Statistical Machine Learning - Homework 01

Vincent Jin

2023-02-09

Homework 01

Persons I worked with: Jing Wang

Linear - ISL 5, 6, 9, 15

Question 5:

Consider the fitted values that result from performing linear regression without an intercept. In this setting, the ith fitted value takes the form

$$\hat{y}_i = x_i\hat{\beta}$$

, where

$$\hat{\beta} = (\sum_{i=1}^{n} x_i y_i)/(\sum_{i'=1}^{n} x_{i'}^2)$$

. Show that we can write

$$\hat{y} = \sum_{i'=1}^{n} a_{i'} y_{i'}$$

What is $a_{i'}$?

Answer:

Since we have $\hat{\beta}$, we can plug in the formula into $\hat{y}$ and get:

$$\hat{y} = x_i * (\sum_{i_1=1}^{n} x_{i_1} y_{i_1})/(\sum_{i_2=1}^{n} x_{i_2}^2) = \sum_{i'=1}^{n} a_{i'} y_{i'}$$

After transformation we can have:

$$\sum_{i_1=1}^{n} \frac{(x_{i_1} y_{i_1}) \times x_i}{\sum_{i_2=1}^{n} x_{i_2}^2} = \sum_{i'=1}^{n} a_{i'} y_{i'}$$

Further transformation:

$$\sum_{i_1=1}^{n} \left(\frac{x_i x_{i_1}}{\sum_{i_2=1}^{n} x_{i_2}^2} \times y_{i_1} \right) = \sum_{i'=1}^{n} a_{i'} y_{i'}$$

Therefore, $a_{i'}$ is equal to:

$$\frac{x_i x_{i_1}}{\sum_{i_2=1}^{n} x_{i_2}^2}$$

Question 6:

Using (3.4), argue that in the case of simple linear regression, the least squares line always passes through the point $(\bar{x}, \bar{y})$.

Answer:

According to (3.4), $y_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ and $\beta_0 = \bar{y} - \hat{\beta}_1 \bar{x}$. When we substitute x_i with $\bar{x}$, we can have: $y_i = \bar{y} - \beta_1 \bar{x} + \beta_1 \bar{x}$. The last part of the equation canceled each other, so that $\hat{y}_i$ will be equal to $\bar{y}$, which means the least squares line will always pass through $(\bar{x}, \bar{y})$.

Question 9:

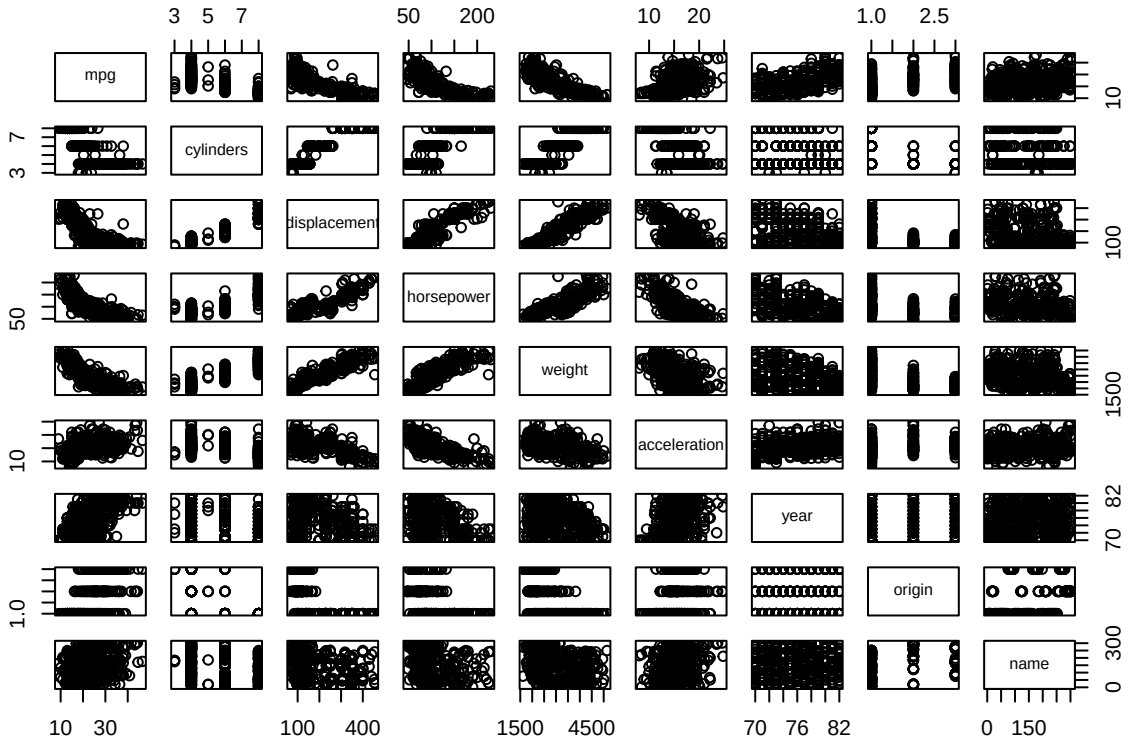
This question involves the use of multiple linear regression on the Auto data set. (a) Produce a scatter plot matrix which includes all of the variables in the data set.

```
#install.packages('ISLR')
library(ISLR)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
```

```
pairs(Auto)
```



(b) Compute the matrix of correlations between the variables using the function cor(). You will need to exclude the name variable, cor() which is qualitative.

```
x <- Auto
x %>% select(-(name)) %>%
  cor()
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
## origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054
##              acceleration       year     origin
## mpg             0.4233285  0.5805410  0.5652088
## cylinders      -0.5046834 -0.3456474 -0.5689316
## displacement   -0.5438005 -0.3698552 -0.6145351
## horsepower     -0.6891955 -0.4163615 -0.4551715
## weight         -0.4168392 -0.3091199 -0.5850054
## acceleration    1.0000000  0.2903161  0.2127458
## year            0.2903161  1.0000000  0.1815277
## origin          0.2127458  0.1815277  1.0000000
```

(c) Use the lm() function to perform a multiple linear regression with mpg as the response and all other variables except name as the predictors. Use the summary() function to print the results. Comment on the output. For instance:

```
x2 <- x %>%
        select(-(name))
reg <- lm(mpg ~ ., data = x2)
summary(reg)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

i. Is there a relationship between the predictors and the response?

Answer:

According to the results of from the summary command, we can see that the associated p-value was less than $2.2 * 10^{-16}$, which was less than 0.05, so that we can reject the null hypothesis of no relationship between the predictors and the response thus conclude that there was a relationship between the predictors and the response variable.

ii. Which predictors appear to have a statistically significant relationship to the response?

Answer:

Based on the regression results, we can see that predictors of: displacement, weight, year, origin were significantly associated with mpg while predictors of: cylinders, horsepower, acceleration were not.

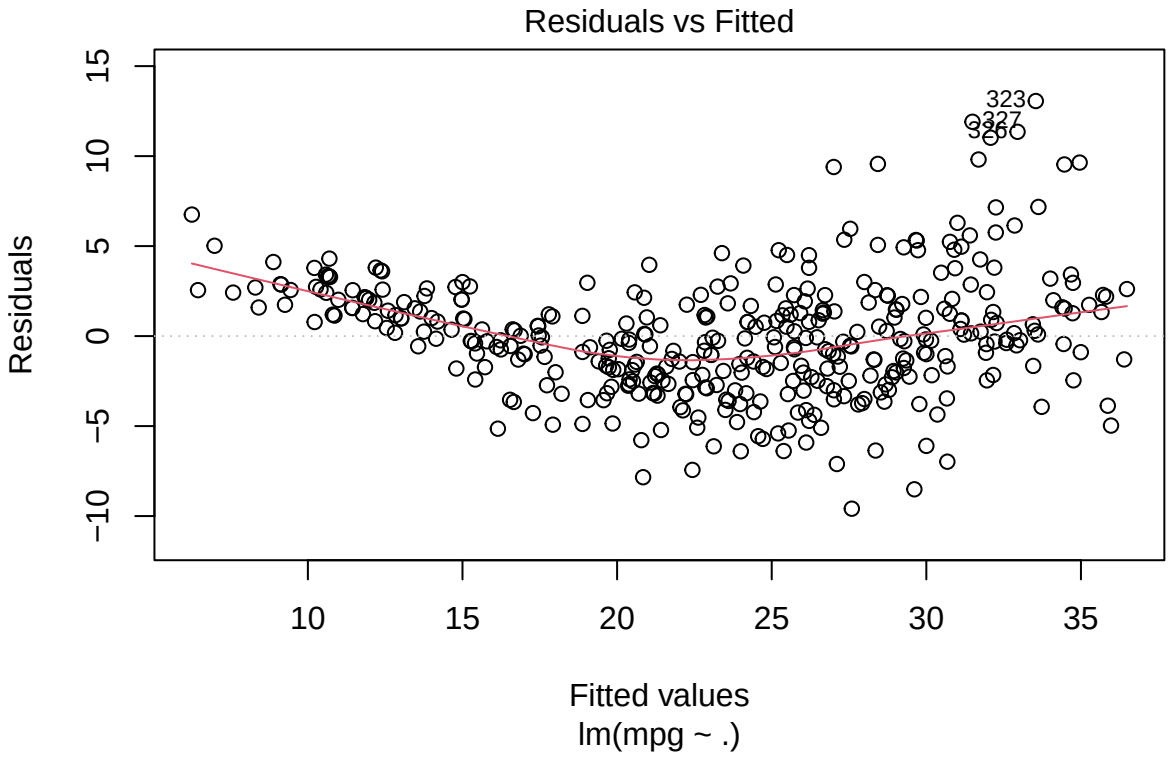
iii. What does the coefficient for the year variable suggest?

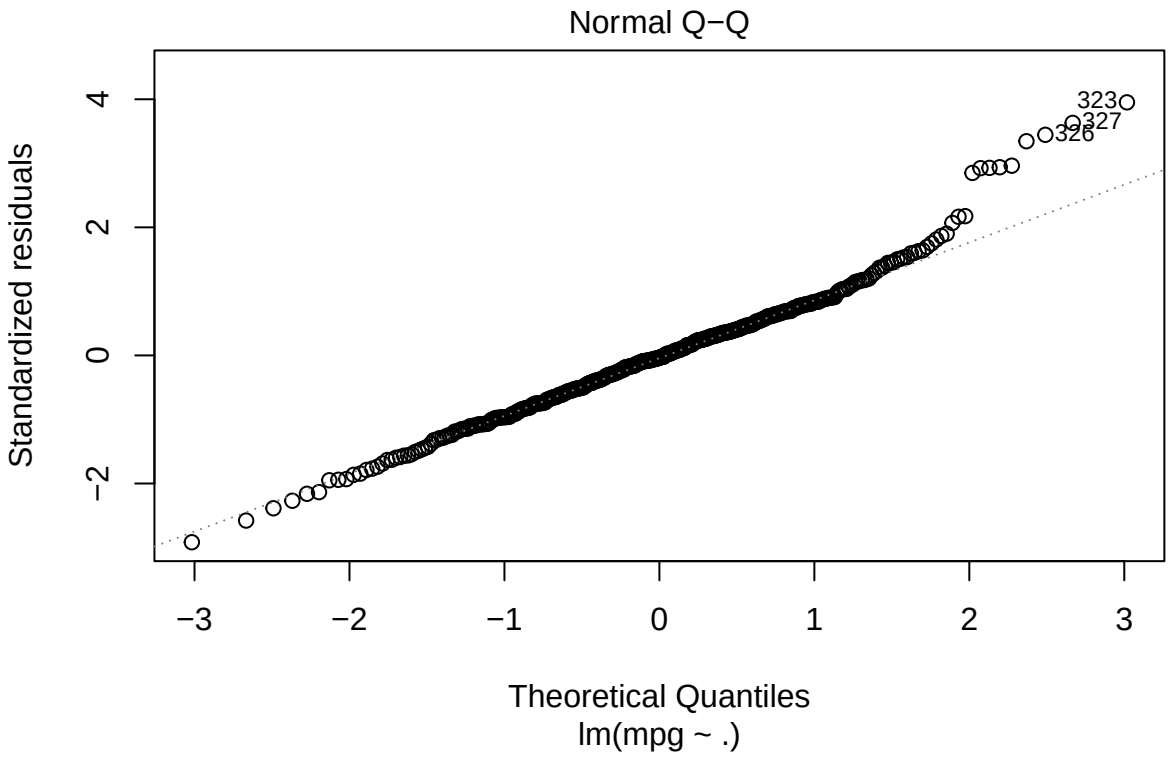
Answer:

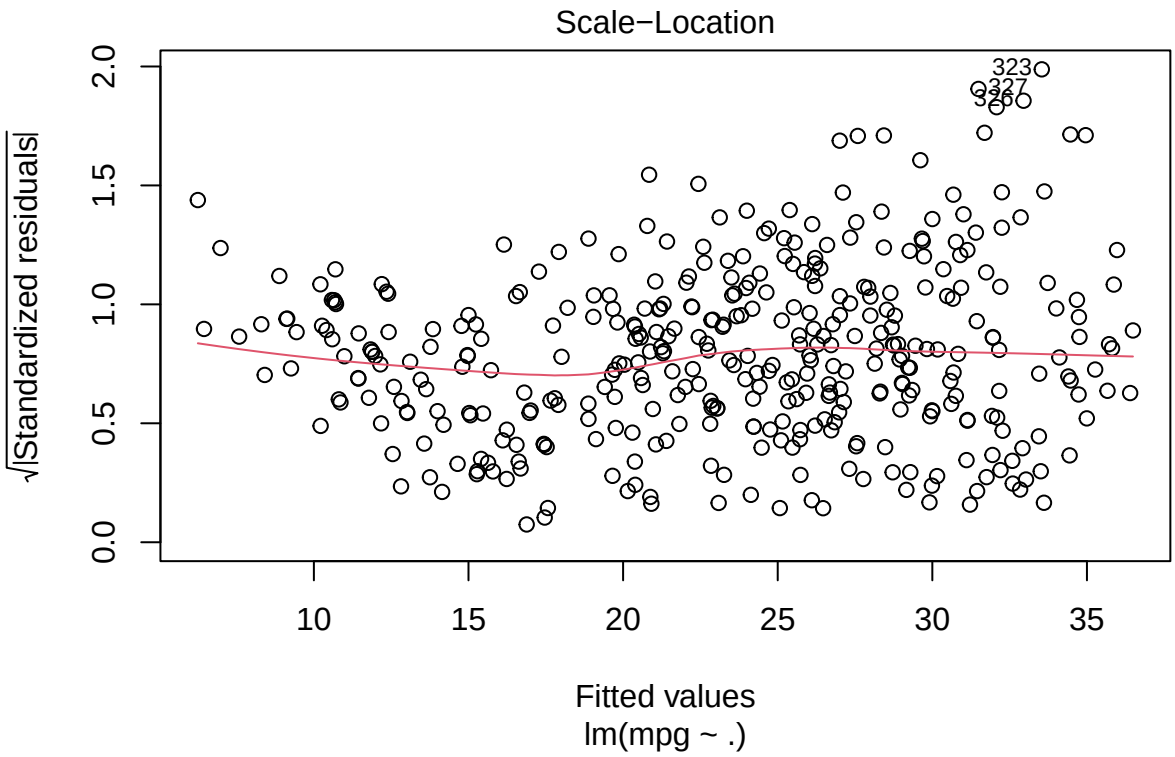
The coefficient for the year variable represents that for every 1 increase in year there is a 0.750773 increase on mpg, adjusting for other variables.

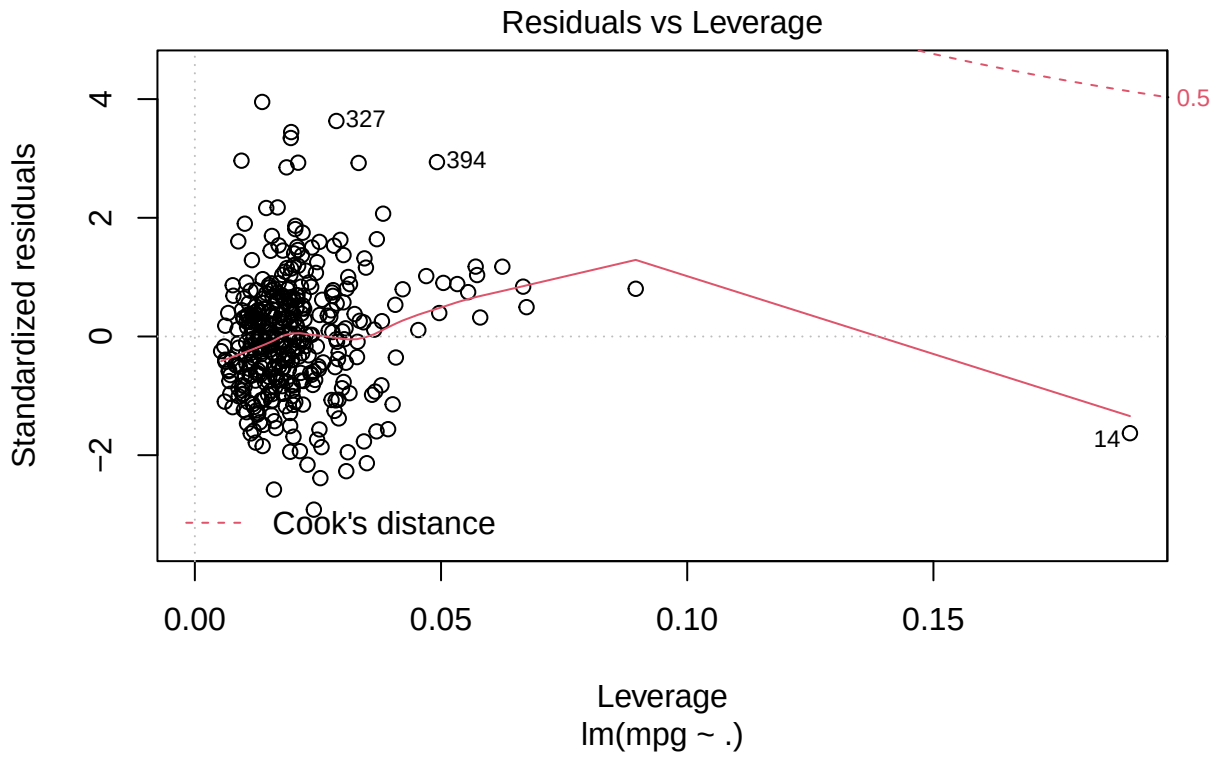
(d) Use the plot() function to produce diagnostic plots of the linear regression fit. Comment on any problems you see with the fit. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusually high leverage?

```
plot(reg)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(mpg ~ .)
323
327
326
−3
−2
−1
0
1
2
3
−2
0
2
4


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
323
327
326
10
15
20
25
30
35
Fitted values
lm(mpg ~ .)



Answer

The residual vs. fitted value plot suggested that there seems to be a pattern which suggests non-linearity in the data. The residual plot also suggested object 323, 326, 327 were three unusal large outliers. The leverage plot also suggested that object 14 had an unusual high leverage.

