Crime Analysis for Boston

2023-08-27

WHAT IS THE PROBLEM ABOUT

This problem involves the Boston data set. Here, I am predicting per capita crime rate using the some of the variables in this data set. In simple words, per capita crime rate is the response, and the other variables are the predictors.

I HAVE DIVIDED THE PROJECT INTO SECTIONS SO THAT IT GIVES CLARITY TO THE VIEWER ON WHAT TO EXPECT IN EACH STAGE

OVERVIEW OF THE FIRST STEPS TAKEN

First, for each predictor, I will fit a simple linear regression model to predict the response. Then you can find my take on the results below. I will also mention in which of the models there is a statistically significant association between the predictor and the response. What will this do? Well, it will help us understand to choose the best model in similar datasets in the future with similar circumstances. Then finally, I will also create some plots to back up your assertions.

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.3.3
```

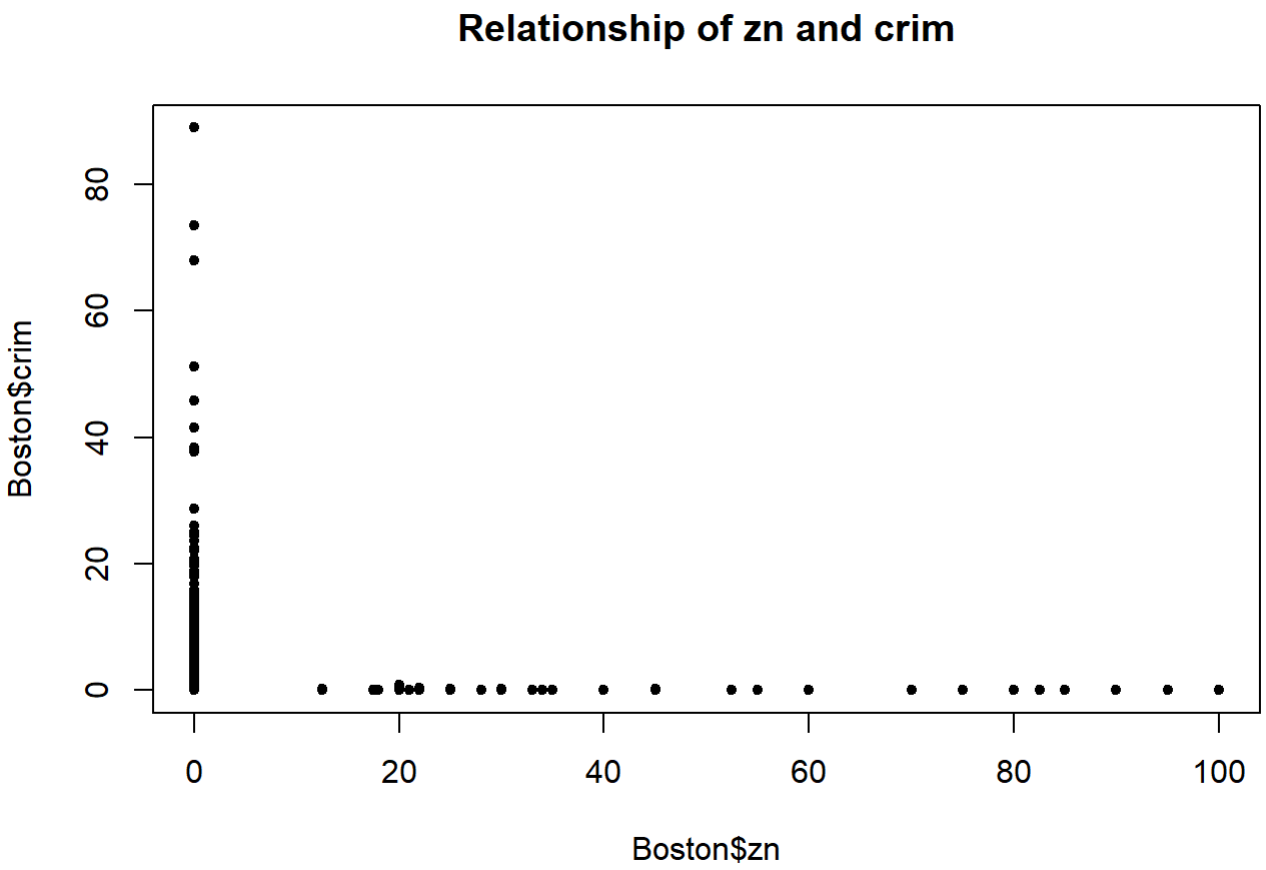
```
data("Boston")
colnames(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

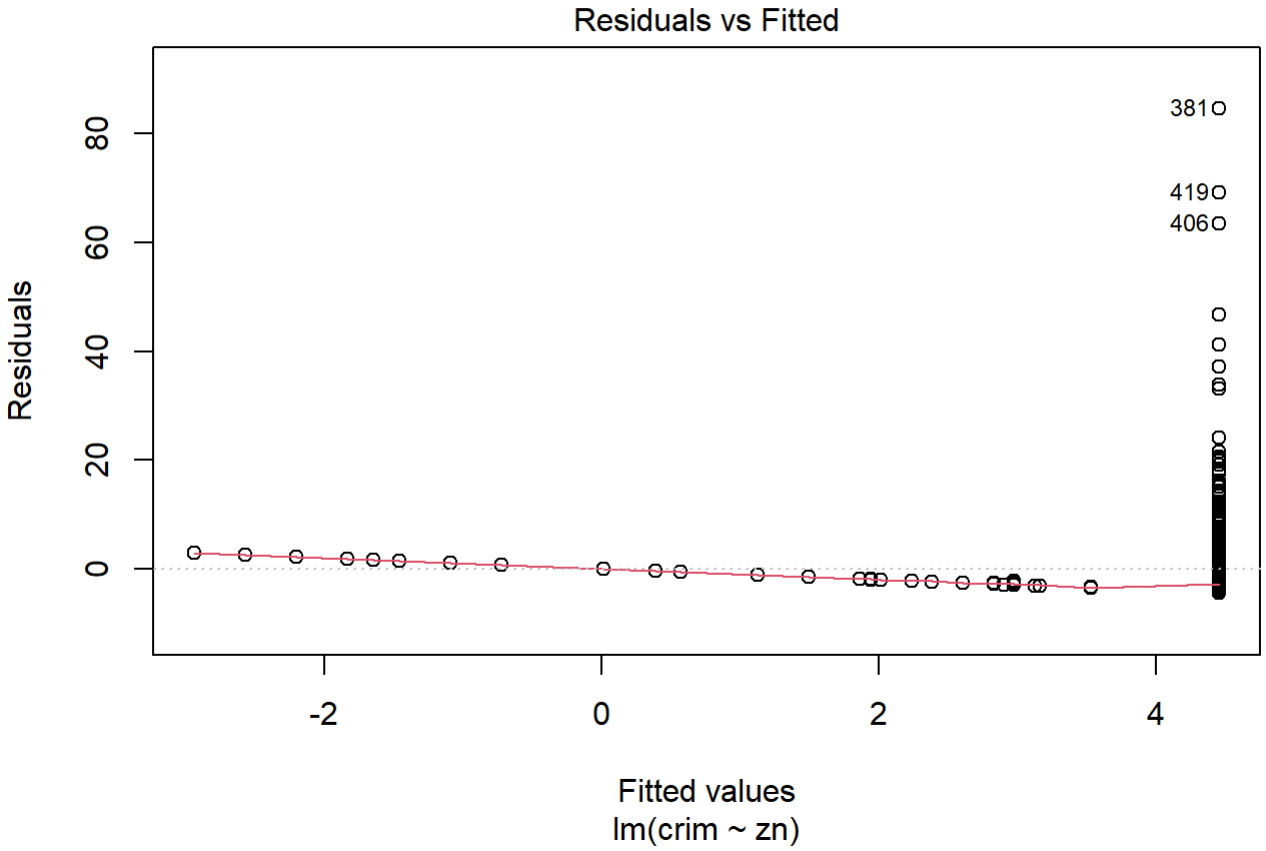
```
lm.fit1<- lm(crim ~ zn, Boston)
summary(lm.fit1)
```

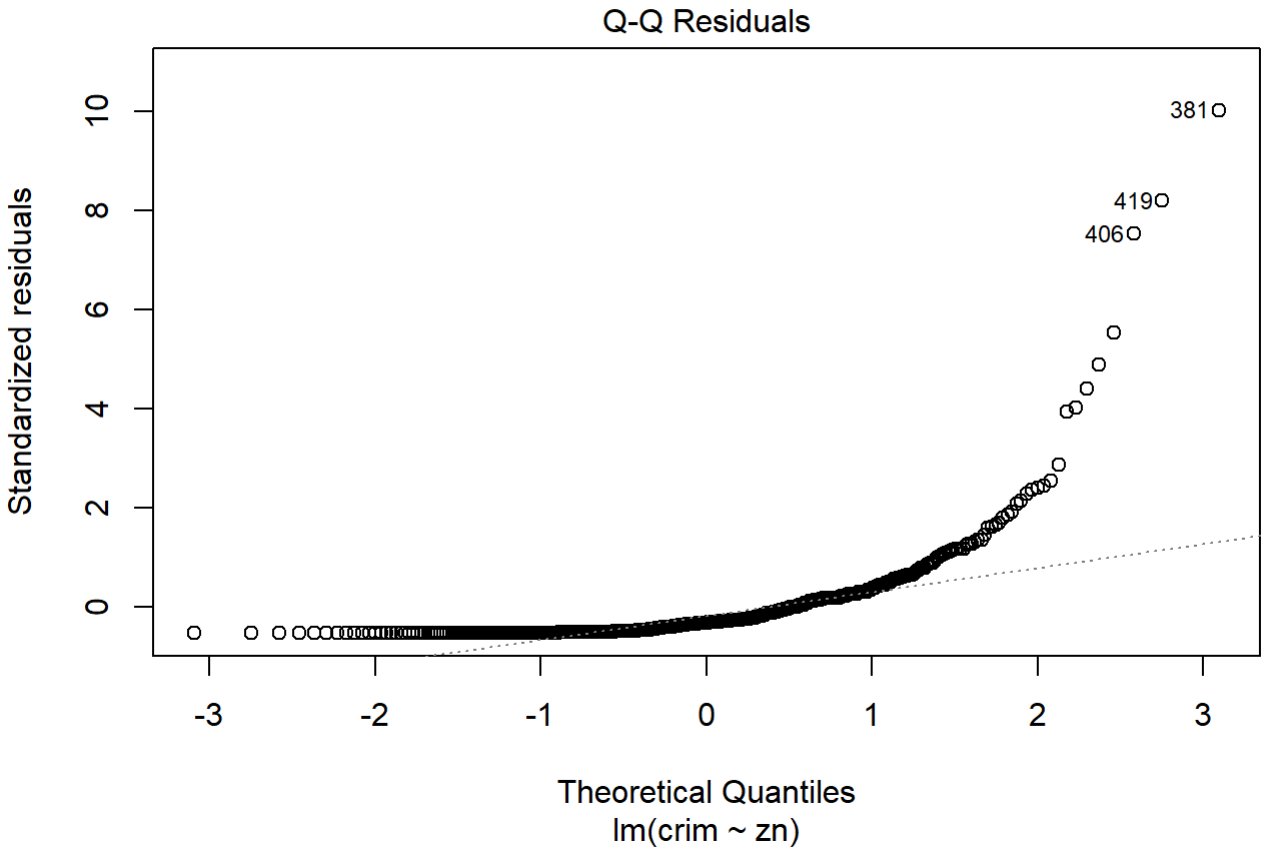
```
##
## Call:
## lm(formula = crim ~ zn, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.429 -4.222 -2.620  1.250 84.523
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.45369    0.41722  10.675  < 2e-16 ***
## zn          -0.07393    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
```

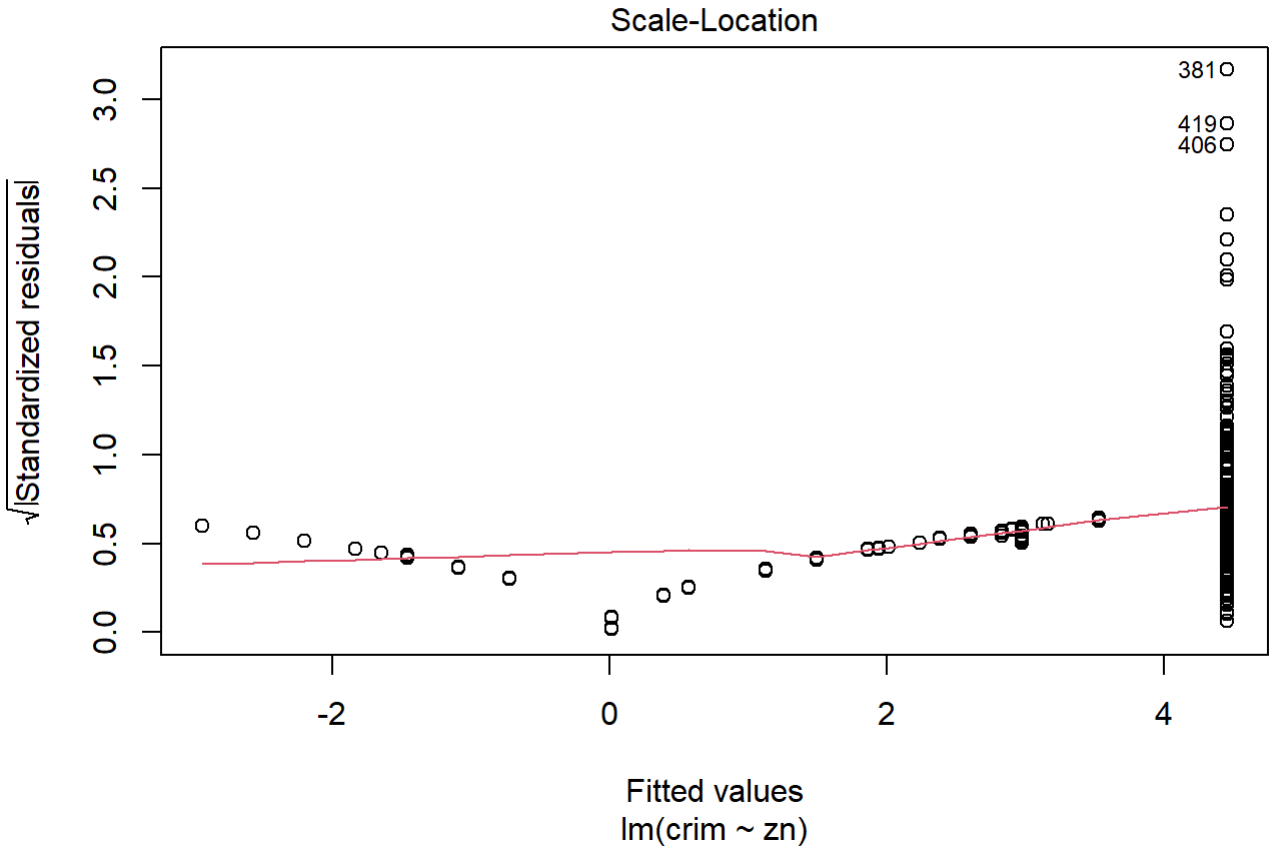
```
plot(Boston$zn, Boston$crim, pch = 20, main = "Relationship of zn and crim")
```

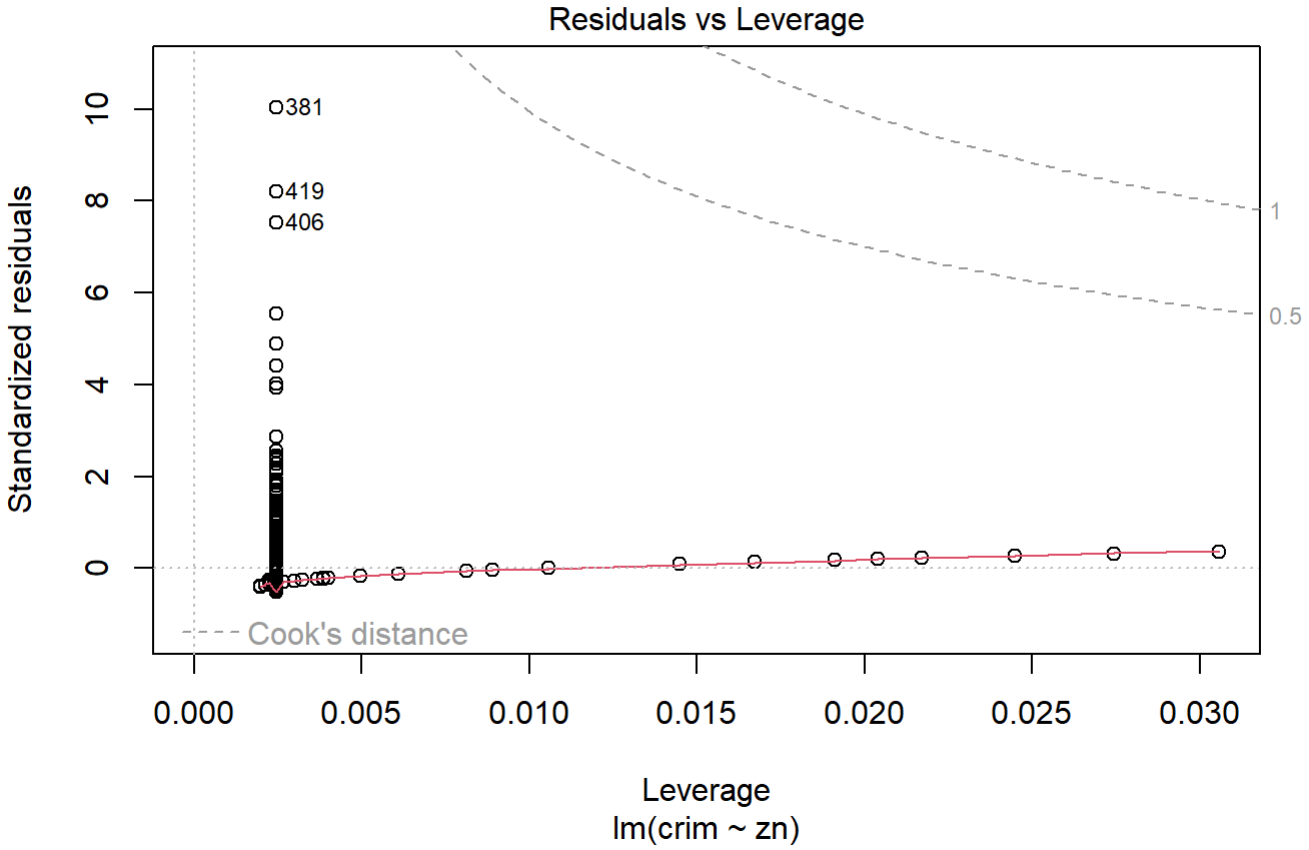



```
plot(lm.fit1)
```


Residuals vs Fitted
Residuals
Fitted values
lm(crim ~ zn)
381
419
406


Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(crim ~ zn)
381
419
406





#Here, we see that the pvalue is low so we can reject null Hyposthesis and determine that there is a significant relationship between zn and crim.

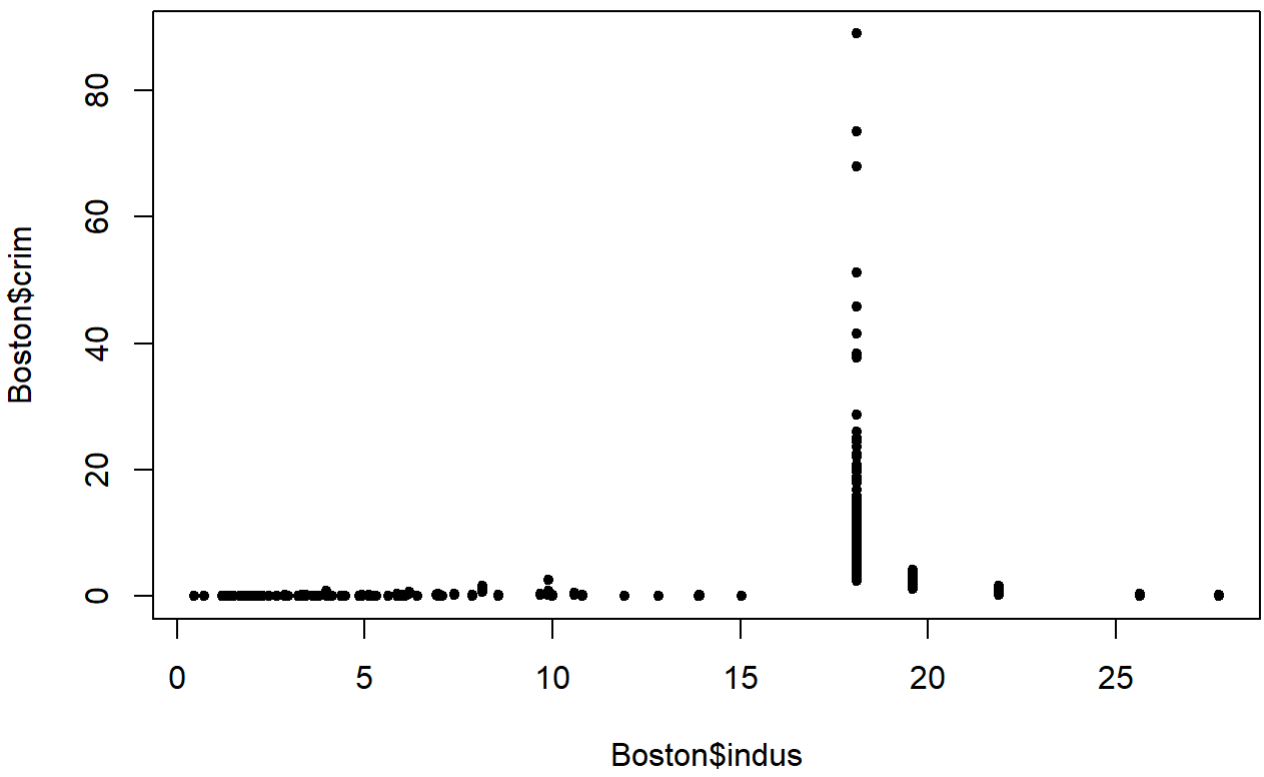
```
library(MASS)
data("Boston")
colnames(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

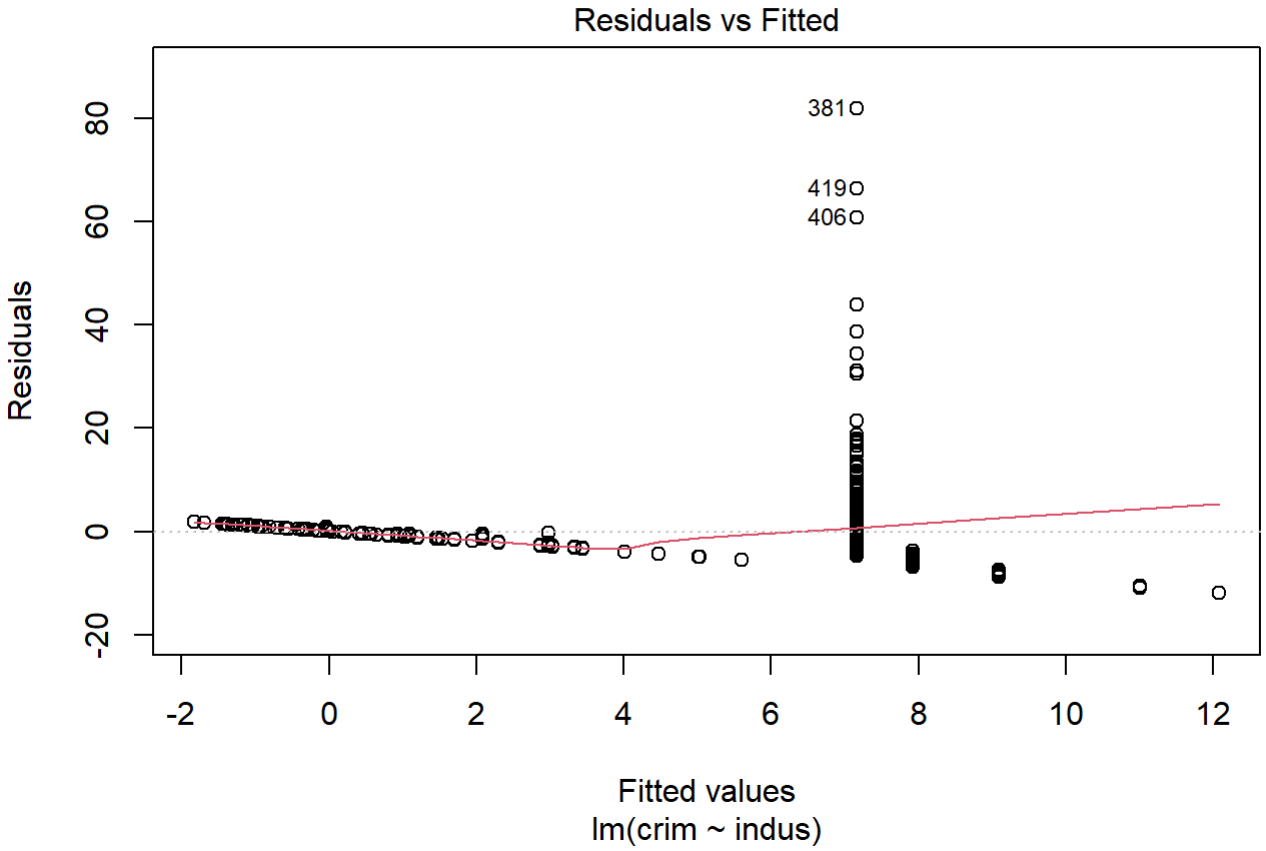
```
lm.fit2<- lm(crim ~ indus, Boston)
summary(lm.fit2)
```

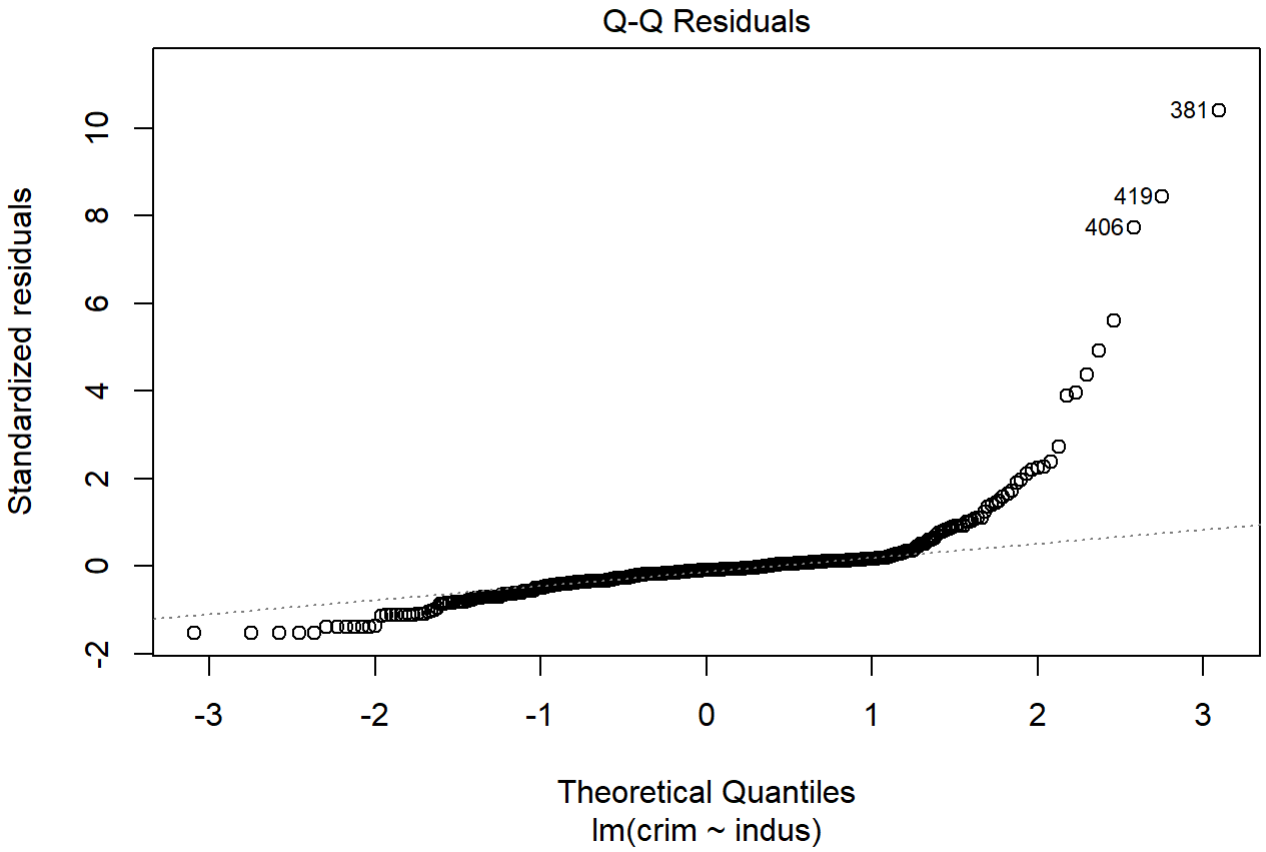
```
##
## Call:
## lm(formula = crim ~ indus, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.972  -2.698  -0.736   0.712  81.813
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.06374    0.66723  -3.093  0.00209 **
## indus        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
```

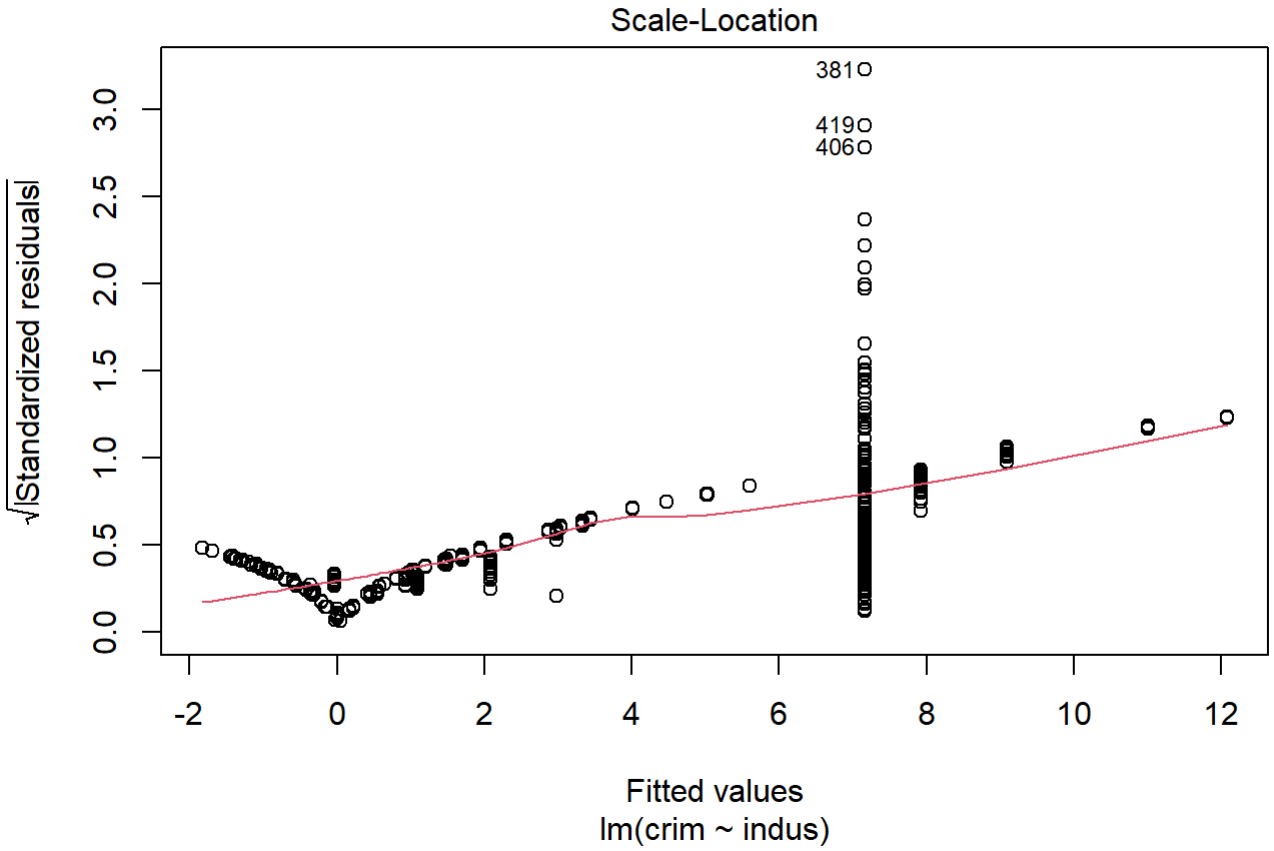
```
plot(Boston$indus, Boston$crim, pch = 20, main = "Relationship of indus and crim")
```

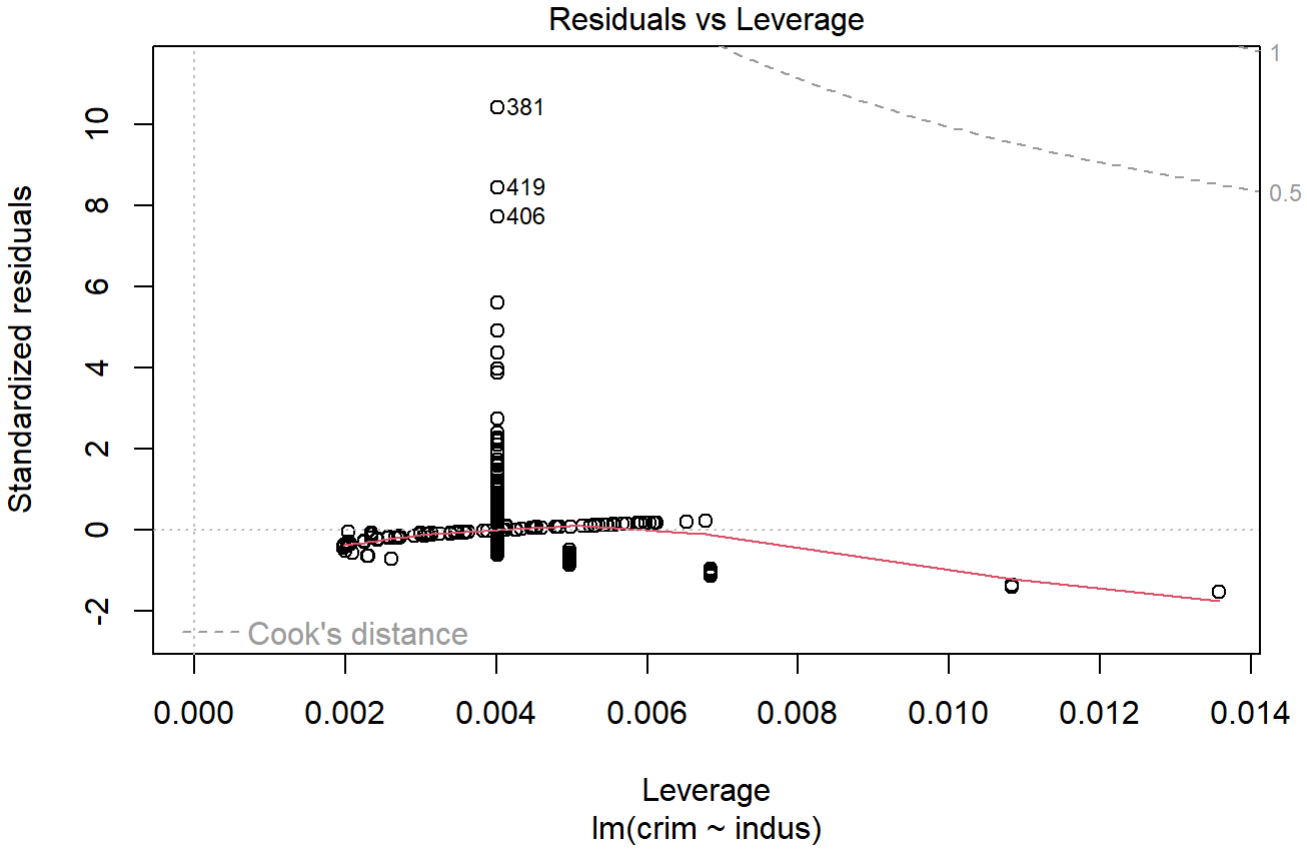

Relationship of indus and crim
Boston$crim
0
20
40
60
80
0
5
10
15
20
25
Boston$indus

