529741
## iter  10 value 95.725015
## iter  20 value 60.553867
## iter  30 value 44.925104
## iter  40 value 41.677245
## iter  50 value 39.739197
## iter  60 value 38.902779
## iter  70 value 37.767167
## iter  80 value 36.176829
## iter  90 value 35.938731
## iter 100 value 35.938013
## final  value 35.937970
## converged
## [1] 9
##
## # weights:  10
## initial  value 1088.741534
## iter  10 value 97.327375
## iter  20 value 62.897333
## iter  30 value 58.408524
## iter  40 value 45.125803
## iter  50 value 34.710876
## iter  60 value 31.570089
## iter  70 value 30.069848
## iter  80 value 29.890360
## iter  90 value 29.787804
## iter 100 value 29.786191
## final  value 29.786124
## converged
## [1] 10
##
## # weights:  10
## initial  value 123.832237
## iter  10 value 93.503417
## iter  20 value 55.730199
## iter  30 value 45.149637
```

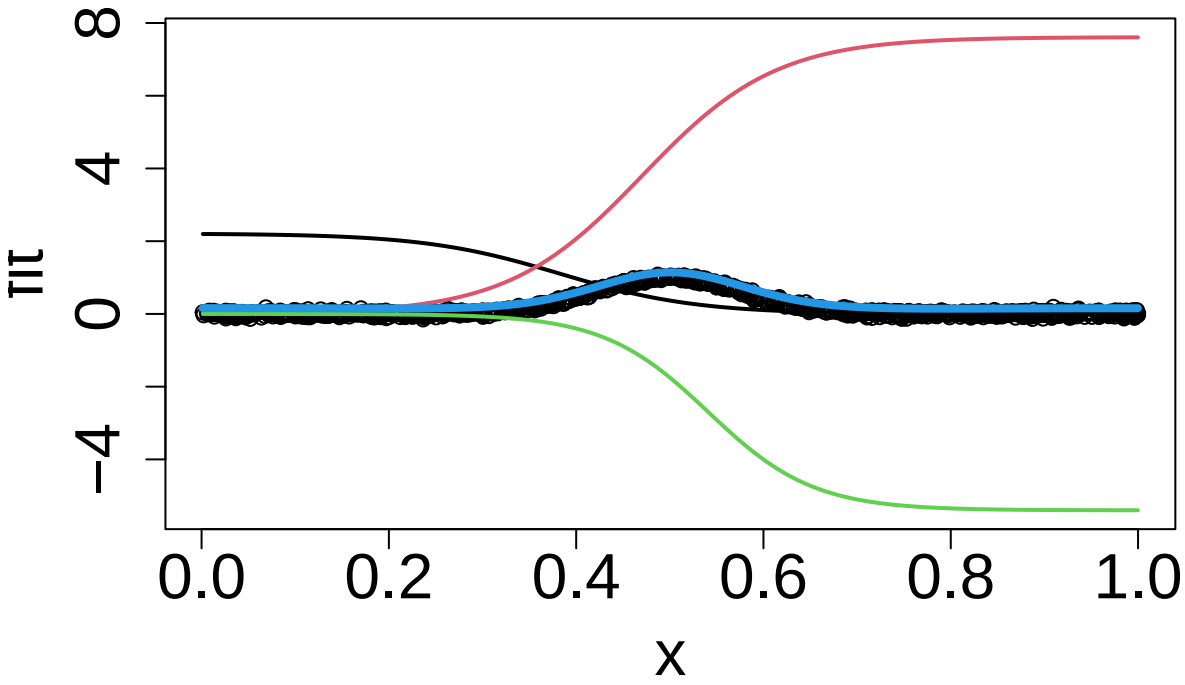
```
## iter  40 value 37.068292
## iter  50 value 26.396158
## iter  60 value 25.314269
## iter  70 value 24.835859
## iter  80 value 24.785966
## final  value 24.785776
## converged
## [1] 11
##
## # weights:  10
## initial  value 941.552226
## iter  10 value 100.917847
## iter  20 value 95.645178
## iter  30 value 40.872195
## iter  40 value 24.362123
## iter  50 value 21.576139
## iter  60 value 20.638054
## iter  70 value 20.375986
## iter  80 value 20.258904
## iter  90 value 20.237636
## iter 100 value 20.221748
## final  value 20.221565
## converged
## [1] 12
##
## # weights:  10
## initial  value 100.954276
## iter  10 value 100.784493
## iter  20 value 94.318711
## iter  30 value 59.559868
## iter  40 value 43.854057
## iter  50 value 30.223812
## iter  60 value 21.625085
## iter  70 value 20.619622
## iter  80 value 18.228011
## iter  90 value 17.840747
## iter 100 value 17.810966
## iter 110 value 17.808866
## final  value 17.808805
## converged
## [1] 13
##
## # weights:  10
## initial  value 182.125022
## iter  10 value 95.939831
## iter  20 value 29.937943
## iter  30 value 19.496491
## iter  40 value 17.283913
## iter  50 value 16.454872
## iter  60 value 15.330165
## iter  70 value 14.720350
## iter  80 value 14.237586
## iter  90 value 14.222353
## iter 100 value 14.210798
```