(e) Use the * and : symbols to fit linear regression models with interaction effects. Do any interactions appear to be statistically significant?

```
reg2 <- lm(mpg ~ cylinders*displacement + displacement*horsepower + displacement*weight, data = x2)
summary(reg2)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders * displacement + displacement *
##     horsepower + displacement * weight, data = x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.1308  -2.1597  -0.3652   1.9001  16.9864
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 5.584e+01  2.569e+00  21.733  < 2e-16 ***
## cylinders                   3.330e-01  8.190e-01   0.407   0.6845
## displacement               -9.524e-02  1.605e-02  -5.935 6.59e-09 ***
```

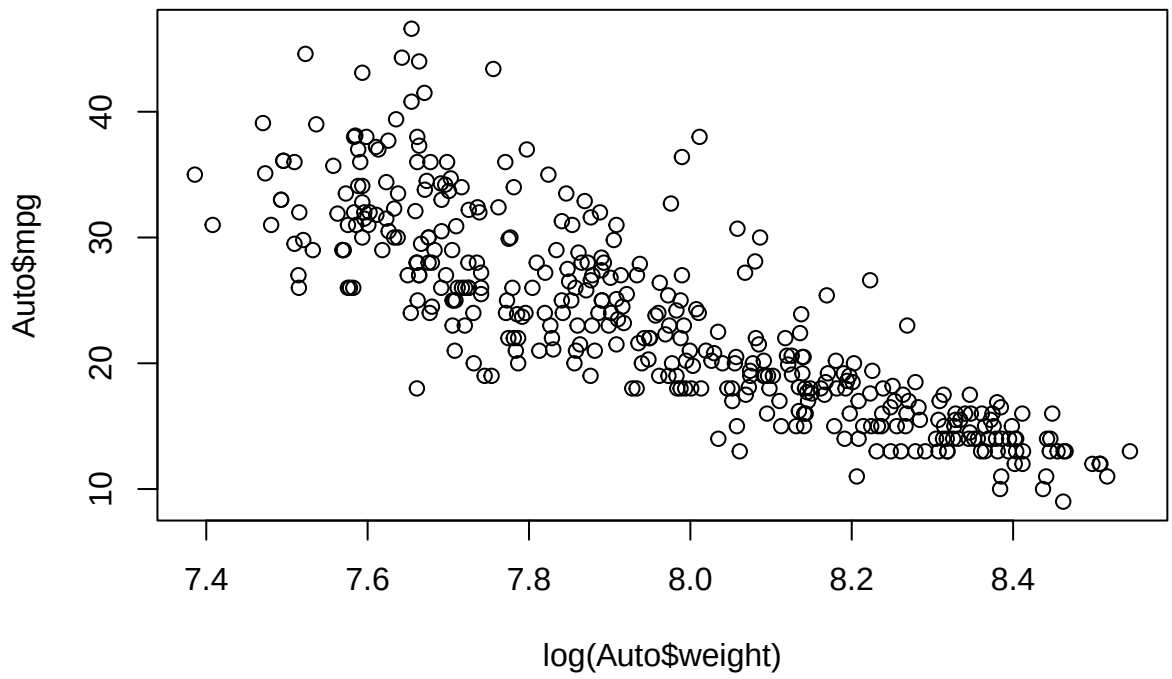
```
## horsepower              -1.844e-01  2.855e-02  -6.460 3.18e-10 ***
## weight                  -3.803e-03  1.589e-03  -2.394   0.0172 *
## cylinders:displacement   1.569e-03  3.581e-03   0.438   0.6615
## displacement:horsepower  4.238e-04  9.786e-05   4.331 1.90e-05 ***
## displacement:weight      4.258e-06  5.555e-06   0.766   0.4439
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.865 on 384 degrees of freedom
## Multiple R-squared:  0.7591, Adjusted R-squared:  0.7547
## F-statistic: 172.9 on 7 and 384 DF,  p-value: < 2.2e-16
```

Answer

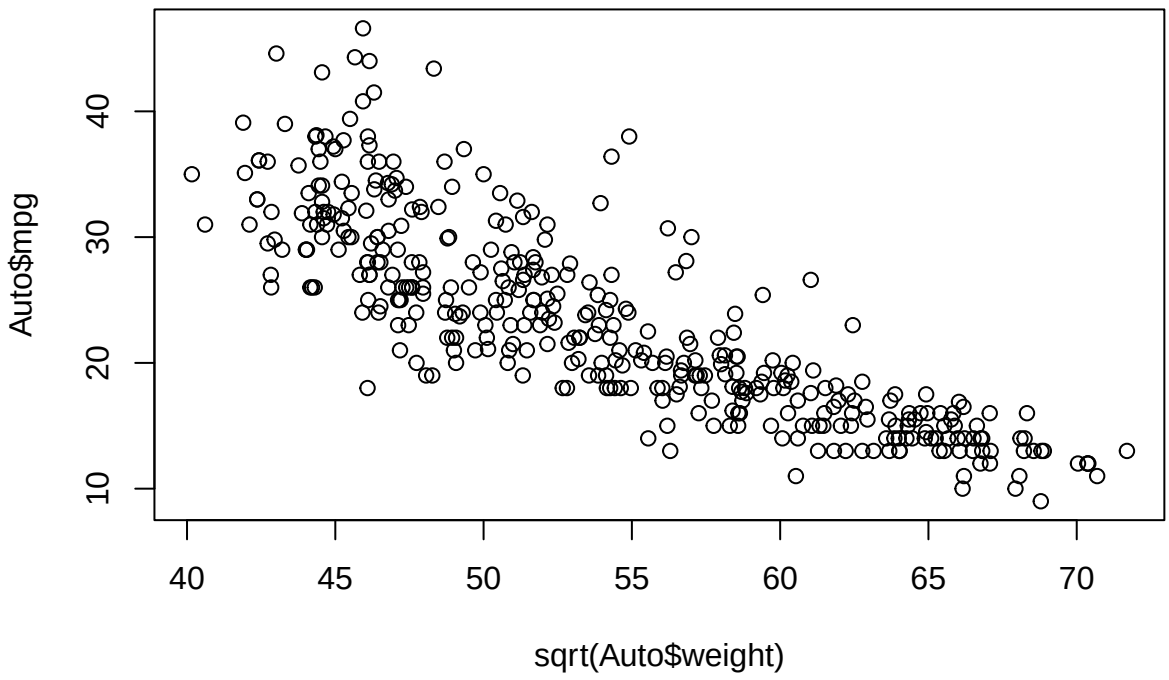
Based on the results, the interaction between displacement and horsepower tened to be statistically significant, while the interaction between cylinders and displacement, and displacement and weight were not significant.

(f) Try a few different transformations of the variables, such aslog(X), $\sqrt{x}$, X^2. Comment on your findings.

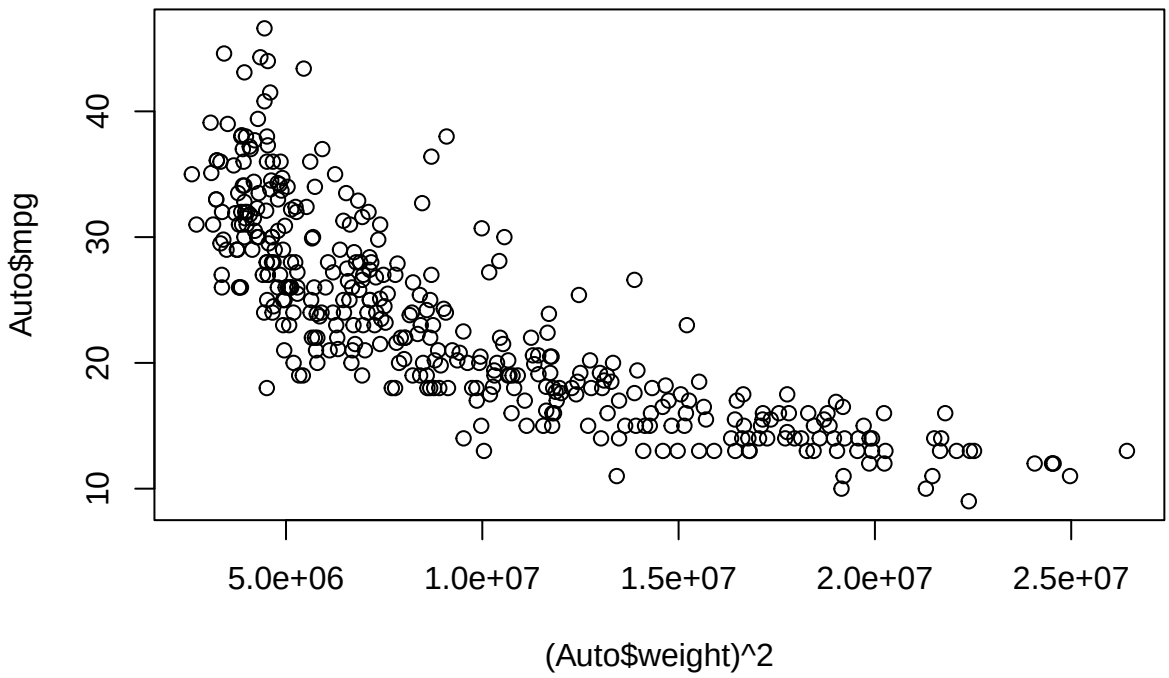
```
plot(log(Auto$weight), Auto$mpg)
```



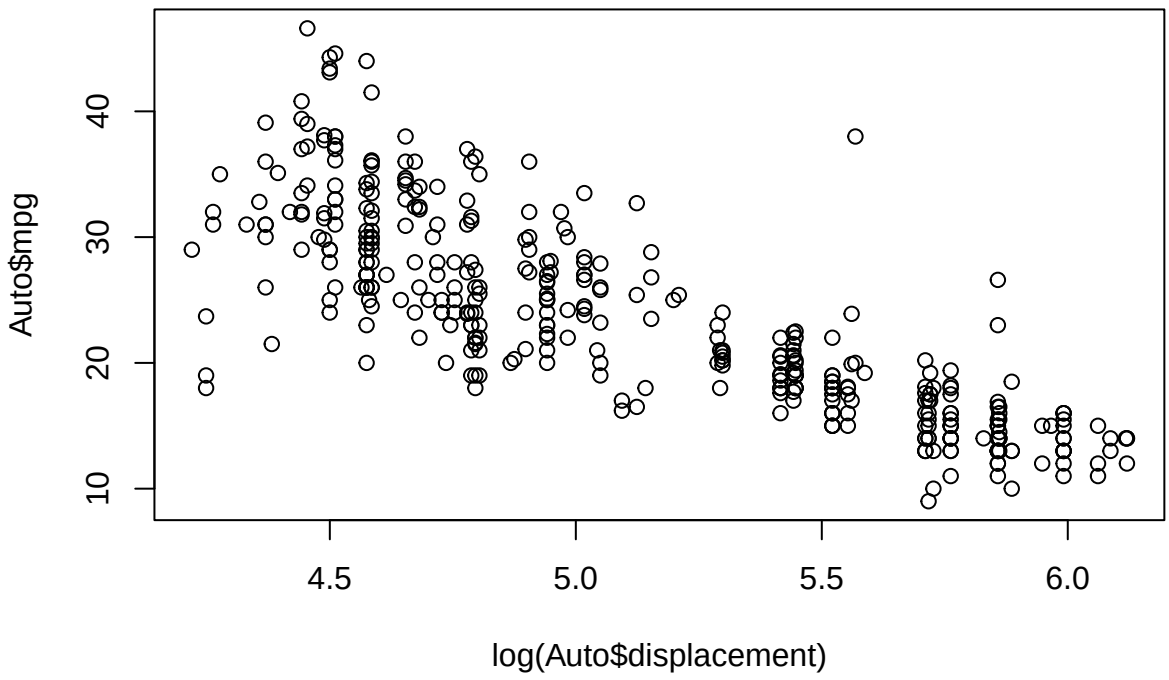

```
plot(sqrt(Auto$weight), Auto$mpg)
```



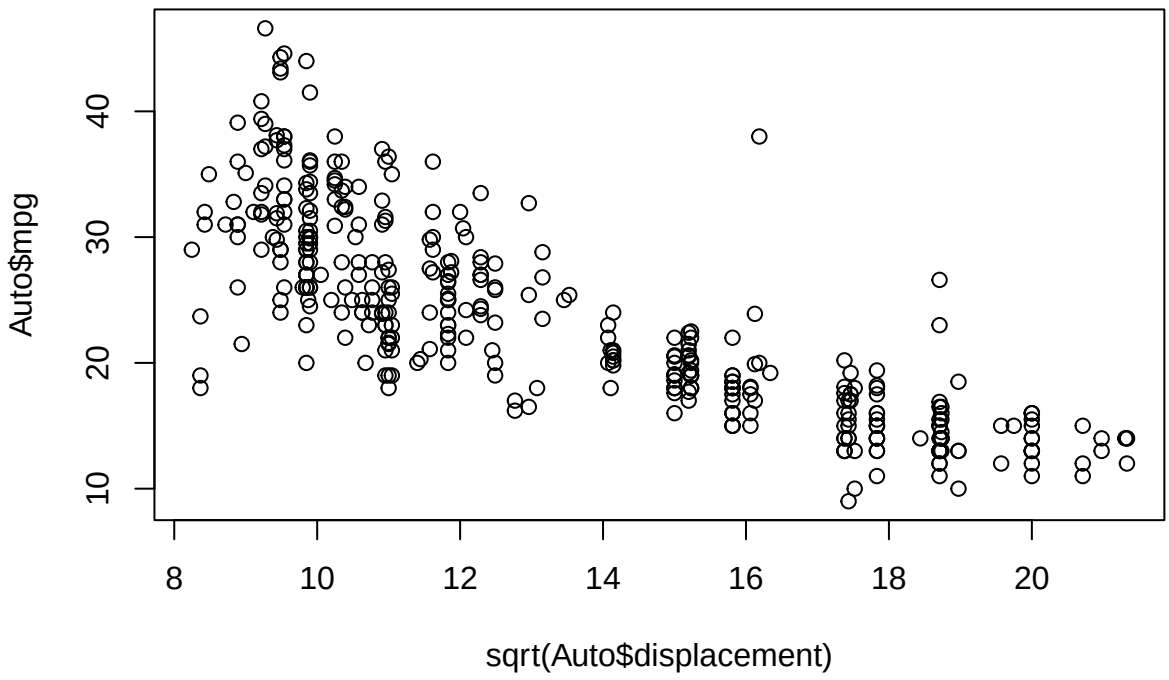

```
plot((Auto$weight)^2, Auto$mpg)
```



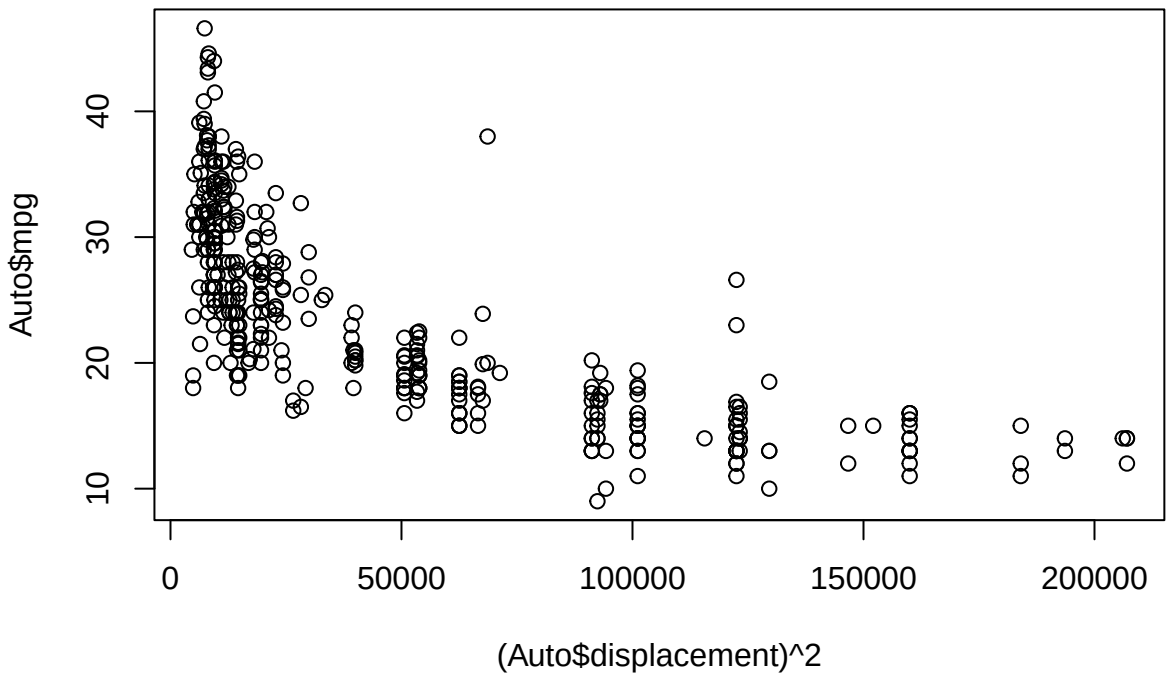

```
plot(log(Auto$displacement), Auto$mpg)
```



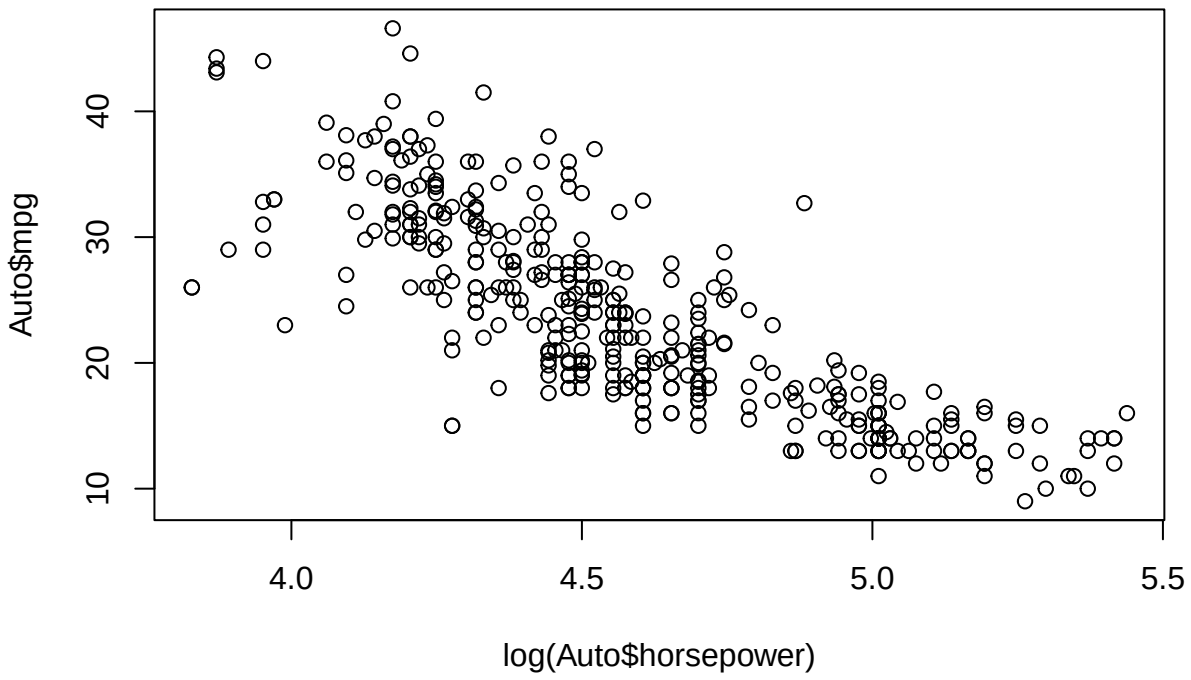

```
plot(sqrt(Auto$displacement), Auto$mpg)
```



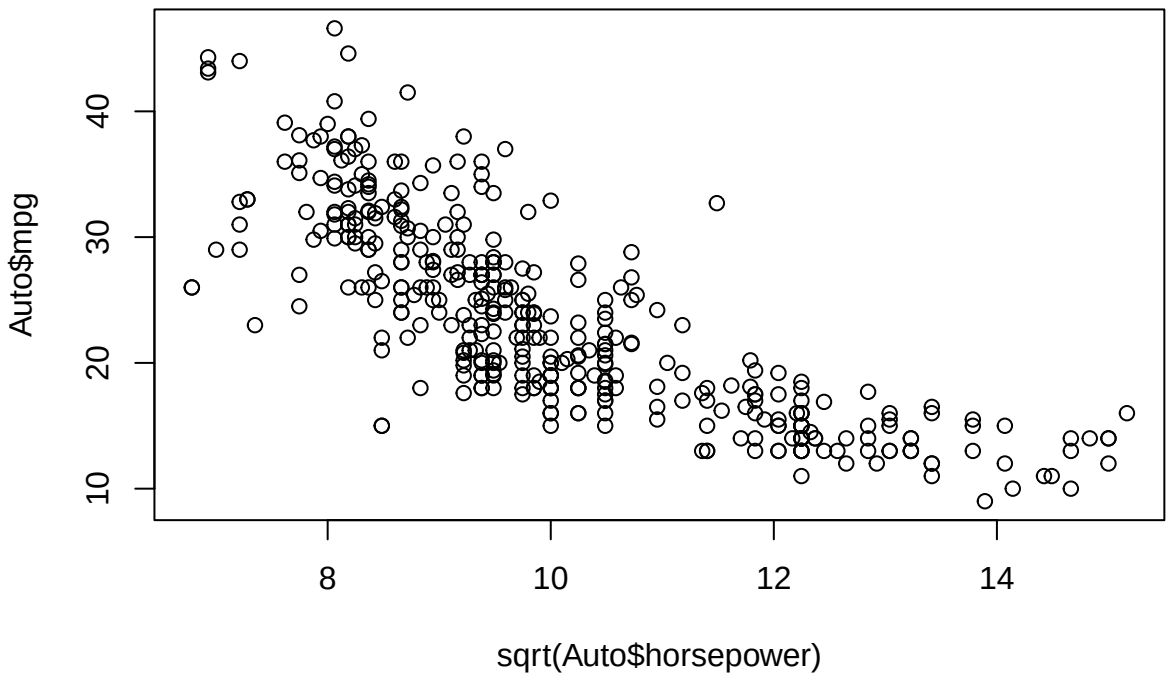

```
plot((Auto$displacement)^2, Auto$mpg)
```



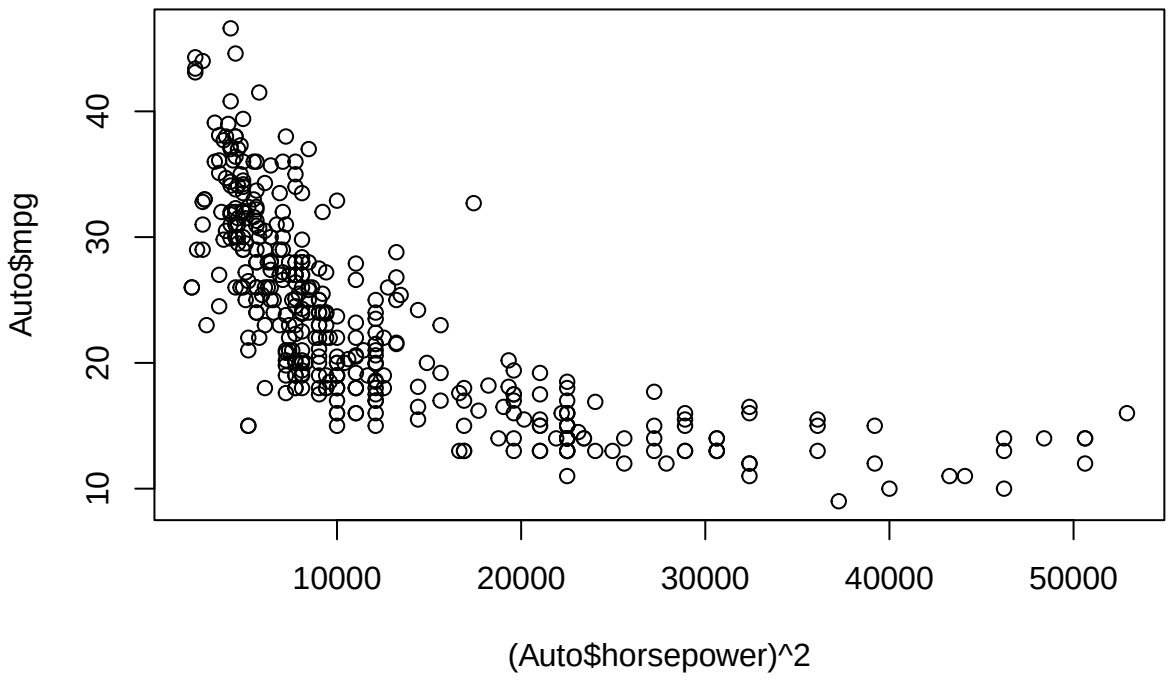

```
plot(log(Auto$horsepower), Auto$mpg)
```




```
plot(sqrt(Auto$horsepower), Auto$mpg)
```




```
plot((Auto$horsepower)^2, Auto$mpg)
```



Answer

After trying transformation on several continuous variables (weight, displacement, horsepower), it looks like the log transformation provides most linear looking at the plot.