```
plot(lm.fit2)
```


Residuals vs Fitted
Residuals
Fitted values
lm(crim ~ indus)
381
419
406


Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(crim ~ indus)
381
419
406





#Here, we see that the pvalue is low so we can reject null Hyposthesis and determine that there is a significant relationship between indus and crim.

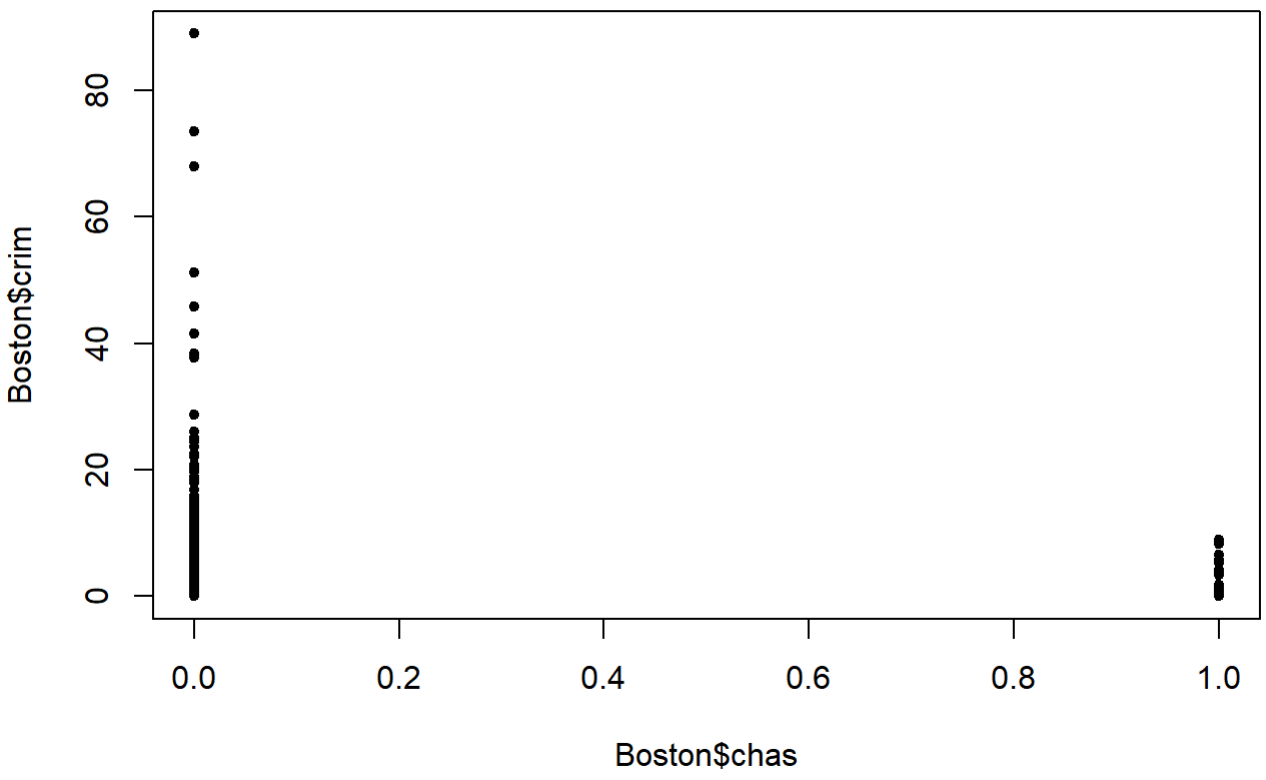
```
library(MASS)
data("Boston")
colnames(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

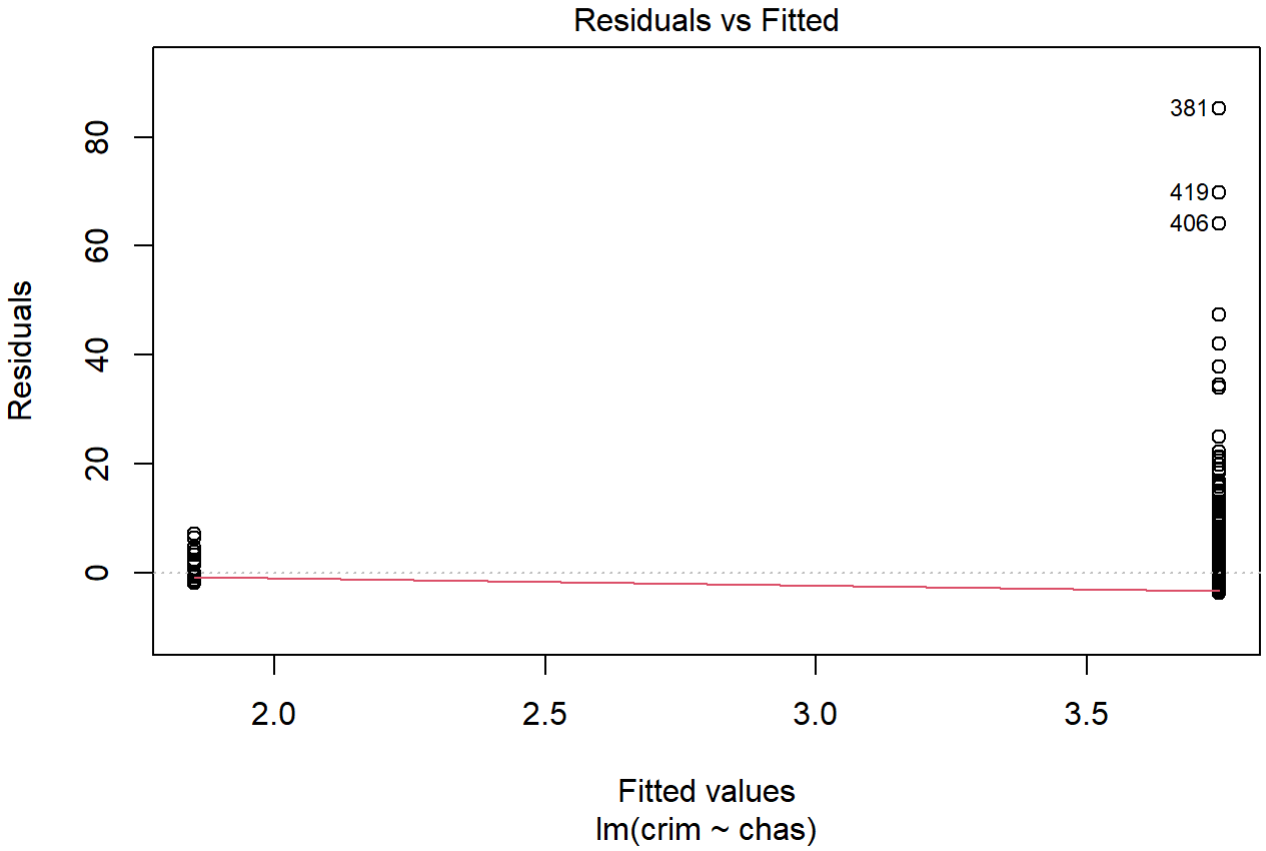
```
lm.fit3<- lm(crim ~ chas, Boston)
summary(lm.fit3)
```

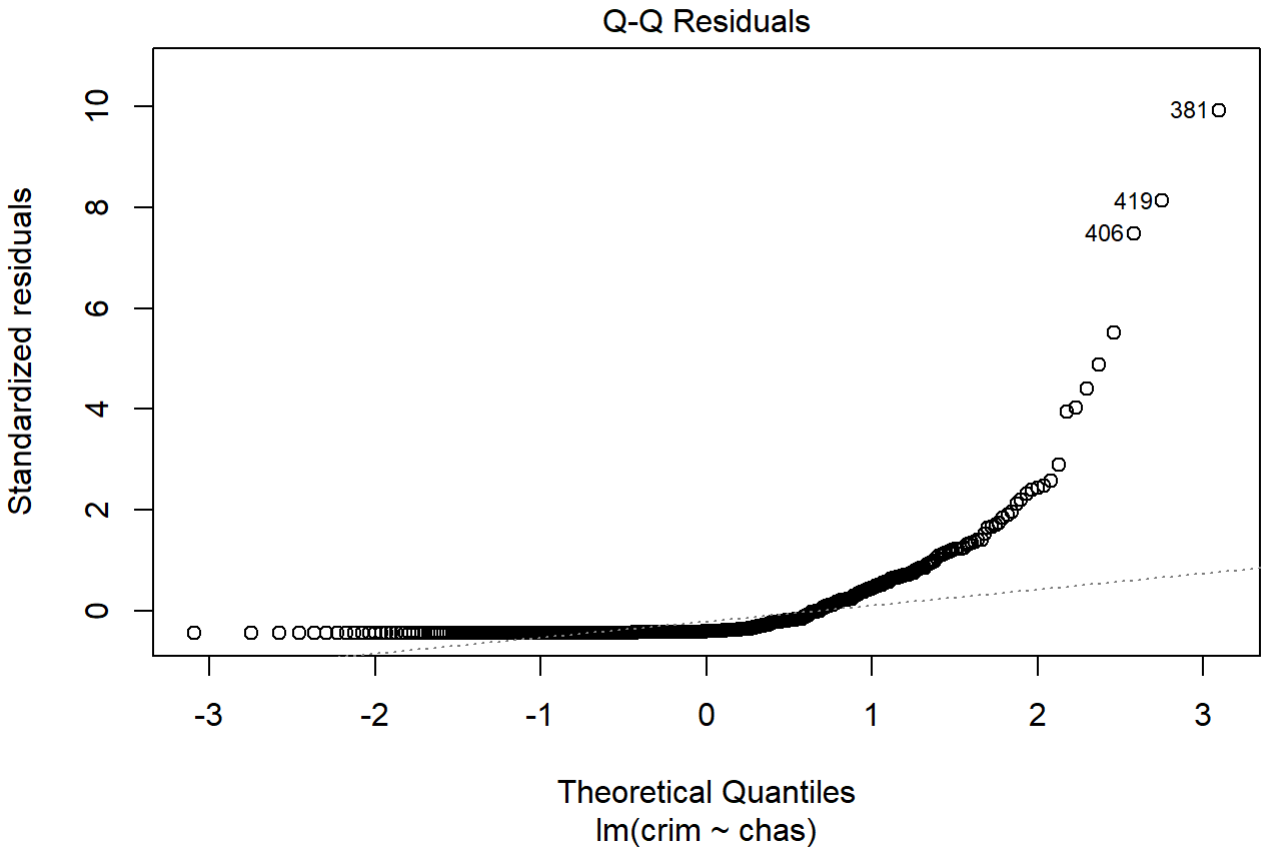
```
##
## Call:
## lm(formula = crim ~ chas, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.738 -3.661 -3.435  0.018 85.232
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7444     0.3961   9.453   <2e-16 ***
## chas         -1.8928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

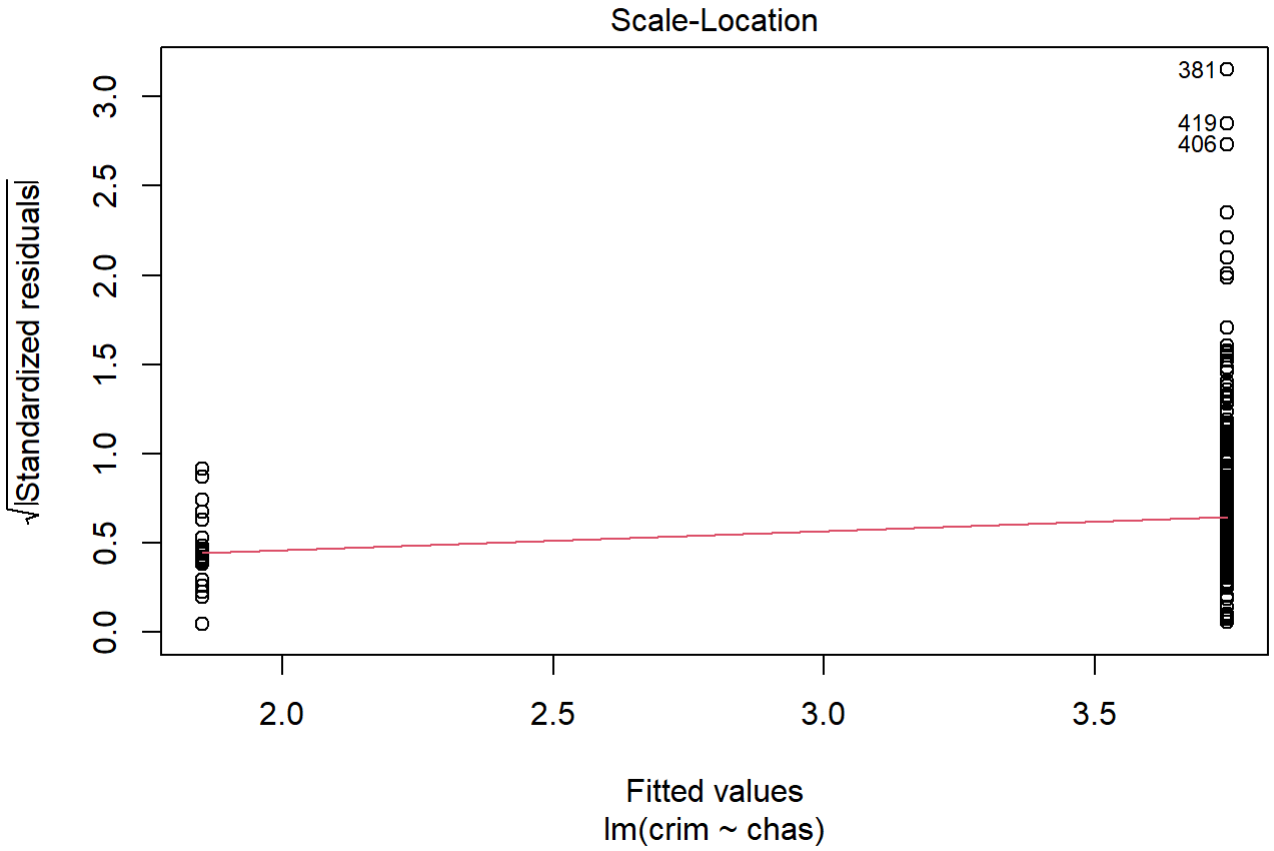
```
plot(Boston$chas, Boston$crim, pch = 20, main = "Relationship of chas and crim")
```

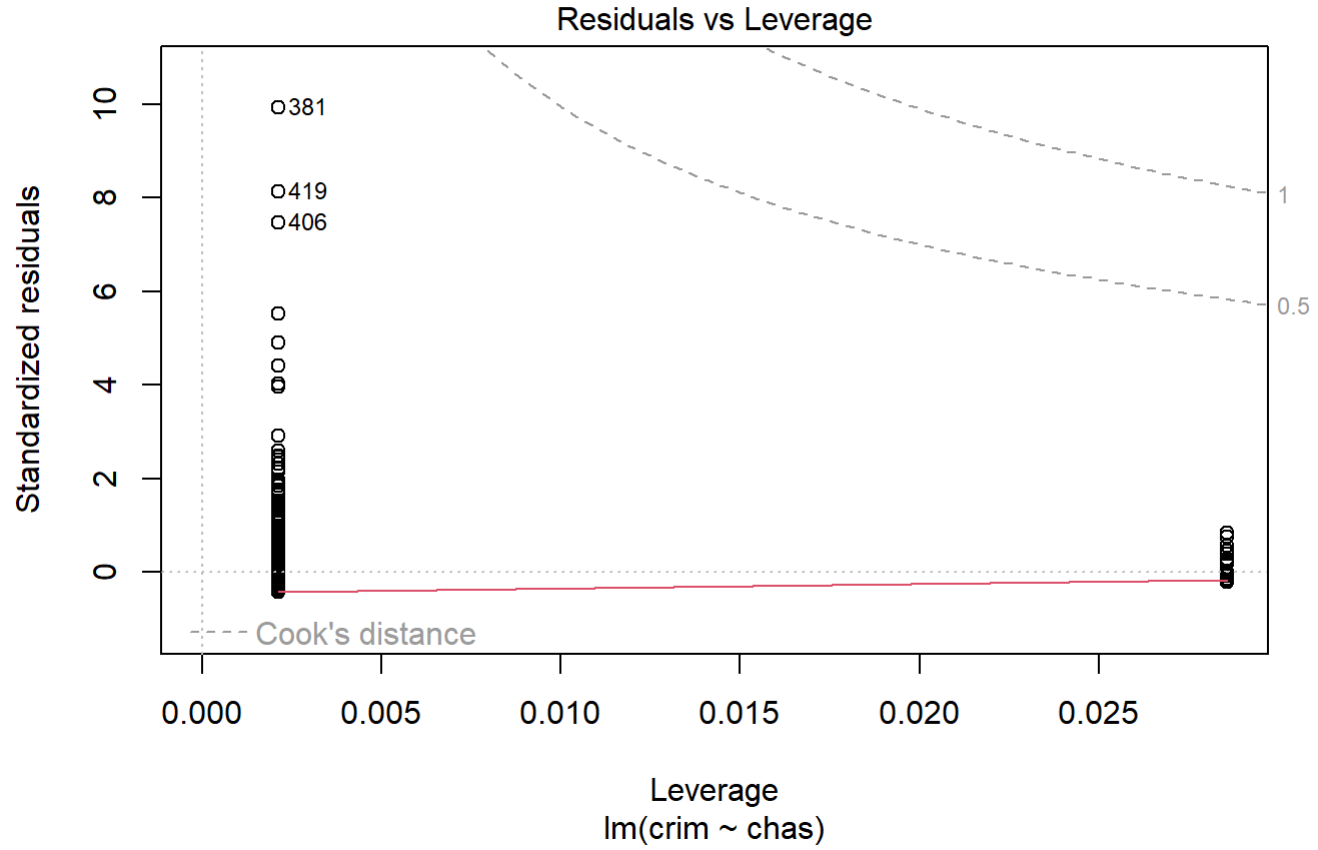

Relationship of chas and crim
Boston$crim
0
20
40
60
80
0.0
0.2
0.4
0.6
0.8
1.0
Boston$chas

```
plot(lm.fit3)
```


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)


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





#Here, p-value is greater than 0.05 so we cannot reject null hypothesis. Thus, there is no significant relationship between chas and crim

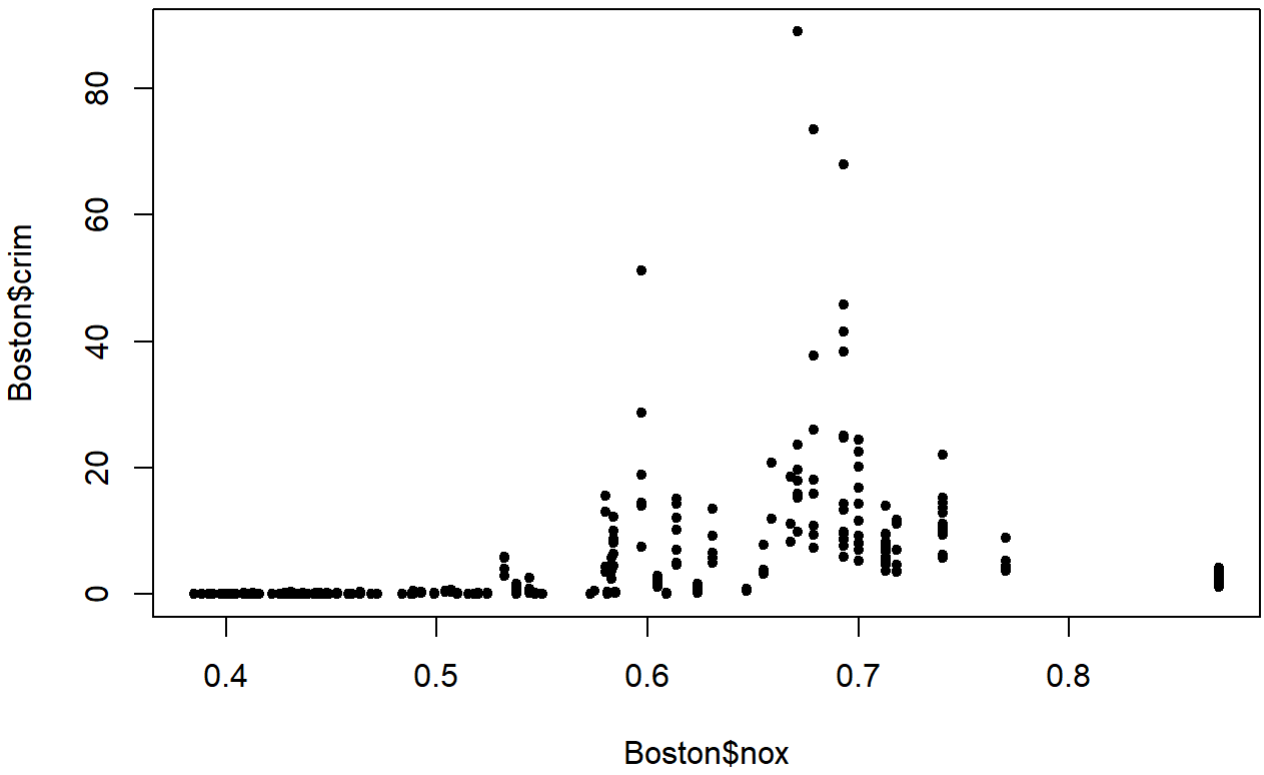
```
library(MASS)
data("Boston")
colnames(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

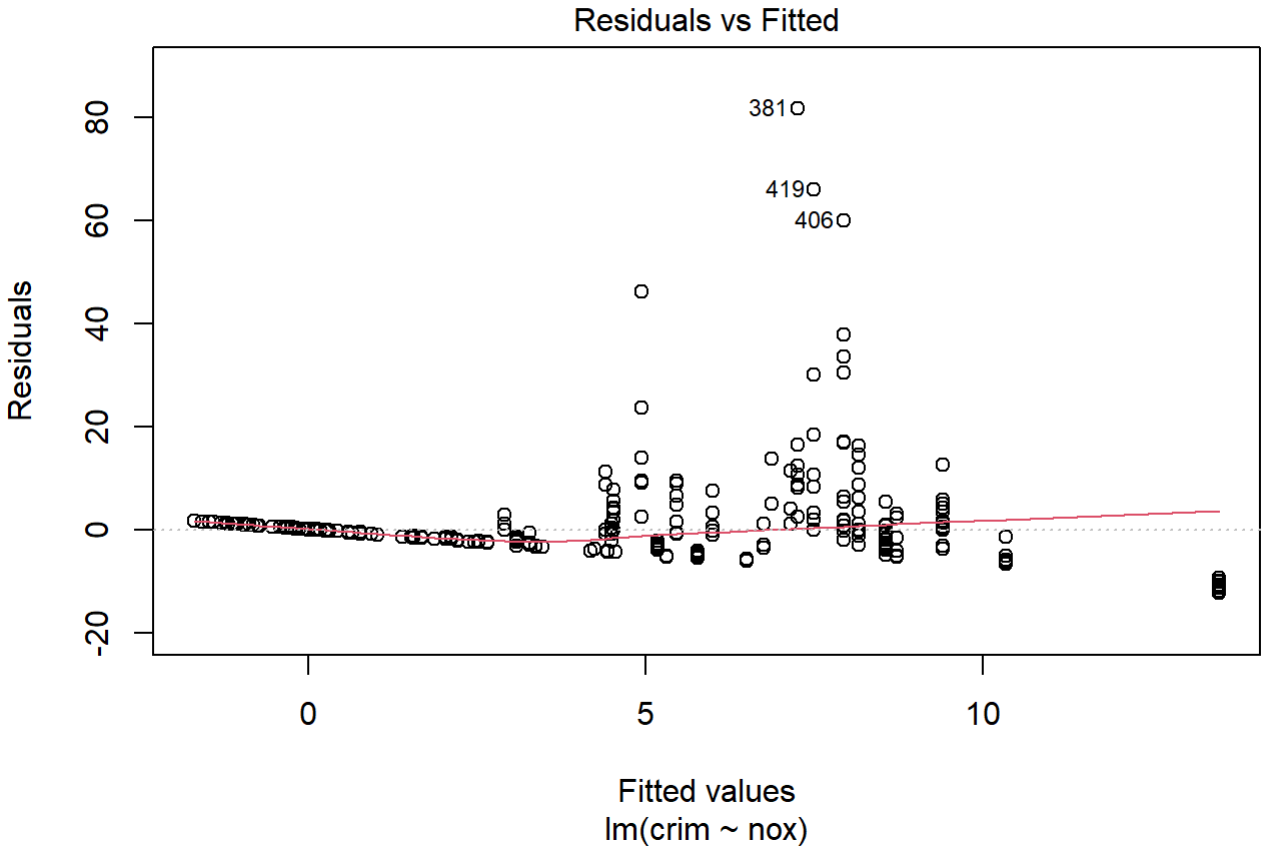
```
lm.fit4<- lm(crim ~ nox, Boston)
summary(lm.fit4)
```

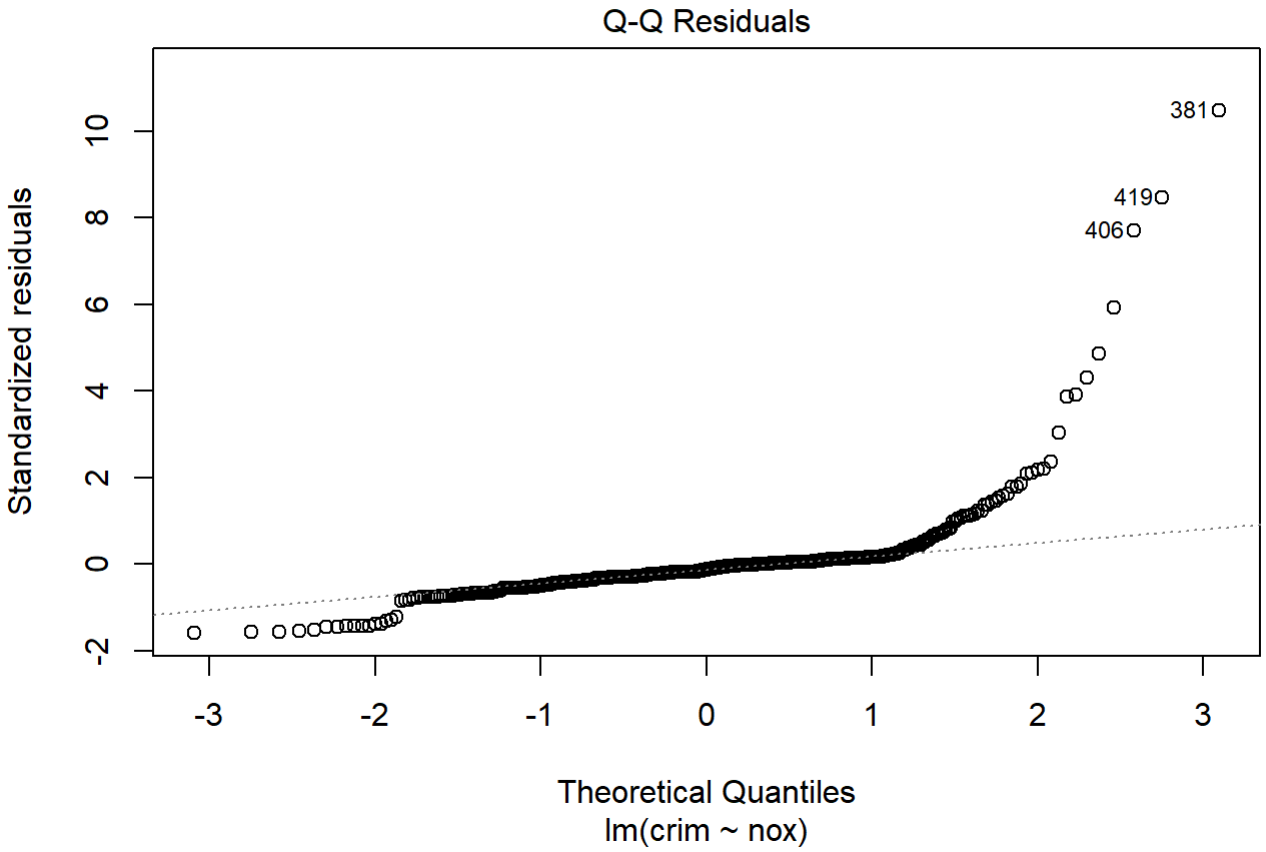
```
##
## Call:
## lm(formula = crim ~ nox, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.371  -2.738  -0.974   0.559  81.728
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -13.720      1.699  -8.073 5.08e-15 ***
## nox           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
```

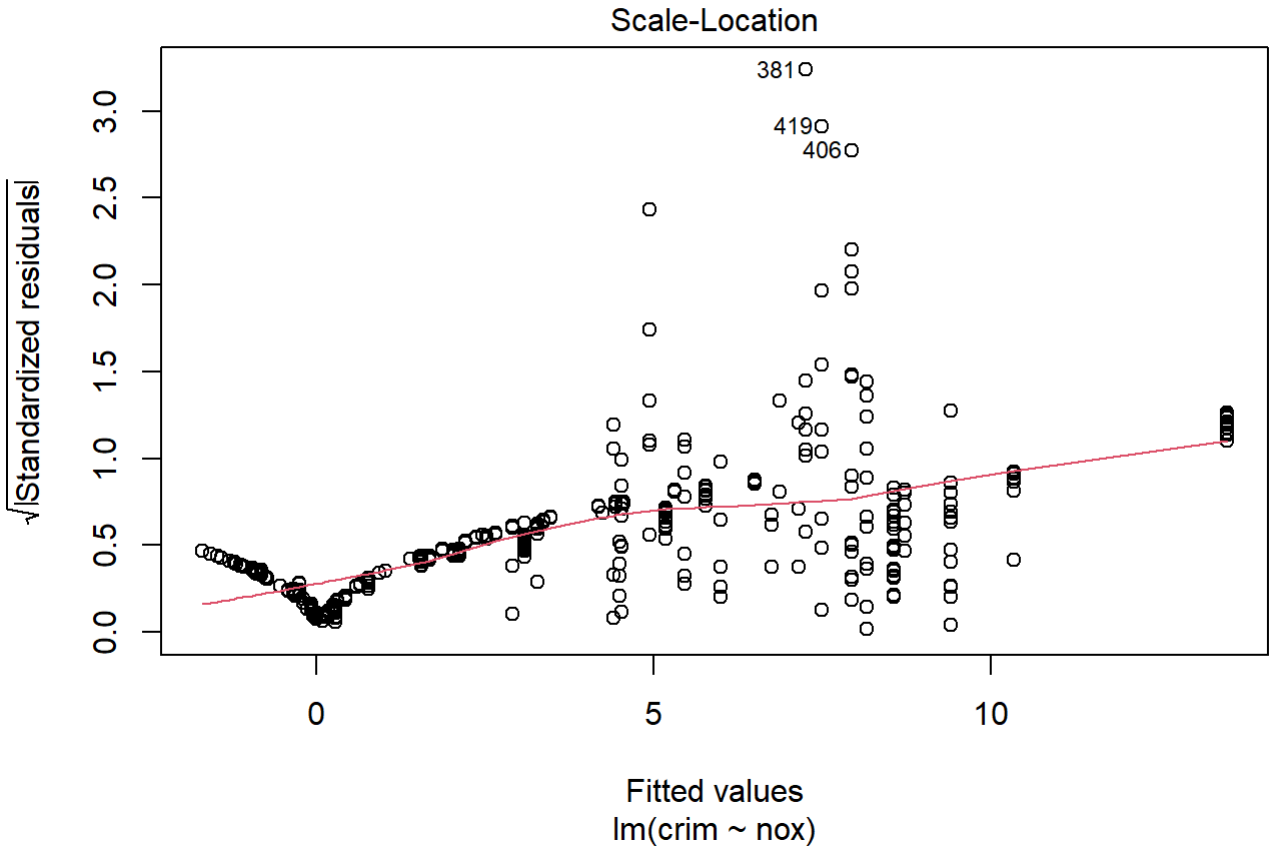
```
plot(Boston$nox, Boston$crim, pch = 20, main = "Relationship of nox and crim")
```

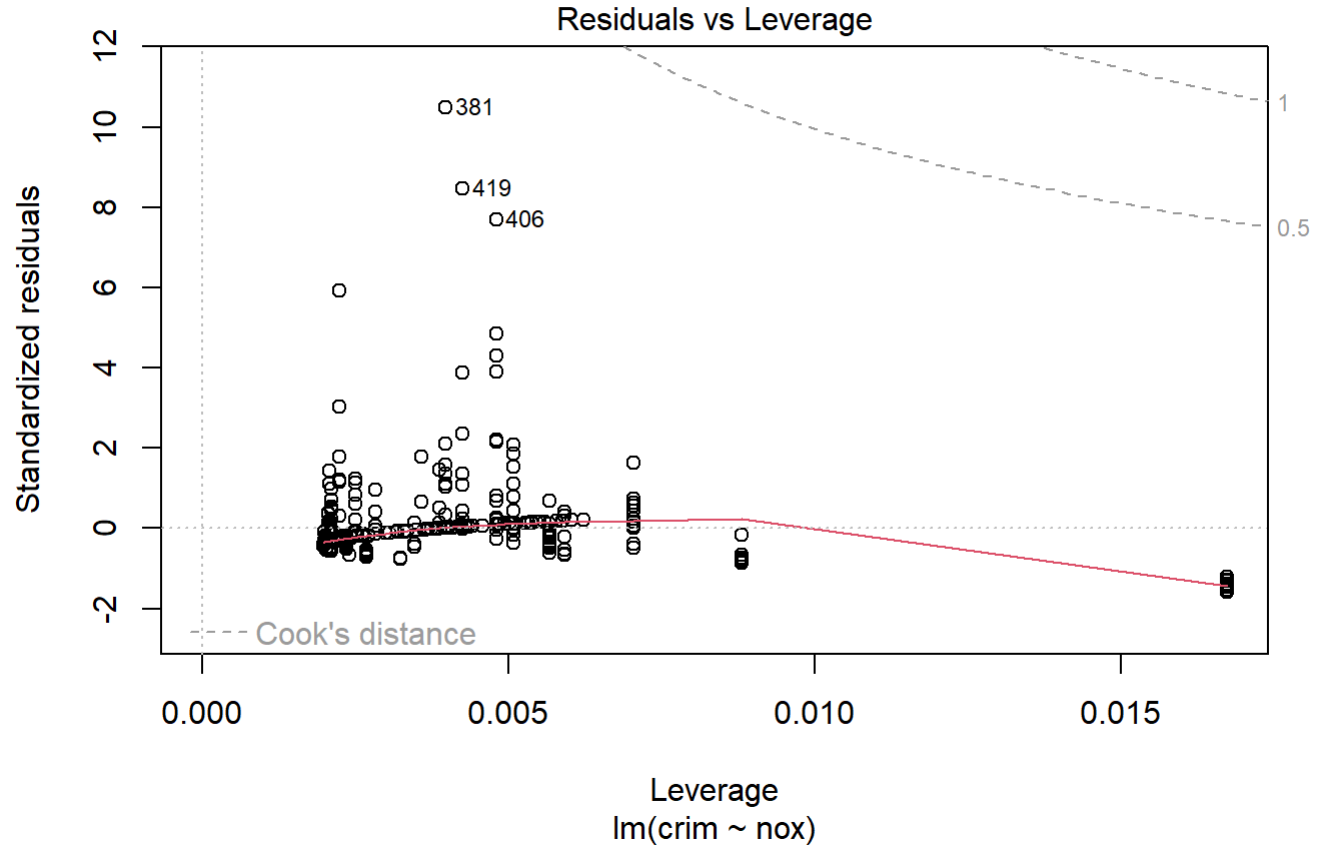

Relationship of nox and crim
Boston$crim
0
20
40
60
80
0.4
0.5
0.6
0.7
0.8
Boston$nox

```
plot(lm.fit4)
```


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


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





#Here, we see that the pvalue is low so we can reject null Hyposthesis and determine that there is a significant relationship between nox and crim.