```
## iter 110 value 14.210084
## final  value 14.210020
## converged
## [1] 14
##
## # weights:  10
## initial  value 207.261262
## iter  10 value 100.794214
## iter  20 value 97.896878
## iter  30 value 67.777669
## iter  40 value 36.111348
## iter  50 value 25.592587
## iter  60 value 12.109255
## iter  70 value 11.749969
## iter  80 value 11.617075
## iter  90 value 11.589561
## iter 100 value 11.584084
## final  value 11.583949
## converged
## [1] 15
##
## # weights:  10
## initial  value 104.282101
## iter  10 value 96.826786
## iter  20 value 54.705298
## iter  30 value 40.571252
## iter  40 value 36.439169
## iter  50 value 33.754562
## iter  60 value 17.557366
## iter  70 value 10.558728
## iter  80 value 9.909745
## iter  90 value 9.730683
## iter 100 value 9.688578
## iter 110 value 9.686636
## iter 120 value 9.686505
## final  value 9.686504
## converged
## [1] 16
##
## # weights:  10
## initial  value 107.517037
## iter  10 value 99.179407
## iter  20 value 74.103263
## iter  30 value 39.576815
## iter  40 value 35.584109
## iter  50 value 34.860364
## iter  60 value 34.531815
## iter  70 value 33.466494
## iter  80 value 30.910451
## iter  90 value 12.687949
## iter 100 value 11.298856
## iter 110 value 10.067692
## iter 120 value 9.703223
## iter 130 value 9.351077
```

```
## iter 140 value 9.184847
## iter 150 value 9.011885
## iter 160 value 8.910900
## iter 170 value 8.830061
## iter 180 value 8.787602
## iter 190 value 8.711741
## iter 200 value 8.684036
## iter 210 value 8.623226
## iter 220 value 8.567492
## iter 230 value 8.542477
## iter 240 value 8.523589
## iter 250 value 8.523494
## final  value 8.523482
## converged
## [1] 17
##
## # weights:  10
## initial  value 532.748497
## iter  10 value 100.461278
## iter  20 value 29.100384
## iter  30 value 8.345113
## iter  40 value 8.237850
## iter  50 value 7.803442
## iter  60 value 7.515991
## iter  70 value 7.364291
## iter  80 value 7.349896
## iter  90 value 7.328494
## iter 100 value 7.313073
## iter 110 value 7.290009
## iter 120 value 7.285413
## iter 130 value 7.271356
## iter 140 value 7.248722
## iter 150 value 7.227985
## iter 160 value 7.163726
## iter 170 value 7.148664
## iter 180 value 7.140695
## iter 190 value 7.140371
## iter 200 value 7.140344
## final  value 7.140344
## converged
## [1] 18
##
## # weights:  10
## initial  value 209.820417
## iter  10 value 100.728480
## iter  20 value 93.561224
## iter  30 value 66.784155
## iter  40 value 14.613506
## iter  50 value 10.120865
## iter  60 value 7.583522
## iter  70 value 7.092579
## iter  80 value 6.874647
## iter  90 value 6.646920
## iter 100 value 6.440176
```

```
## iter 110 value 6.315671
## iter 120 value 6.242634
## iter 130 value 6.168678
## iter 140 value 6.146759
## iter 150 value 6.112820
## iter 160 value 6.084322
## iter 170 value 6.036342
## iter 180 value 6.033408
## iter 190 value 6.032952
## final  value 6.032942
## converged
## [1] 19
##
## # weights:  10
## initial  value 173.792531
## iter  10 value 100.754535
## iter  20 value 89.667530
## iter  30 value 39.589511
## iter  40 value 7.078267
## iter  50 value 5.799601
## iter  60 value 5.680030
## iter  70 value 5.590441
## iter  80 value 5.500487
## iter  90 value 5.462176
## iter 100 value 5.421297
## iter 110 value 5.417246
## iter 120 value 5.417005
## final  value 5.416988
## converged
## [1] 20

## # weights:  10
## initial  value 465.933841
## iter  10 value 100.787554
## iter  20 value 98.638855
## iter  30 value 64.470715
## iter  40 value 53.295894
## iter  50 value 39.429427
## iter  60 value 25.916367
## iter  70 value 25.174796
## iter  80 value 22.049713
## iter  90 value 21.303681
## iter 100 value 21.106118
## iter 110 value 21.086510
## iter 120 value 21.076069
## final  value 21.075860
## converged
```


y
0.0
0.2
0.4
0.6
0.8
1.0
x
0.0
0.2
0.4
0.6
0.8
1.0



Problem 5:

On the team project, our group was very collaborative throughout the duration of the assignment. Through a total of 7 meetings, each lasting 2 to 3 hours, we worked on every step of the project as a single unit. In these meetings, I helped select the dataset with which we completed the assignment with. I also led a workshop on how to use R to generate trees by going through the Chapter 8 lab in the book step-by-step with my team. We then used this knowledge as a foundation to work on our project. I worked on the code for fitting of trees on my own, and went through the code with the group who assisted in editing it to meet the standards of all members. Together, we also figured out how to conduct a Random Forest and Bagging model. Once our codes were to our satisfaction, I ensured I attended office hours held by Pedro to run our code and thought processes by him. He advised additions and changes that would supplement our code, and using this insight I was able to guide my team to a more accurate and complete project. We then fine-tuned our code, and once it was to our liking, we began working on our presentation. I helped each team member write their talking points by dissecting each line of code with the team member assigned to a particular section for the presentation. We then created the slides as a team, and we held multiple run-throughs together and individually. Taking all of this into account, I would say our team dynamic was extremely unique in that our project was truly done as a team each step of the way. There were moments in which we each were able to take leadership and play an integral role, which is how collaboration in its most efficient and productive form should work. I am very pleased with my performance as a team member, and believe I exceeded all expectations that were set for me by my teammates and myself coming into the project. I truly grew a lot given this was the first group assignment I had completed at the Master's level, and I am eager to continue this momentum throughout the rest of my time in the MSBA cohort and beyond.