Question 15

This problem involves the Boston data set, which we saw in the lab for this chapter. We will now try to predict per capita crime rate using the other variables in this data set. In other words, per capita crime rate is the response, and the other variables are the predictors.

```
#install.packages('MASS')
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

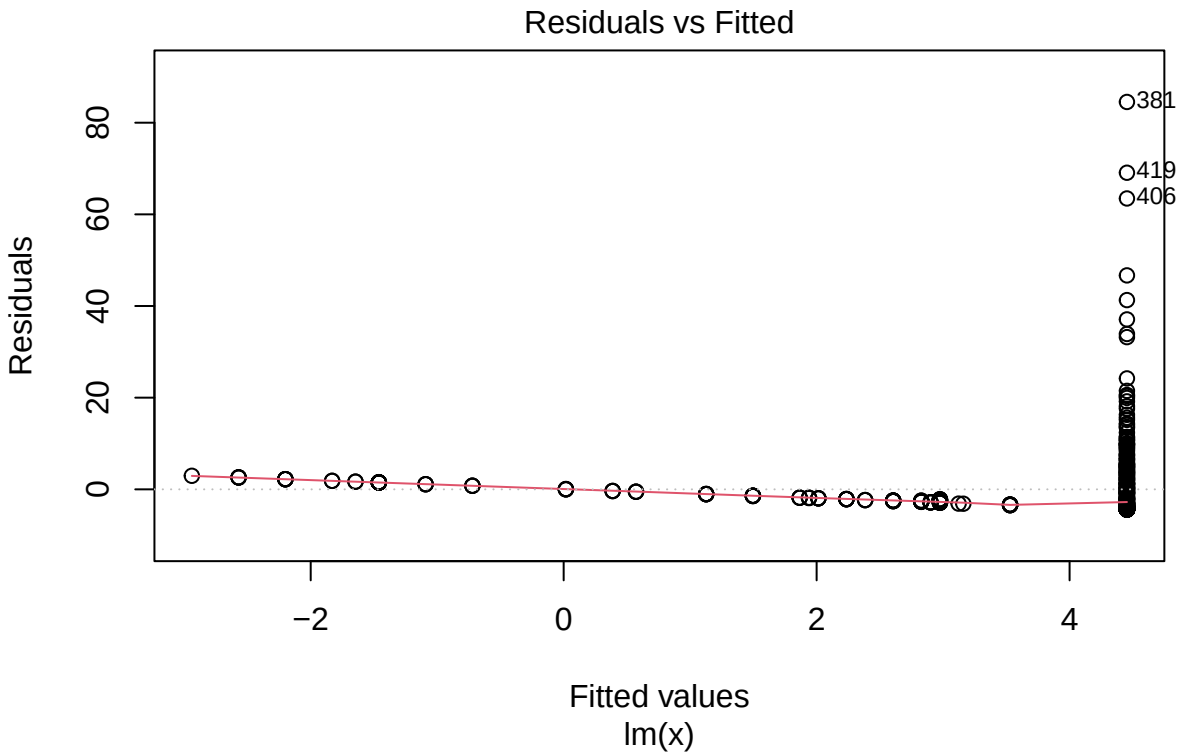
```
## The following object is masked from 'package:dplyr':
##
##     select
```

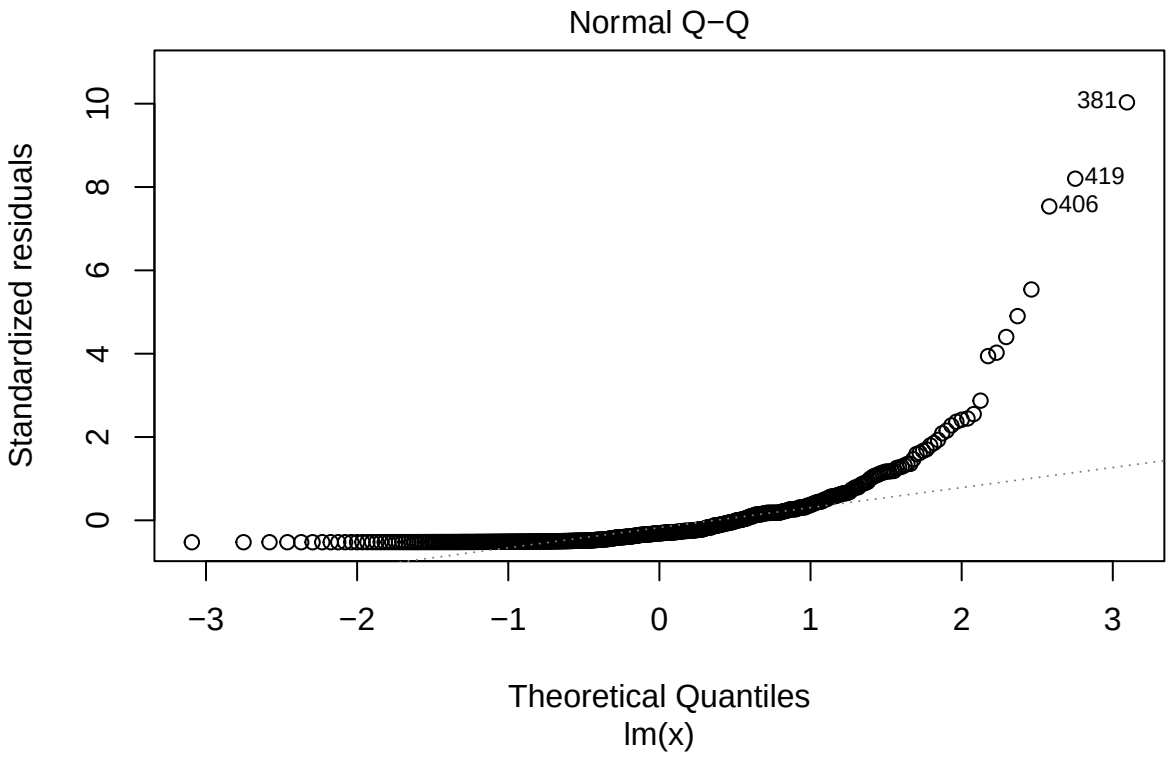
```
library(tidyverse)
boston <- Boston
```

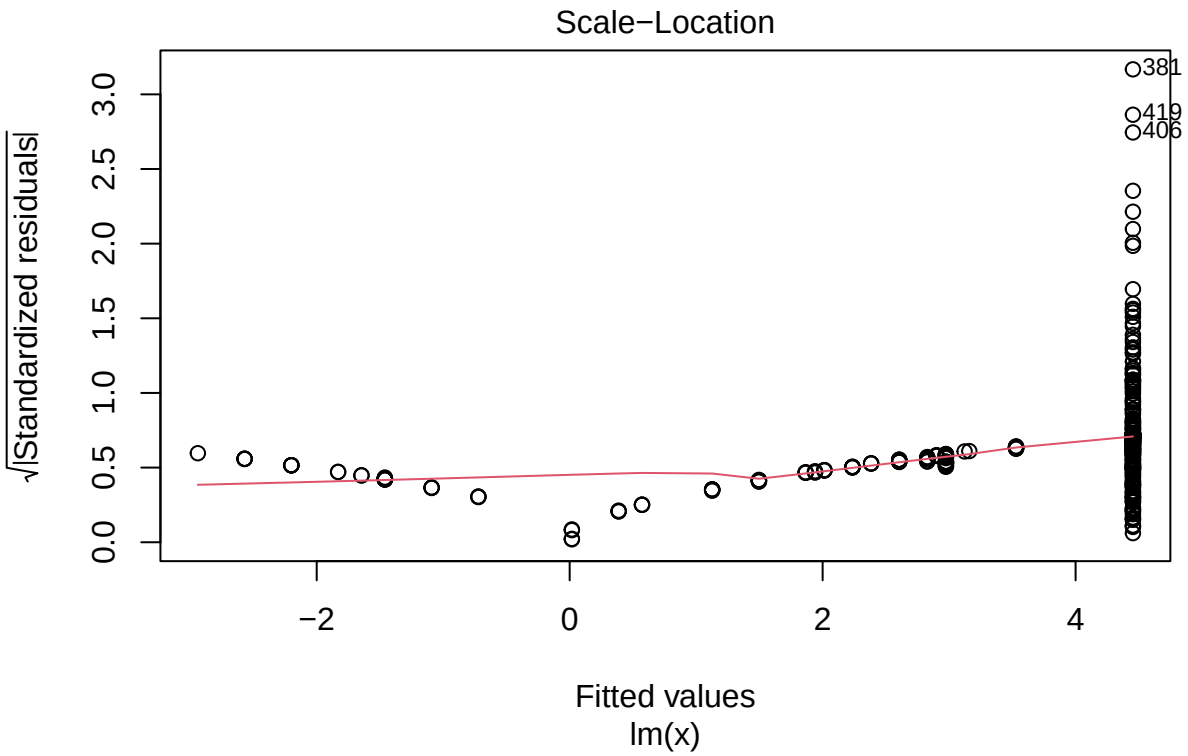
(a) For each predictor, fit a simple linear regression model to predict the response. Describe your results. In which of the models is there a statistically significant association between the predictor and the response? Create some plots to back up your assertions.

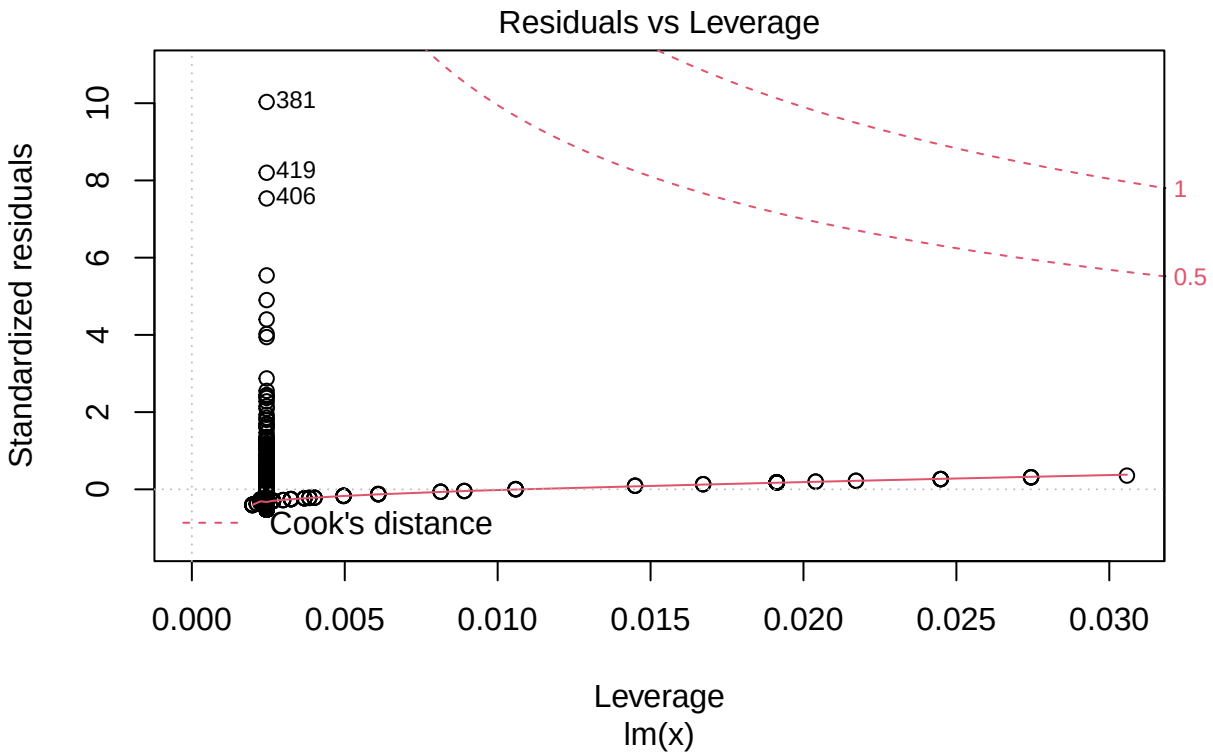
```
var_names <- boston %>% names()
var_names <- var_names[2:14]
allModelsList <- lapply(paste("crim ~", var_names), as.formula)
allModelsResults <- lapply(allModelsList, function(x) lm(x, data= boston))
for (n in 1:13) {
  print(summary(allModelsResults[[n]]))
  cat(paste('plots for ', var_names[n], '\n'))
  plot(allModelsResults[[n]])
}
```

```
##
## Call:
## lm(formula = x, 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
##
## plots for  zn
```

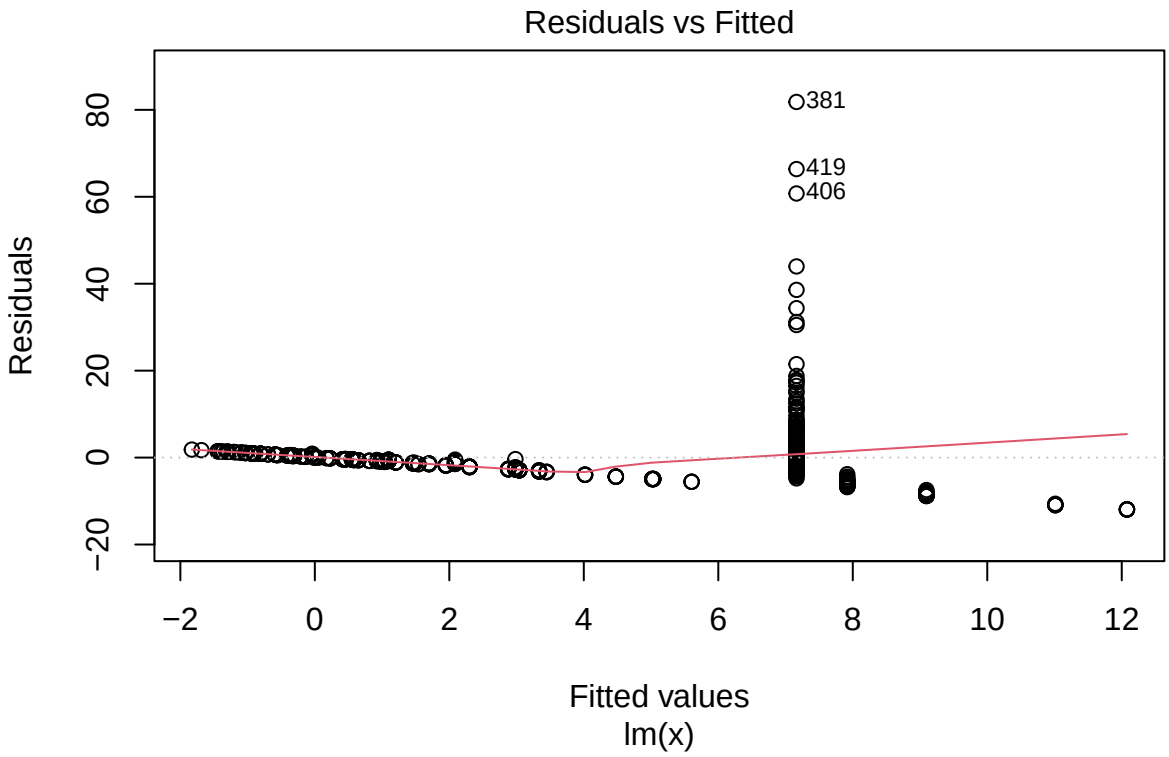

Residuals vs Fitted
Residuals
0
20
40
60
80
381
419
406
−2
0
2
4
Fitted values
lm(x)

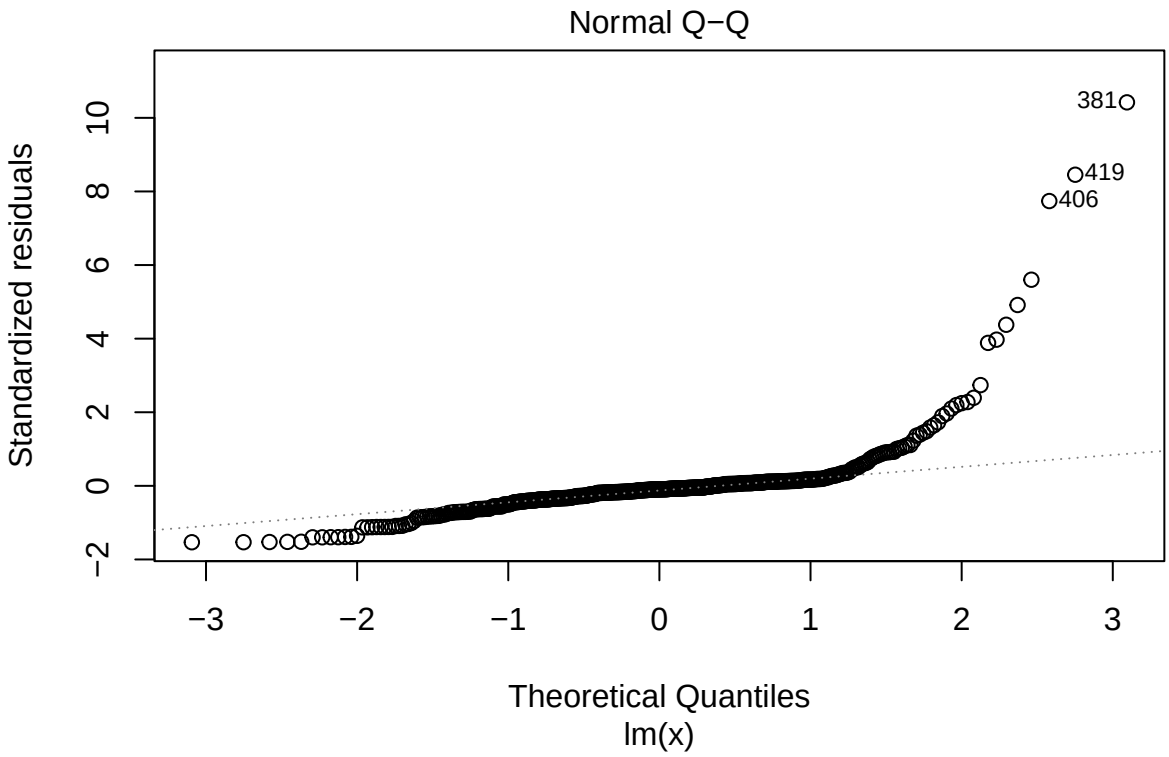

Normal Q−Q
Standardized residuals
0
2
4
6
8
10
381
419
406
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(x)

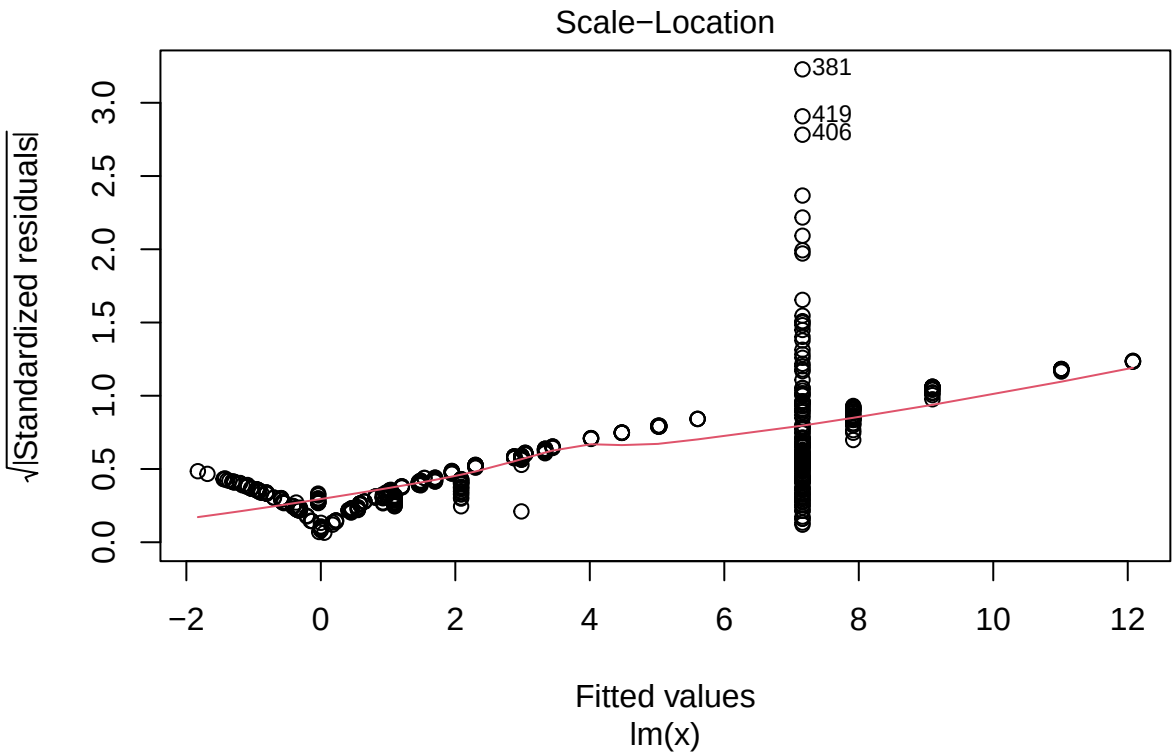

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
381
419
406
−2
0
2
4
Fitted values
lm(x)

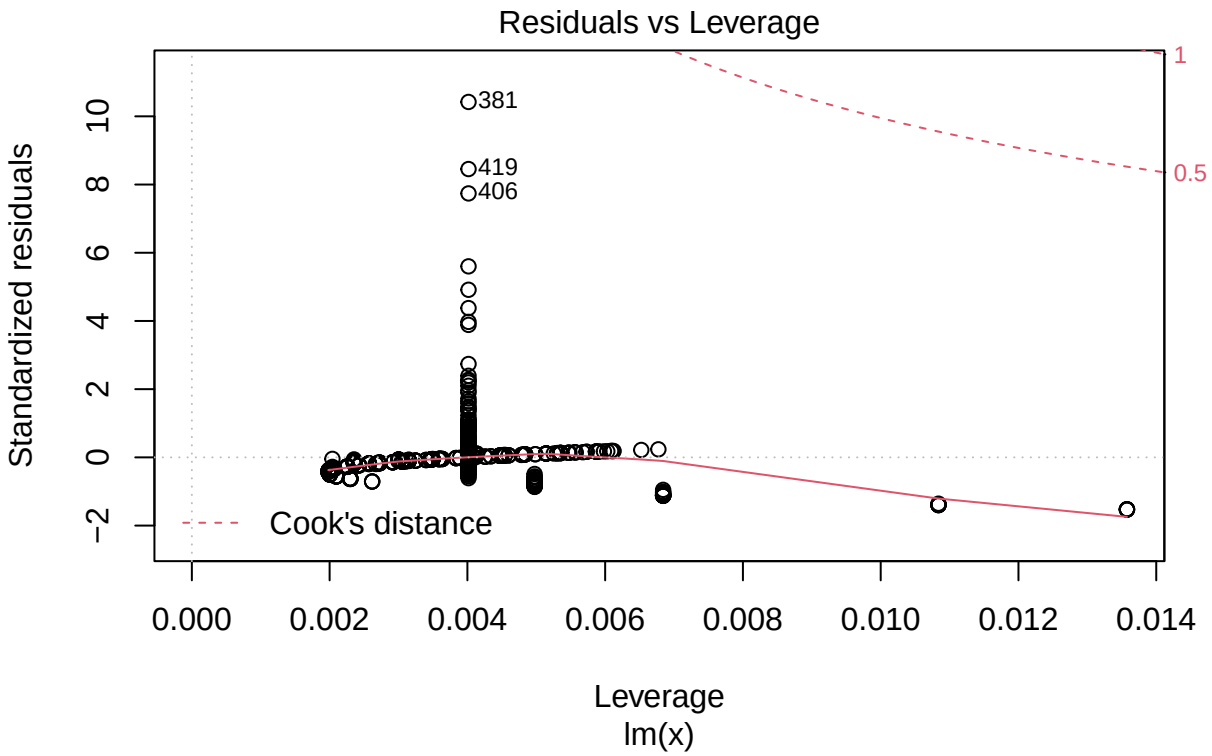