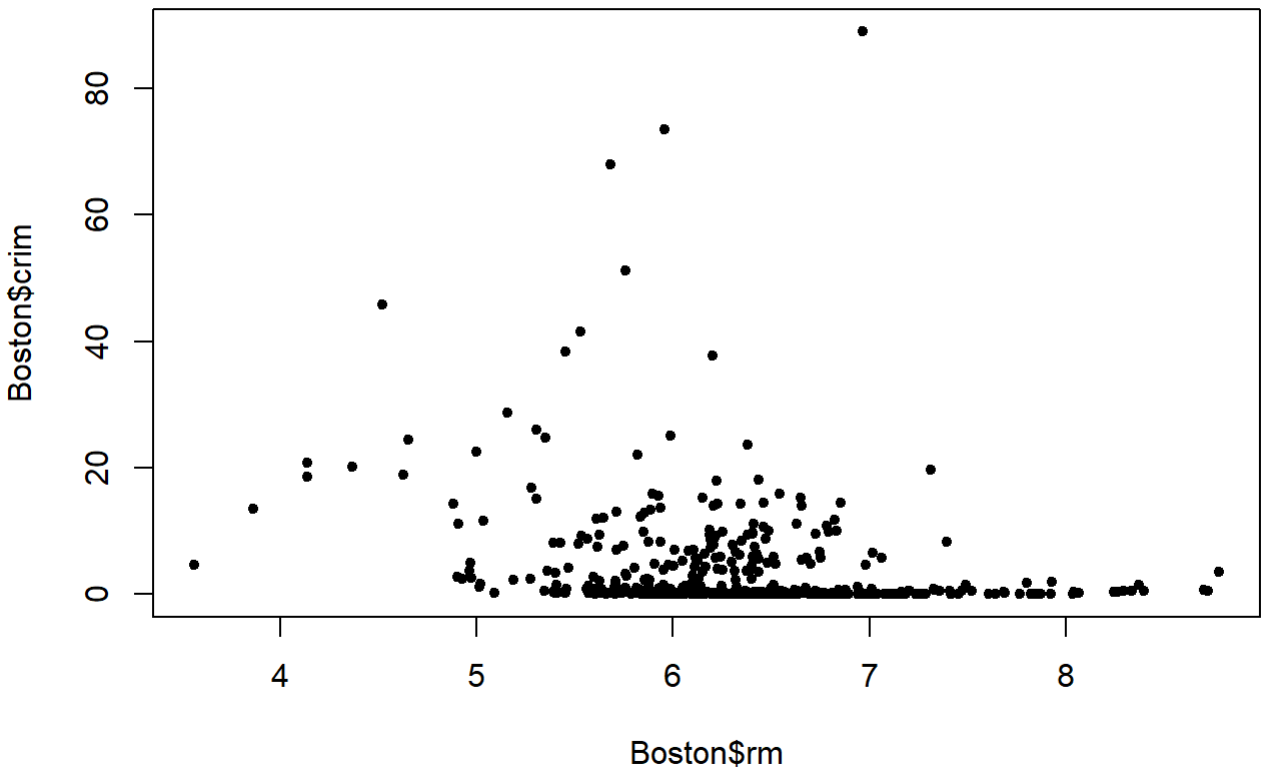
```
library(MASS)
data("Boston")
colnames(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

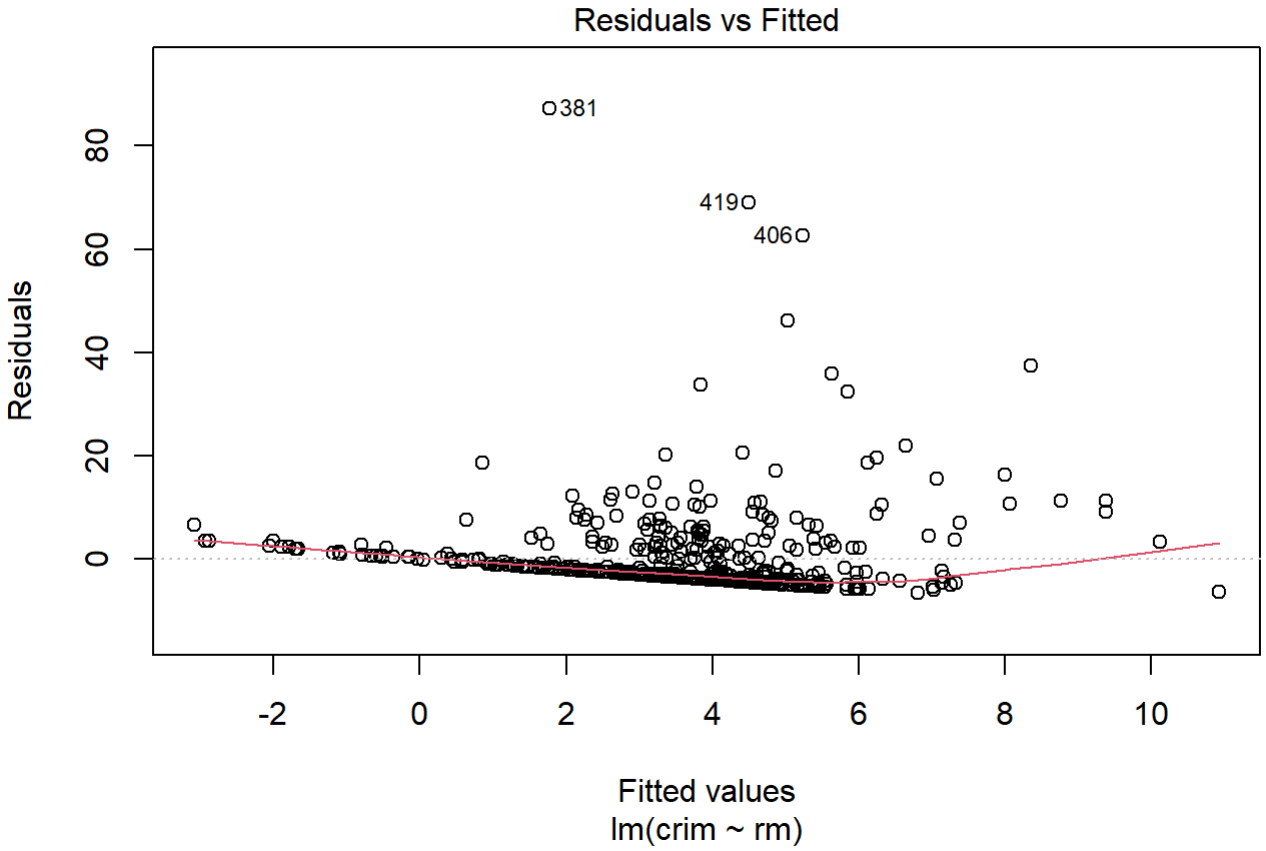
```
lm.fit5<- lm(crim ~ rm, Boston)
summary(lm.fit5)
```

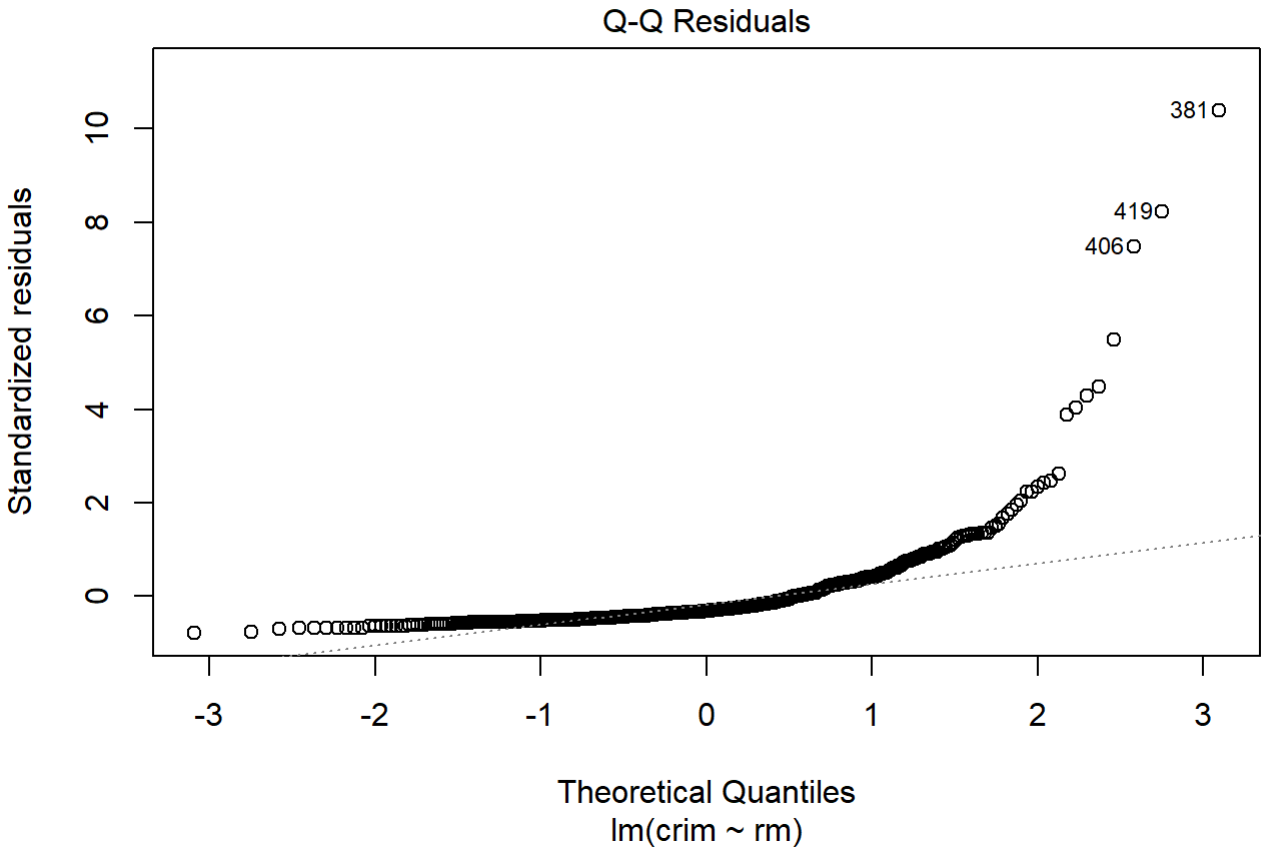
```
##
## Call:
## lm(formula = crim ~ rm, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.604 -3.952 -2.654  0.989 87.197
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.482      3.365   6.088 2.27e-09 ***
## rm            -2.684      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
```

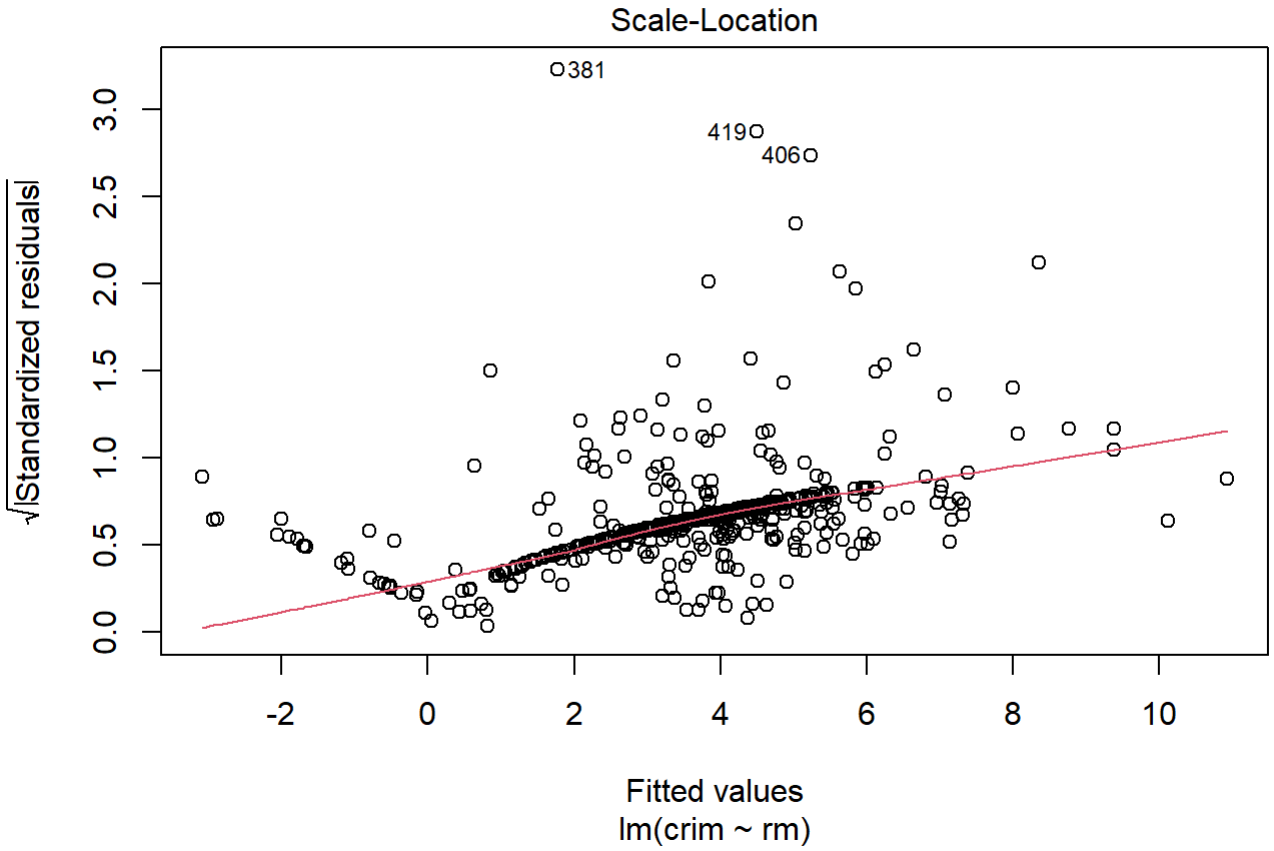
```
plot(Boston$rm, Boston$crim, pch = 20, main = "Relationship of rm and crim")
```

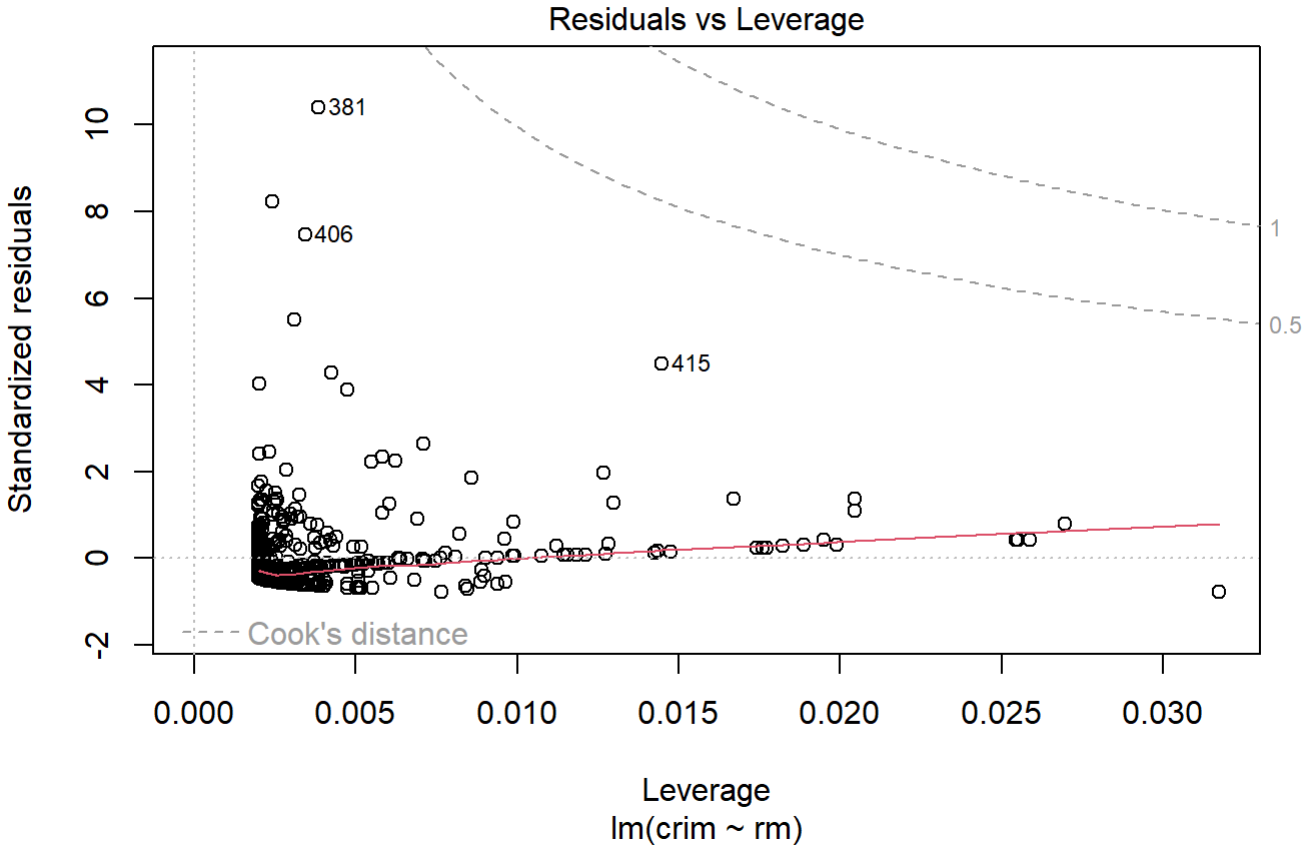

Relationship of rm and crim
Boston$crim
0
20
40
60
80
4
5
6
7
8
Boston$rm

```
plot(lm.fit5)
```


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


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





#Here, p

value was low so null hypothesis could be rejected, but R squared value and adjusted R squared value is also low so relationship between rm and crim is not significant.

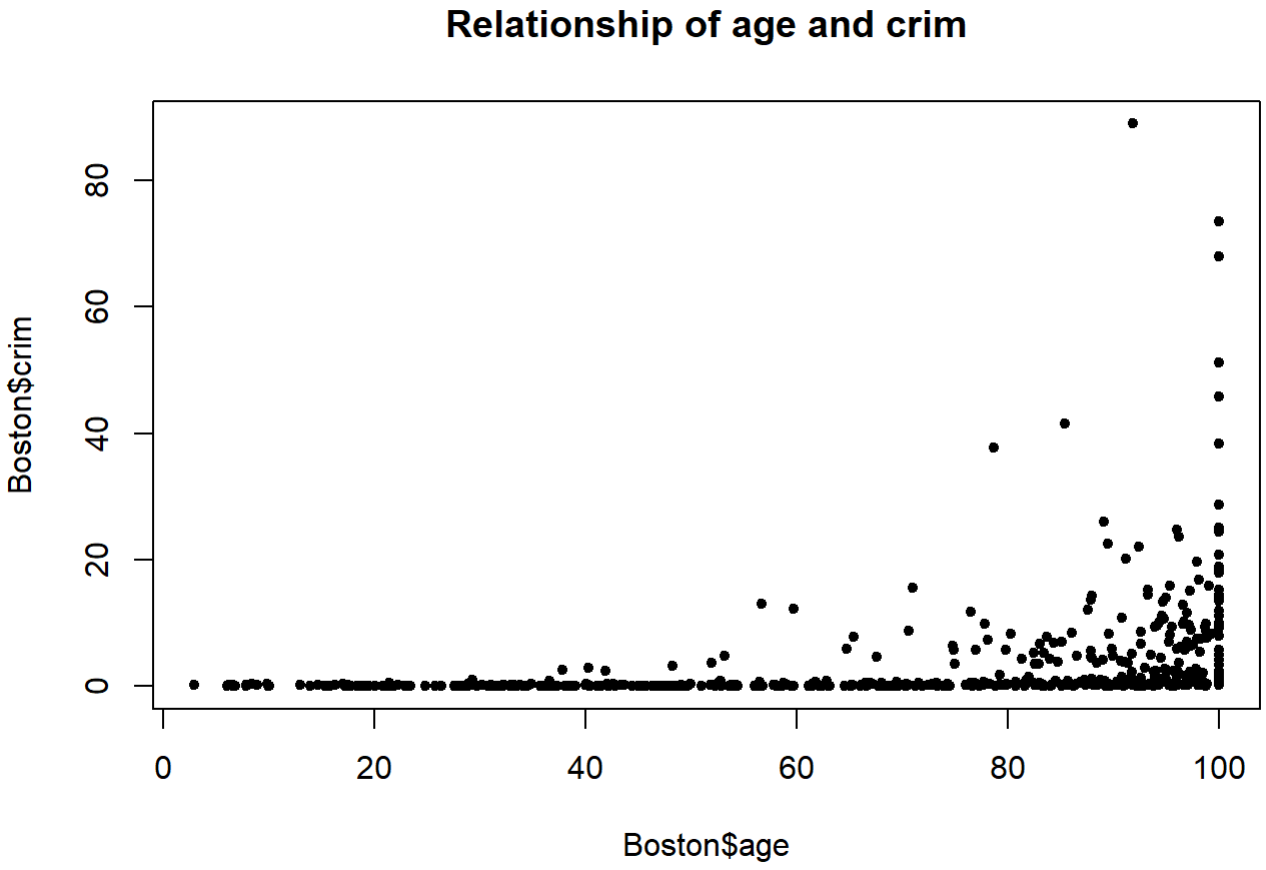
```
library(MASS)
data("Boston")
colnames(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

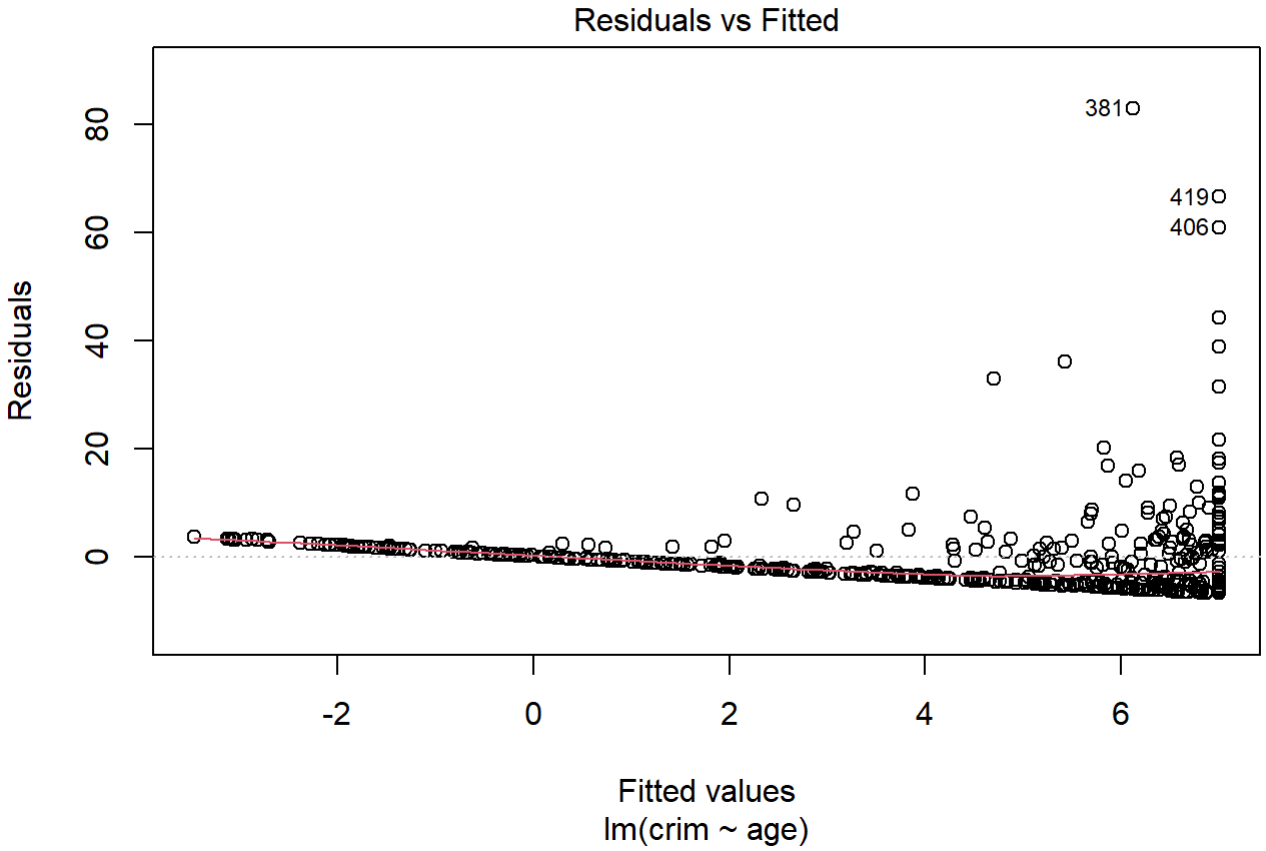
```
lm.fit6<- lm(crim ~ age, Boston)
summary(lm.fit6)
```

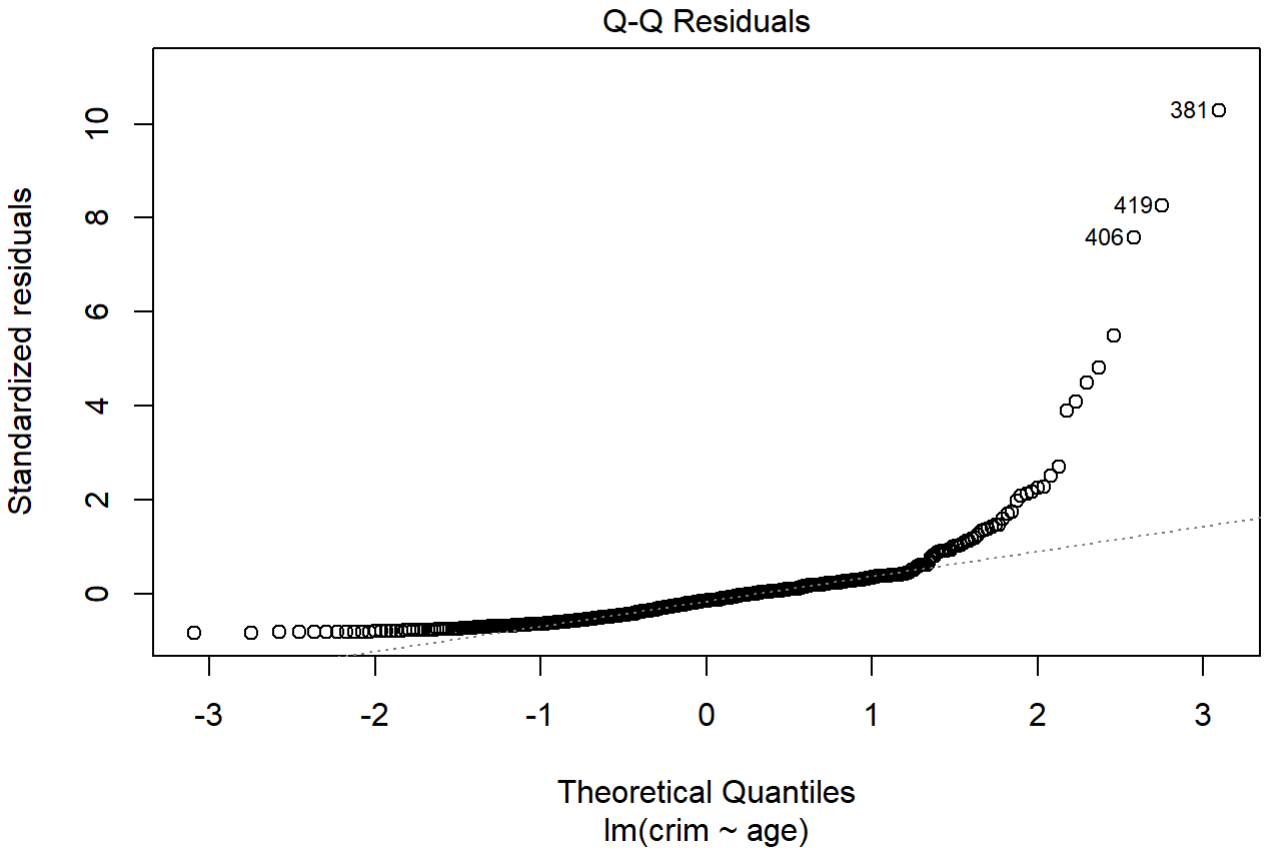
```
##
## Call:
## lm(formula = crim ~ age, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.789 -4.257 -1.230  1.527 82.849
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.77791    0.94398  -4.002 7.22e-05 ***
## age          0.10779    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
```

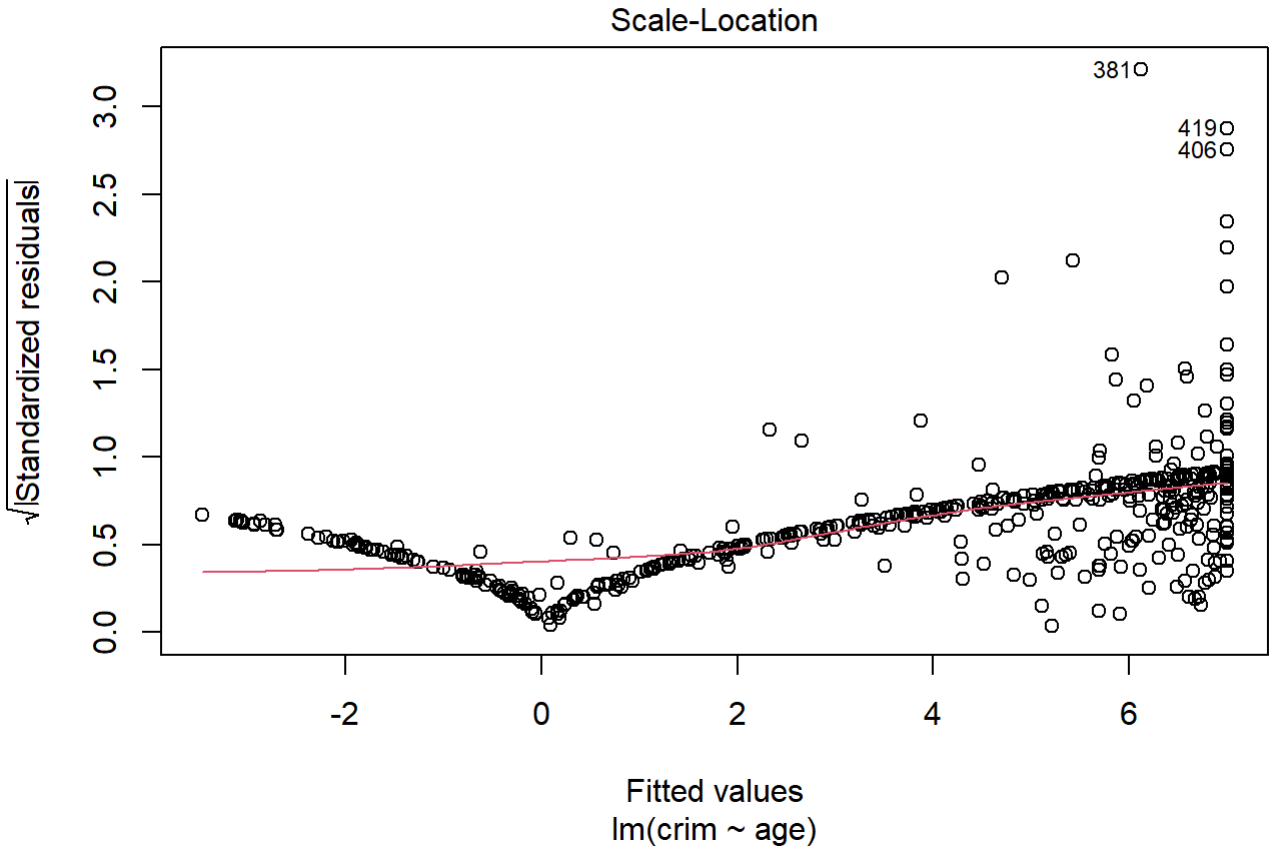
```
plot(Boston$age, Boston$crim, pch = 20, main = "Relationship of age and crim")
```

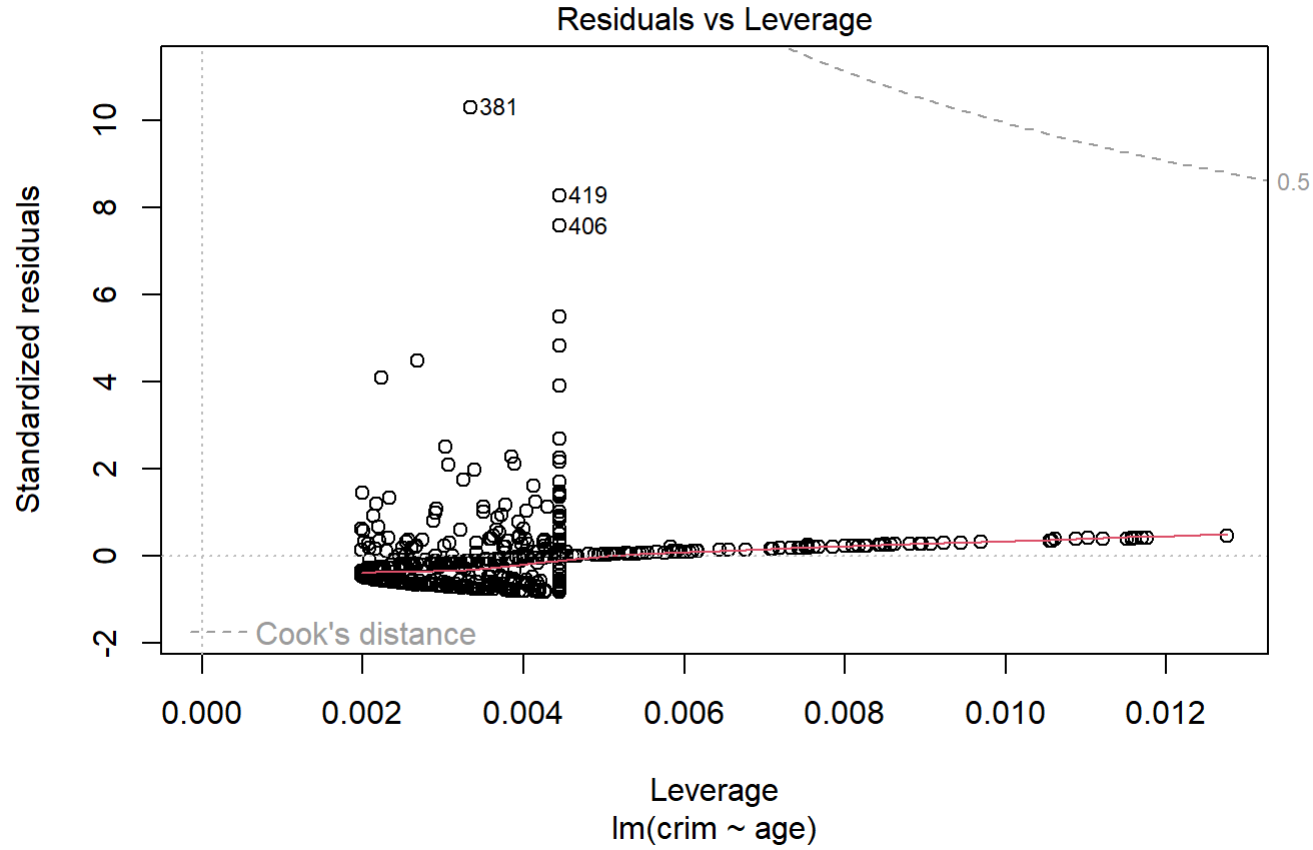

Relationship of age and crim
Boston$crim
0
20
40
60
80
Boston$age
0
20
40
60
80
100

```
plot(lm.fit6)
```


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


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





#Here, p

value was low so null hypothesis could be rejected, but R squared value and adjusted R squared value is also low so relationship between age and crim is not significant.