```
##
## Call:
## lm(formula = x, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.972  -2.698  -0.736   0.712  81.813
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.06374    0.66723  -3.093  0.00209 **
## indus        0.50978    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
##
## plots for  indus
```

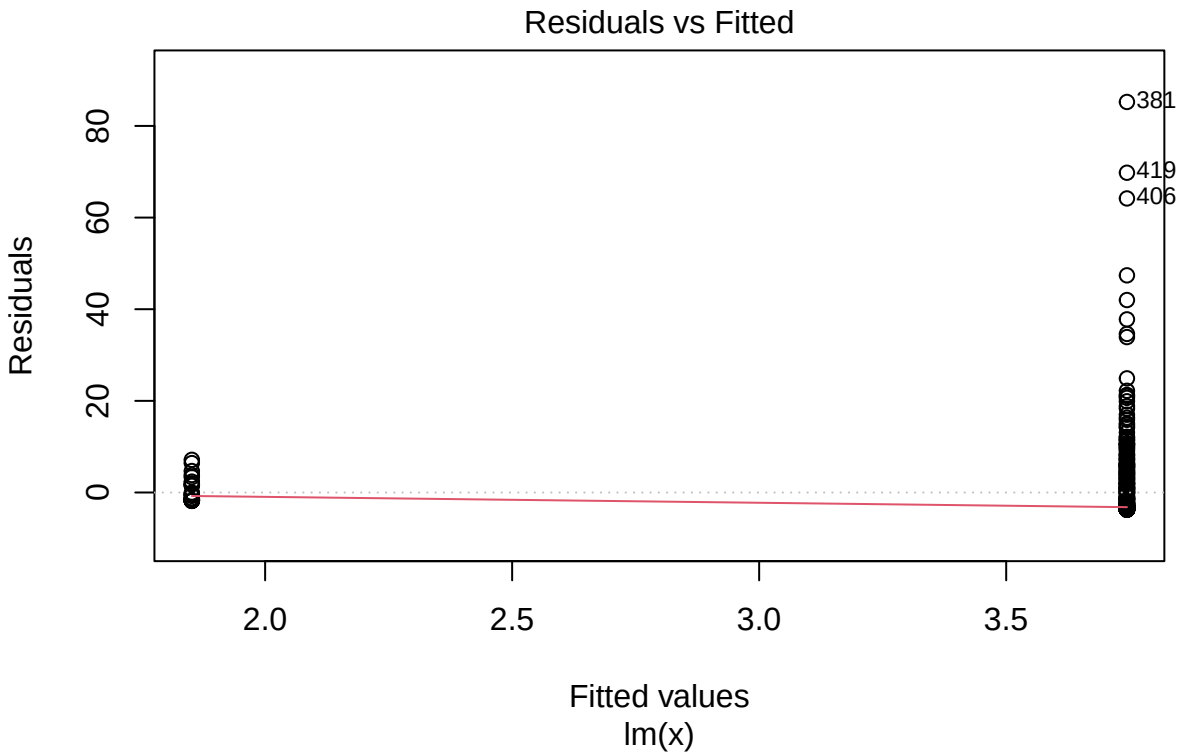

Residuals vs Fitted
Residuals
Fitted values
lm(x)
381
419
406
−20
0
20
40
60
80
−2
0
2
4
6
8
10
12

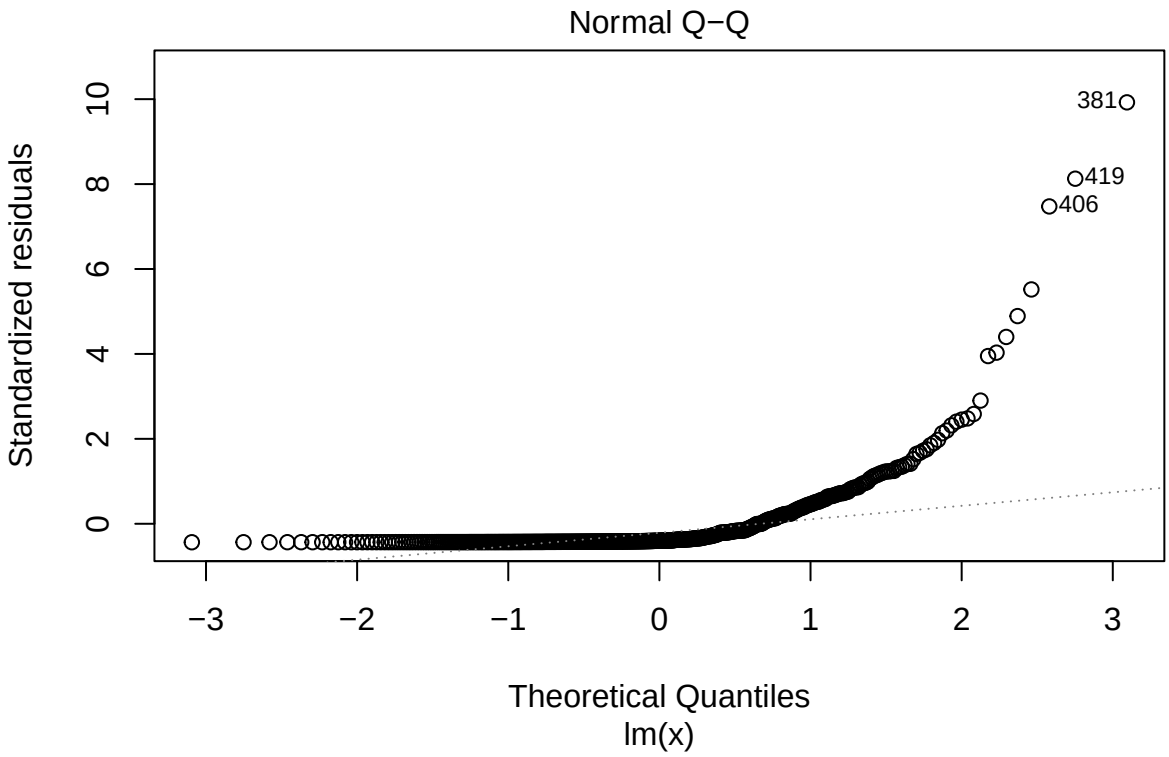

Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(x)
381
419
406
−2
0
2
4
6
8
10
−3
−2
−1
0
1
2
3

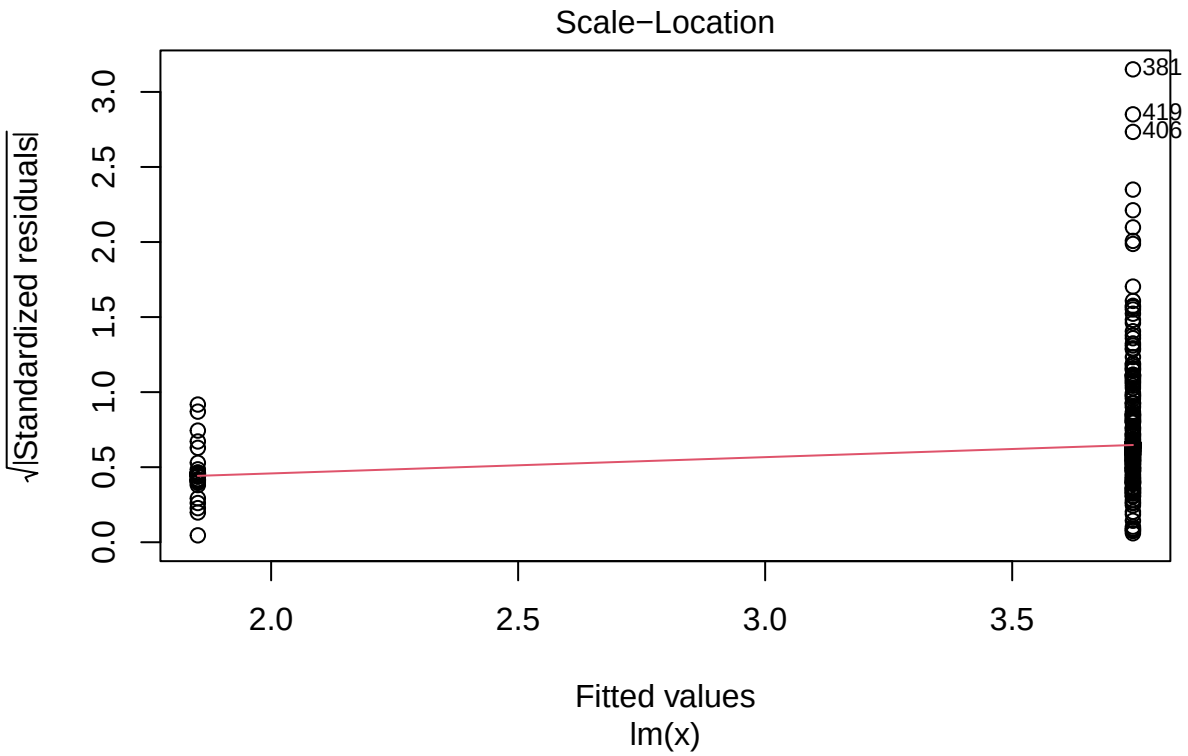

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
381
419
406
−2
0
2
4
6
8
10
12
Fitted values
lm(x)

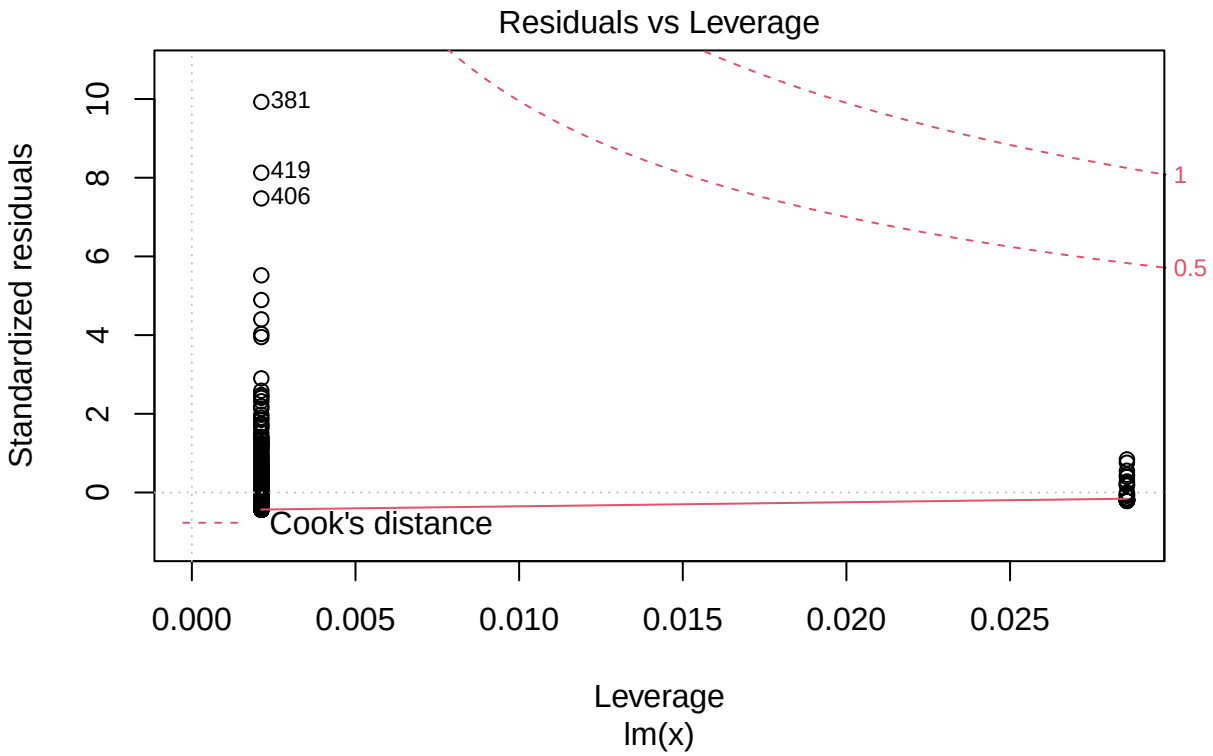