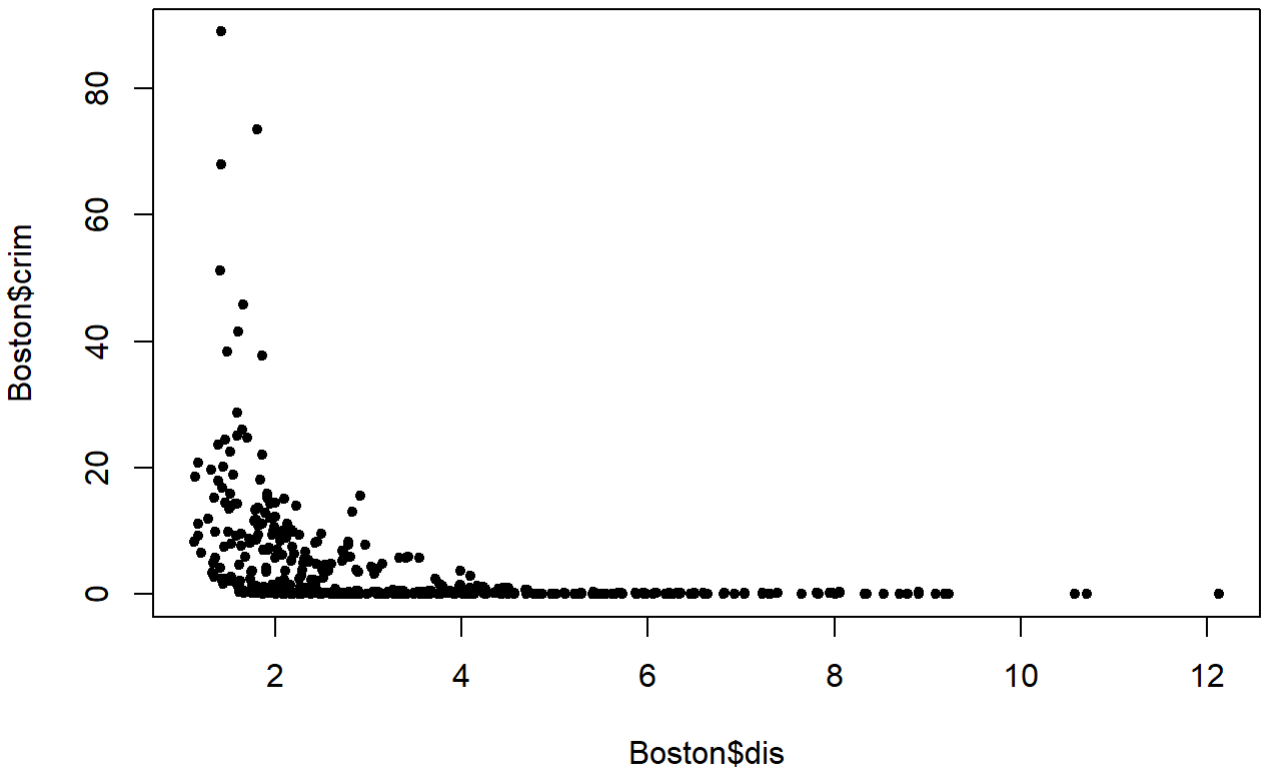
```
library(MASS)
data("Boston")
colnames(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

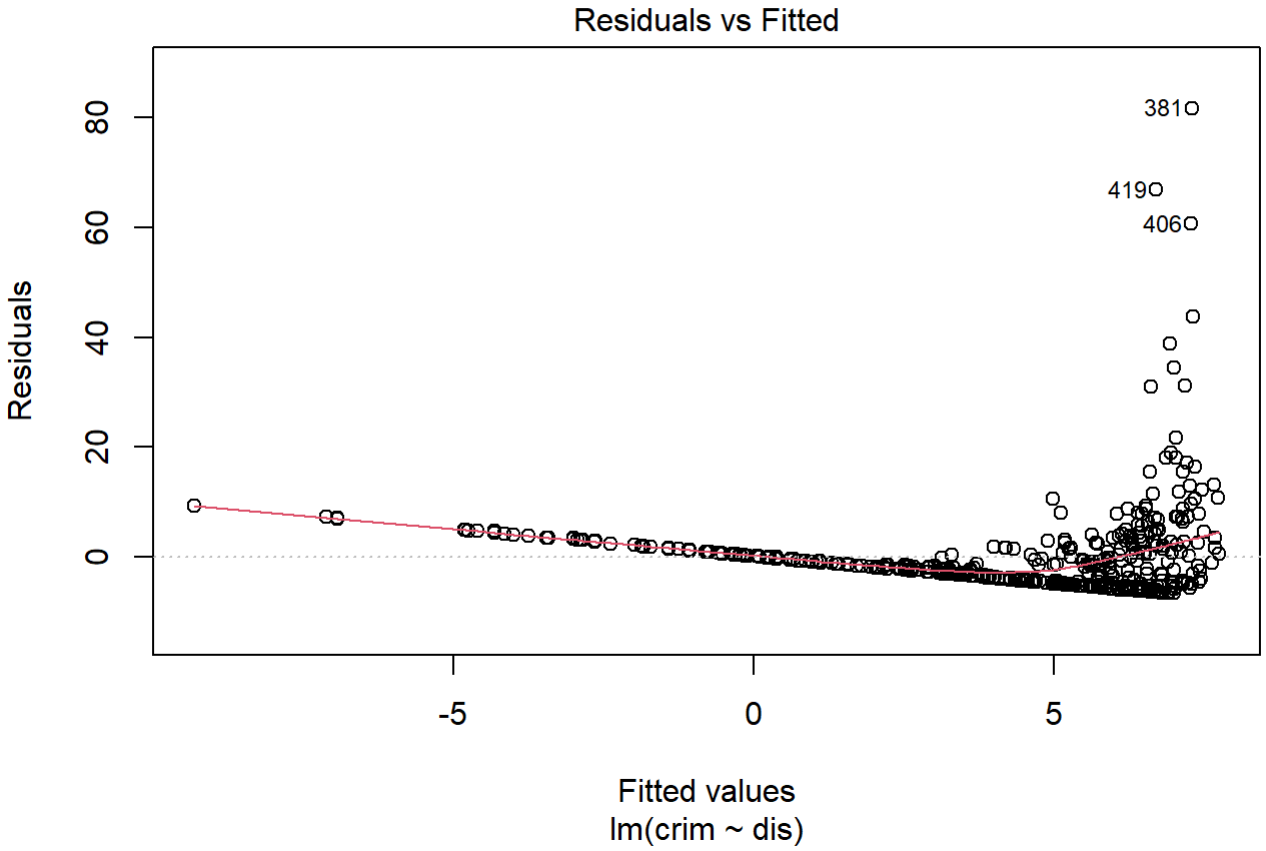
```
lm.fit7<- lm(crim ~ dis, Boston)
summary(lm.fit7)
```

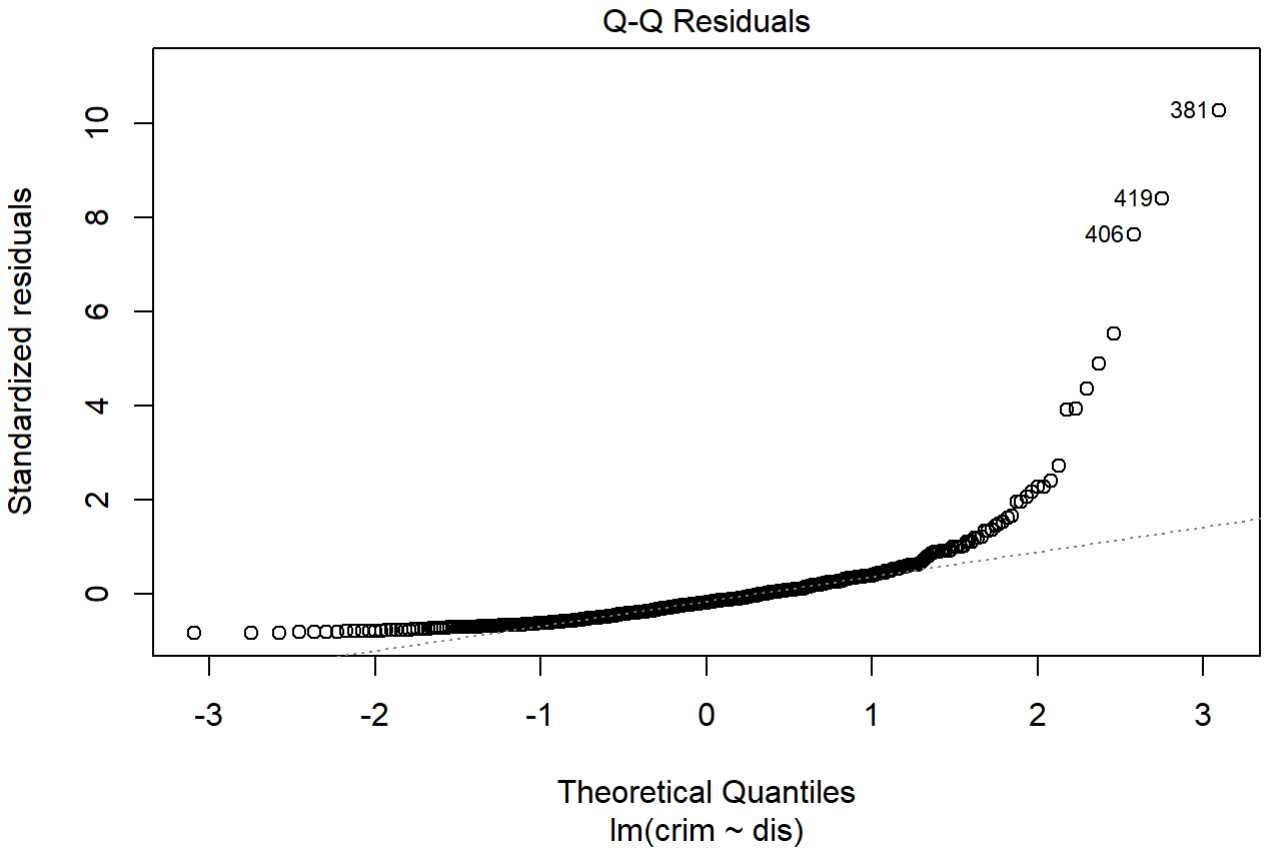
```
##
## Call:
## lm(formula = crim ~ dis, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.708 -4.134 -1.527  1.516 81.674
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.4993     0.7304  13.006   <2e-16 ***
## dis          -1.5509     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
```

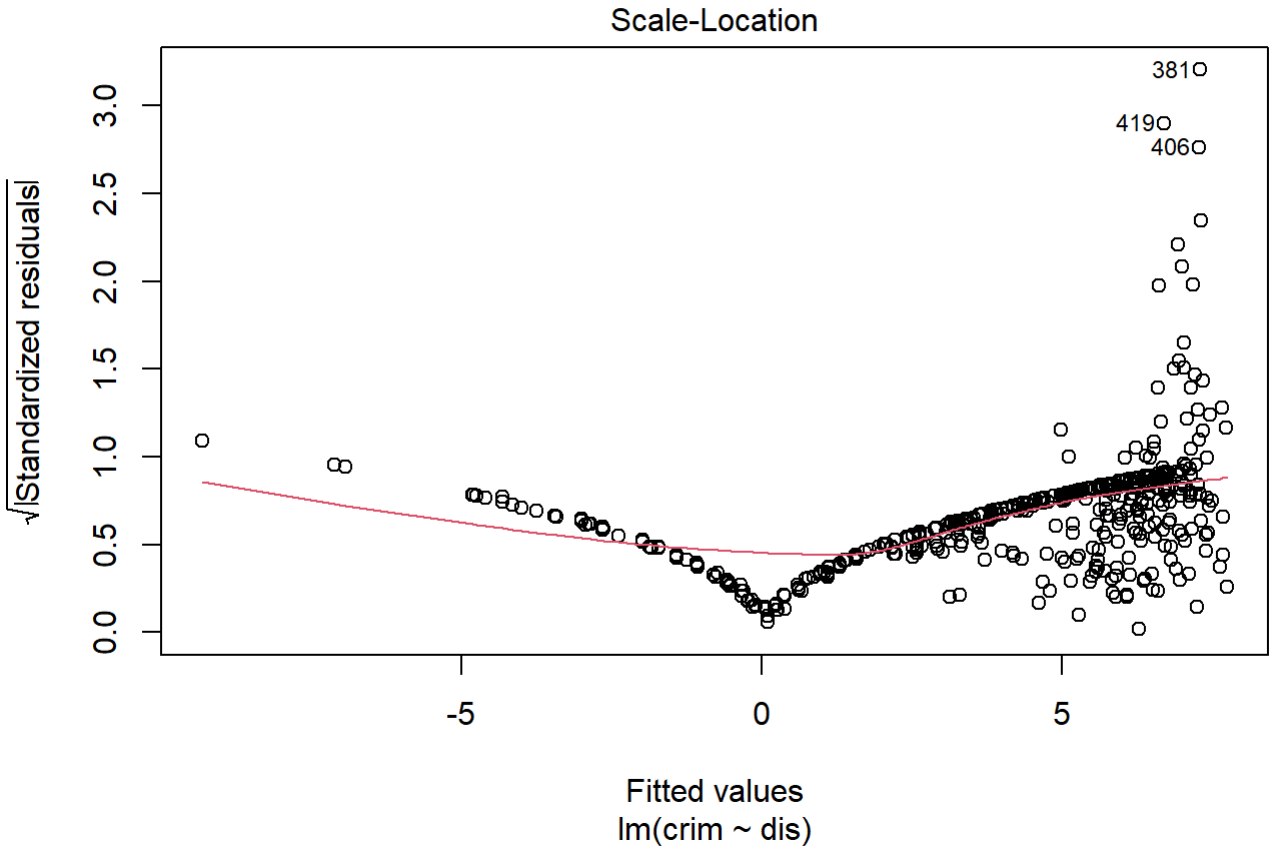
```
plot(Boston$dis, Boston$crim, pch = 20, main = "Relationship of dis and crim")
```

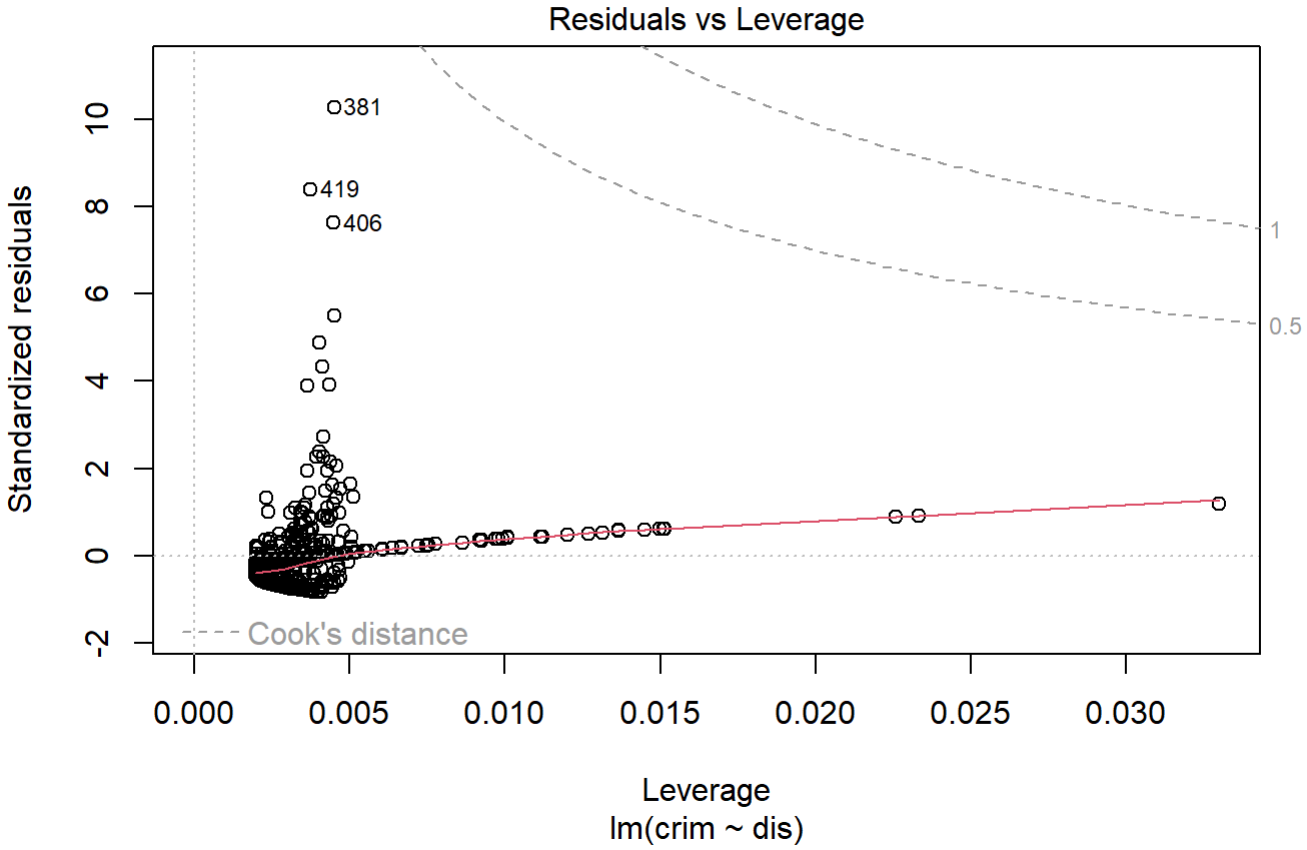

Relationship of dis and crim
Boston$crim
0
20
40
60
80
2
4
6
8
10
12
Boston$dis

```
plot(lm.fit7)
```


Residuals vs Fitted
Residuals
Fitted values
lm(crim ~ dis)
381
419
406


Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(crim ~ dis)
381
419
406





#Here, p

value was low so null hypothesis could be rejected, but R squared value and adjusted R squared value is also low so relationship between dis and crim is not significant.

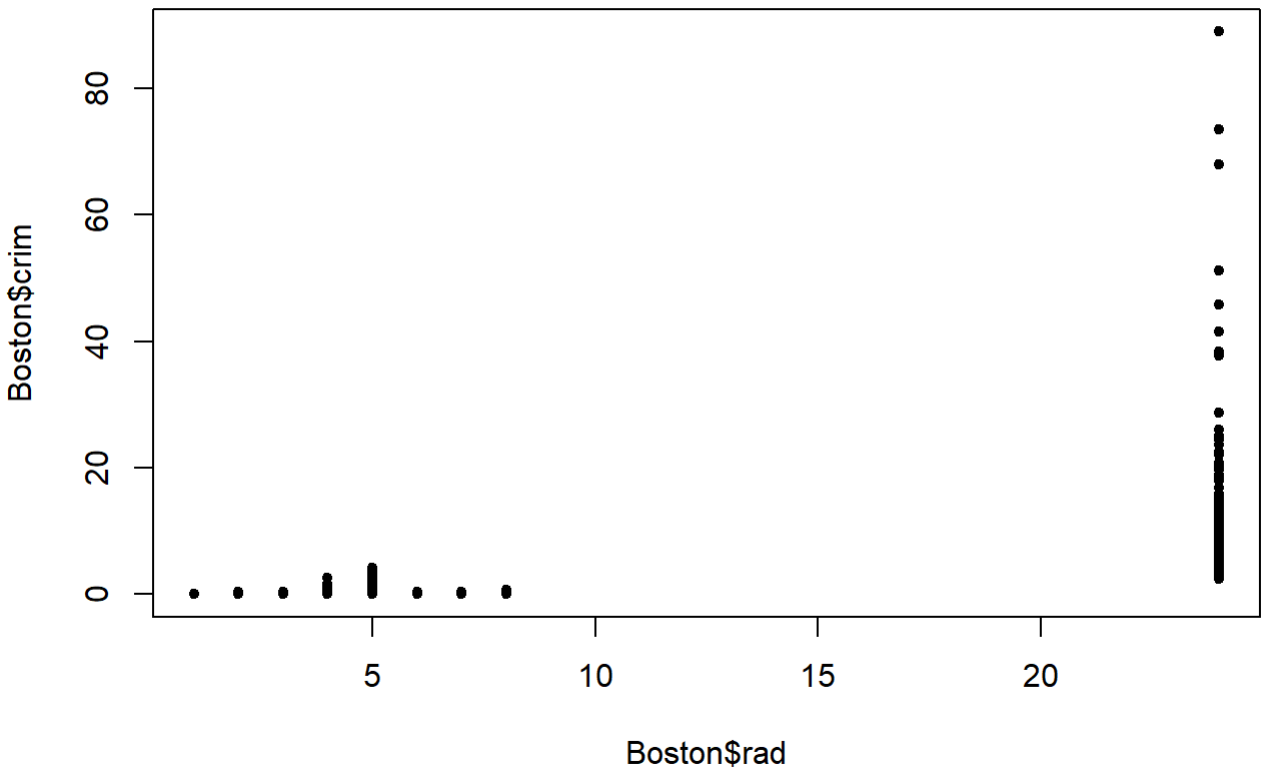
```
library(MASS)
data("Boston")
colnames(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

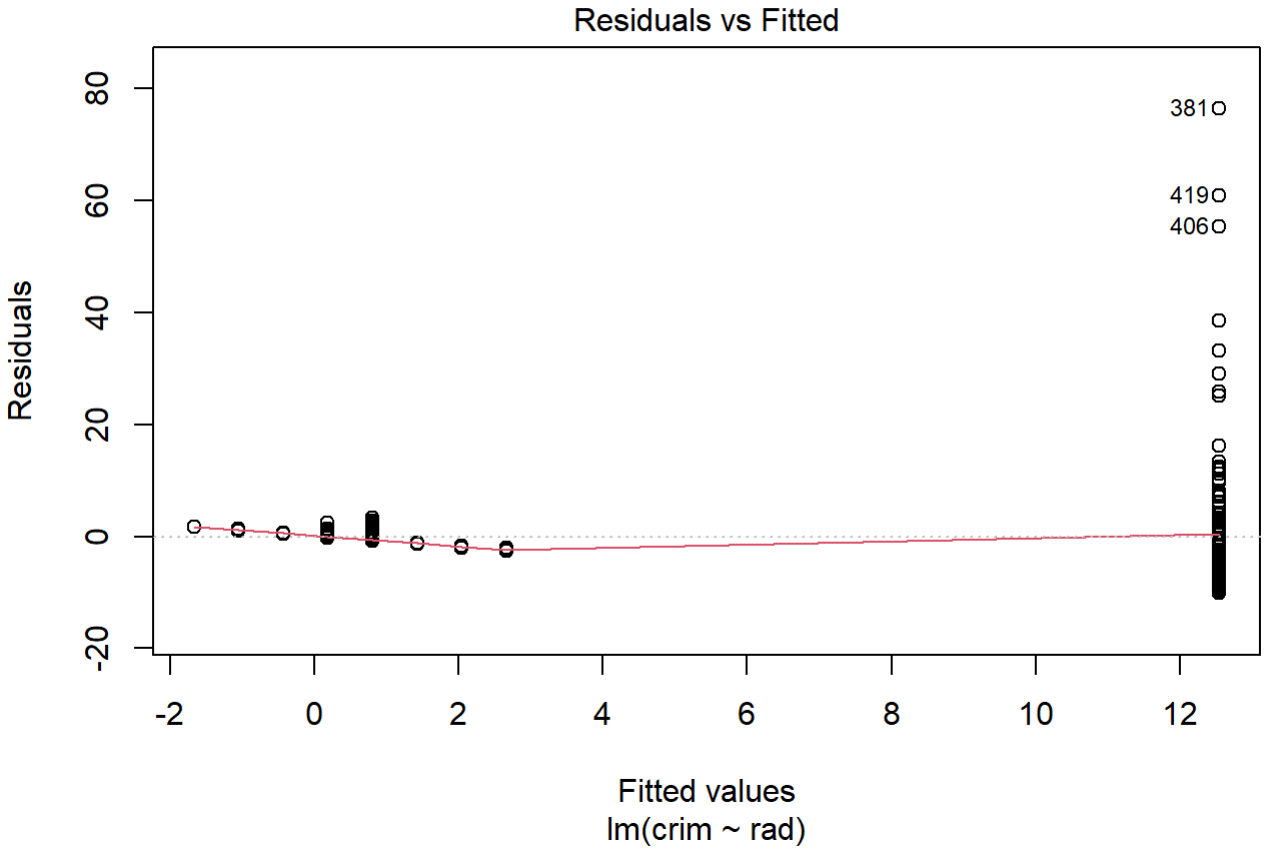
```
lm.fit8<- lm(crim ~ rad, Boston)
summary(lm.fit8)
```

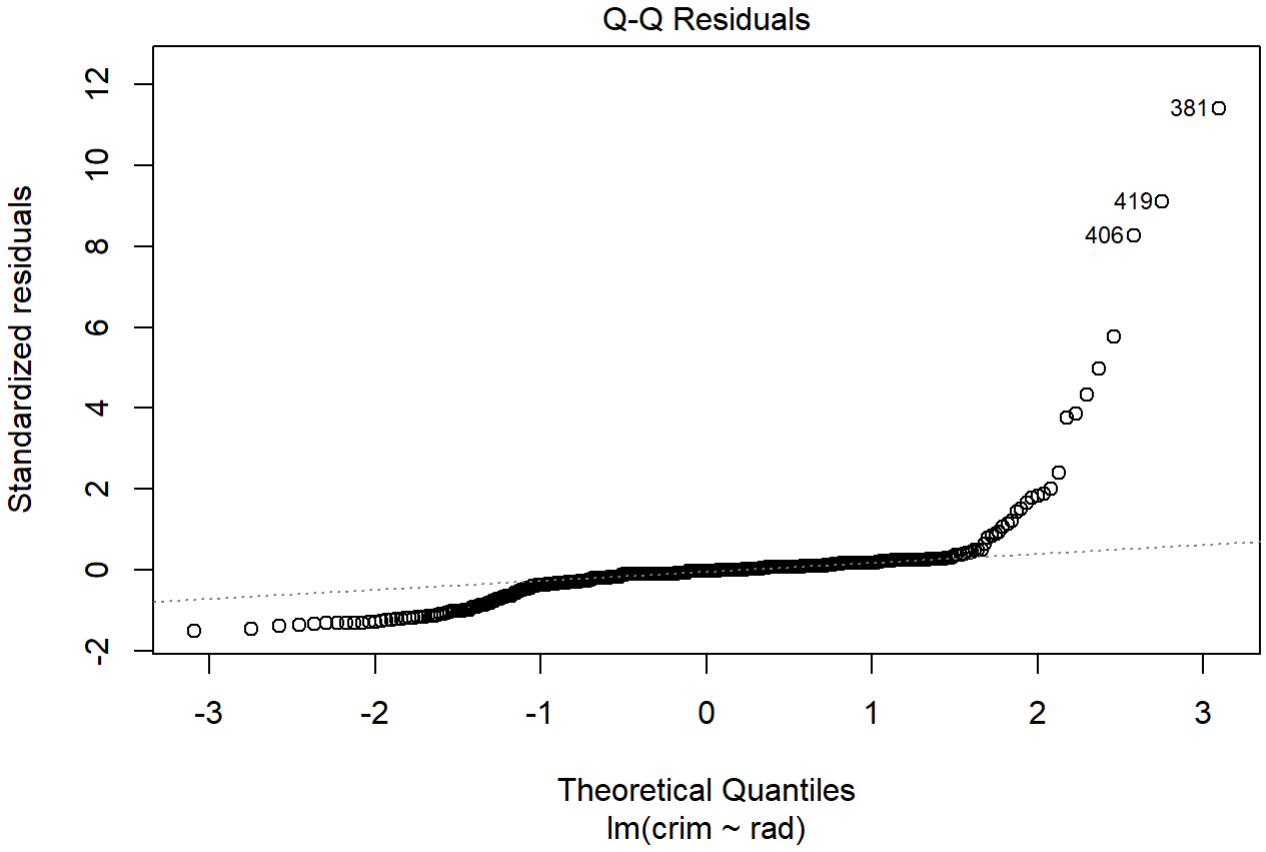
```
##
## Call:
## lm(formula = crim ~ rad, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.164  -1.381  -0.141   0.660  76.433
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.28716    0.44348  -5.157 3.61e-07 ***
## rad          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
```

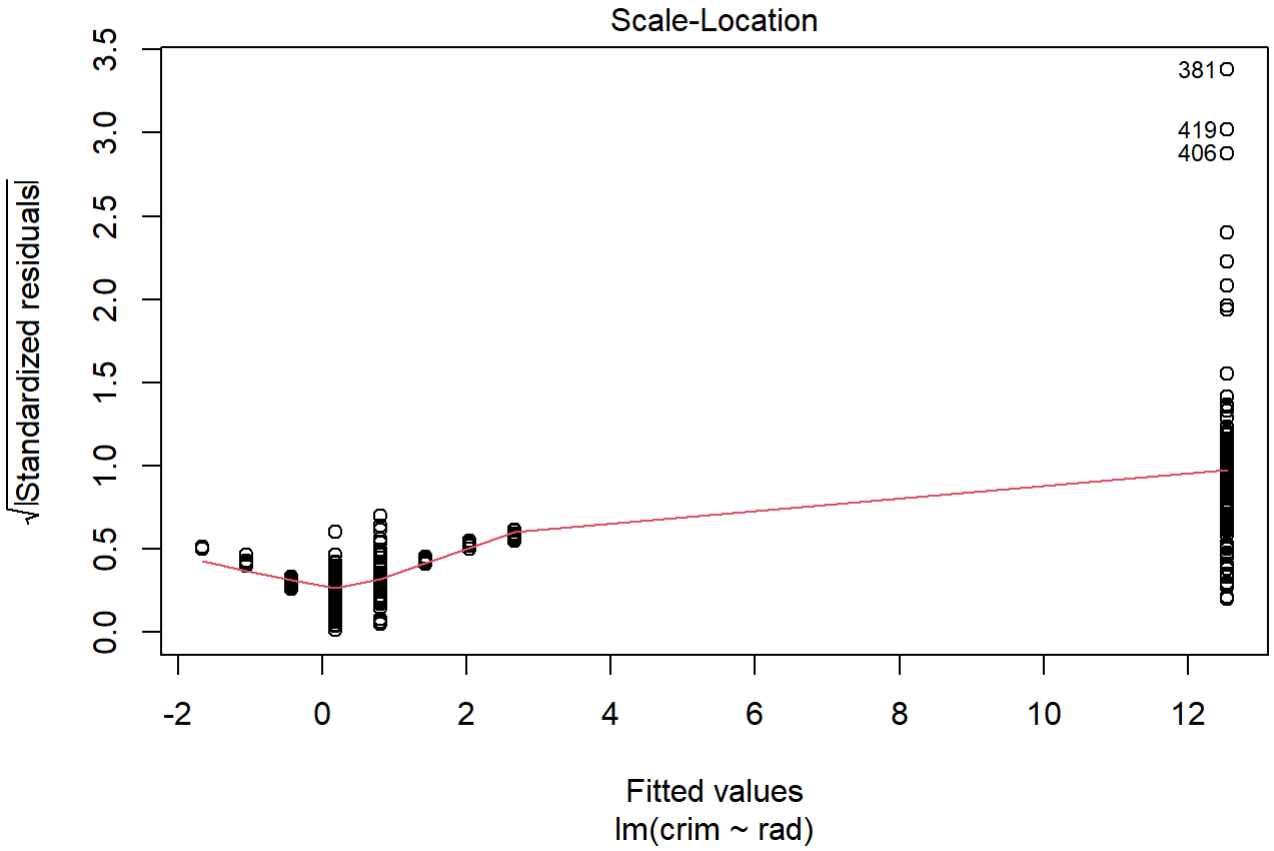
```
plot(Boston$rad, Boston$crim, pch = 20, main = "Relationship of rad and crim")
```

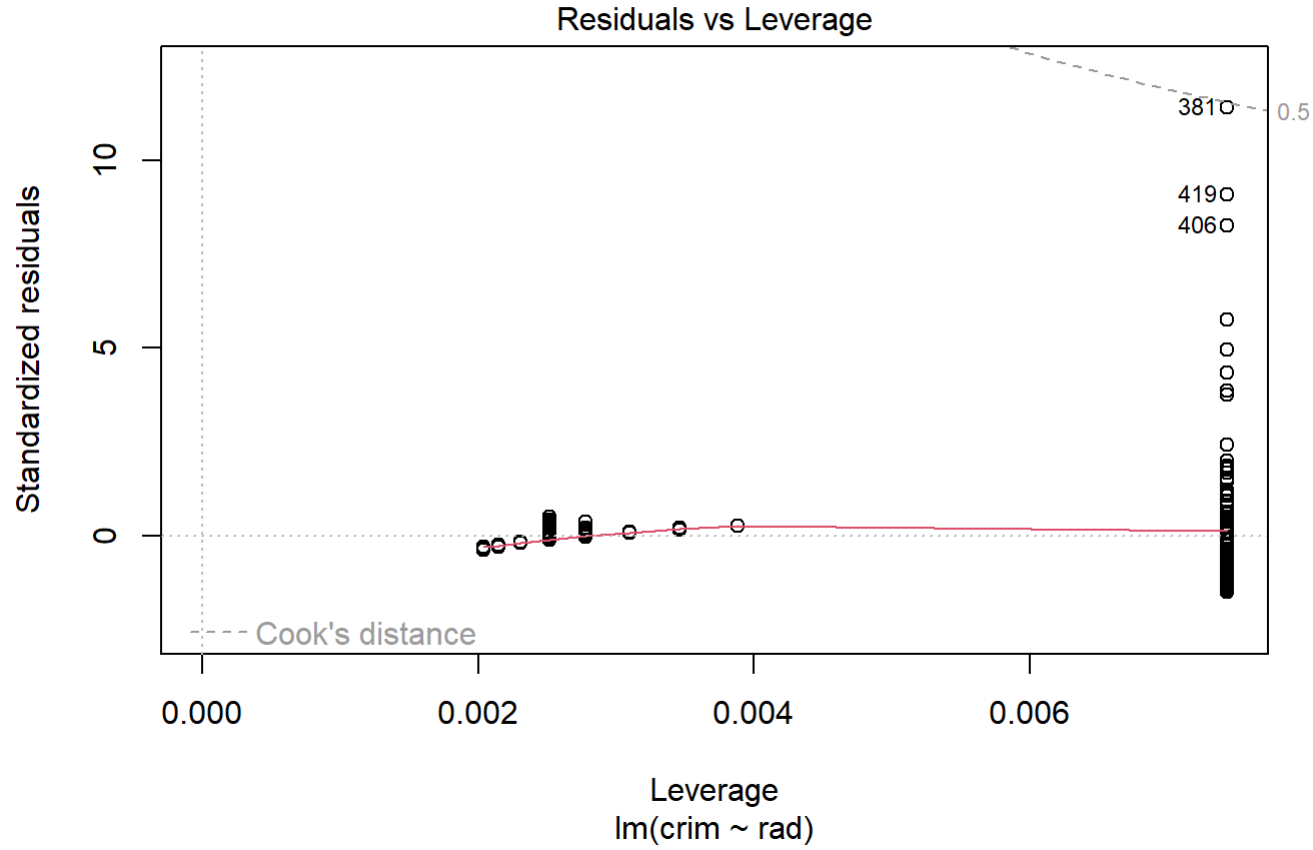

Relationship of rad and crim
Boston$crim
0
20
40
60
80
5
10
15
20
Boston$rad

```
plot(lm.fit8)
```


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


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





#Here, p

value was low so null hypothesis could be rejected, but R squared value and adjusted R squared value is also low so relationship between rad and crim is not significant.

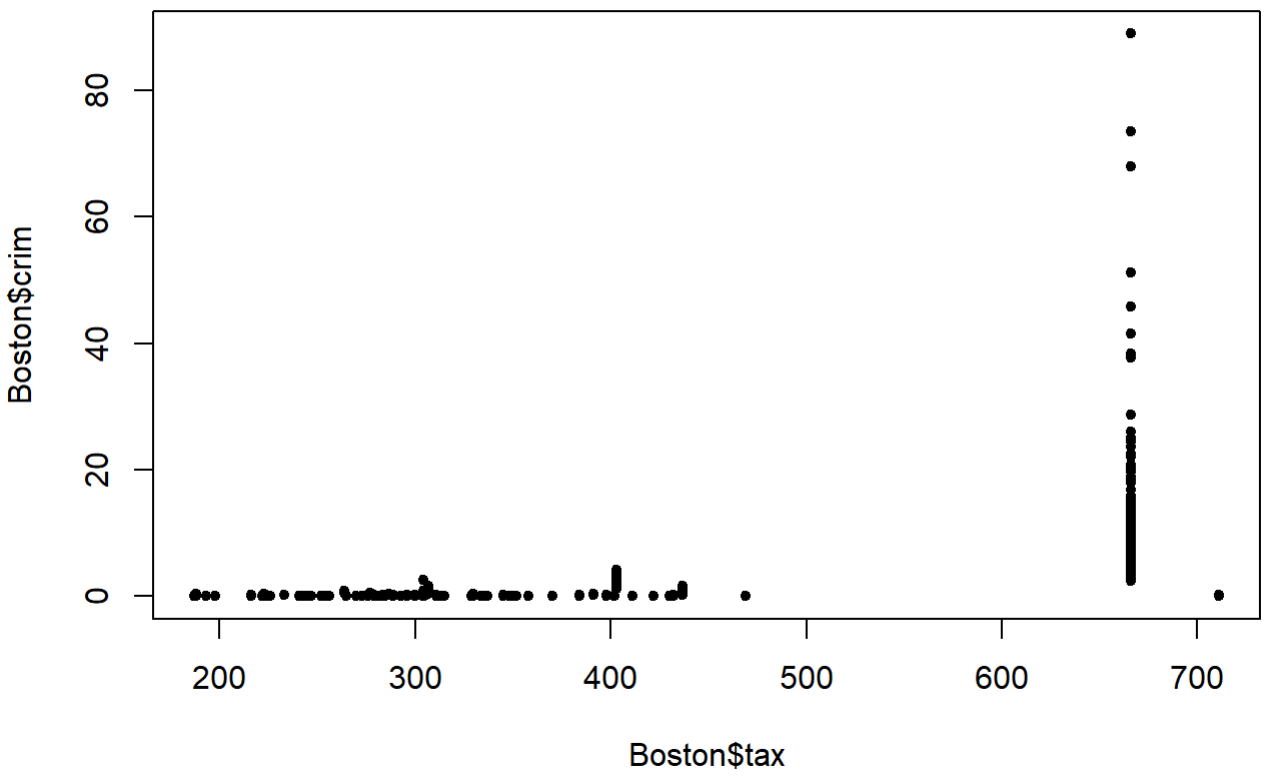
```
library(MASS)
data("Boston")
colnames(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

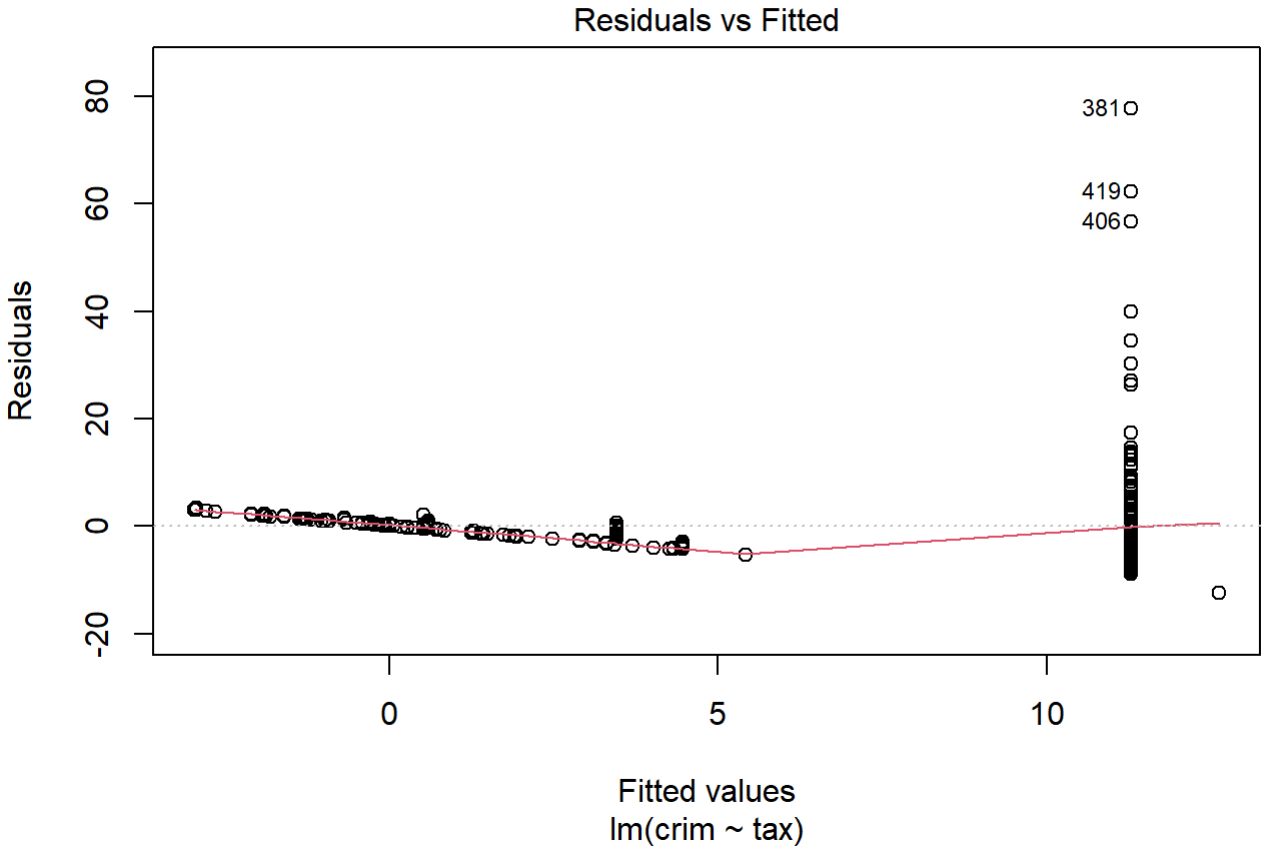
```
lm.fit9<- lm(crim ~ tax, Boston)
summary(lm.fit9)
```

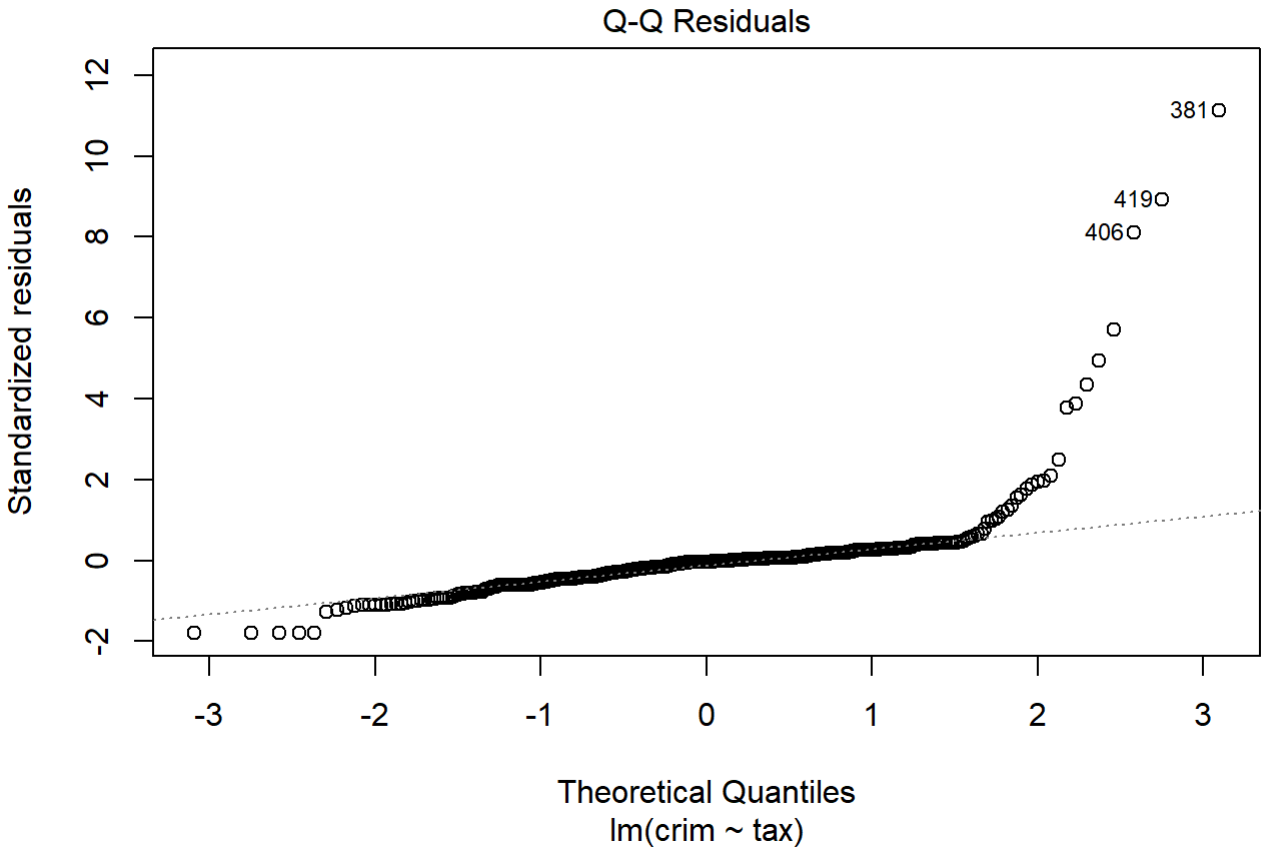
```
##
## Call:
## lm(formula = crim ~ tax, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.513  -2.738  -0.194   1.065  77.696
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.528369   0.815809  -10.45   <2e-16 ***
## tax          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
```

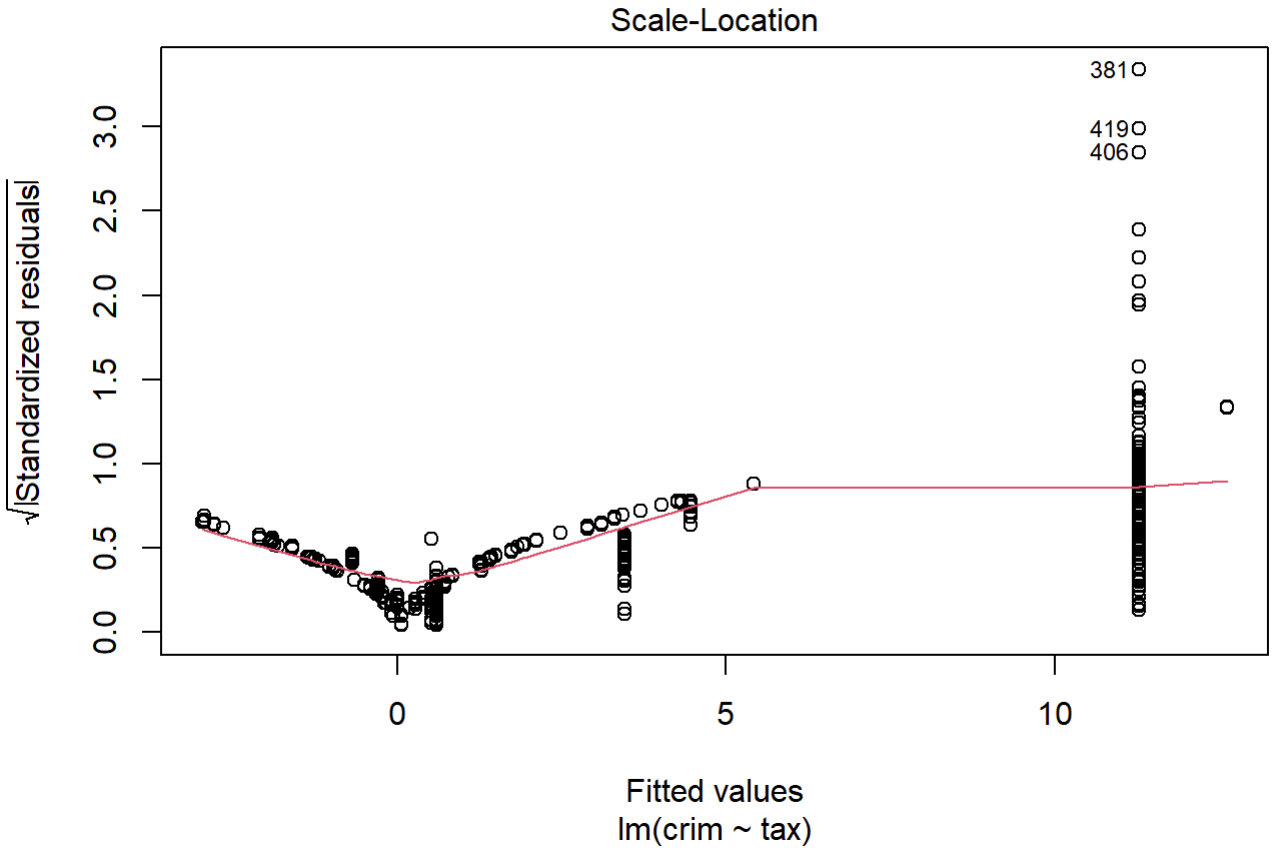
```
plot(Boston$tax, Boston$crim, pch = 20, main = "Relationship of tax and crim")
```

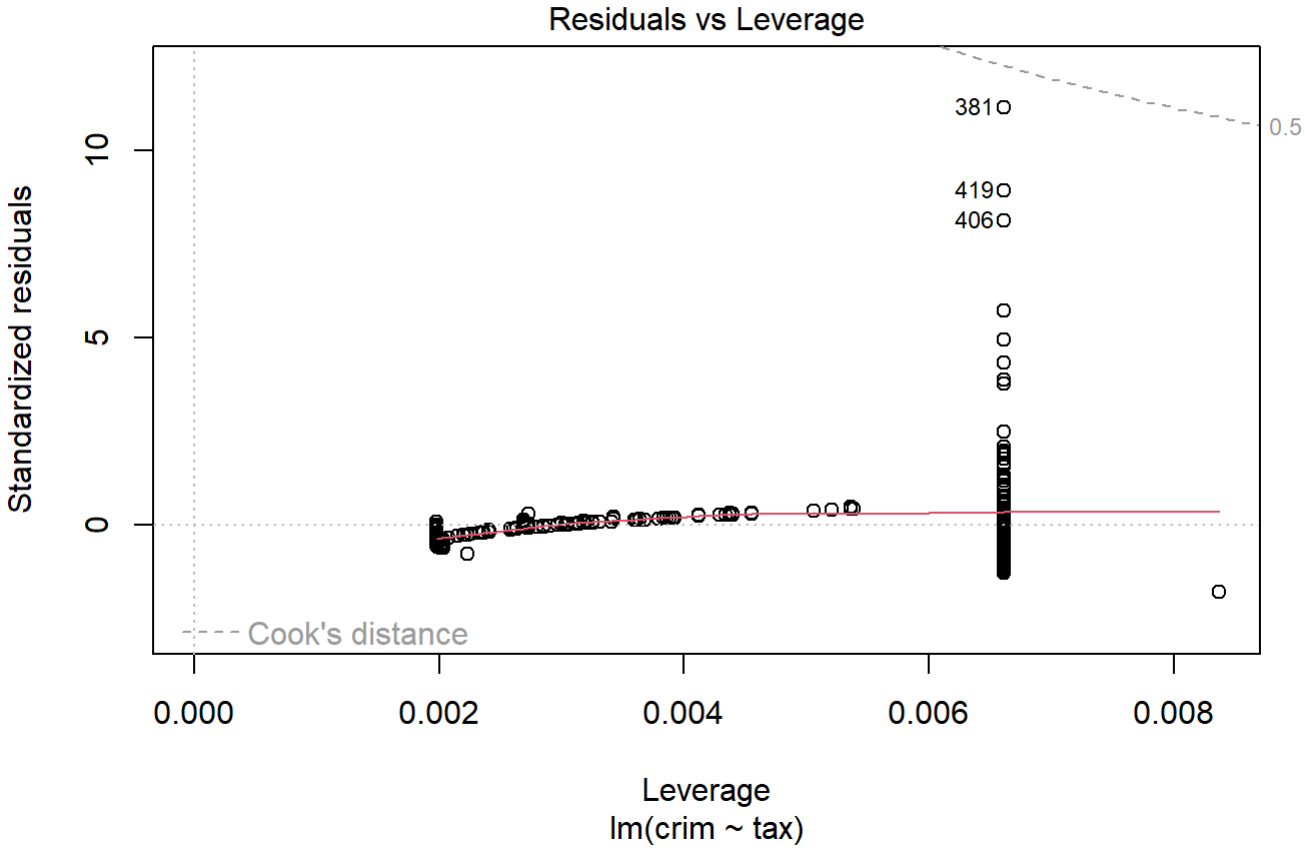

Relationship of tax and crim
Boston$crim
0
20
40
60
80
200
300
400
500
600
700
Boston$tax

```
plot(lm.fit9)
```


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


Q-Q Residuals
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





#Here, p

value was low so null hypothesis could be rejected, but R squared value and adjusted R squared value is also low so relationship between tax and crim is not significant.

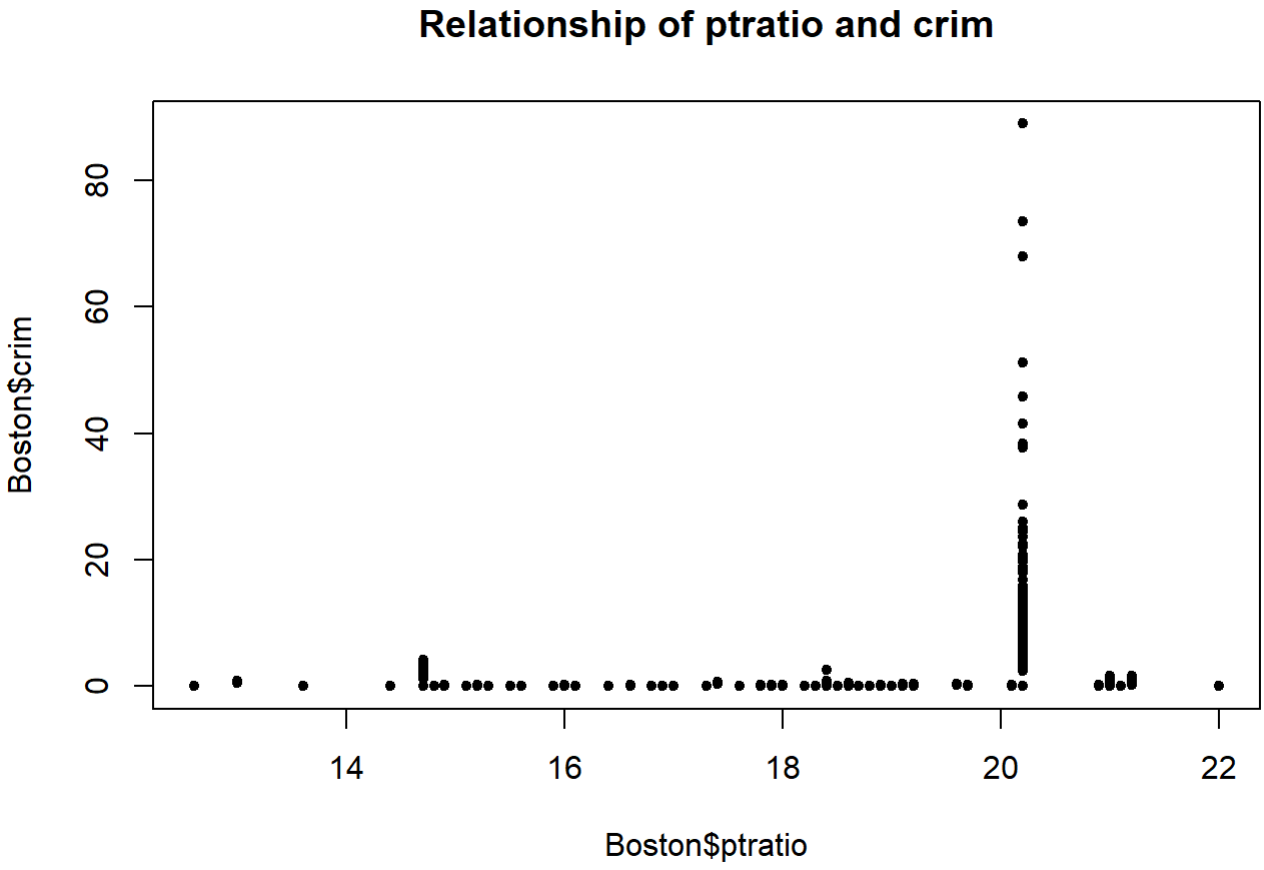
```
library(MASS)
data("Boston")
colnames(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

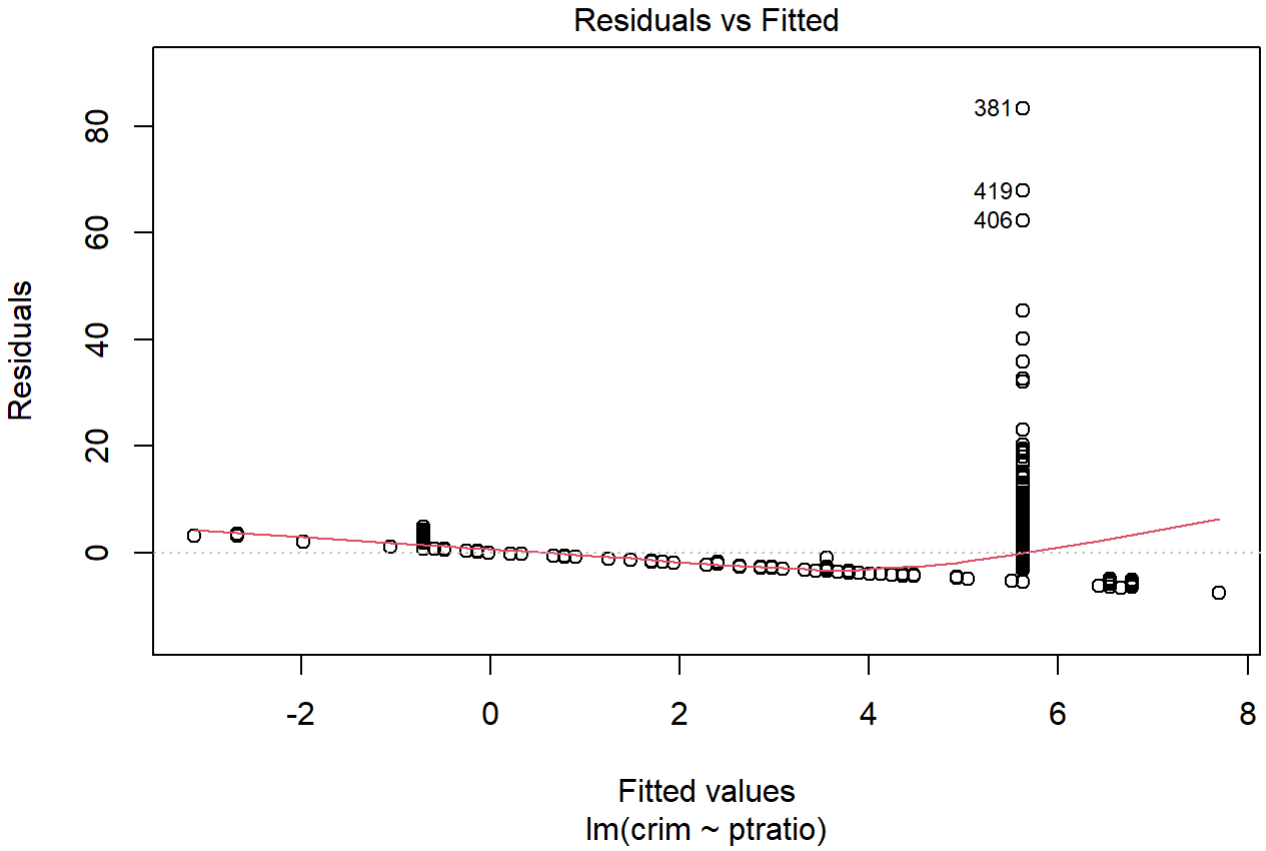
```
lm.fit10<- lm(crim ~ ptratio, Boston)
summary(lm.fit10)
```

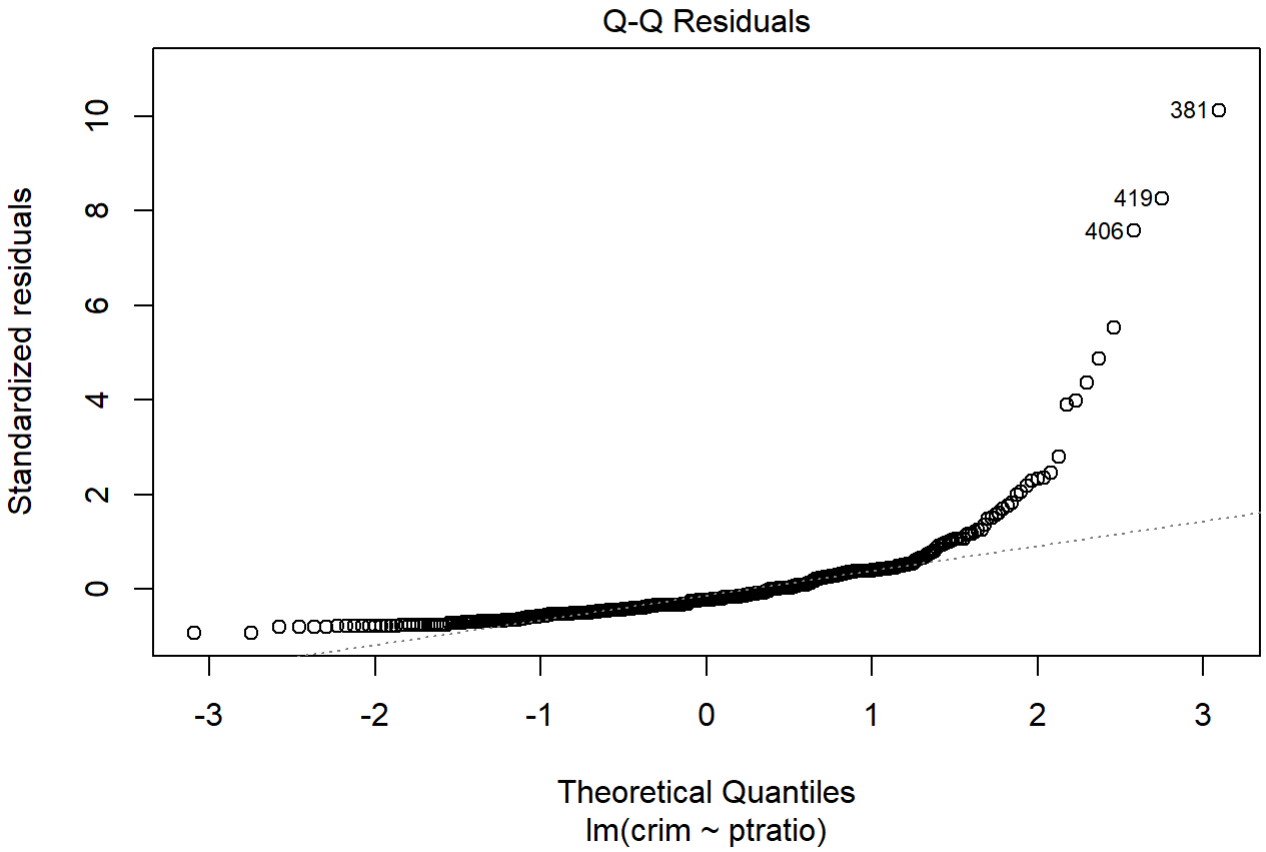
```
##
## Call:
## lm(formula = crim ~ ptratio, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.654 -3.985 -1.912  1.825 83.353
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.6469     3.1473  -5.607 3.40e-08 ***
## ptratio       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
```

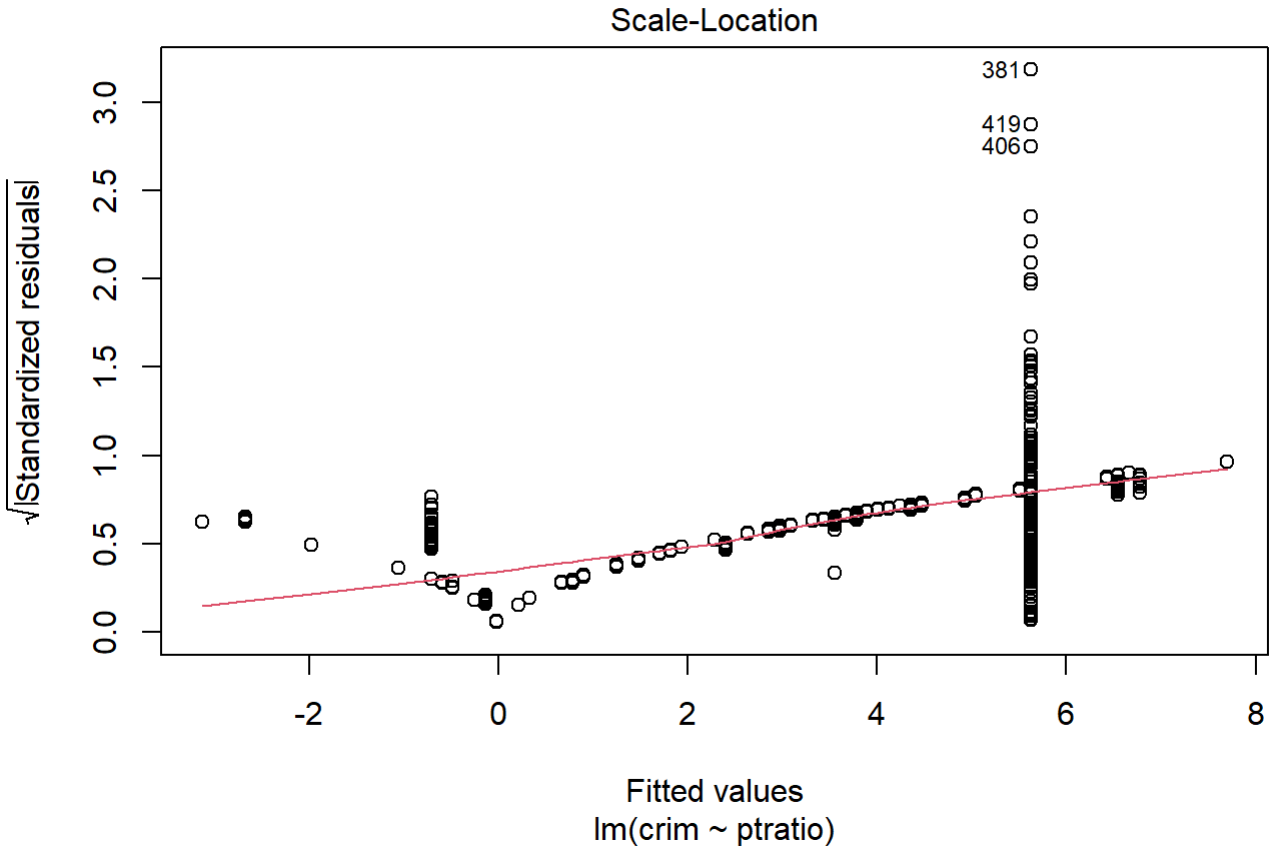
```
plot(Boston$ptratio, Boston$crim, pch = 20, main = "Relationship of ptratio and crim")
```

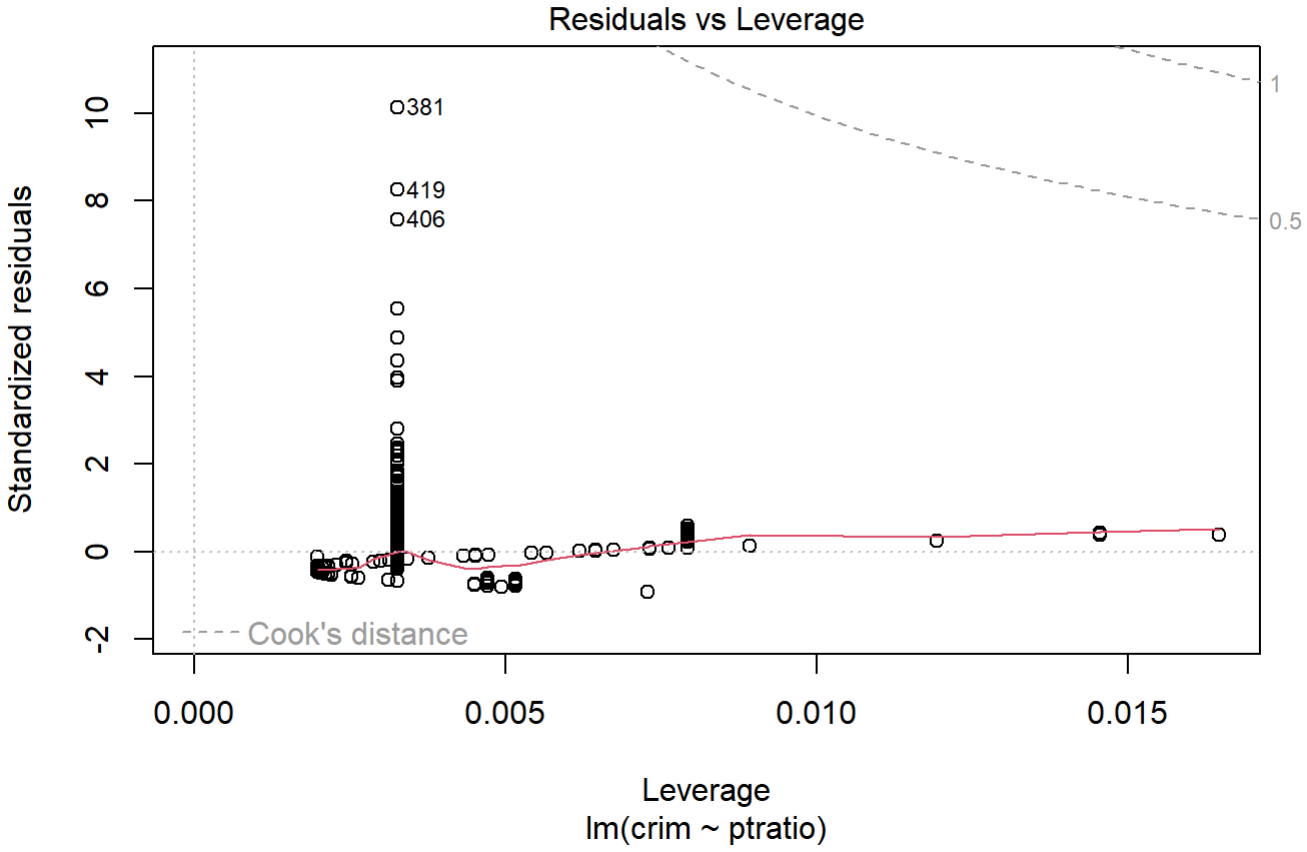



```
plot(lm.fit10)
```


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


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





#Here, p

value was low so null hypothesis could be rejected, but R squared value and adjusted R squared value is also low so relationship between ptratio and crim is not significant.