```
##
## Call:
## lm(formula = x, data = boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.738 -3.661 -3.435  0.018 85.232
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7444     0.3961   9.453   <2e-16 ***
## chas         -1.8928     1.5061  -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
##
## plots for  chas
```

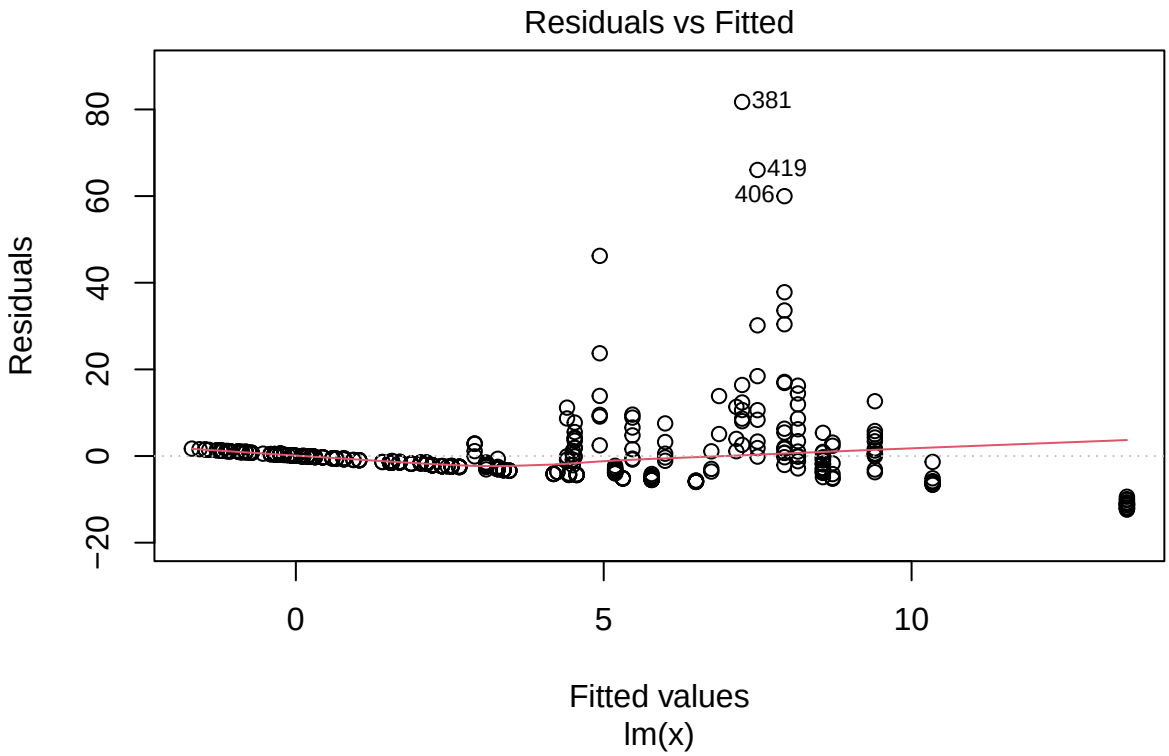

Residuals vs Fitted
Residuals
0
20
40
60
80
381
419
406
2.0
2.5
3.0
3.5
Fitted values
lm(x)

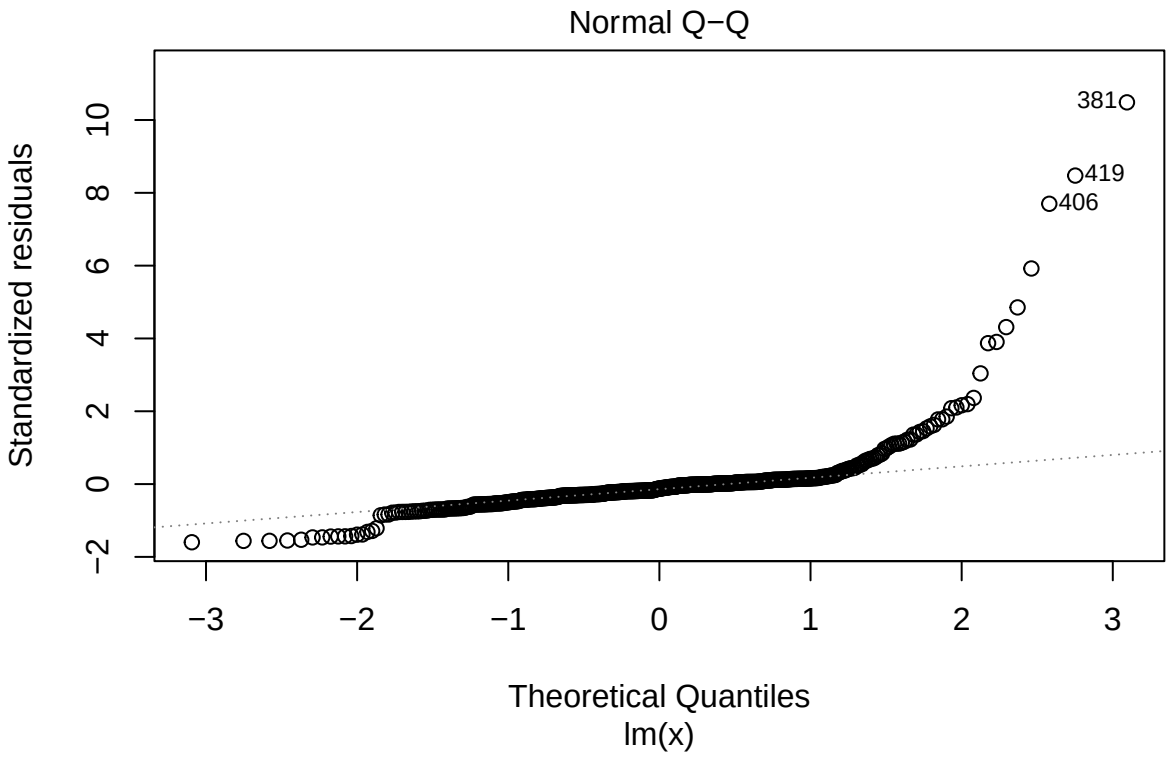

Normal Q−Q
Standardized residuals
0
2
4
6
8
10
381
419
406
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(x)

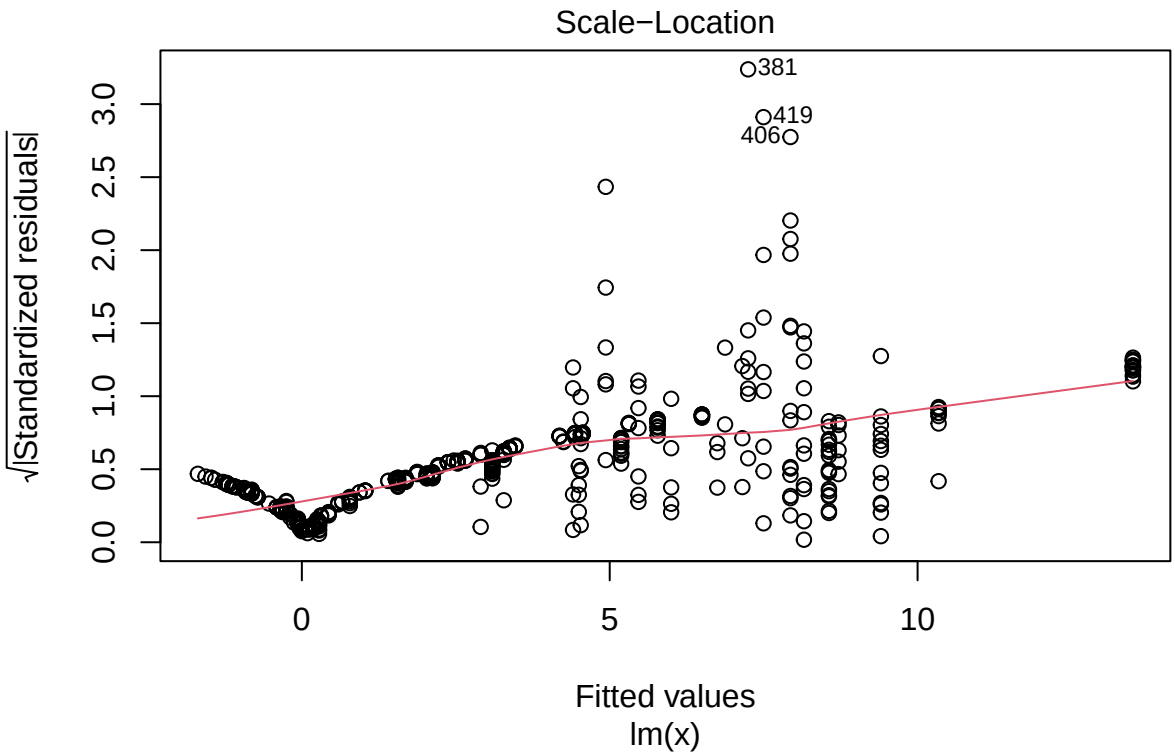

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
381
419
406
2.0
2.5
3.0
3.5
Fitted values
lm(x)

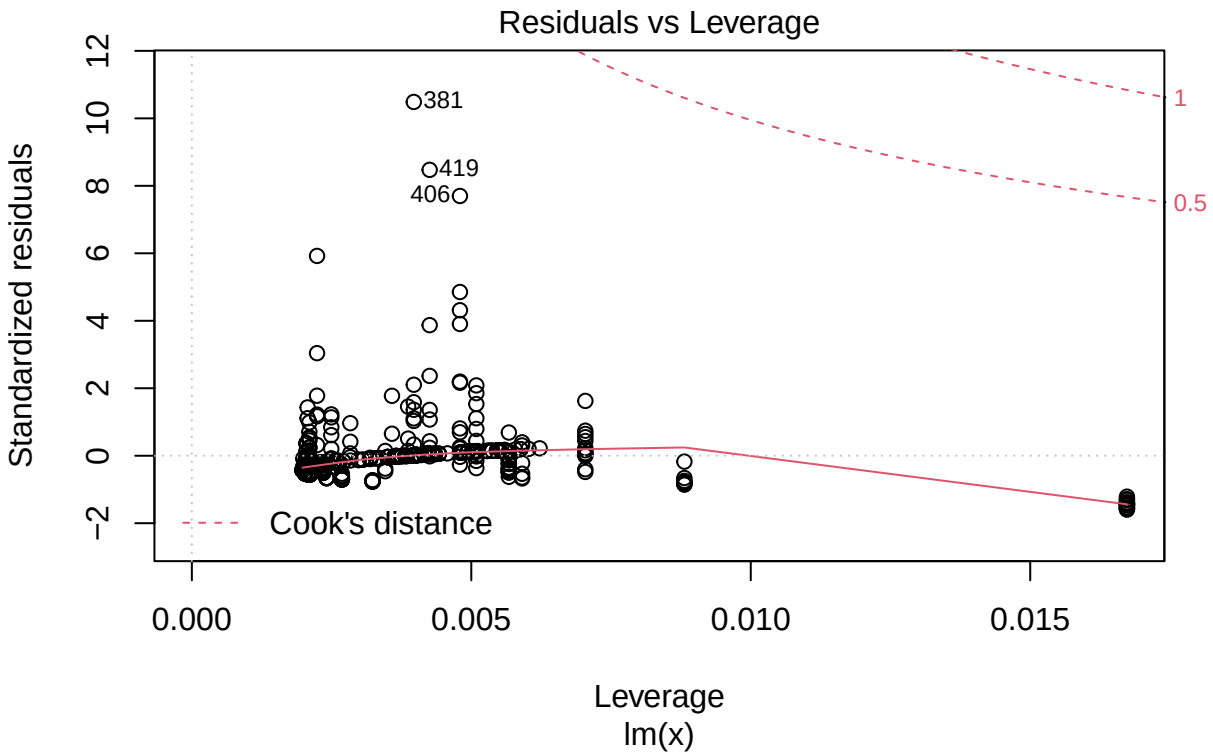



```
##
## Call:
## lm(formula = x, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.371  -2.738  -0.974   0.559  81.728
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -13.720      1.699  -8.073 5.08e-15 ***
## nox           31.249      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
##
## plots for  nox
```

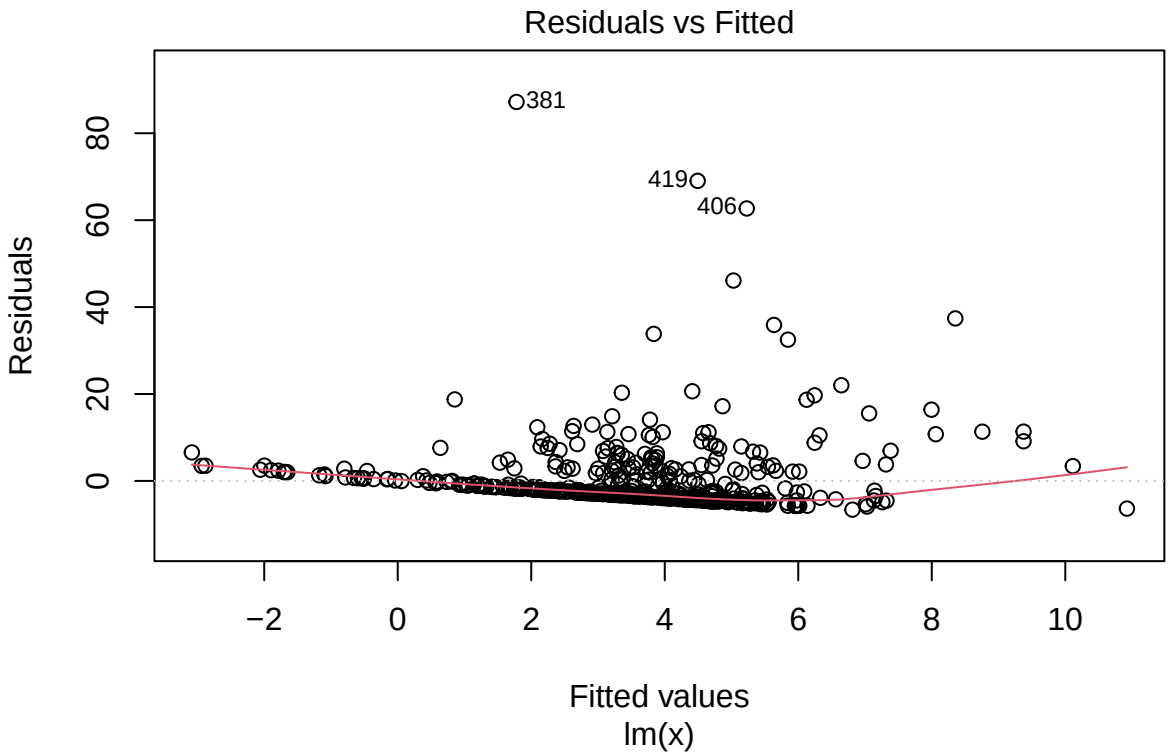

Residuals vs Fitted
Residuals
Fitted values
lm(x)
381
419
406
−20
0
20
40
60
80
0
5
10

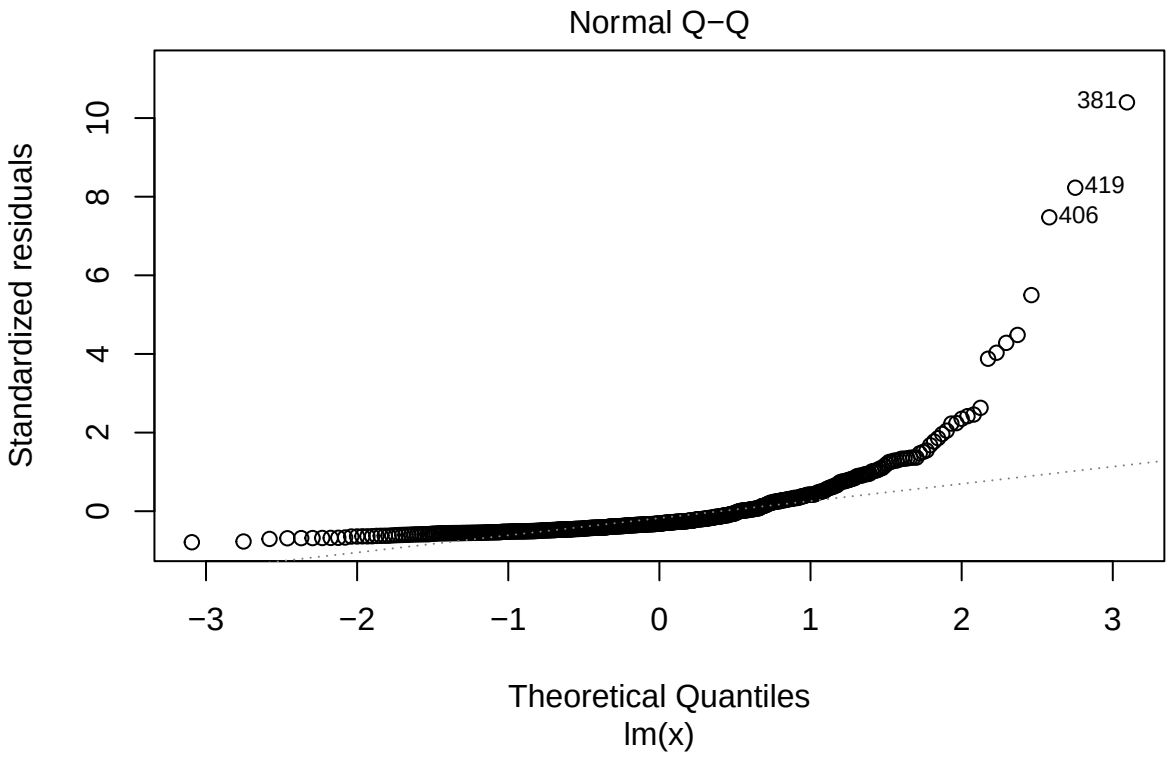

Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(x)
381
419
406
−2
0
2
4
6
8
10
−3
−2
−1
0
1
2
3

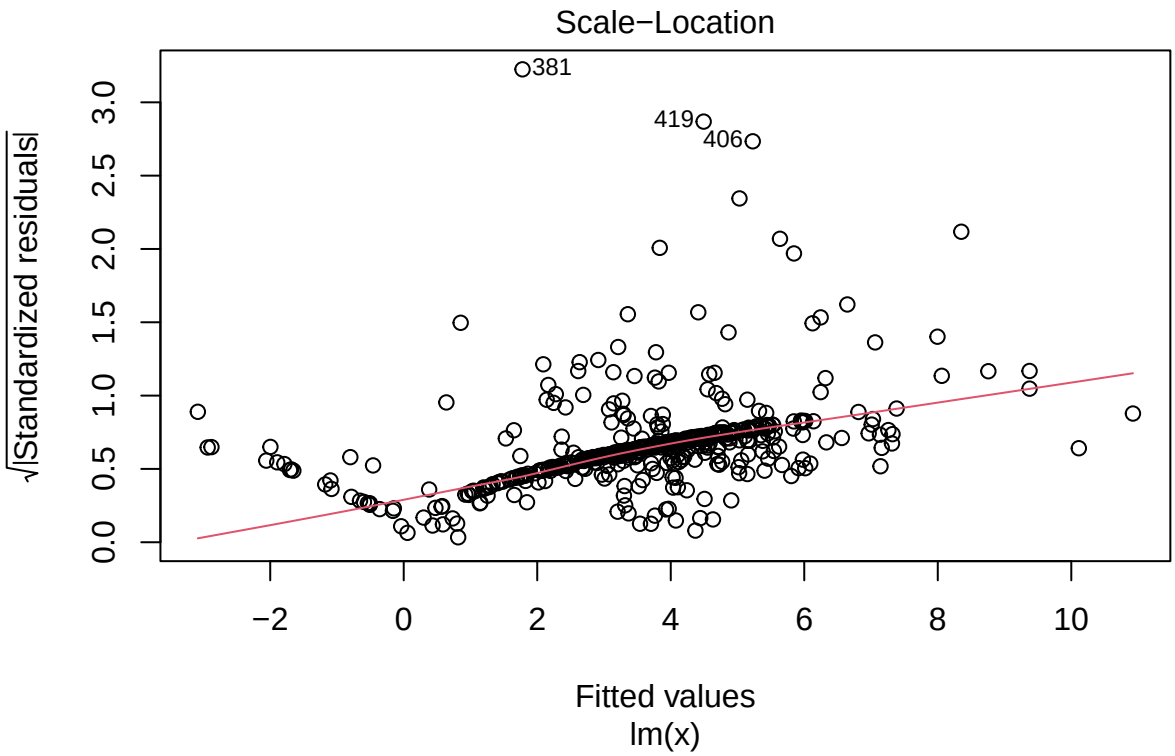

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
381
419
406
0
5
10
Fitted values
lm(x)

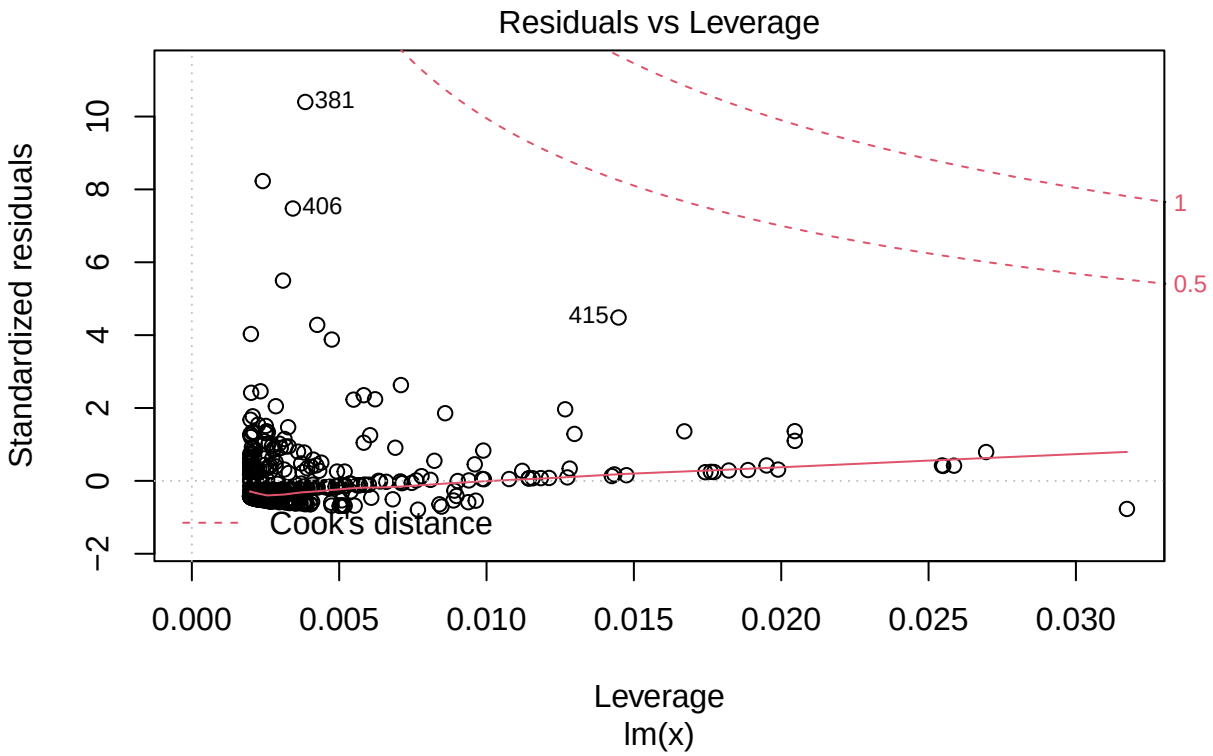