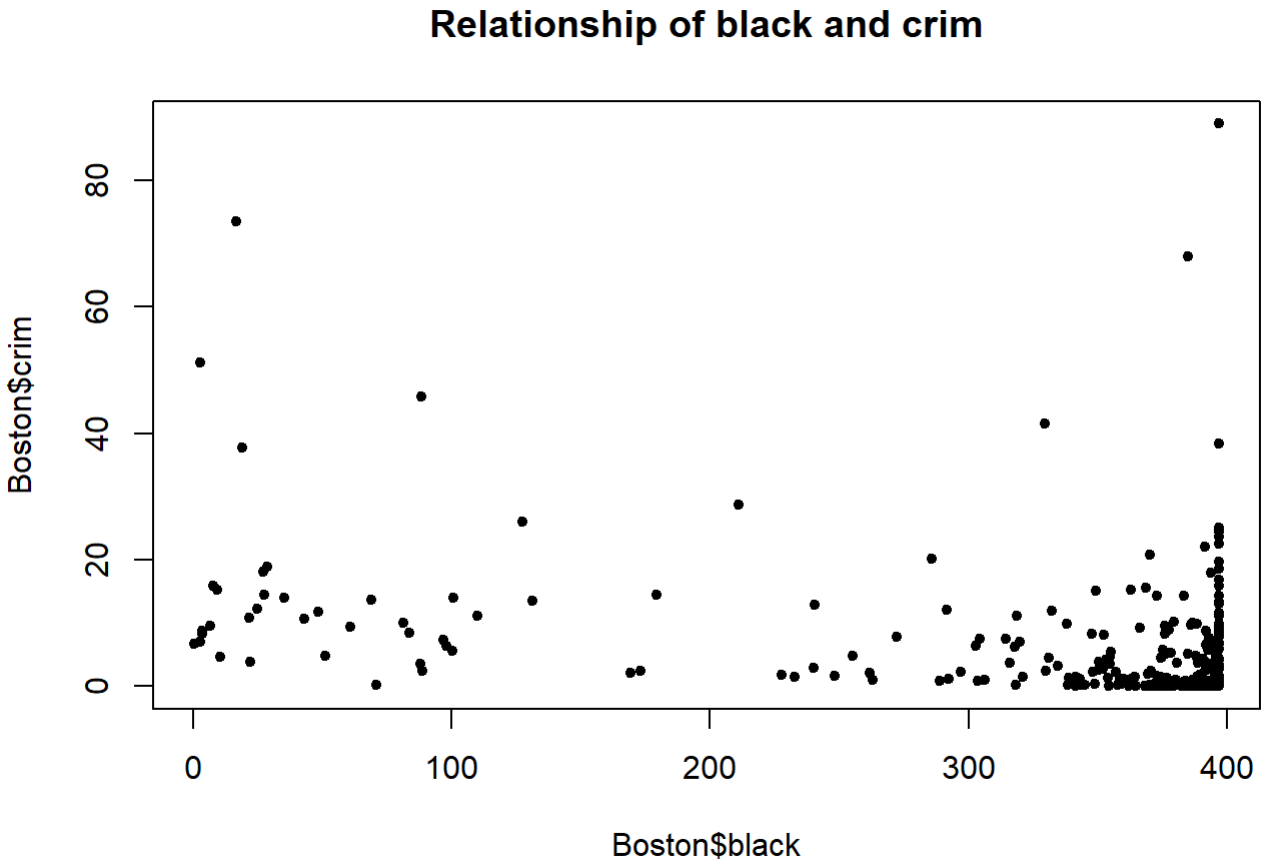
```
library(MASS)
data("Boston")
colnames(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

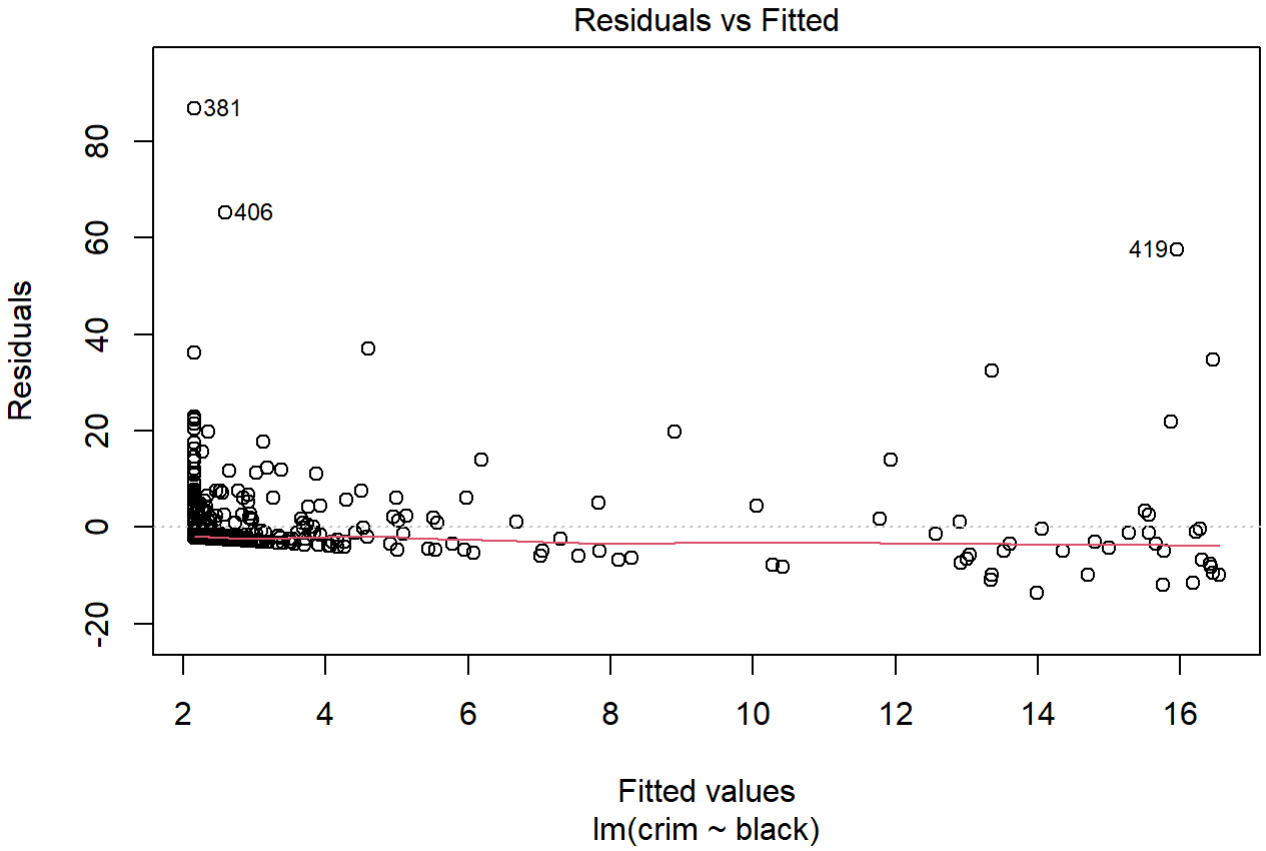
```
lm.fit11<- lm(crim ~ black, Boston)
summary(lm.fit11)
```

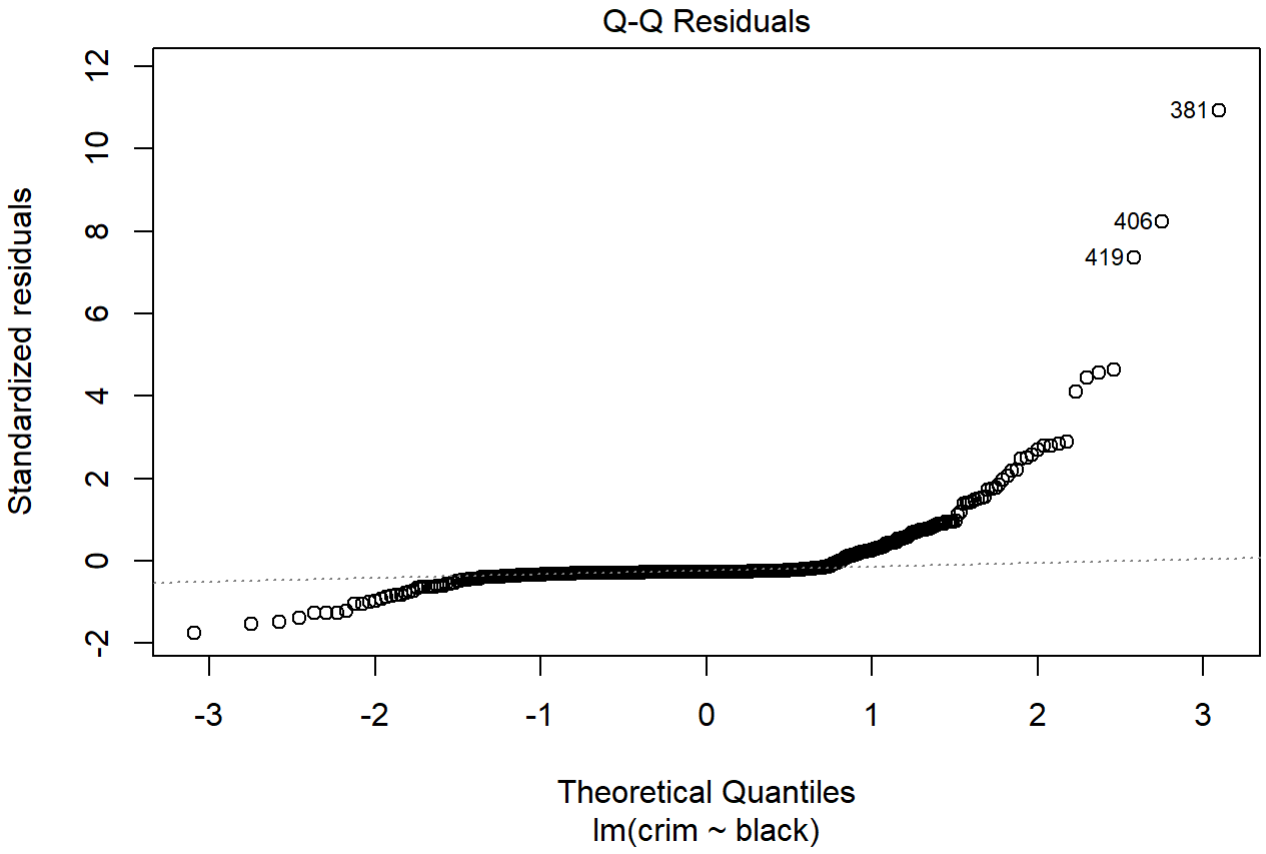
```
##
## Call:
## lm(formula = crim ~ black, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.756  -2.299  -2.095  -1.296  86.822
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.553529   1.425903  11.609   <2e-16 ***
## black       -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
```

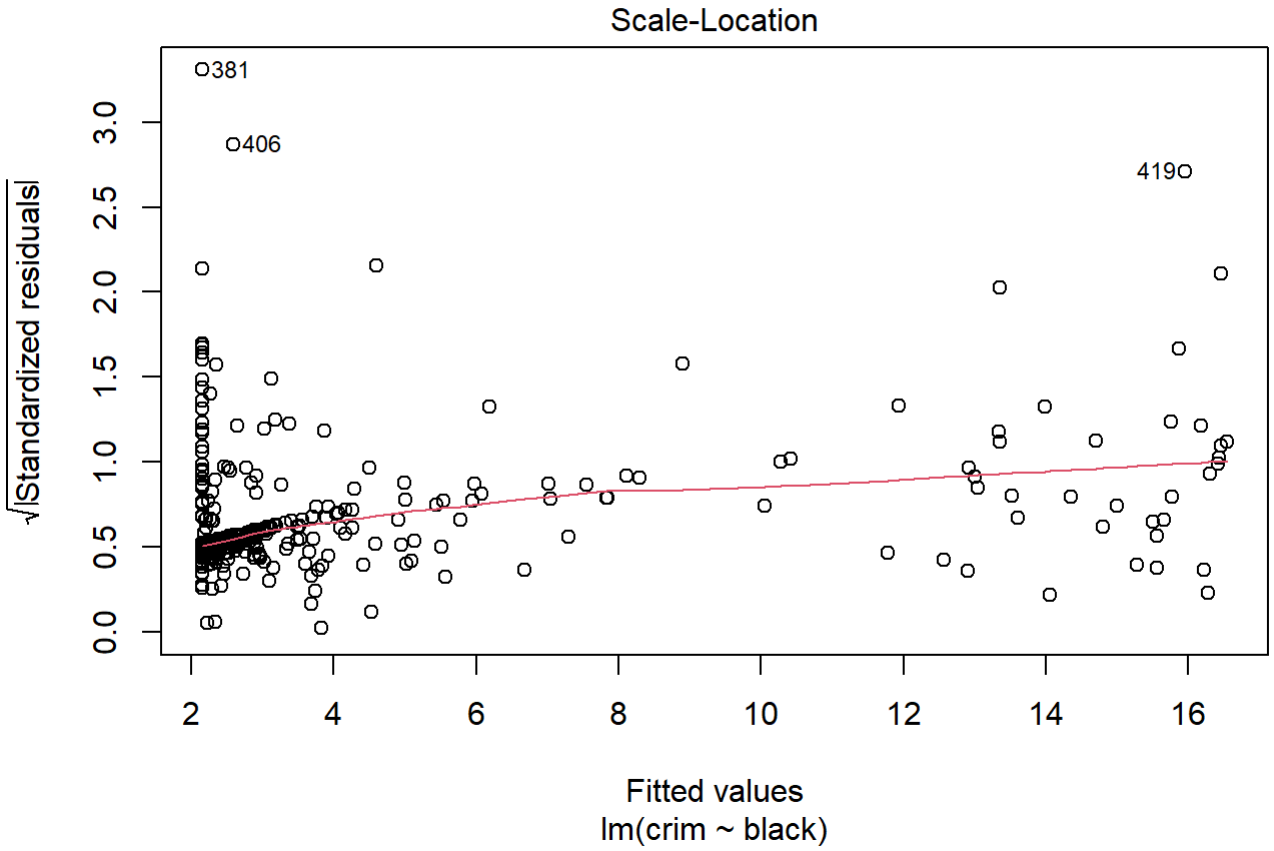
```
plot(Boston$black, Boston$crim, pch = 20, main = "Relationship of black and crim")
```

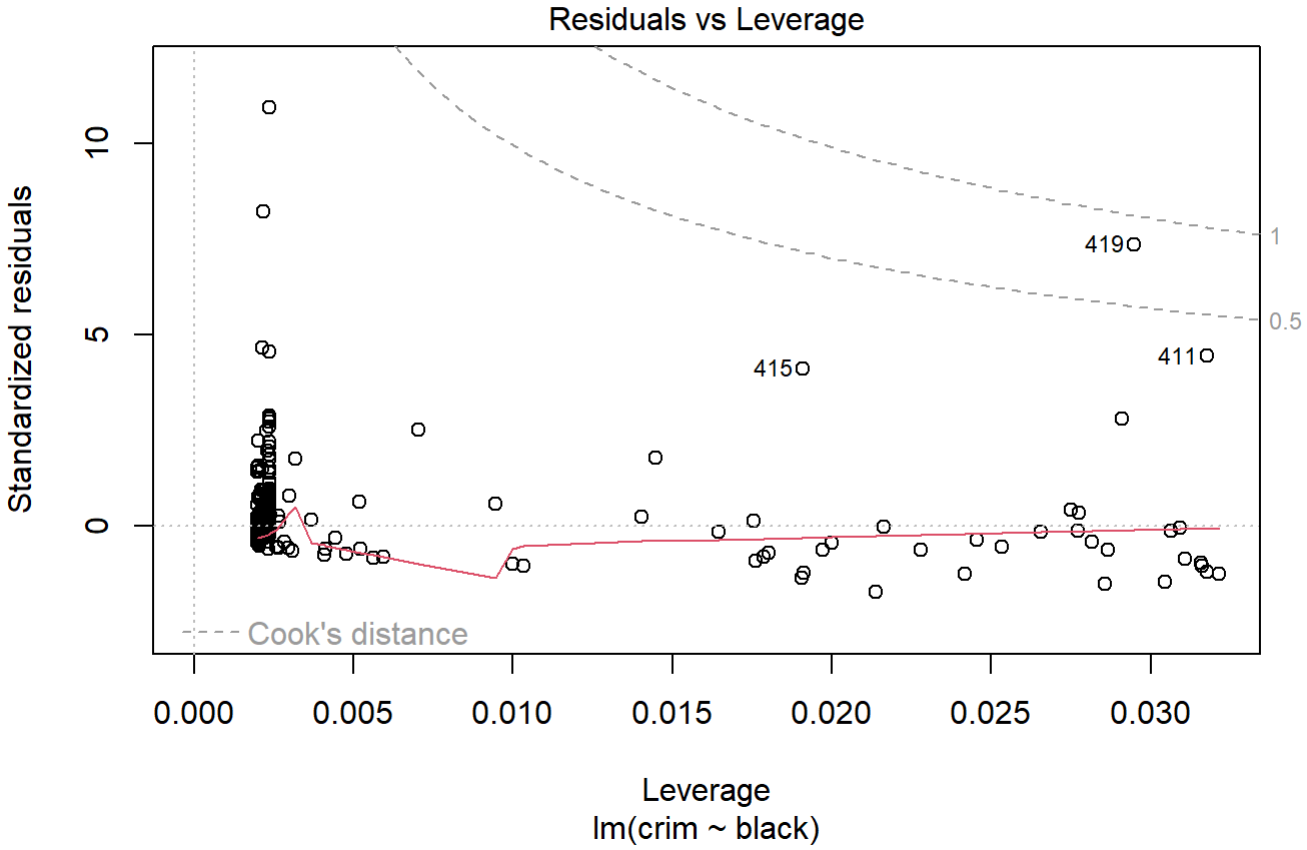



```
plot(lm.fit11)
```


Residuals vs Fitted
Residuals
Fitted values
lm(crim ~ black)
381
406
419


Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(crim ~ black)
381
406
419





#Here, p

value was low so null hypothesis could be rejected, but R squared value and adjusted R squared value is also low so relationship between rm and crim is not significant.

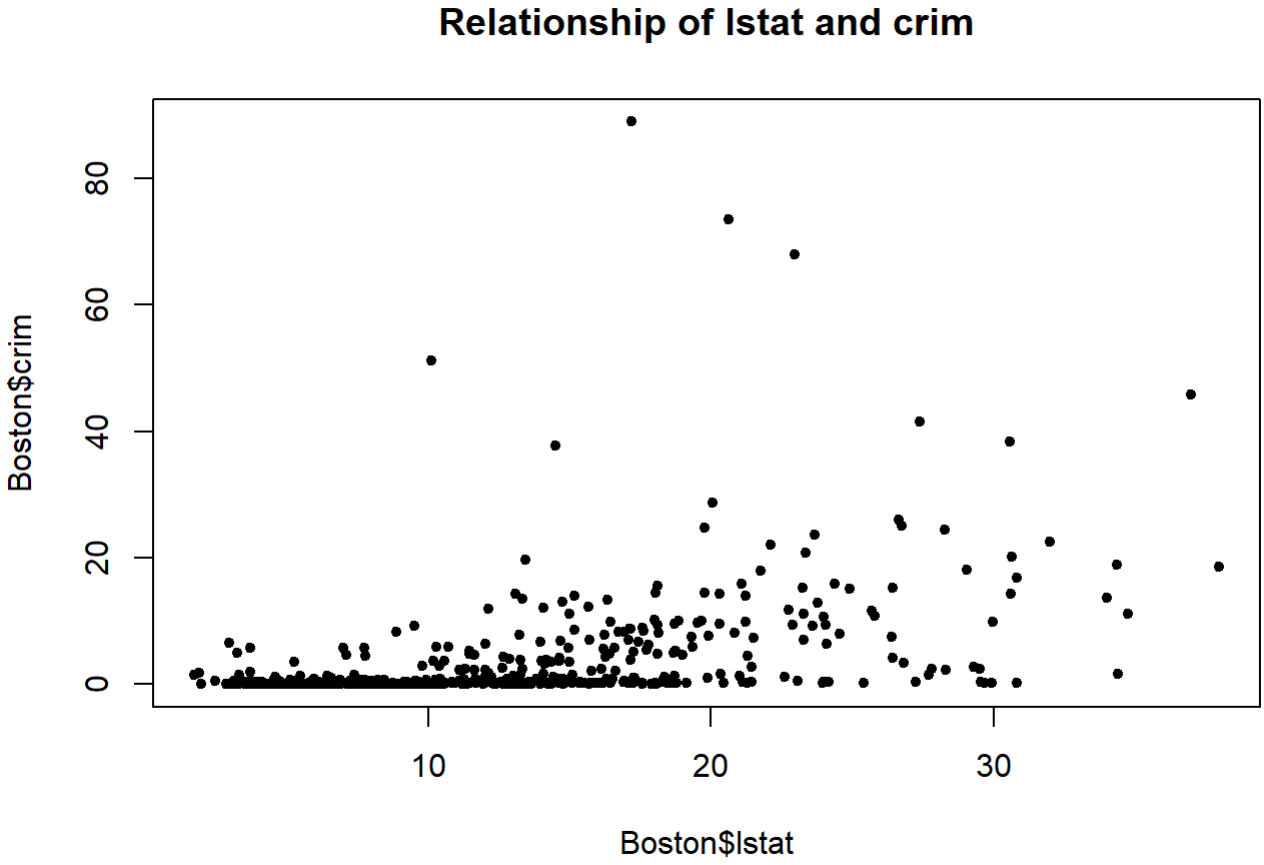
```
library(MASS)
data("Boston")
colnames(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

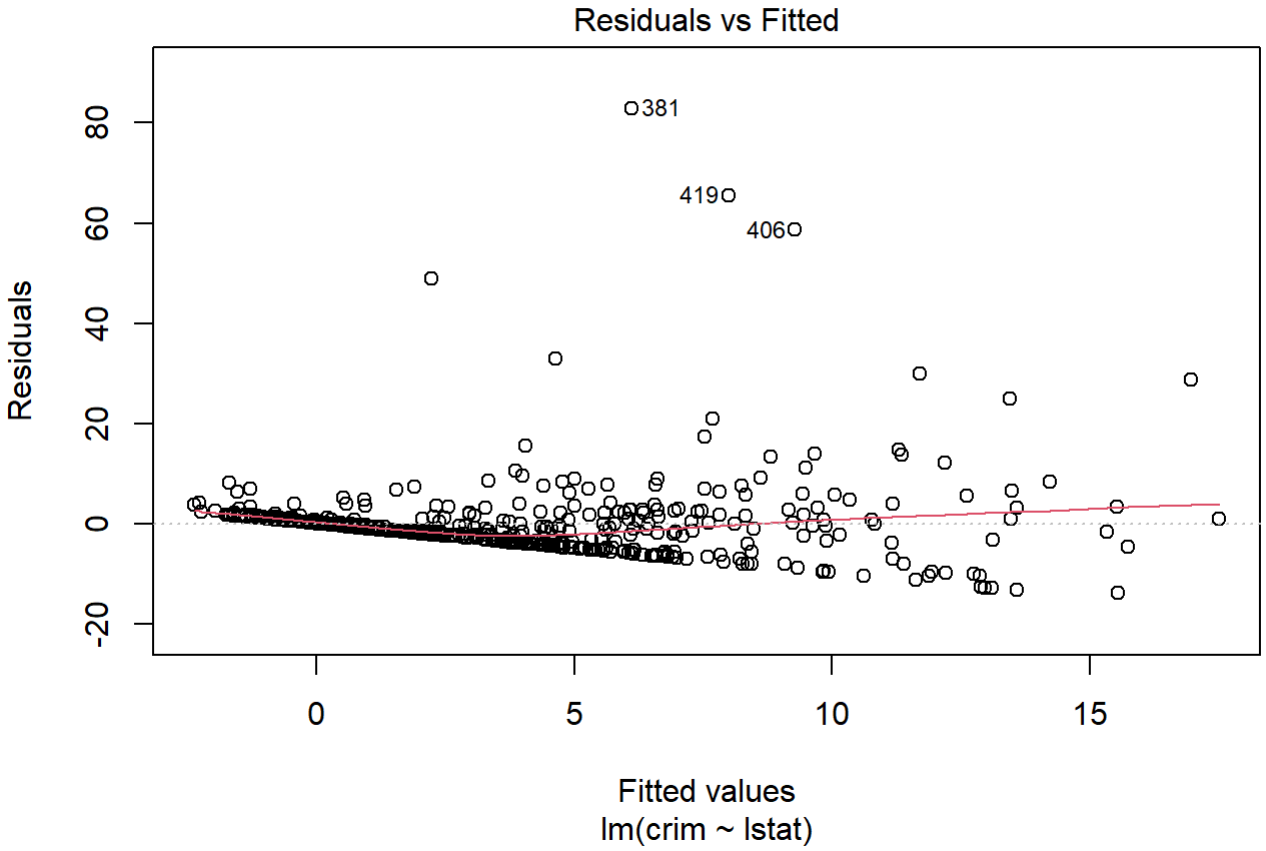
```
lm.fit12<- lm(crim ~ lstat, Boston)
summary(lm.fit12)
```

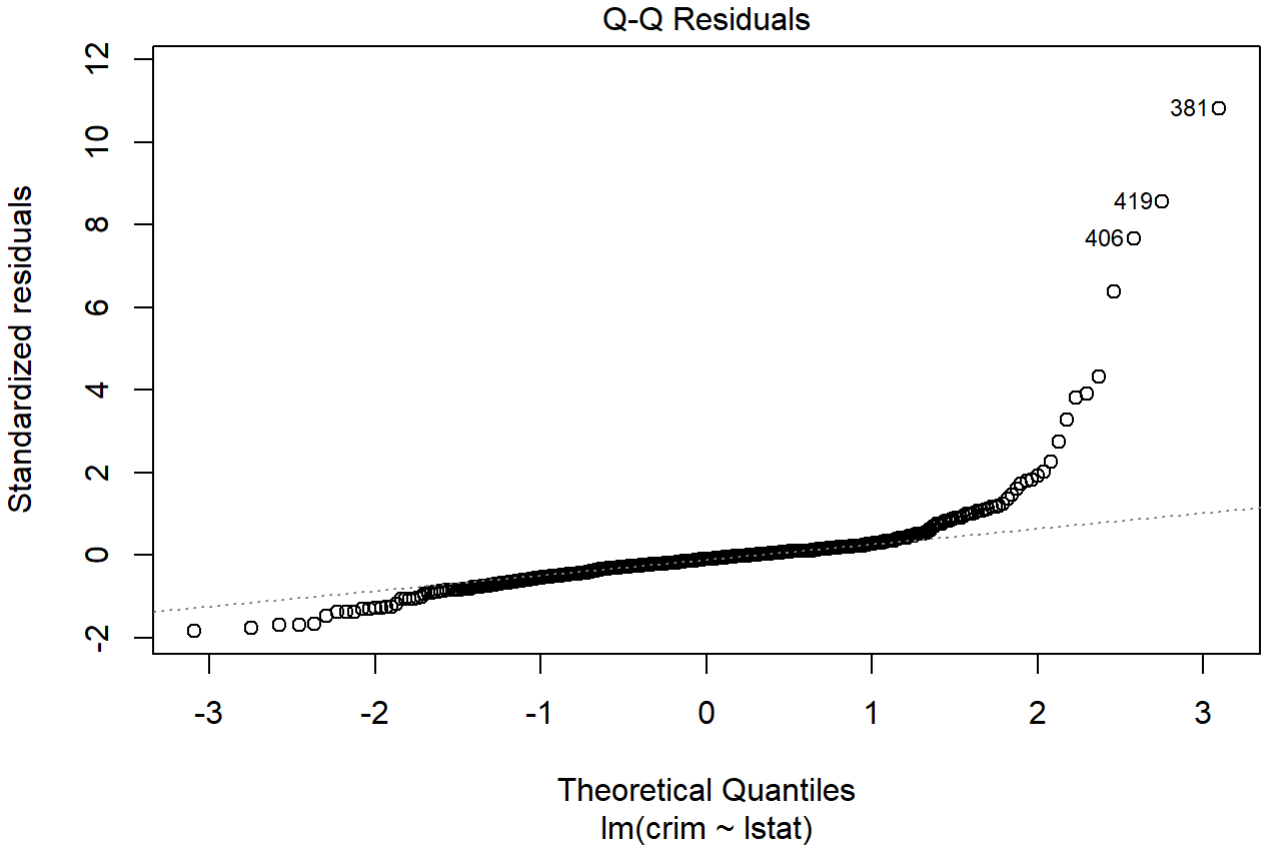
```
##
## Call:
## lm(formula = crim ~ lstat, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.925  -2.822  -0.664   1.079  82.862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.33054    0.69376  -4.801 2.09e-06 ***
## lstat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

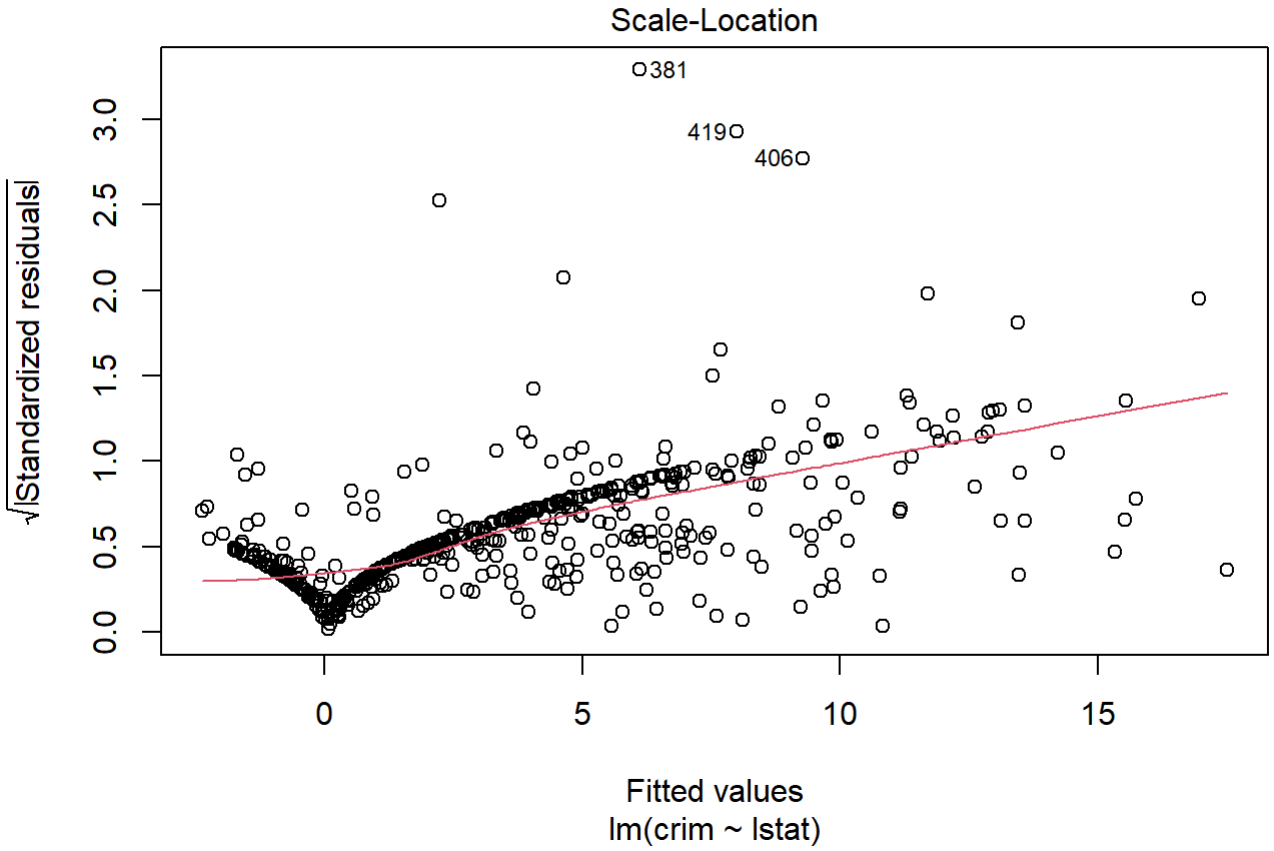
```
plot(Boston$lstat, Boston$crim, pch = 20, main = "Relationship of lstat and crim")
```

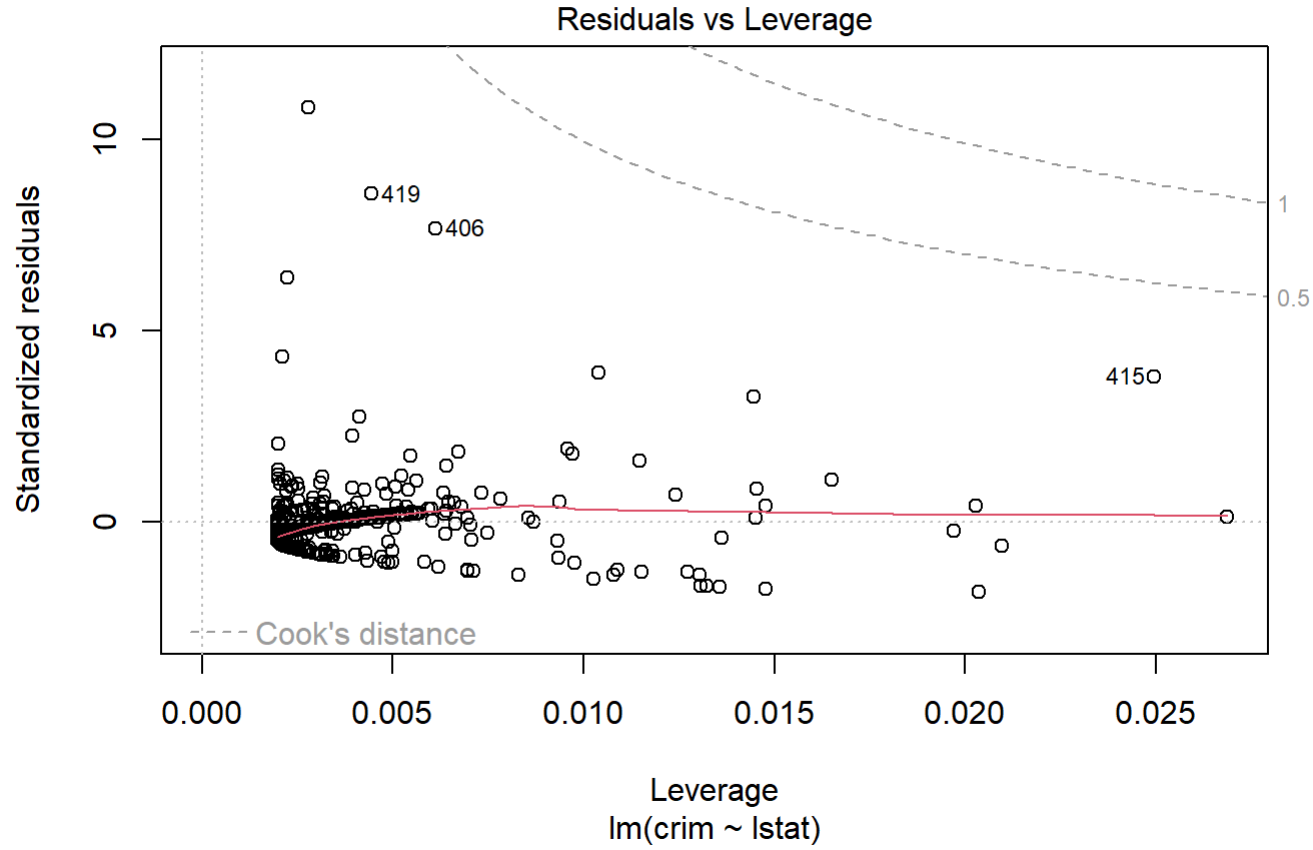