```
##
## Call:
## lm(formula = x, data = boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.604 -3.952 -2.654  0.989 87.197
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.482      3.365   6.088 2.27e-09 ***
## rm            -2.684      0.532  -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
##
## plots for  rm
```

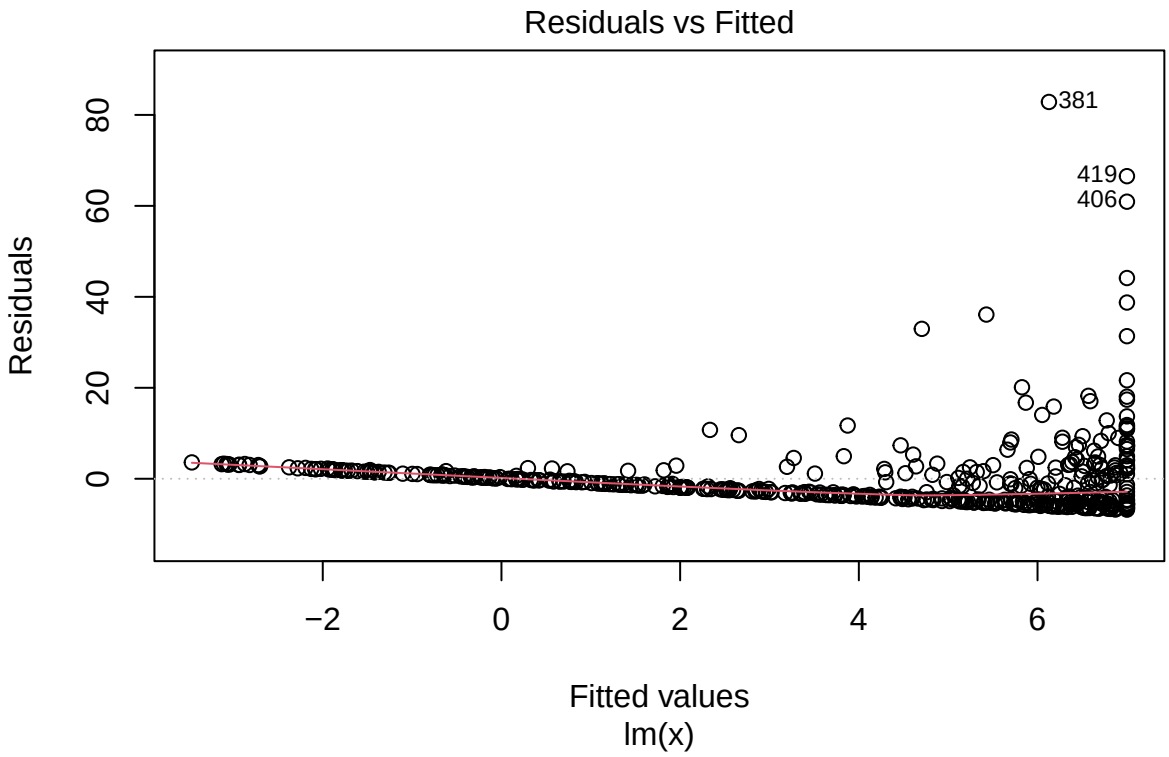

Residuals vs Fitted
381
419
406
Residuals
0
20
40
60
80
−2
0
2
4
6
8
10
Fitted values
lm(x)

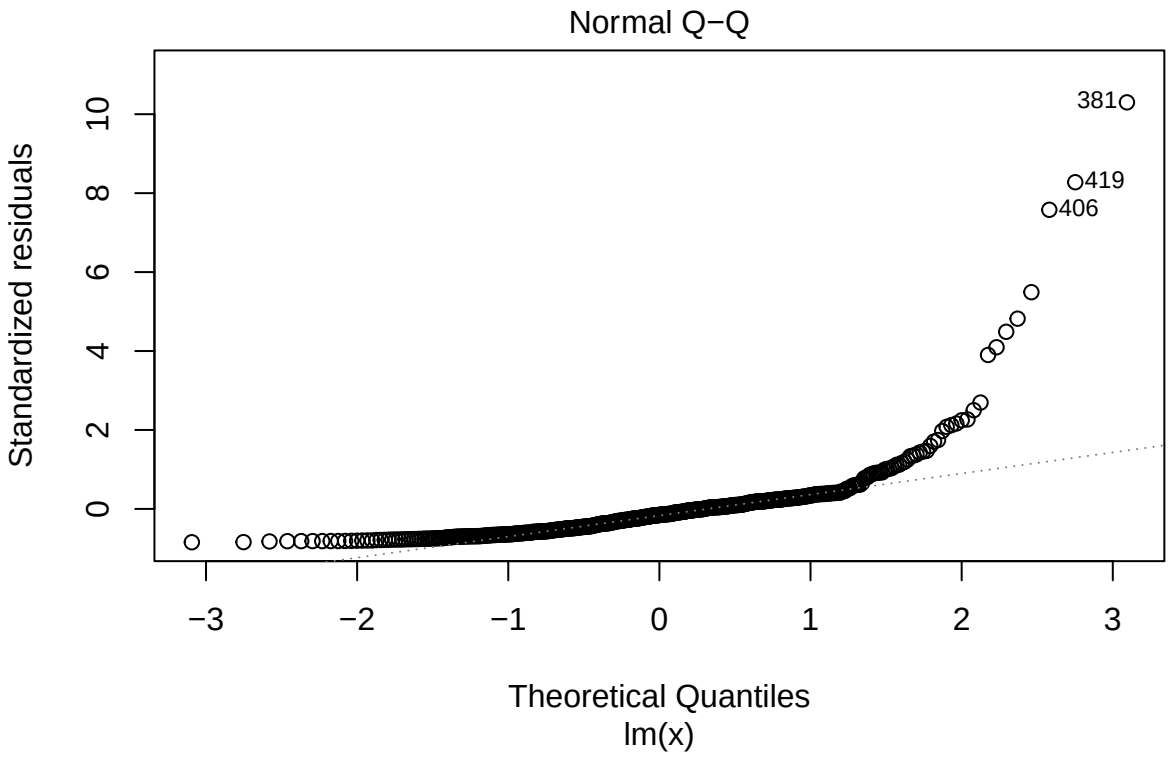

Normal Q−Q
Standardized residuals
0
2
4
6
8
10
381
419
406
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(x)

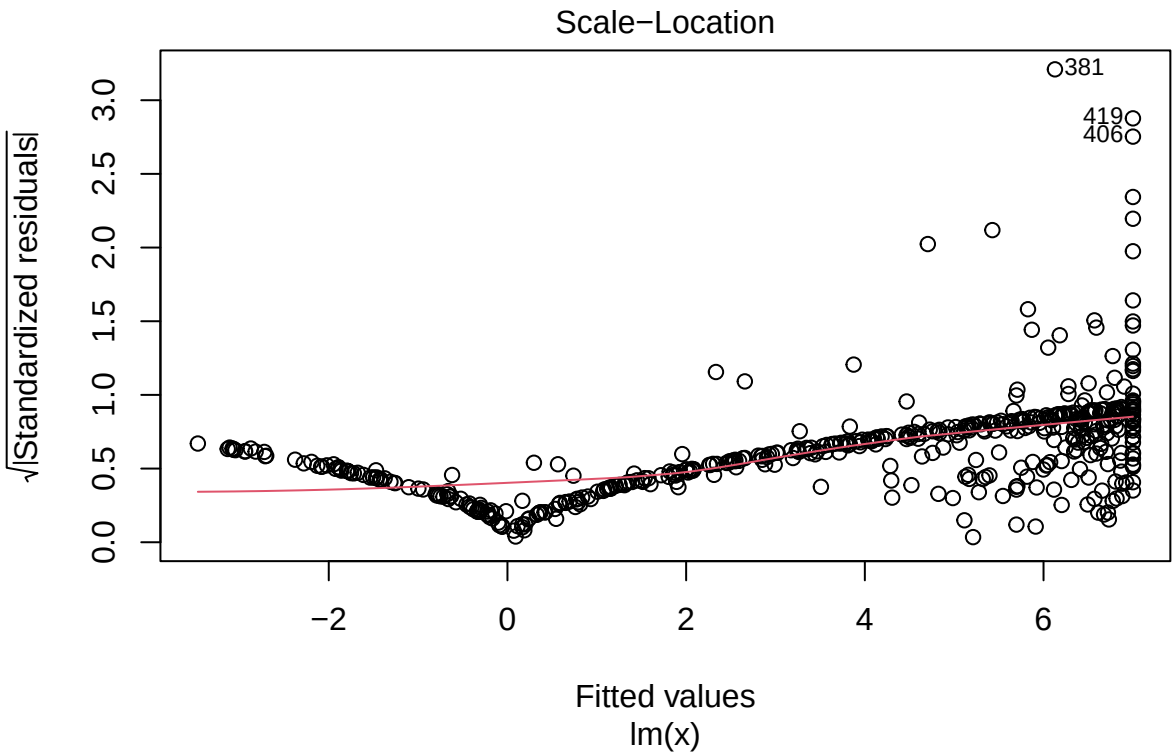

Scale−Location
381
419
406
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
−2
0
2
4
6
8
10
Fitted values
lm(x)

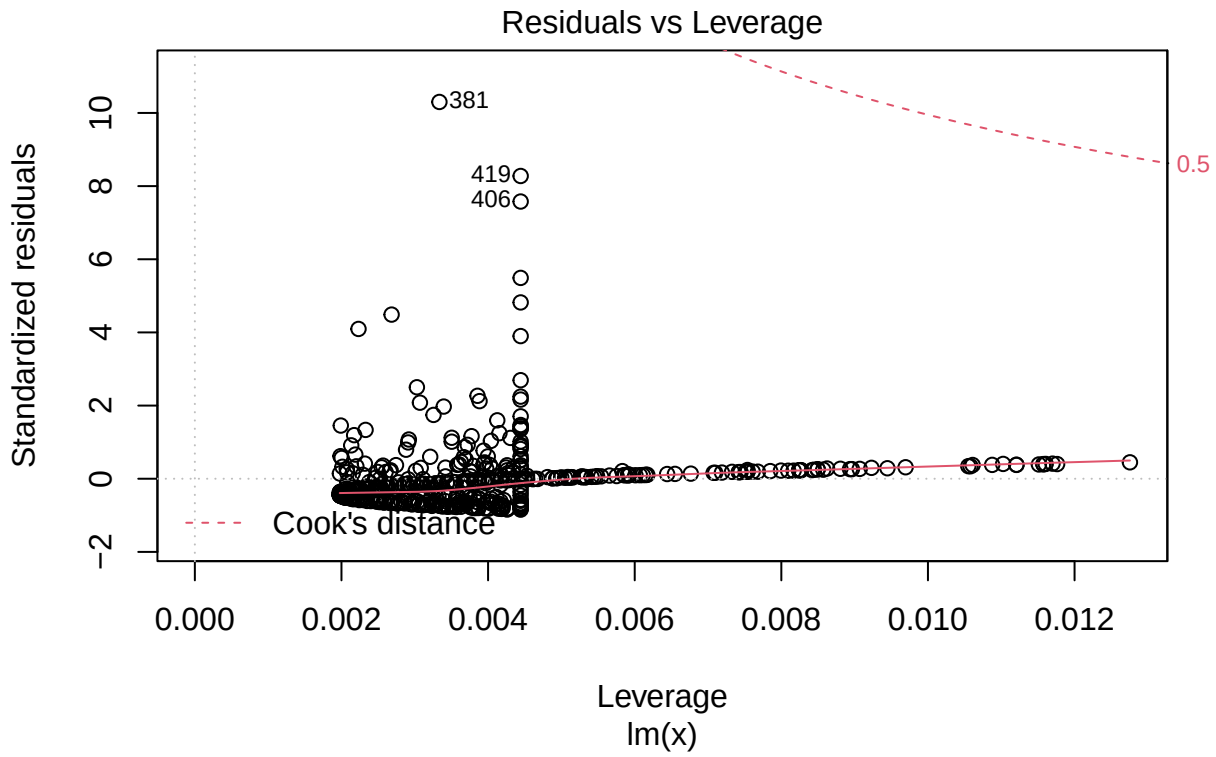



```
##
## Call:
## lm(formula = x, data = boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.789 -4.257 -1.230  1.527 82.849
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.77791    0.94398  -4.002 7.22e-05 ***
## age          0.10779    0.01274   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
##
## plots for  age
```

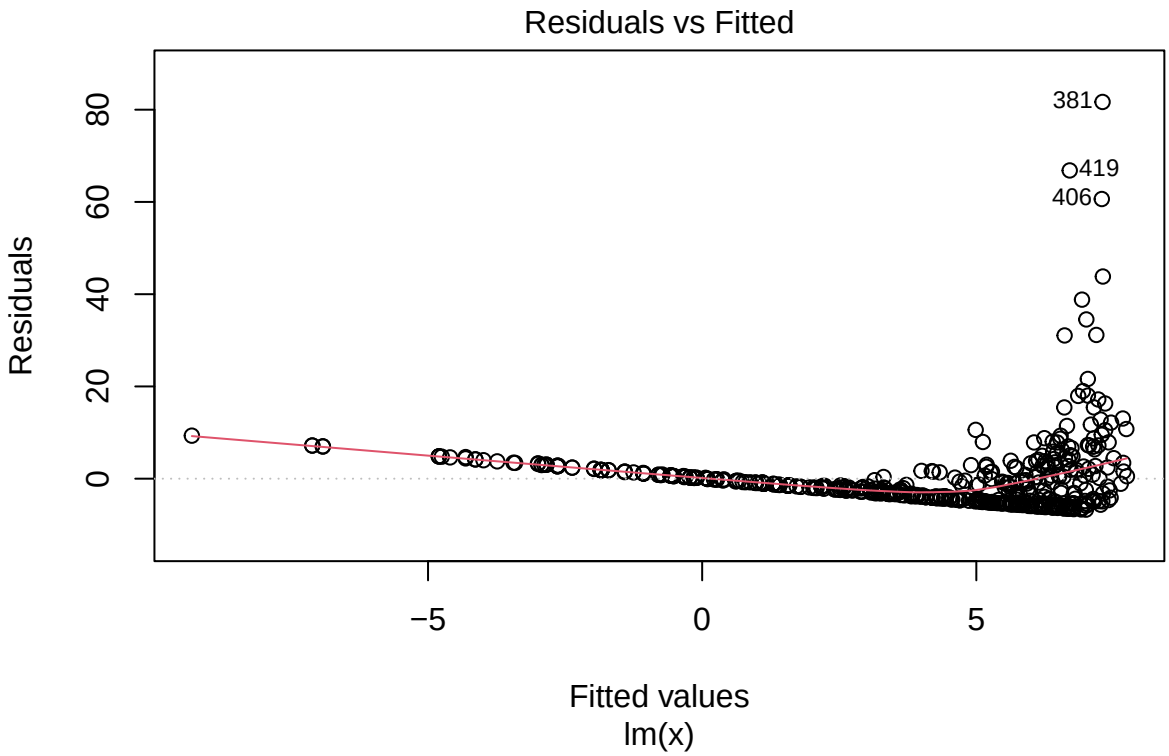

Residuals vs Fitted
Residuals
0
20
40
60
80
381
419
406
−2
0
2
4
6
Fitted values
lm(x)

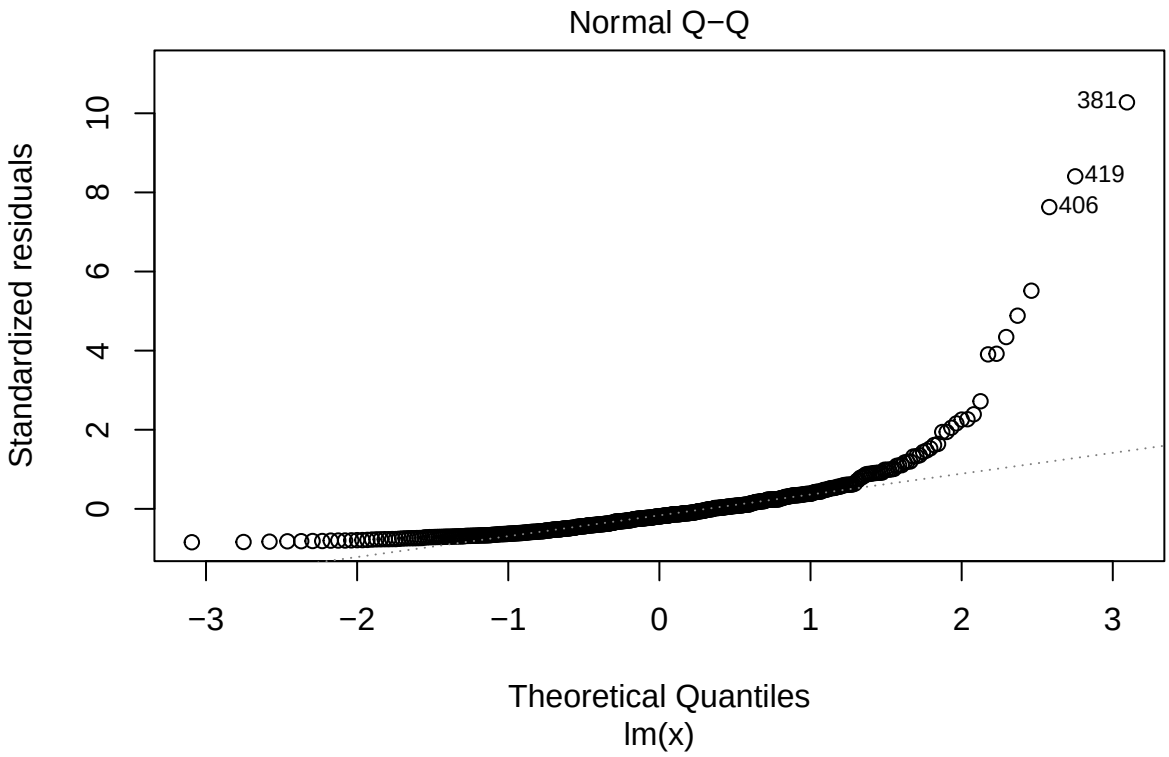

Normal Q−Q
Standardized residuals
0
2
4
6
8
10
381
419
406
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(x)

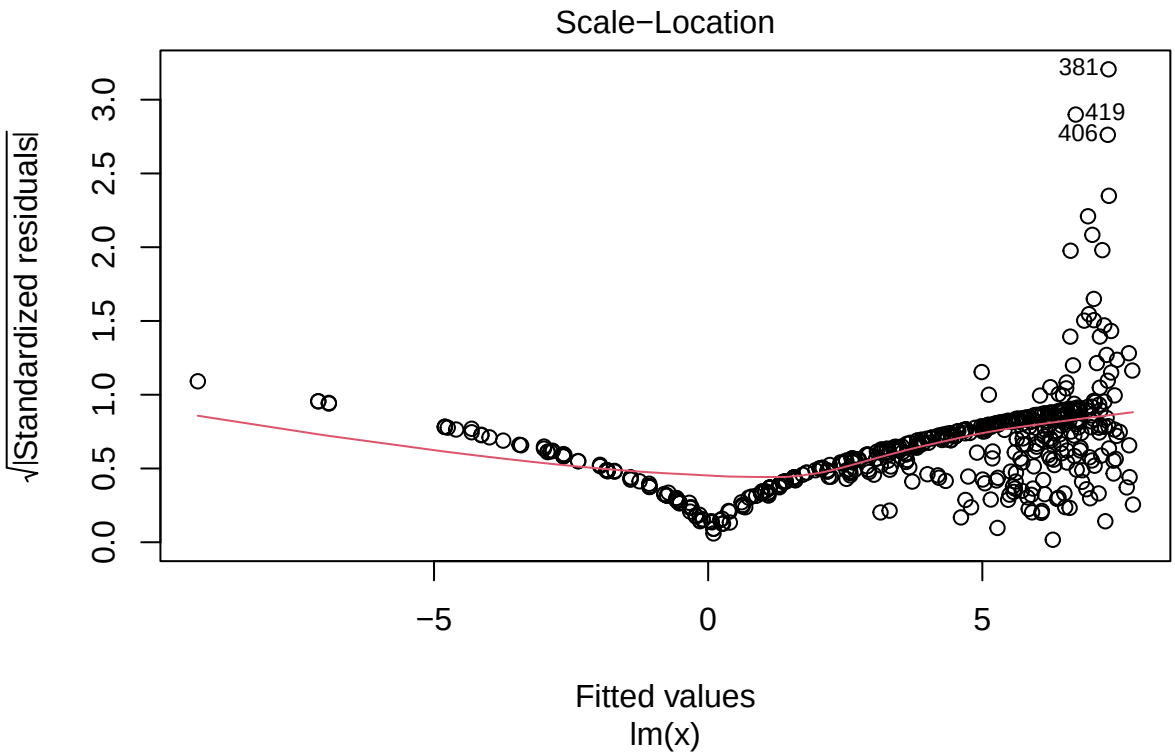

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
381
419
406
−2
0
2
4
6
Fitted values
lm(x)

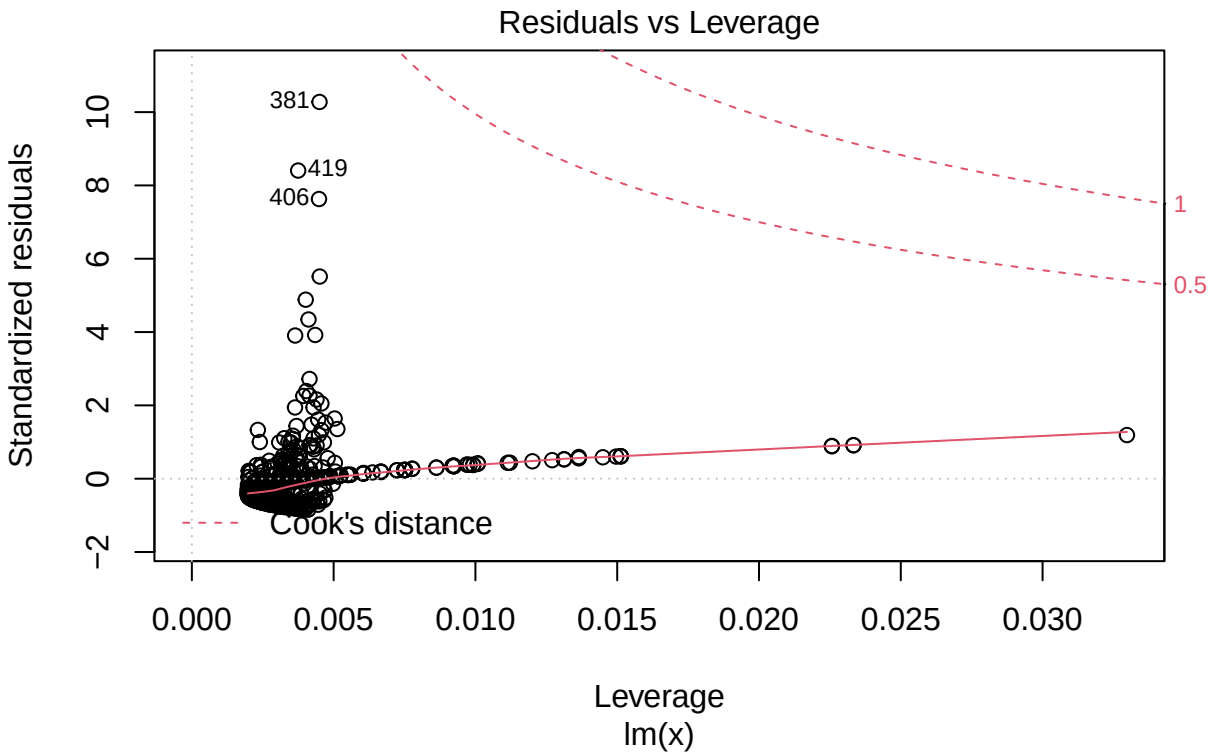