```
plot(lm.fit12)
```


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


Q-Q Residuals
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





#Here, p

value was low so null hypothesis could be rejected, but R squared value and adjusted R squared value is also low so relationship between lstat and crim is not significant.

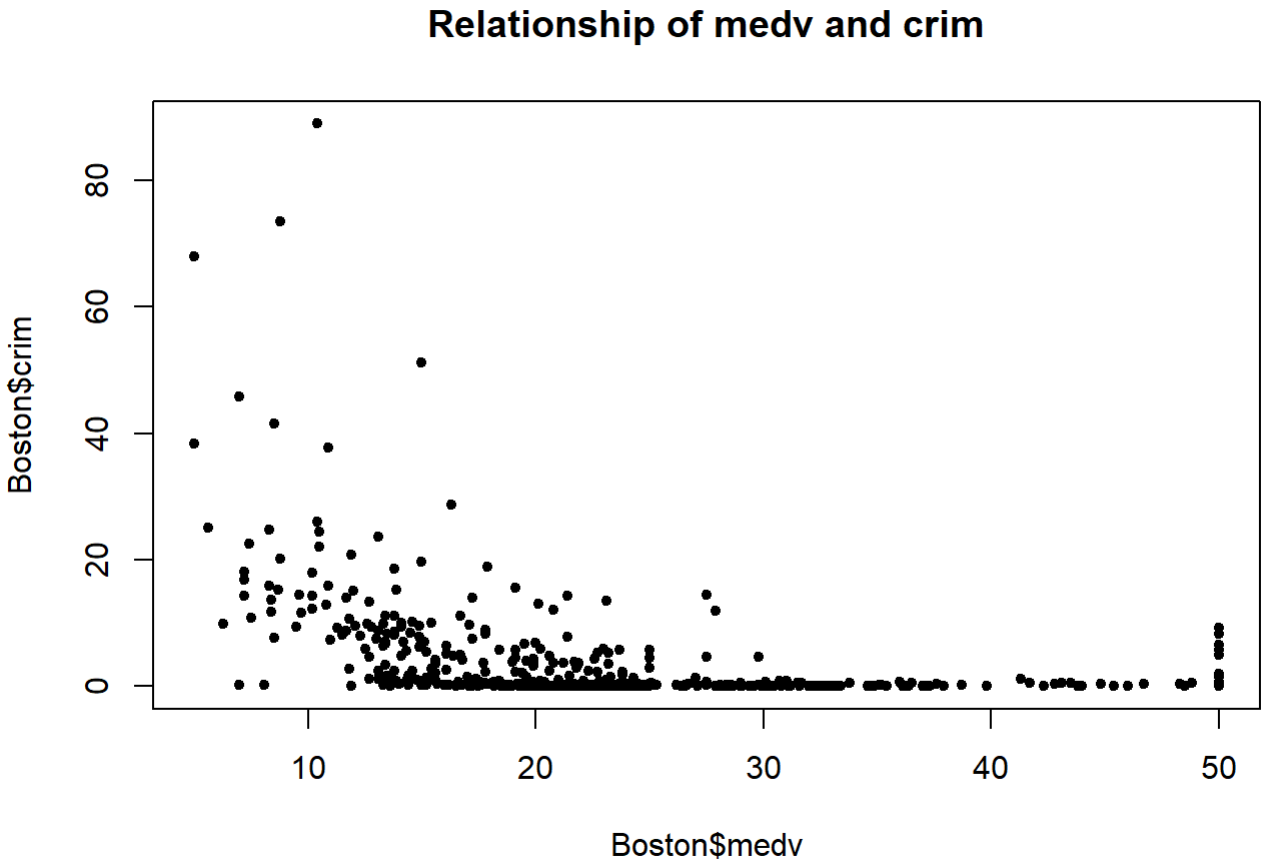
```
library(MASS)
data("Boston")
colnames(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

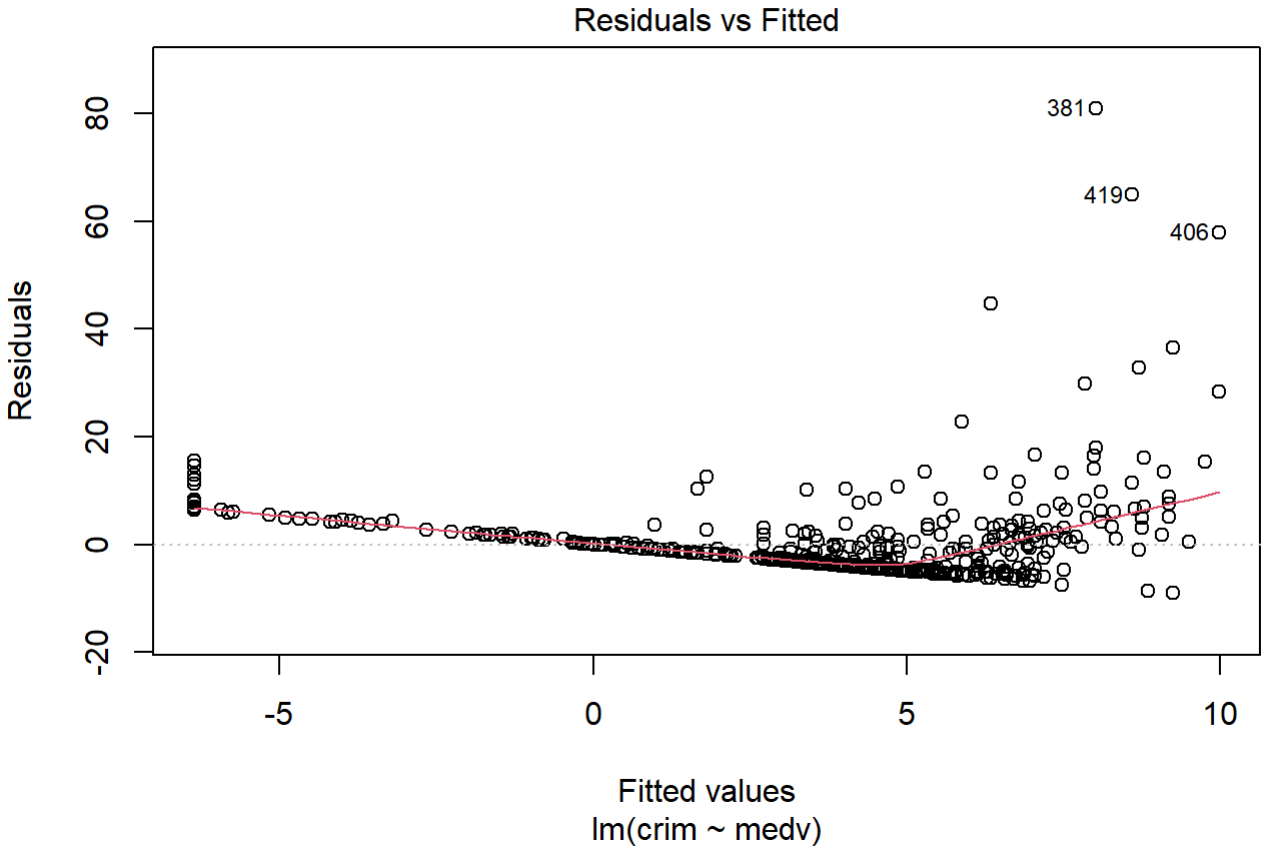
```
lm.fit13<- lm(crim ~ medv, Boston)
summary(lm.fit13)
```

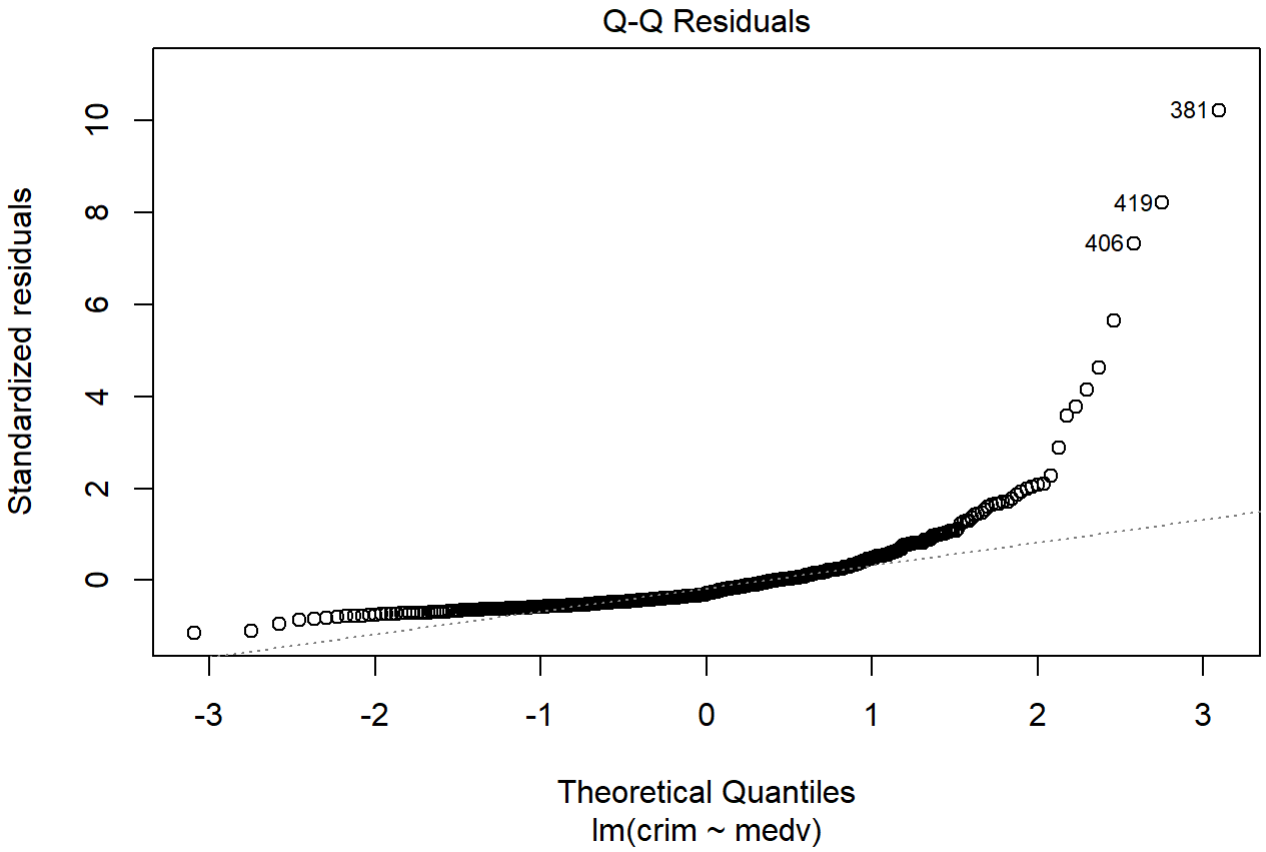
```
##
## Call:
## lm(formula = crim ~ medv, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.071 -4.022 -2.343  1.298 80.957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.79654    0.93419   12.63   <2e-16 ***
## medv        -0.36316    0.03839   -9.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.934 on 504 degrees of freedom
## Multiple R-squared:  0.1508, Adjusted R-squared:  0.1491
## F-statistic: 89.49 on 1 and 504 DF,  p-value: < 2.2e-16
```

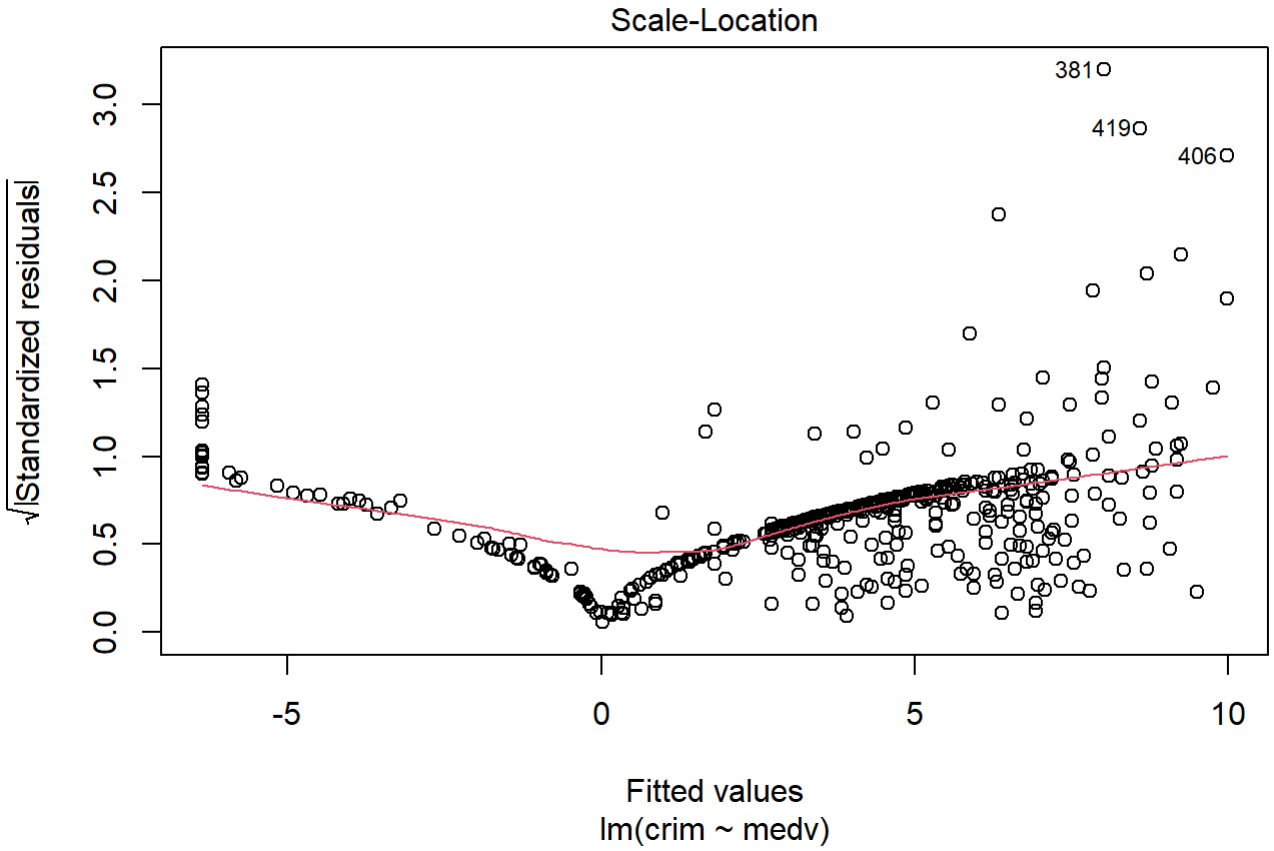
```
plot(Boston$medv, Boston$crim, pch = 20, main = "Relationship of medv and crim")
```

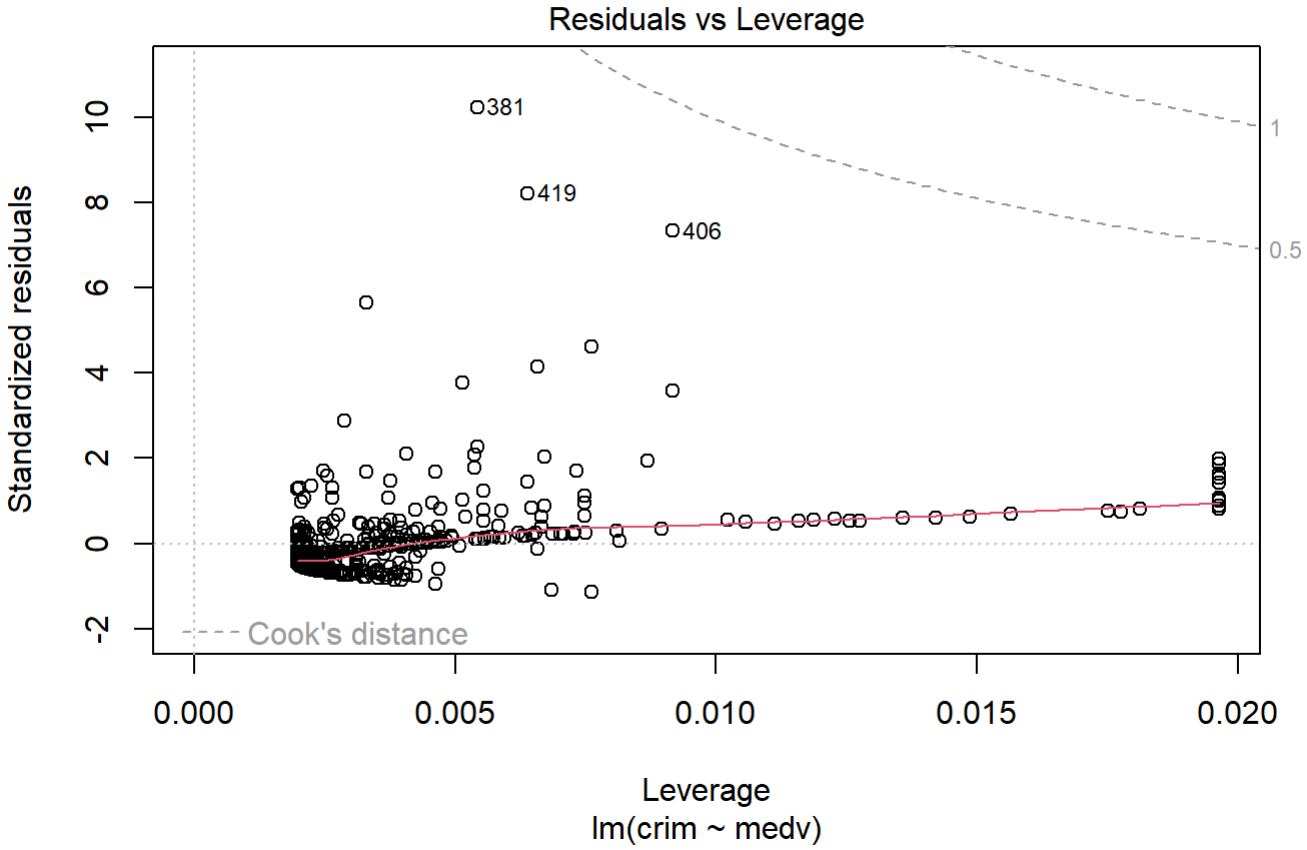



```
plot(lm.fit13)
```


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


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





#Here, p

value was low so null hypothesis could be rejected, but R squared value and adjusted R squared value is also low so relationship between medv and crim is not significant.

OVERVIEW OF THE NEXT STEPS TAKEN

I will fit a multiple regression model to predict the response using all of the predictors and then I will also write a small explanation of the results. Finally, I will determine for which predictors can we reject the null hypothesis H0 : βj = 0? What will it do, well it will help us determines which variables are important predictors.

```
lm.fitmultiple <- lm(crim~.,data = Boston)
summary(lm.fitmultiple)
```

```
##
## 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
```

#we see that the predictors “zn”, “dis”, “rad”, “black” and “medv” are statistically significant becuase of their low p-value. Hence we can reject the null-hypothesis for these predictors.

OVERVIEW OF THE THIRD PART

Then comparing how the results from (a) compare to the results from (b)? Then I will create a plot displaying the univariate regression coefficients from (a) on the x-axis, and the multiple regression coefficients from (b) on the y-axis. That is, each predictor will be displayed as a single point in the plot. Its coefficient in a simple linear regression model will be shown on the x-axis, and its coefficient estimate in the multiple linear regression model will be shown on the y-axis.

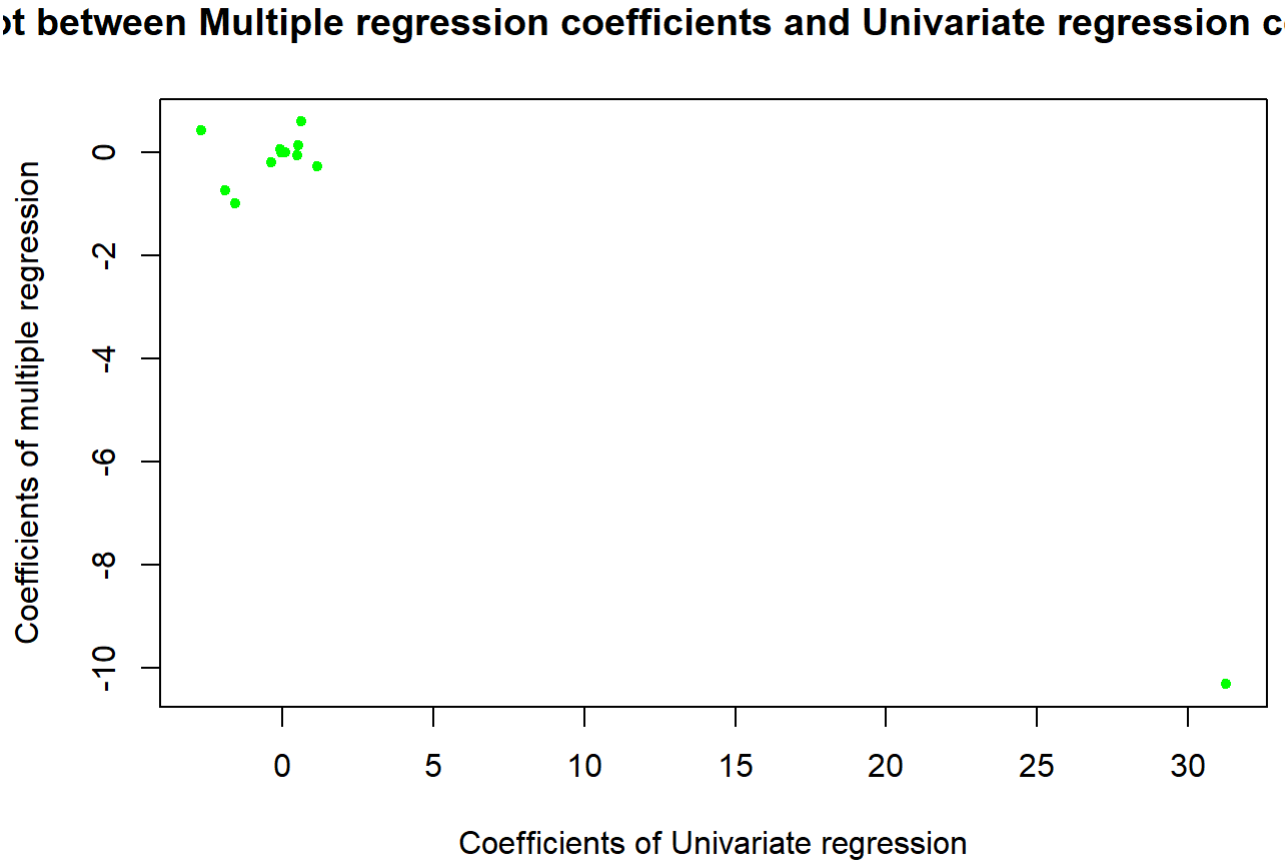
```
#First lets create a vector with the coefficients of all the simple regression models in a.
simplecoef <- vector("numeric", 0)
simplecoef <- c(simplecoef, lm.fit1$coefficients[2])
simplecoef <- c(simplecoef, lm.fit2$coefficients[2])
simplecoef <- c(simplecoef, lm.fit3$coefficients[2])
simplecoef <- c(simplecoef, lm.fit4$coefficients[2])
simplecoef <- c(simplecoef, lm.fit5$coefficients[2])
simplecoef <- c(simplecoef, lm.fit6$coefficients[2])
simplecoef <- c(simplecoef, lm.fit7$coefficients[2])
simplecoef <- c(simplecoef, lm.fit8$coefficients[2])
simplecoef <- c(simplecoef, lm.fit9$coefficients[2])
simplecoef <- c(simplecoef, lm.fit10$coefficients[2])
simplecoef <- c(simplecoef, lm.fit11$coefficients[2])
simplecoef <- c(simplecoef, lm.fit12$coefficients[2])
simplecoef <- c(simplecoef, lm.fit13$coefficients[2])
simplecoef
```

```
##          zn       indus        chas         nox          rm         age
## -0.07393498  0.50977633 -1.89277655 31.24853120 -2.68405122  0.10778623
##         dis         rad         tax     ptratio       black       lstat
## -1.55090168  0.61791093  0.02974225  1.15198279 -0.03627964  0.54880478
##        medv
## -0.36315992
```

```
#Now, creating a vector for the multiple regression coefficients
multiplecoef <- vector("numeric", 0)
multiplecoef <- c(multiplecoef, lm.fitmultiple$coefficients)
multiplecoef <- multiplecoef[-1]
multiplecoef
```

```
##            zn         indus          chas           nox            rm
##   0.044855215  -0.063854824  -0.749133611 -10.313534912   0.430130506
##           age           dis           rad           tax       ptratio
##   0.001451643  -0.987175726   0.588208591  -0.003780016  -0.271080558
##         black         lstat          medv
##  -0.007537505   0.126211376  -0.198886821
```

```
#Now plotting,
plot(simplecoef, multiplecoef, col = "green", pch = 20, ylab = "Coefficients of multiple regression", x
lab = "Coefficients of Univariate regression", main = "Plot between Multiple regression coefficients an
d Univariate regression coefficients")
```



#Now

explaining the first part of the question, we see that the coefficients in (a) or Univariate regression coefficients and the coefficients in (b) or multiple regression coefficients have a striking difference. We see that the according to multiple regression, per capita crime has almost no relationship with the a lot of the predictors if not all. However, in the simple regression it is not the case as there are association of per capita crime with a lot of predictors.

OVERVIEW OF THE NEXT PART

I am checking if there is evidence of non-linear association between any of the predictors and the response? To answer this question, for each predictor X, I will fit a model of the form Y = β0 + β1X + β2X2 + β3X3 + ϵ.

```
#crim=ϐ0+ϐ1(zn)+ϐ2(zn)2+ϐ3(zn)3+ϵ
lm.zn <- lm(crim ~ zn + I(zn^2) + I(zn^3), data = Boston)
summary(lm.zn)
```

```
##
## Call:
## lm(formula = crim ~ zn + I(zn^2) + I(zn^3), data = Boston)
##
## 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)  4.846e+00  4.330e-01  11.192  < 2e-16 ***
## zn          -3.322e-01  1.098e-01  -3.025  0.00261 **
## I(zn^2)      6.483e-03  3.861e-03   1.679  0.09375 .
## I(zn^3)     -3.776e-05  3.139e-05  -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
```

#Zn and crim does not have a non-linear association as the p-values for the degree 2 term and the degree three term are large.

```
#crim=β0+β1(indus)+β2(indus)2+β3(indus)3+ϵ
lmindus <- lm(crim ~ indus + I(indus^2) + I(indus^3), data = Boston)
summary(lmindus)
```

```
##
## Call:
## lm(formula = crim ~ indus + I(indus^2) + I(indus^3), data = Boston)
##
## 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)  3.6625683  1.5739833   2.327   0.0204 *
## indus       -1.9652129  0.4819901  -4.077 5.30e-05 ***
## I(indus^2)   0.2519373  0.0393221   6.407 3.42e-10 ***
## I(indus^3)  -0.0069760  0.0009567  -7.292 1.20e-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
```

#Since the p values of both the degree 2 term and degree3 term are small, there is a non-linear association with crim.

```
#crim=β0+β1(chas)+β2(chas)2+β3(chas)3+ϵ
lmchas <- lm(crim ~ chas + I(chas^2) + I(chas^3), data = Boston)
summary(lmchas)
```

```
##
## Call:
## lm(formula = crim ~ chas + I(chas^2) + I(chas^3), data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.738 -3.661 -3.435  0.018 85.232
##
## Coefficients: (2 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7444     0.3961   9.453   <2e-16 ***
## chas         -1.8928     1.5061  -1.257    0.209
## I(chas^2)         NA         NA      NA       NA
## I(chas^3)         NA         NA      NA       NA
## ---
## 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
```

#it shows NA as chas is a factor so it does not affect the crime rate.

```
#crim=β0+β1(nox)+β2(nox)2+β3(nox)3+ϵ
lmnox <- lm(crim ~ nox + I(nox^2) + I(nox^3), data = Boston)
summary(lmnox)
```

```
##
## Call:
## lm(formula = crim ~ nox + I(nox^2) + I(nox^3), data = Boston)
##
## 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)   233.09      33.64   6.928 1.31e-11 ***
## nox         -1279.37     170.40  -7.508 2.76e-13 ***
## I(nox^2)     2248.54     279.90   8.033 6.81e-15 ***
## I(nox^3)    -1245.70     149.28  -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
```

#as per p-values, nox does have a non-linear association with crim

```
#crim=β0+β1(rm)+β2(rm)2+β3(rm)3+ϵ
lmrm <- lm(crim ~ rm + I(rm^2) + I(rm^3), data = Boston)
summary(lmrm)
```

```
##
## Call:
## lm(formula = crim ~ rm + I(rm^2) + I(rm^3), data = Boston)
##
## 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) 112.6246    64.5172   1.746   0.0815 .
## rm          -39.1501    31.3115  -1.250   0.2118
## I(rm^2)       4.5509     5.0099   0.908   0.3641
## I(rm^3)      -0.1745     0.2637  -0.662   0.5086
## ---
## 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
```

#as per p-value, rm does NOT have a non-linear association with crim

```
#crim=β0+β1(age)+β2(age)2+β3(age)3+ϵ
lmage <- lm(crim ~ age + I(age^2) + I(age^3), data = Boston)
summary(lmage)
```

```
##
## Call:
## lm(formula = crim ~ age + I(age^2) + I(age^3), data = Boston)
##
## 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) -2.549e+00  2.769e+00  -0.920  0.35780
## age          2.737e-01  1.864e-01   1.468  0.14266
## I(age^2)    -7.230e-03  3.637e-03  -1.988  0.04738 *
## I(age^3)     5.745e-05  2.109e-05   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
```

#as per p-values, age has a non-linear association with crim

```
#crim=β0+β1(dis)+β2(dis)2+β3(dis)3+ϵ
lmdis <- lm(crim ~ dis + I(dis^2) + I(dis^3), data = Boston)
summary(lmdis)
```

```
##
## Call:
## lm(formula = crim ~ dis + I(dis^2) + I(dis^3), data = Boston)
##
## 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)  30.0476     2.4459  12.285  < 2e-16 ***
## dis         -15.5543     1.7360  -8.960  < 2e-16 ***
## I(dis^2)      2.4521     0.3464   7.078 4.94e-12 ***
## I(dis^3)     -0.1186     0.0204  -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
```

#as per the p values, dis has a non-linear association with crim

```
#crim=β0+β1(rad)+β2(rad)2+β3(rad)3+ϵ
lmrad <- lm(crim ~ rad + I(rad^2) + I(rad^3), data = Boston)
summary(lmrad)
```

```
##
## Call:
## lm(formula = crim ~ rad + I(rad^2) + I(rad^3), data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.381  -0.412  -0.269   0.179  76.217
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.605545   2.050108  -0.295    0.768
## rad          0.512736   1.043597   0.491    0.623
## I(rad^2)    -0.075177   0.148543  -0.506    0.613
## I(rad^3)     0.003209   0.004564   0.703    0.482
##
## Residual standard error: 6.682 on 502 degrees of freedom
## Multiple R-squared:    0.4,  Adjusted R-squared:  0.3965
## F-statistic: 111.6 on 3 and 502 DF,  p-value: < 2.2e-16
```

#as per the p values, rad does not have a non-linear association with crim

```
#crim=β0+β1(tax)+β2(tax)2+β3(tax)3+ϵ
lmtax <- lm(crim ~ tax + I(tax^2) + I(tax^3), data = Boston)
summary(lmtax)
```

```
##
## Call:
## lm(formula = crim ~ tax + I(tax^2) + I(tax^3), data = Boston)
##
## 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)  1.918e+01  1.180e+01   1.626    0.105
## tax         -1.533e-01  9.568e-02  -1.602    0.110
## I(tax^2)     3.608e-04  2.425e-04   1.488    0.137
## I(tax^3)    -2.204e-07  1.889e-07  -1.167    0.244
##
## 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
```

#as per p value, tax does not have a non-linear association with crim

```
#crim=β0+β1(ptratio)+β2(ptratio)2+β3(ptratio)3+ϵ
lmptratio <- lm(crim ~ ptratio + I(ptratio^2) + I(ptratio^3), data = Boston)
summary(lmptratio)
```

```
##
## Call:
## lm(formula = crim ~ ptratio + I(ptratio^2) + I(ptratio^3), data = Boston)
##
## 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)  477.18405  156.79498   3.043  0.00246 **
## ptratio      -82.36054   27.64394  -2.979  0.00303 **
## I(ptratio^2)   4.63535    1.60832   2.882  0.00412 **
## I(ptratio^3)  -0.08476    0.03090  -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
```

#as per p value, ptratio shows a non-linear association with crim

```
#crim=β0+β1(black)+β2(black)2+β3(black)3+ϵ
lmblack <- lm(crim ~ black + I(black^2) + I(black^3), data = Boston)
summary(lmblack)
```

```
##
## Call:
## lm(formula = crim ~ black + I(black^2) + I(black^3), data = Boston)
##
## 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)  1.826e+01  2.305e+00   7.924  1.5e-14 ***
## black       -8.356e-02  5.633e-02  -1.483    0.139
## I(black^2)   2.137e-04  2.984e-04   0.716    0.474
## I(black^3)  -2.652e-07  4.364e-07  -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
```

#as per p values, black does not have a non-linear association with crim

```
#crim=β0+β1(lstat)+β2(lstat)2+β3(lstat)3+ϵ
lmlstat <- lm(crim ~ lstat + I(lstat^2) + I(lstat^3), data = Boston)
summary(lmlstat)
```

```
##
## Call:
## lm(formula = crim ~ lstat + I(lstat^2) + I(lstat^3), data = Boston)
##
## 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)  1.2009656  2.0286452   0.592   0.5541
## lstat       -0.4490656  0.4648911  -0.966   0.3345
## I(lstat^2)   0.0557794  0.0301156   1.852   0.0646 .
## I(lstat^3)  -0.0008574  0.0005652  -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
```

#as per the p-value, lstat does not have a non-linear association with crim

```
#crim=β0+β1(medv)+β2(medv)2+β3(medv)3+ϵ
lmmedv <- lm(crim ~ medv + I(medv^2) + I(medv^3), data = Boston)
summary(lmmedv)
```

```
##
## Call:
## lm(formula = crim ~ medv + I(medv^2) + I(medv^3), data = Boston)
##
## 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) 53.1655381  3.3563105  15.840  < 2e-16 ***
## medv        -5.0948305  0.4338321 -11.744  < 2e-16 ***
## I(medv^2)    0.1554965  0.0171904   9.046  < 2e-16 ***
## I(medv^3)   -0.0014901  0.0002038  -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
```

#based on p-values, medv has a non-linear association with crim.

OVERVIEW OF THE NEXT SECTION

We will now try to predict per capita crime rate.

For that we will try out some of the regression methods, such as best subset selection, the lasso, ridge regression, and PCR. Then we will present and discuss results for the approaches that we consider.

#1) subset selection - exhaustive

```
library(MASS)
library(glmnet)
```

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

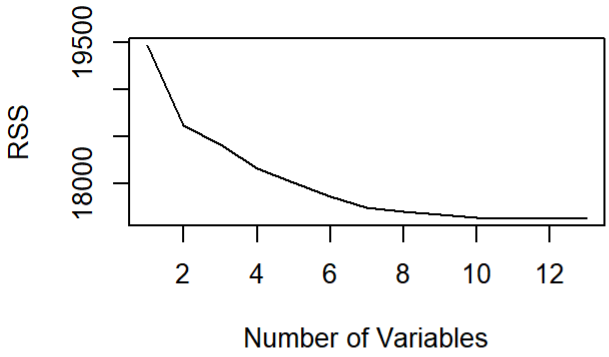
```
## Warning: package 'Matrix' was built under R version 4.3.3
```

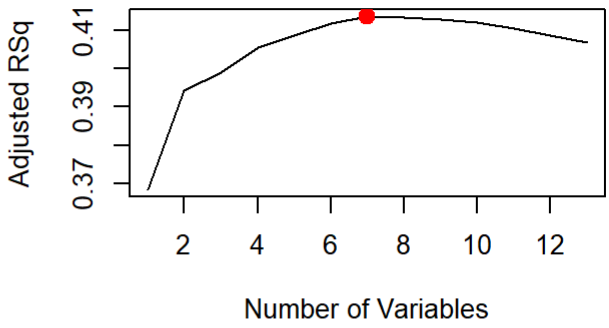
```
## Loaded glmnet 4.1-8
```

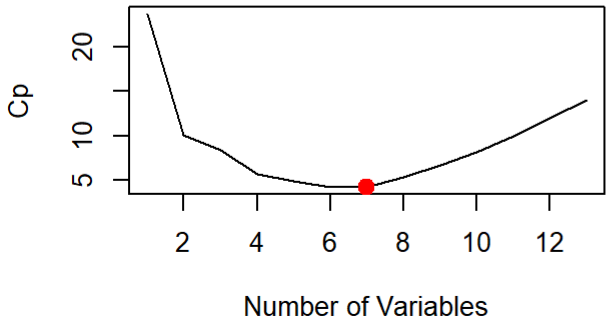
```
library(leaps)
data(Boston)
head(Boston)
```

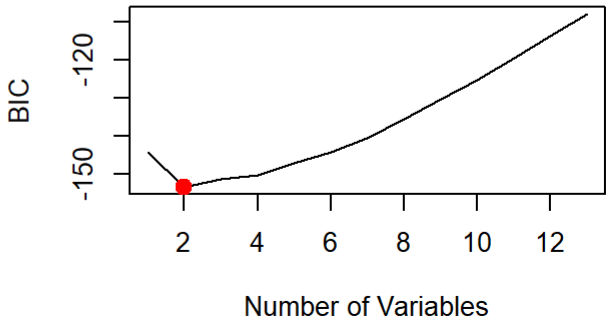
```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio  black lstat
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21
##   medv
## 1 24.0
## 2 21.6
## 3 34.7
## 4 33.4
## 5 36.2
## 6 28.7
```

```
set.seed(123)


indis <- sample(1:nrow(Boston), round(2/3*nrow(Boston)), replace = FALSE)
boston_train <- Boston[indis, ]
boston_test <- Boston[-indis, ]


#using exhaustive selection method
regfit.full <- regsubsets(crim~., data = boston_train, nbest = 1, nvmax = 13, method = "exhaustive")
my_sum <- summary(regfit.full)


par(mfrow=c(2,2))
plot(my_sum$rss,xlab="Number of Variables",ylab="RSS",type="l")

plot(my_sum$adjr2,xlab="Number of Variables",ylab="Adjusted RSq",type="l")
p = which.max(my_sum$adjr2)
points(p,my_sum$adjr2[p], col="red",cex=2,pch=20)

plot(my_sum$cp,xlab="Number of Variables",ylab="Cp",type='l')
p = which.min(my_sum$cp)
points(p,my_sum$cp[p],col="red",cex=2,pch=20)

plot(my_sum$bic,xlab="Number of Variables",ylab="BIC",type='l')
p = which.min(my_sum$bic)
points(p,my_sum$bic[p],col="red",cex=2,pch=20)
```










```
#identifying the optimal models
which(my_sum$cp == min(my_sum$cp))
```

```
## [1] 7
```

```
which(my_sum$bic == min(my_sum$bic))
```

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

```
which(my_sum$rss == min(my_sum$rss))
```

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

```
which(my_sum$adjr2 == max(my_sum$adjr2))
```

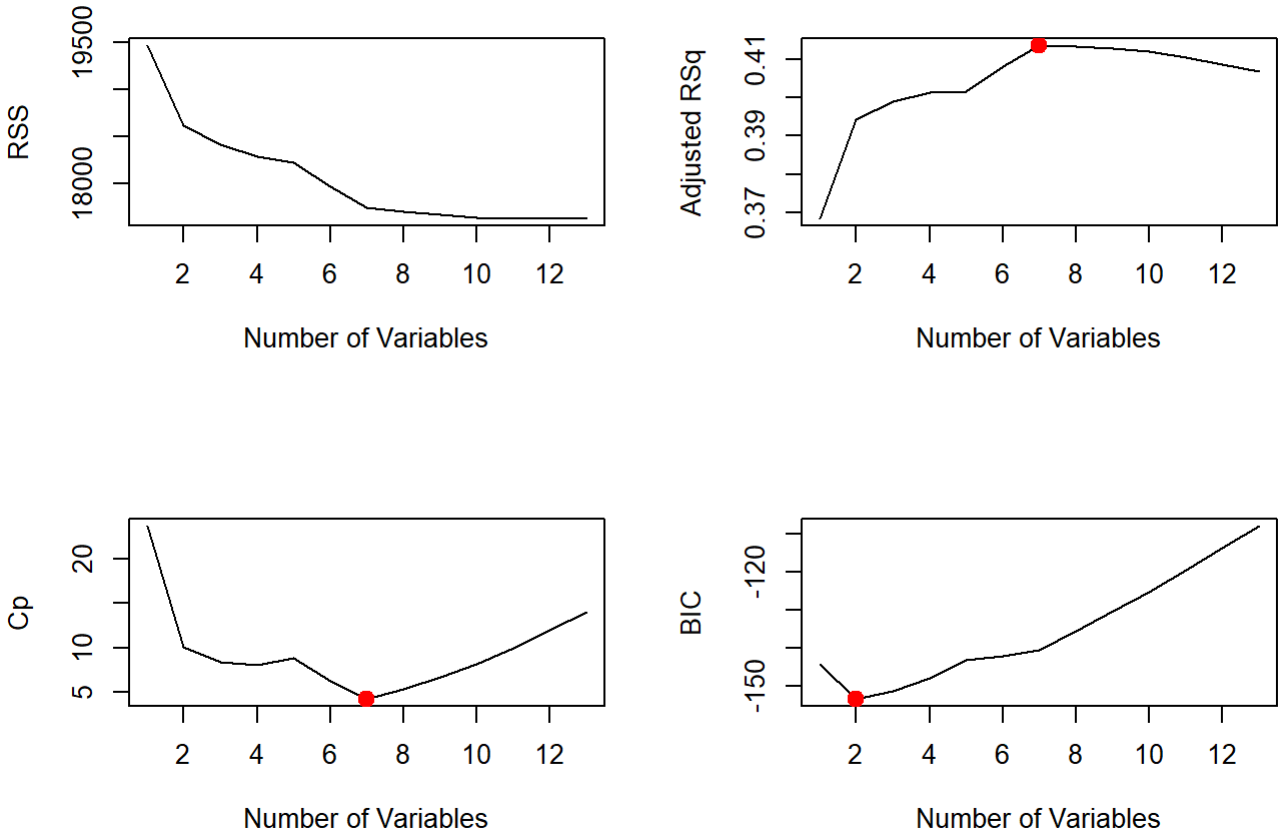
```
## [1] 7
```

#subset selection - forward

```
regfit.fwd <- regsubsets(crim~., data = boston_train, nbest = 1, nvmax = 13, method = "forward")

summary_fwd <- summary(regfit.fwd)


# examining the best "p" variables models
summary_fwd$outmat
```

```
##           zn  indus chas nox rm  age dis rad tax ptratio black lstat medv
## 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 ) "*" "*"   "*"  "*" "*" "*" "*" "*" "*" "*"     "*"   "*"   "*"
```

```
par(mfrow=c(2,2))
plot(summary_fwd$rss,xlab="Number of Variables",ylab="RSS",type="l")

plot(summary_fwd$adjr2,xlab="Number of Variables",ylab="Adjusted RSq",type="l")
p = which.max(summary_fwd$adjr2)
points(p,summary_fwd$adjr2[p], col="red",cex=2,pch=20)

plot(summary_fwd$cp,xlab="Number of Variables",ylab="Cp",type='l')
p = which.min(summary_fwd$cp)
points(p,summary_fwd$cp[p],col="red",cex=2,pch=20)

plot(summary_fwd$bic,xlab="Number of Variables",ylab="BIC",type='l')
p = which.min(summary_fwd$bic)
points(p,summary_fwd$bic[p],col="red",cex=2,pch=20)
```




```
#identifying the optimal models
which(summary_fwd$cp == min(summary_fwd$cp))
```

```
## [1] 7
```

```
which(summary_fwd$bic == min(summary_fwd$bic))
```

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

```
which(summary_fwd$rss == min(summary_fwd$rss))
```

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

```
which(summary_fwd$adjr2 == max(summary_fwd$adjr2))
```

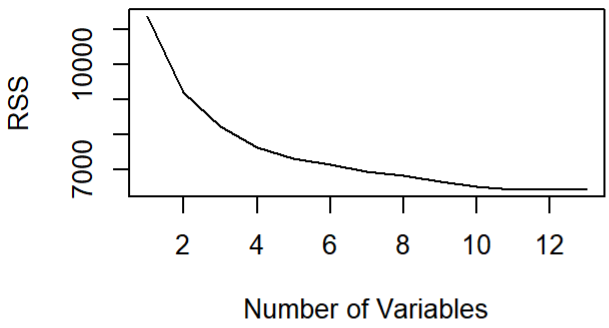
```
## [1] 7
```

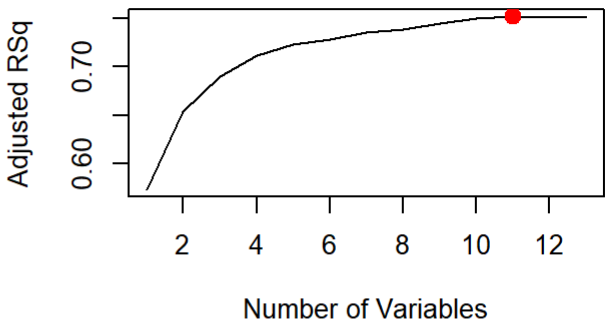
#subset selection - backward

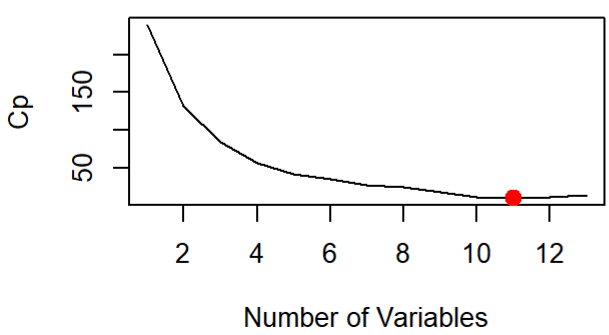
```
regfit.bwd <- regsubsets(medv~., data = boston_train, nbest = 1, nvmax = 13, method = "backward")

summary_bwd <- summary(regfit.bwd)

# examine the best "p" variables models
summary_bwd$outmat
```

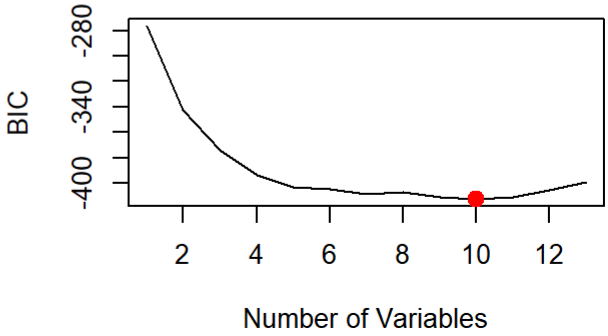
```
##           crim zn  indus chas nox rm  age dis rad tax ptratio black lstat
## 1  ( 1 )  " "  " " " "   " "  " " " " " " " " " " " "     " "   "*"
## 2  ( 1 )  " "  " " " "   " "  " " "*" " " " " " " " " "     " "   "*"
## 3  ( 1 )  " "  " " " "   " "  " " "*" " " " " " " " " "*"     " "   "*"
## 4  ( 1 )  " "  " " " "   "*"  " " "*" " " " " " " " " "*"     " "   "*"
## 5  ( 1 )  " "  " " " "   "*"  " " "*" " " "*" " " " " "*"     " "   "*"
## 6  ( 1 )  "*"  " " " "   "*"  " " "*" " " "*" " " " " "*"     " "   "*"
## 7  ( 1 )  "*"  "*" " "   "*"  " " "*" " " "*" " " " " "*"     " "   "*"
## 8  ( 1 )  "*"  "*" " "   "*"  " " "*" " " "*" " " "*" "*"     " "   "*"
## 9  ( 1 )  "*"  "*" " "   "*"  " " "*" " " "*" "*" "*" "*"     " "   "*"
## 10  ( 1 ) "*"  "*" " "   "*"  "*" "*" " " "*" "*" "*" "*"     " "   "*"
## 11  ( 1 ) "*"  "*" " "   "*"  "*" "*" " " "*" "*" "*" "*"     "*"   "*"
## 12  ( 1 ) "*"  "*" "*"   "*"  "*" "*" " " "*" "*" "*" "*"     "*"   "*"
## 13  ( 1 ) "*"  "*" "*"   "*"  "*" "*" "*" "*" "*" "*" "*"     "*"   "*"
```

```
Test_error <- mean((mean(boston_test$crim)-boston_test$crim)^2)

par(mfrow=c(2,2))
plot(summary_bwd$rss,xlab="Number of Variables",ylab="RSS",type="l")

plot(summary_bwd$adjr2,xlab="Number of Variables",ylab="Adjusted RSq",type="l")
p = which.max(summary_bwd$adjr2)
points(p,summary_bwd$adjr2[p], col="red",cex=2,pch=20)

plot(summary_bwd$cp,xlab="Number of Variables",ylab="Cp",type='l')
p = which.min(summary_bwd$cp)
points(p,summary_bwd$cp[p],col="red",cex=2,pch=20)

plot(summary_bwd$bic,xlab="Number of Variables",ylab="BIC",type='l')
p = which.min(summary_bwd$bic)
points(p,summary_bwd$bic[p],col="red",cex=2,pch=20)
```










```
#identifying the optimal models
which(summary_bwd$cp == min(summary_bwd$cp))
```

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

```
which(summary_bwd$bic == min(summary_bwd$bic))
```

```
## [1] 10
```

```
which(summary_bwd$rss == min(summary_bwd$rss))
```

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

```
which(summary_bwd$adjr2 == max(summary_bwd$adjr2))
```

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

```
coef(regfit.full, 13)
```

```
##   (Intercept)            zn         indus          chas           nox
##  18.493261157   0.052112653  -0.048166230  -0.047529794 -11.566021839
##            rm           age           dis           rad           tax
##   0.781952562   0.002362890  -1.040371125   0.649095528  -0.003889975
##       ptratio         black         lstat          medv
##  -0.326085312  -0.007464261   0.078377954  -0.280845963
```

```
coef(regfit.fwd, 13)
```

```
##   (Intercept)            zn         indus          chas           nox
##  18.493261157   0.052112653  -0.048166230  -0.047529794 -11.566021839
##            rm           age           dis           rad           tax
##   0.781952562   0.002362890  -1.040371125   0.649095528  -0.003889975
##       ptratio         black         lstat          medv
##  -0.326085312  -0.007464261   0.078377954  -0.280845963
```

```
coef(regfit.bwd, 13)
```

```
##   (Intercept)          crim            zn         indus          chas
##  32.530243820  -0.102260447   0.051222999  -0.027924835   4.176966501
##           nox            rm           age           dis           rad
## -10.463214426   3.465141843  -0.003972019  -1.412593289   0.242808424
##           tax       ptratio         black         lstat
##  -0.010587055  -0.722413616   0.006561157  -0.622407786
```

#Linear regression

```
lm.fit <- lm(crim~., data = boston_train)
lm_pred <- predict(lm.fit, boston_test )
Test_error_linear <- mean((boston_test$crim - lm_pred)^2)
Test_error_linear
```

```
## [1] 17.89069
```

```
#plotting the same using the best subset from exhaustive, forward and backward which are the optiamal m
odels as the predictors now
lm.fit1 = lm(crim~medv+dis+indus+black+ptratio, data=boston_train)
lm_pred1 <- predict(lm.fit1, boston_test )
Test_error1_linear_with_bestsubset <- mean((boston_test$crim - lm_pred1)^2)
Test_error1_linear_with_bestsubset
```

```
## [1] 26.54651
```

#Performing LASSO

```
set.seed(123)
X_train = model.matrix(crim~., data = boston_train)
X_test = model.matrix(crim~., data = boston_test)
#Choosing lambda using cross-validation
cv.out = cv.glmnet(X_train, boston_train$crim, alpha=1)
sel = cv.out$lambda.min
sel
```

```
## [1] 0.07355166
```

```
lasso_mod = glmnet(X_train, boston_train$crim, alpha=1, lambda=sel)
#Make predictions
lasso_pred = predict(lasso_mod, s=sel, newx=X_test)
Test_error_lasso <- mean((lasso_pred - boston_test$crim)^2)
Test_error_lasso
```

```
## [1] 17.48612
```

#RIDGE

```
cv.out = cv.glmnet(X_train, boston_train$crim, alpha=0)
sel2 = cv.out$lambda.min
sel2
```

```
## [1] 0.5828843
```

```
ridge_mod = glmnet(X_train, boston_train$crim,alpha = 0)
#Make predictions
ridge_pred = predict(ridge_mod,s=sel2, newx =X_test, lambda=sel2)
#Calculate test error
Test_error_ridge <- mean((ridge_pred - boston_test$crim)^2)
Test_error_ridge
```

```
## [1] 17.0651
```

#PCR

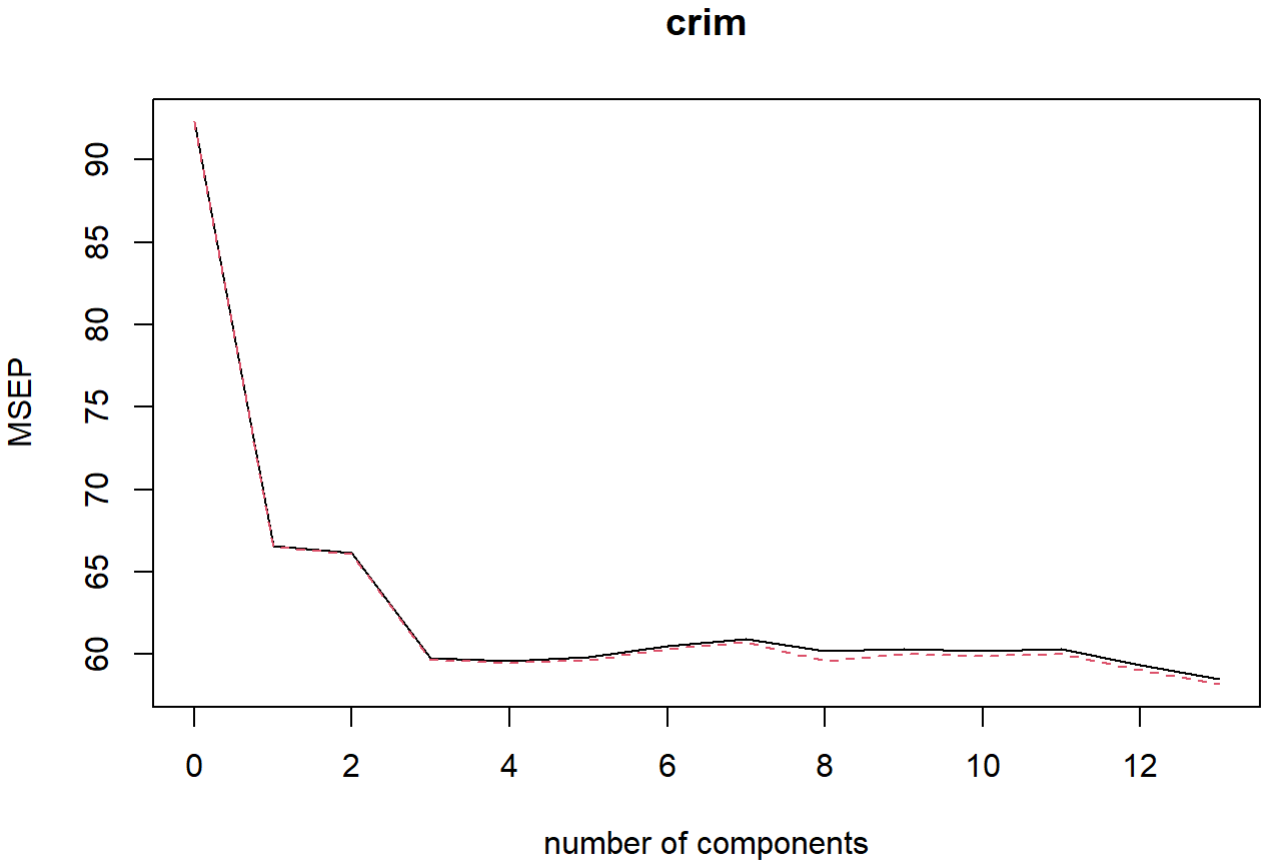
```
library(pls)
```

```
## Warning: package 'pls' was built under R version 4.3.3
```

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

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

```
set.seed(123)

pcrfit <- pcr(crim~., data=boston_train, scale=TRUE, validation="CV")

validationplot(pcrfit, val.type = "MSEP")
```




```
summary(pcrfit)
```

```
## Data:    X dimension: 337 13
##  Y dimension: 337 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           9.607    8.159    8.133    7.733    7.720    7.734    7.780
## adjCV        9.607    8.155    8.130    7.725    7.712    7.726    7.769
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       7.807    7.758    7.767     7.759     7.766     7.706     7.650
## adjCV    7.793    7.720    7.749     7.739     7.748     7.685     7.629
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X       48.58    61.29    70.65    77.41    83.38    88.32    91.54    93.62
## crim    28.70    29.25    36.83    37.26    37.43    37.62    38.06    39.75
##       9 comps  10 comps  11 comps  12 comps  13 comps
## X       95.66     97.25     98.54     99.54    100.00
## crim    40.08     40.39     40.45     41.88     42.97
```

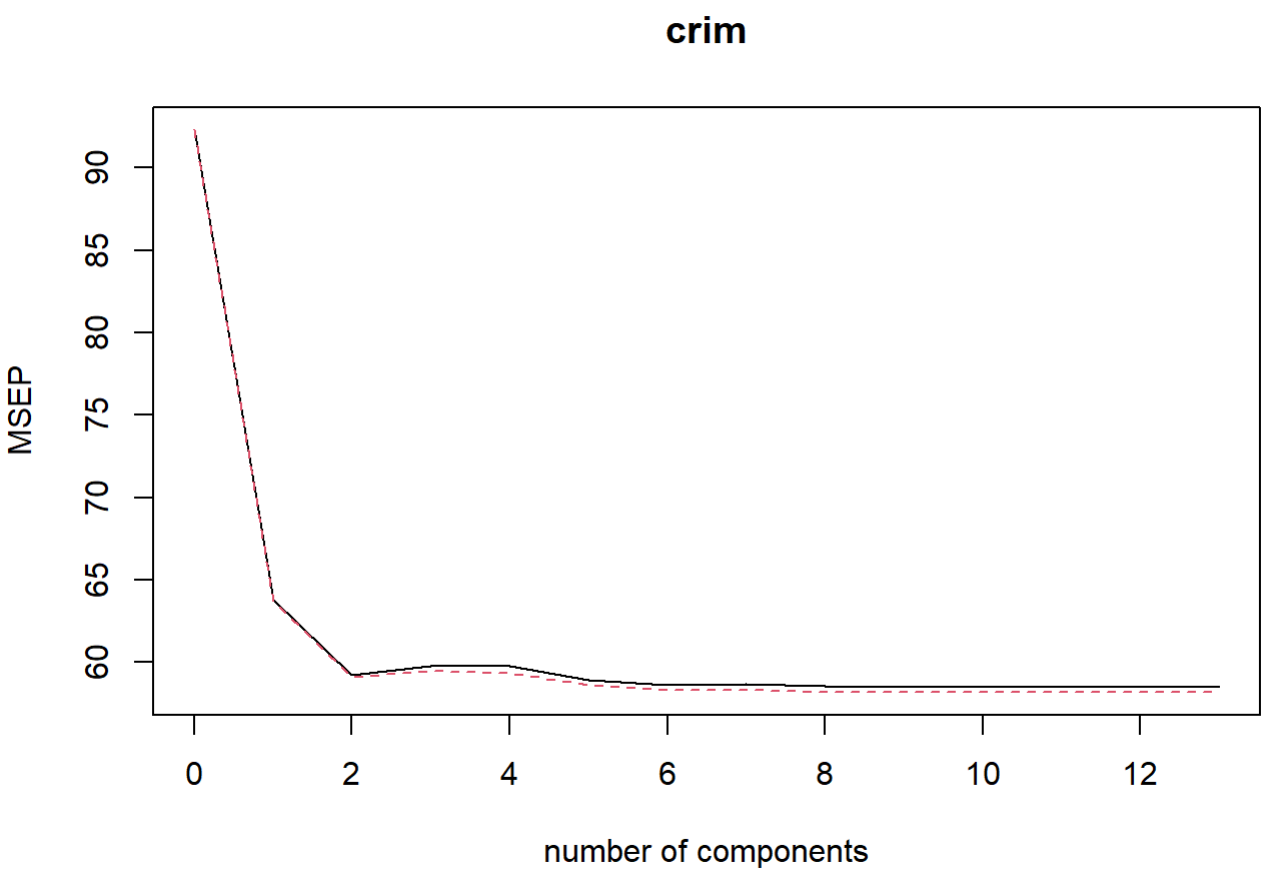
```
#From the summary and the plot, the lowest MSEP occur at M = 13.
pcrfit1 <- pcr(crim~., data=boston_train, scale=TRUE, ncomp=13)
prediction <- predict(pcrfit1, boston_test, ncomp=13)

#test error
Test_error_pcr <- mean((prediction-boston_test$crim)^2)
Test_error_pcr
```

```
## [1] 17.89069
```

#also confirmed the M value by changing the value of ncomp value from 2 to 3 and 8 to 13 and got the minimum value of test error at 13. Hence, considered M = 13 in the final answer.

#PLS

```
set.seed(123)
#Fit and determine M based on CV results
plsfit = plsr(crim~., data=boston_train, scale=TRUE, validation="CV")
validationplot(plsfit, val.type = "MSEP")
```




```
summary(plsfit)
```

```
## Data:    X dimension: 337 13
##  Y dimension: 337 1
## Fit method: kernelpls
## 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           9.607    7.988    7.698    7.731    7.730    7.676    7.658
## adjCV        9.607    7.984    7.688    7.712    7.705    7.654    7.637
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       7.660    7.651    7.650     7.650     7.650     7.650     7.650
## adjCV    7.638    7.630    7.628     7.628     7.629     7.629     7.629
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X       48.12    57.92    64.47    72.49    77.70    80.96    84.77    88.25
## crim    32.18    39.33    41.21    42.12    42.51    42.79    42.90    42.94
##       9 comps  10 comps  11 comps  12 comps  13 comps
## X       91.24     95.05     96.65     98.24    100.00
## crim    42.96     42.97     42.97     42.97     42.97
```

```
#From the summary and the plot, the lowest MSEP occur at M = 13.
plsfit1 = plsr(crim~., data=boston_train, scale=TRUE, ncomp=13)
prediction = predict(plsfit1, boston_test, ncomp = 13)
#test error
Test_error_pls <- mean((prediction - boston_test$crim)^2)
Test_error_pls
```

```
## [1] 17.89069
```

#also confirmed the M value by changing the value of ncomp value from 2 to 3 and 8 to 13 and got the minimum value of test error at 13. Hence, considered M = 13 in the final answer.

#Now to discuss the results of the approaches that I performed

```
Test_error
```

```
## [1] 38.00492
```

```
Test_error_linear
```

```
## [1] 17.89069
```

```
Test_error1_linear_with_bestsubset
```

```
## [1] 26.54651
```

```
Test_error_lasso
```

```
## [1] 17.48612
```

```
Test_error_ridge
```

```
## [1] 17.0651
```

```
Test_error_pcr
```

```
## [1] 17.89069
```

```
Test_error_pls
```

```
## [1] 17.89069
```

#Comapring the results, we see that the linear model, Pcr and pls model perform similarly producing the same test error. The linear model here, performs a little better with all the predictors then with the best selected predictors. Overall, the ridge model here performs slightly better. However, the difference is negligible.

Now lets propose a model (or set of models) that seem to perform well on this data set, and justify my answer.

Here, I am Making sure that I am evaluating model performance using validation set error, crossvalidation, or some other reasonable alternative, as opposed to using training error.

So, I will propose Ridge model as it has a slightly lower test error than other models and as it also stabilizes the model and prevents overcomplicating. However, even Lasso can be considered depending on the requirement and context as it also produces low test error. But since here no context or requiremnet is mentioned, I am proposing Ridge based on its lowest test_error values observed. Hence the best model to predict college applications is Ridge. Following it will be the Lasso model. This evaluation has been made by performing cross-validation and validation set error as that is how the test error values were obtained.

Again, does my chosen model involve all of the features in the data set, let see

Well Ridge Regression, the chosen model, does use all the features (predictors) in the dataset. Ridge Regression uses a technique called L2 regularization, which penalizes the model based on the square of the coefficients. Unlike Lasso, Ridge doesn’t force coefficients to become zero aggressively. Instead, it reduces their impact, helping to stabilize the model and avoid overfitting. Ridge usually includes all the features in the model but manages their influence to create a more balanced and reliable prediction preventing any single feature from dominating the prediction. This ensures a more stable and less prone to overfitting model. It is more about finding the right balance between the best predictors.