```
##
## Call:
## lm(formula = x, data = boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.708 -4.134 -1.527  1.516 81.674
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.4993     0.7304  13.006   <2e-16 ***
## dis          -1.5509     0.1683  -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
##
## plots for  dis
```

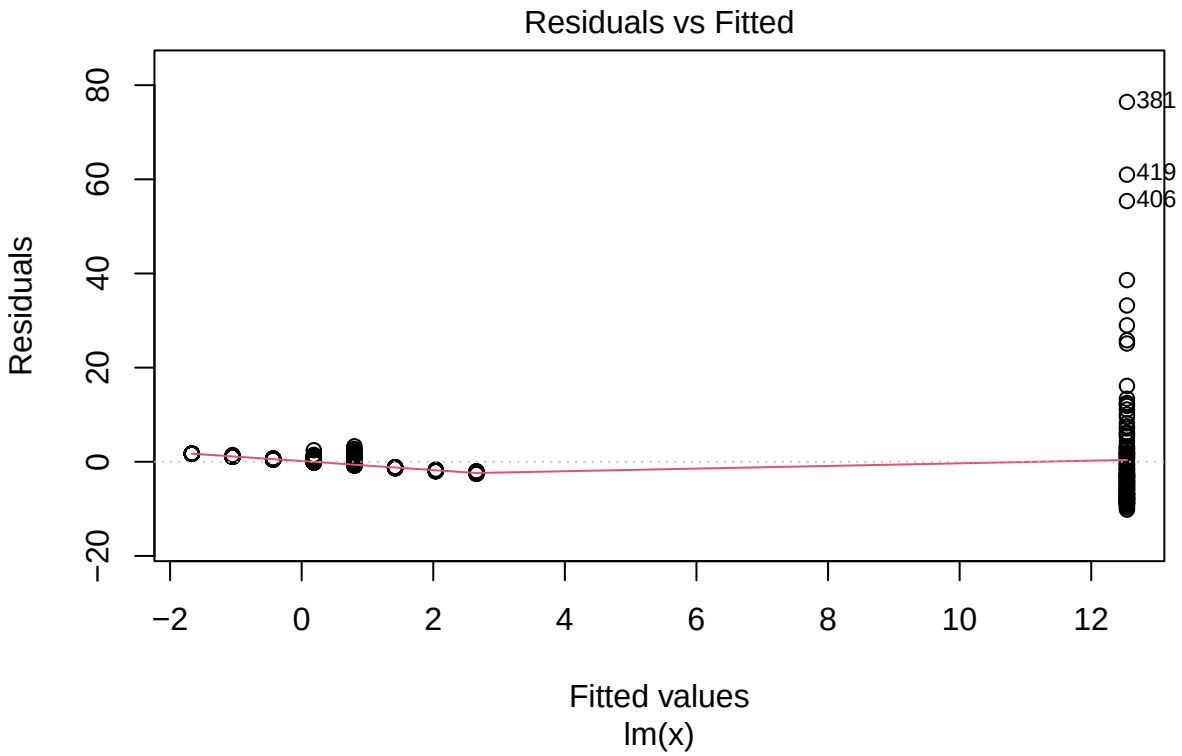

Residuals vs Fitted
Residuals
0
20
40
60
80
381
419
406
−5
0
5
Fitted values
lm(x)

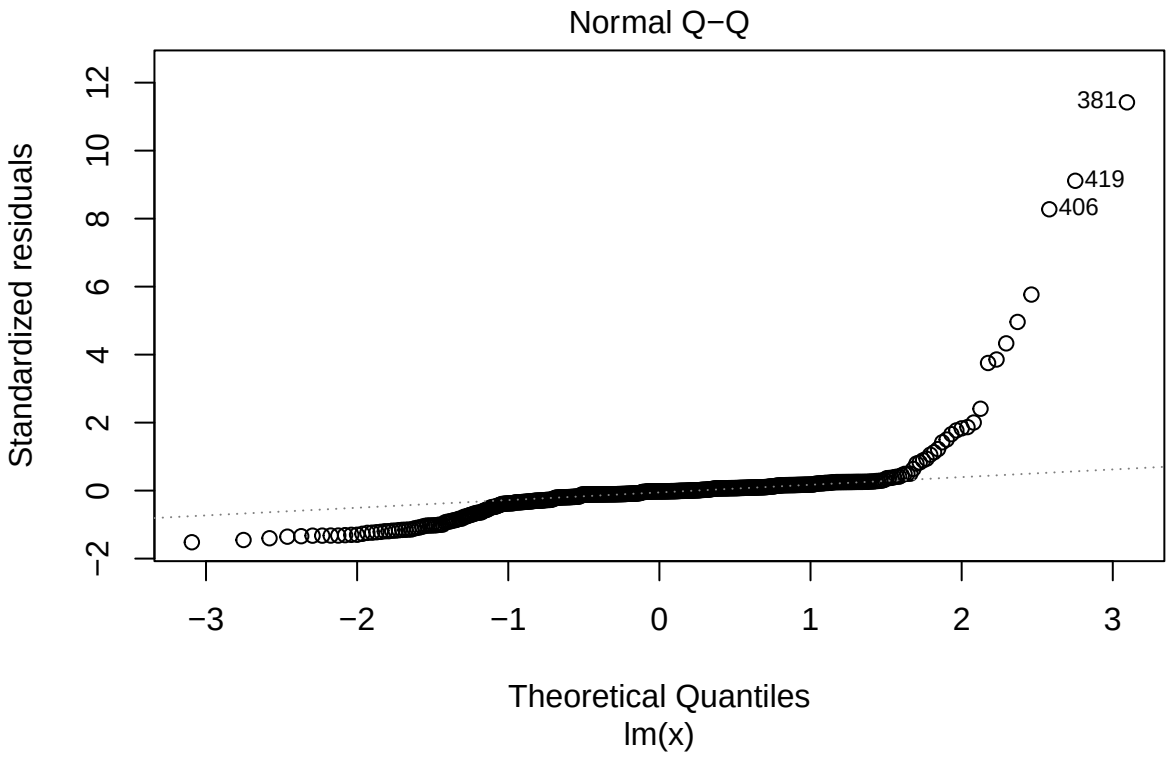

Normal Q−Q
Standardized residuals
0
2
4
6
8
10
381
419
406
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(x)

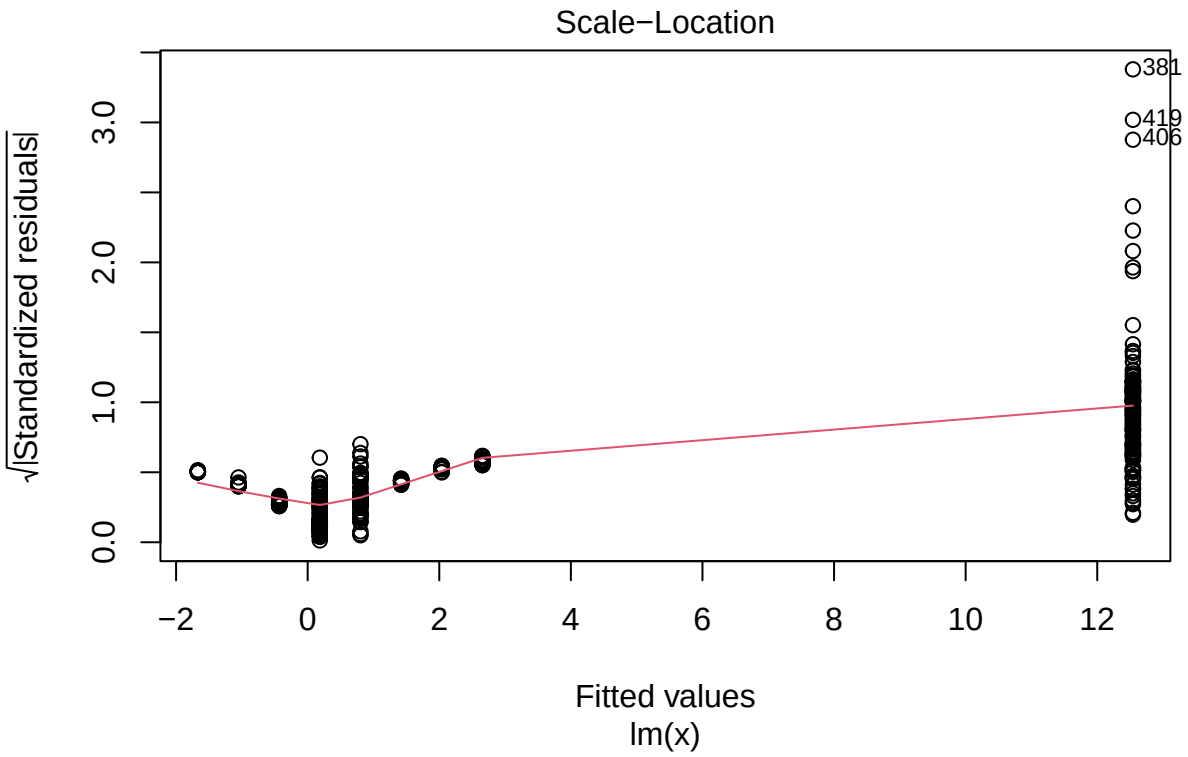

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
381
419
406
−5
0
5
Fitted values
lm(x)

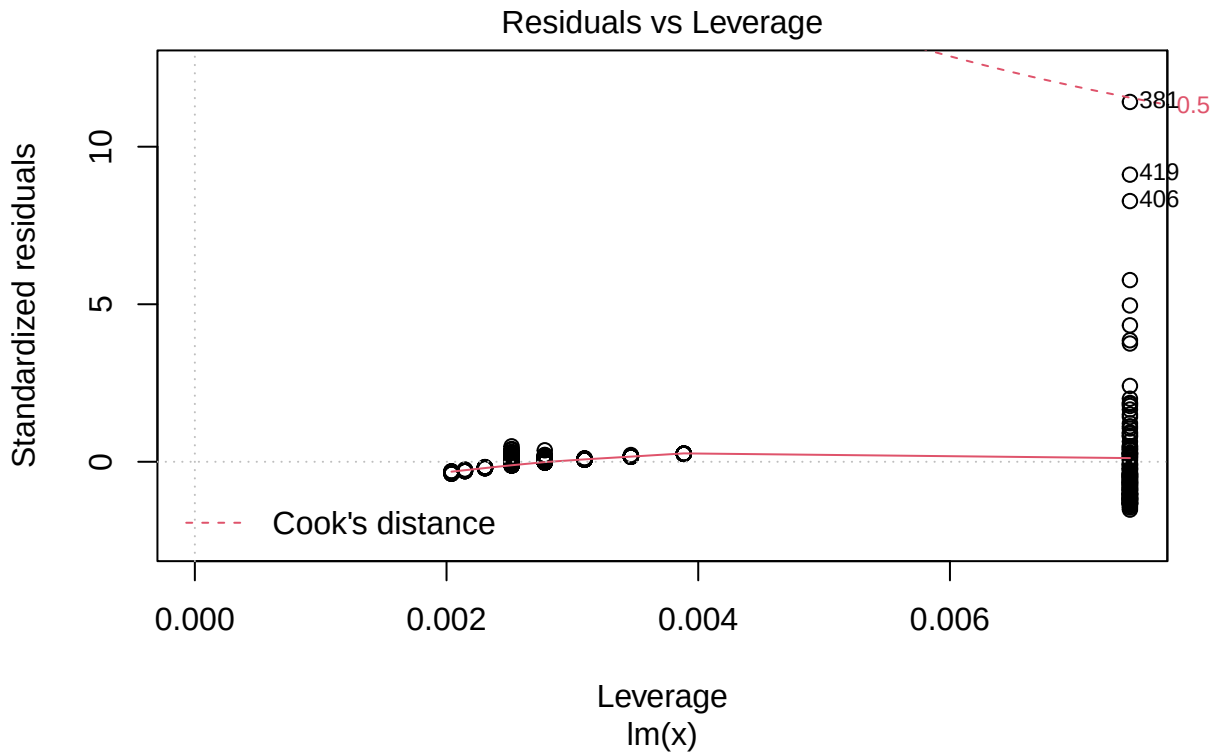



```
##
## Call:
## lm(formula = x, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.164  -1.381  -0.141   0.660  76.433
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.28716    0.44348  -5.157 3.61e-07 ***
## rad          0.61791    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
##
## plots for  rad
```

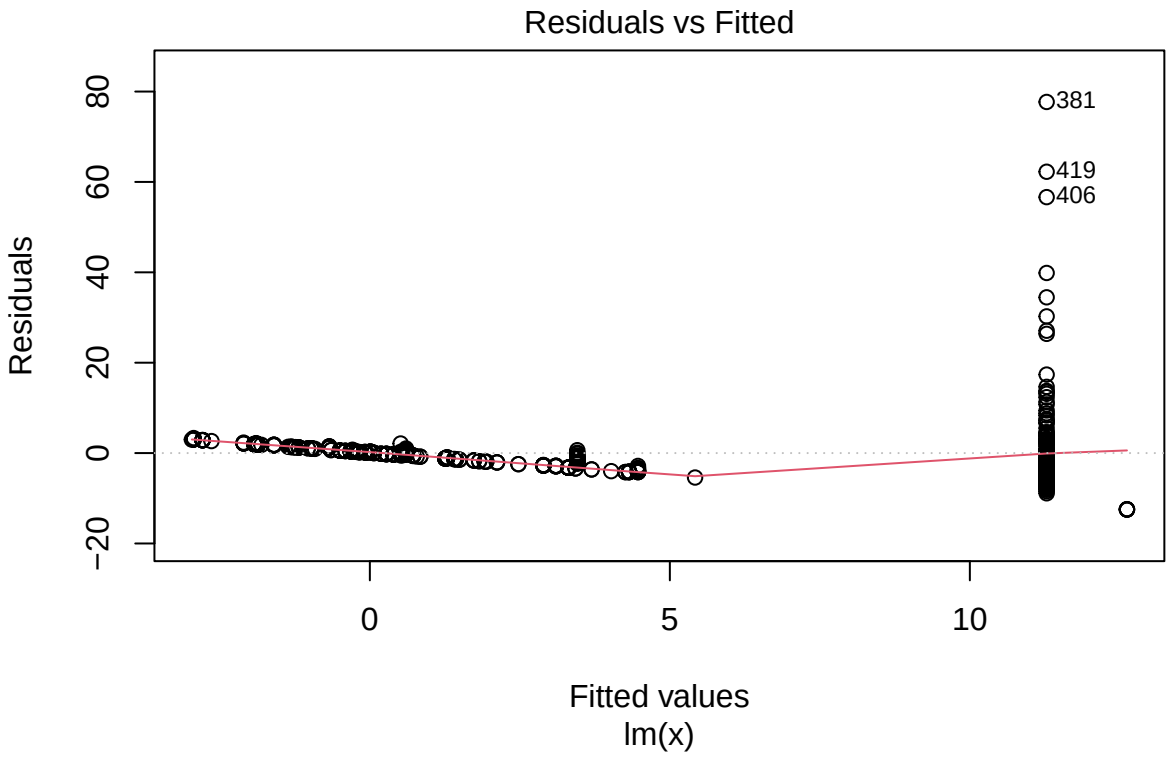

Residuals vs Fitted
Residuals
Fitted values
lm(x)
381
419
406
−20
0
20
40
60
80
−2
0
2
4
6
8
10
12

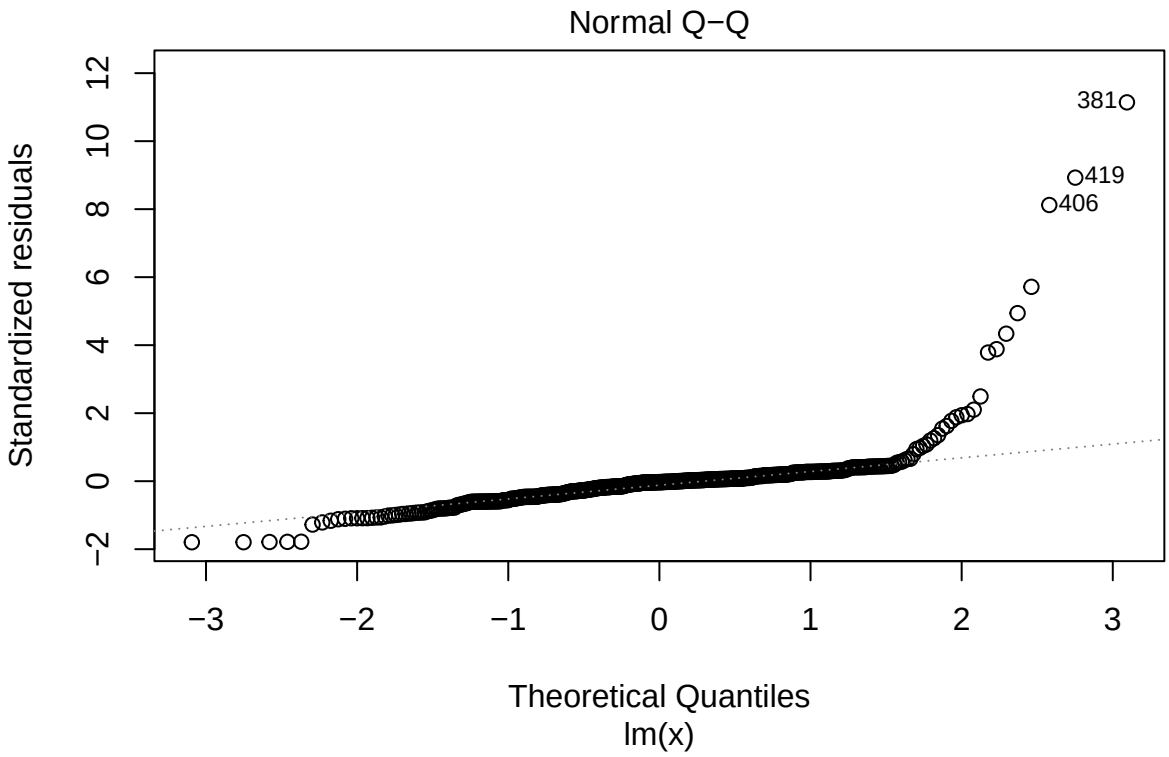

Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(x)
381
419
406
−2
0
2
4
6
8
10
12
−3
−2
−1
0
1
2
3

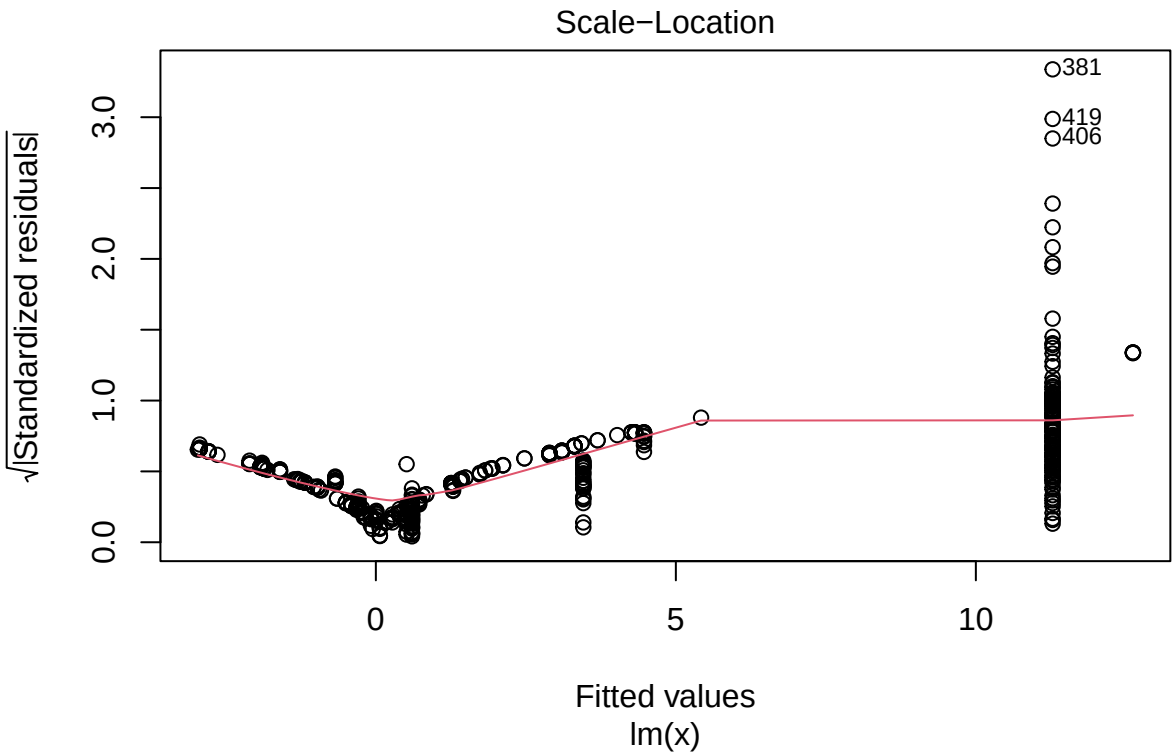

Scale−Location
√|Standardized residuals|
0.0
1.0
2.0
3.0
−2
0
2
4
6
8
10
12
381
419
406
Fitted values
lm(x)

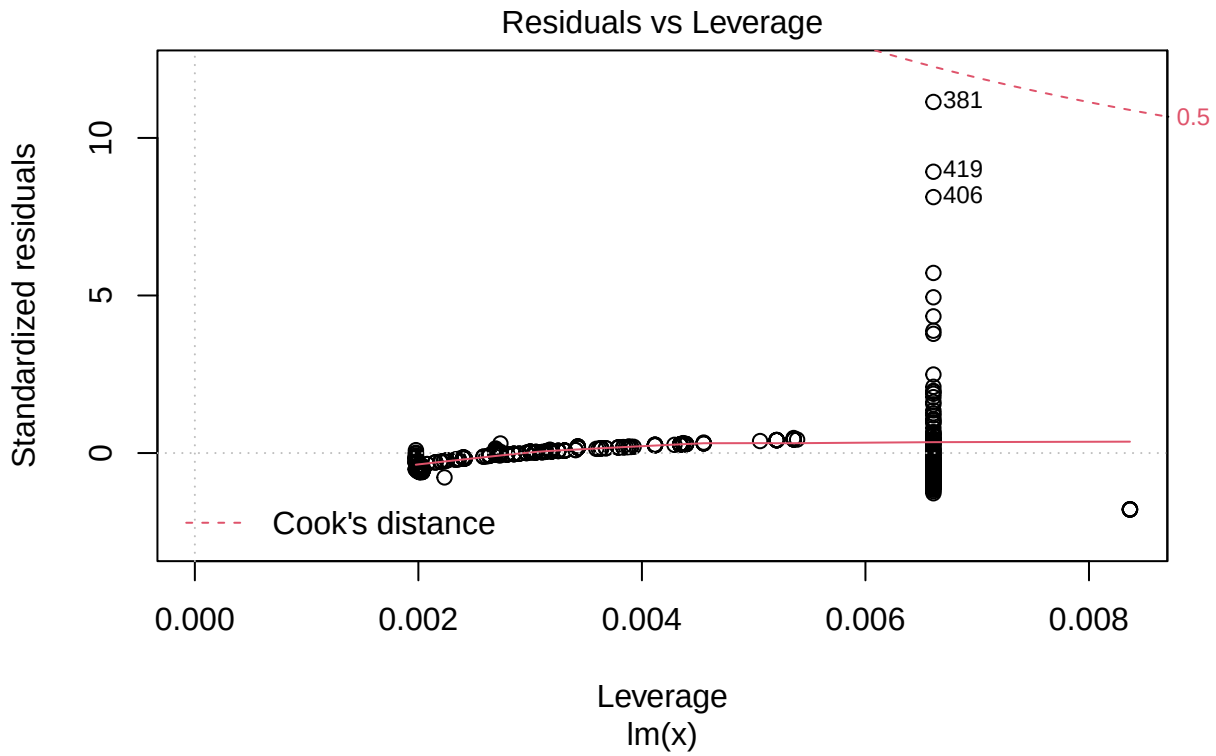



```
##
## Call:
## lm(formula = x, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.513  -2.738  -0.194   1.065  77.696
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.528369   0.815809  -10.45   <2e-16 ***
## tax          0.029742   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
##
## plots for  tax
```

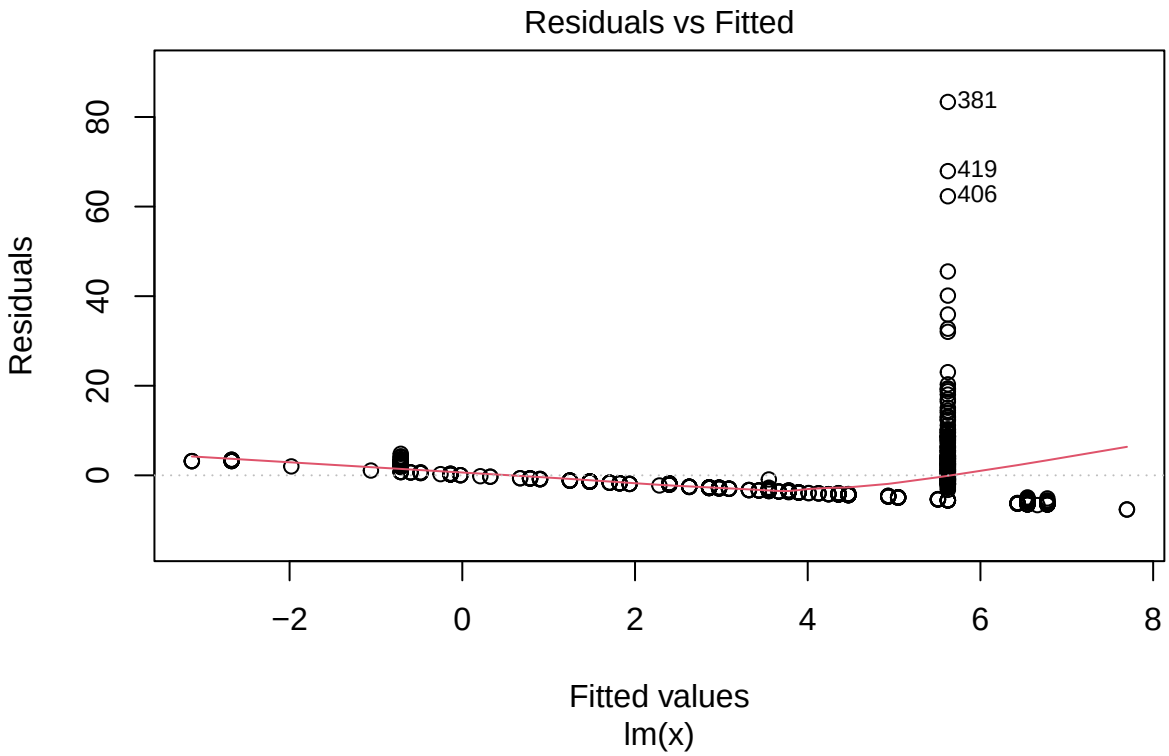

Residuals vs Fitted
Residuals
Fitted values
lm(x)
381
419
406
−20
0
20
40
60
80
0
5
10

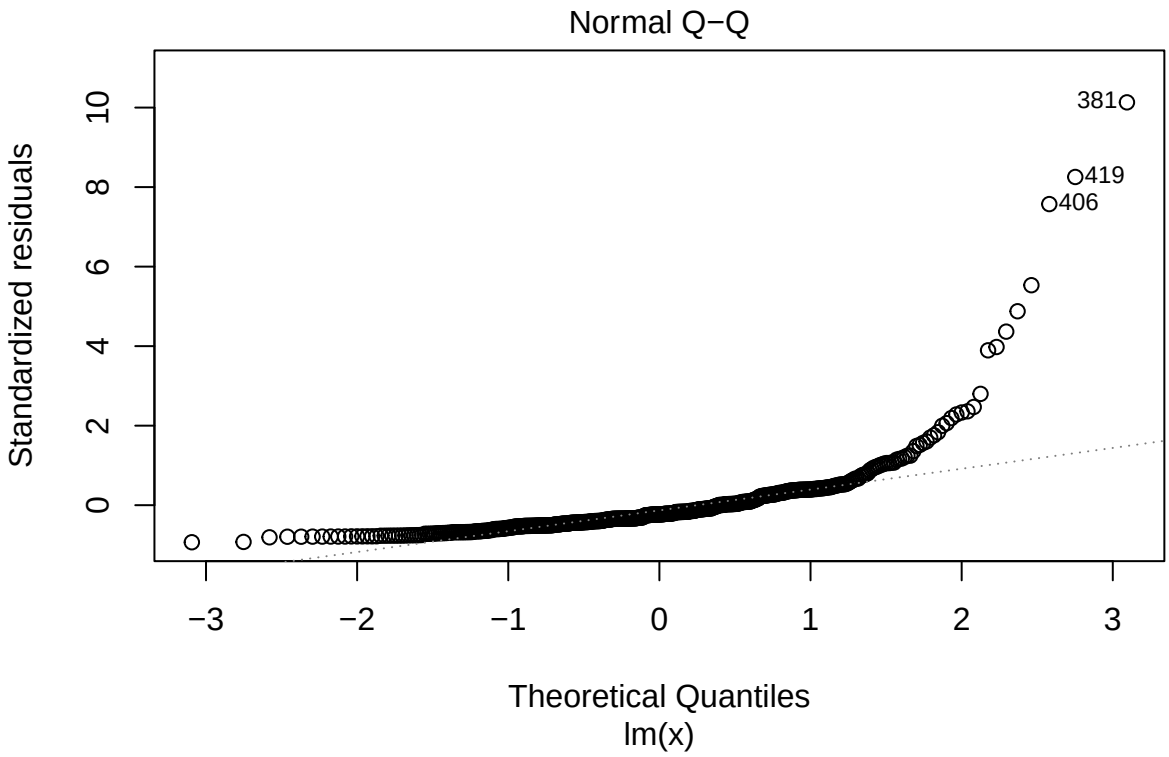

Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(x)
381
419
406
−2
0
2
4
6
8
10
12
−3
−2
−1
0
1
2
3

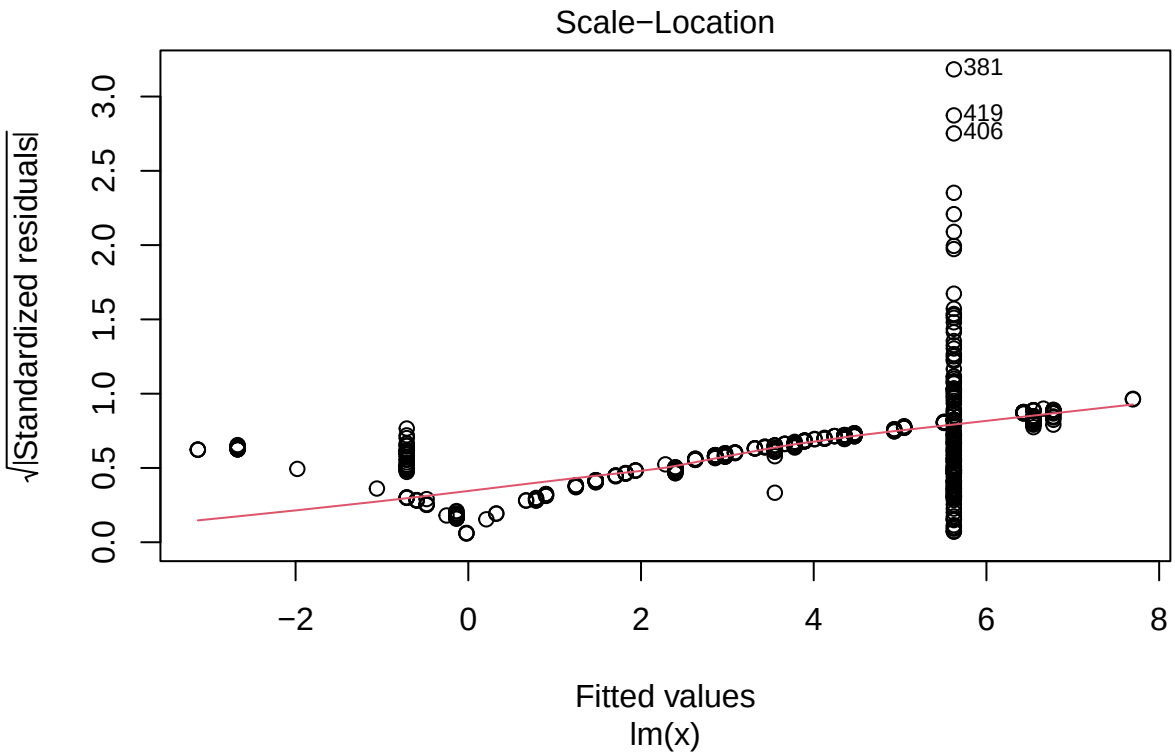

Scale−Location
√|Standardized residuals|
0.0
1.0
2.0
3.0
381
419
406
0
5
10
Fitted values
lm(x)

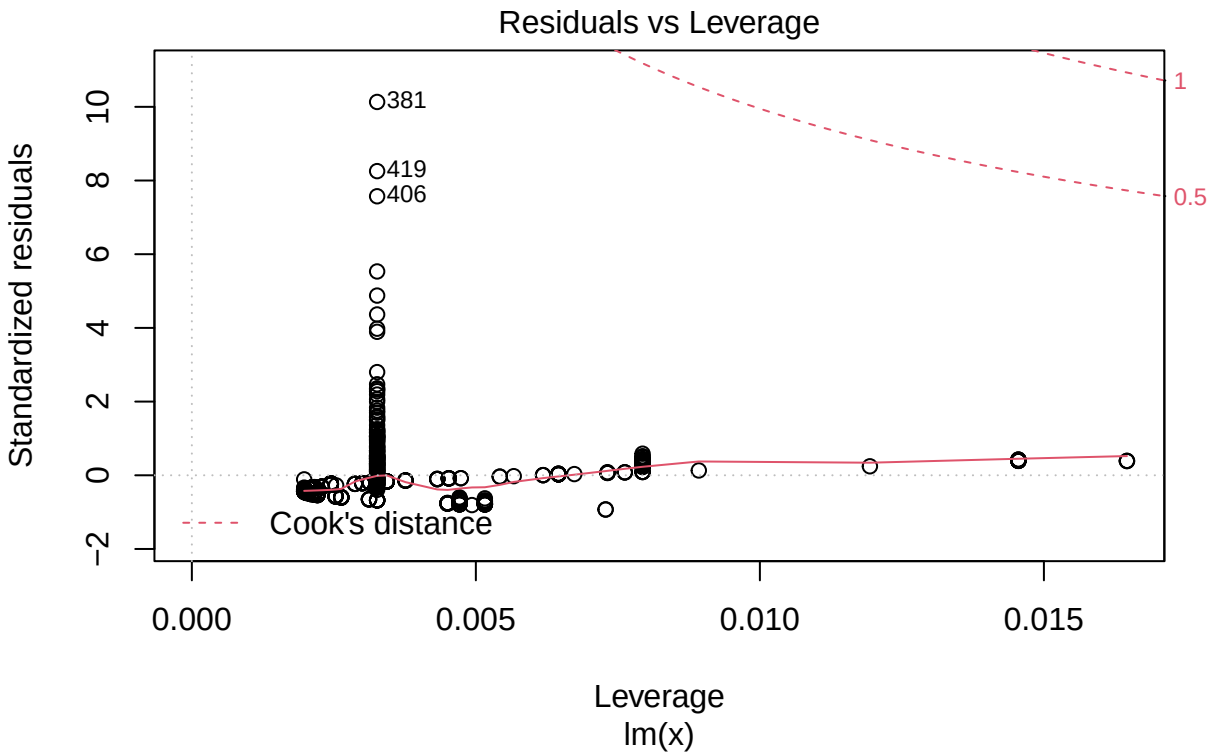



```
##
## Call:
## lm(formula = x, data = boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.654 -3.985 -1.912  1.825 83.353
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.6469     3.1473  -5.607 3.40e-08 ***
## ptratio       1.1520     0.1694   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
##
## plots for  ptratio
```

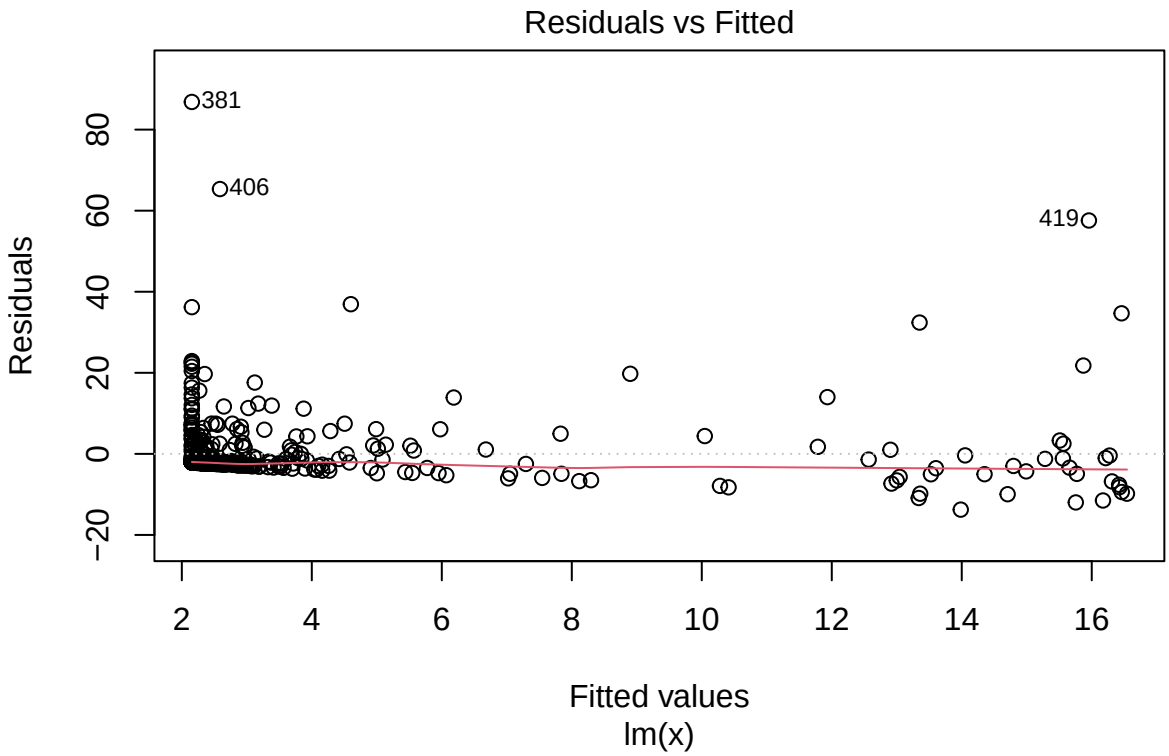

Residuals vs Fitted
Residuals
0
20
40
60
80
381
419
406
−2
0
2
4
6
8
Fitted values
lm(x)

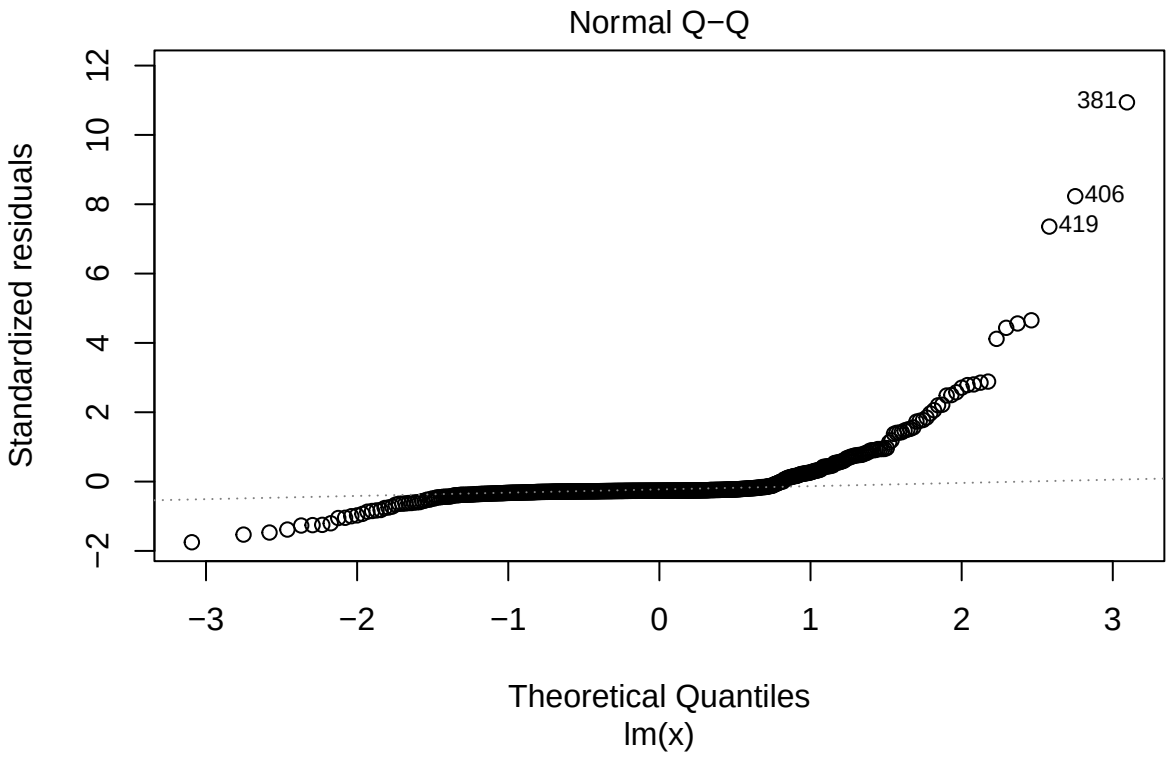

Normal Q−Q
Standardized residuals
0
2
4
6
8
10
381
419
406
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(x)

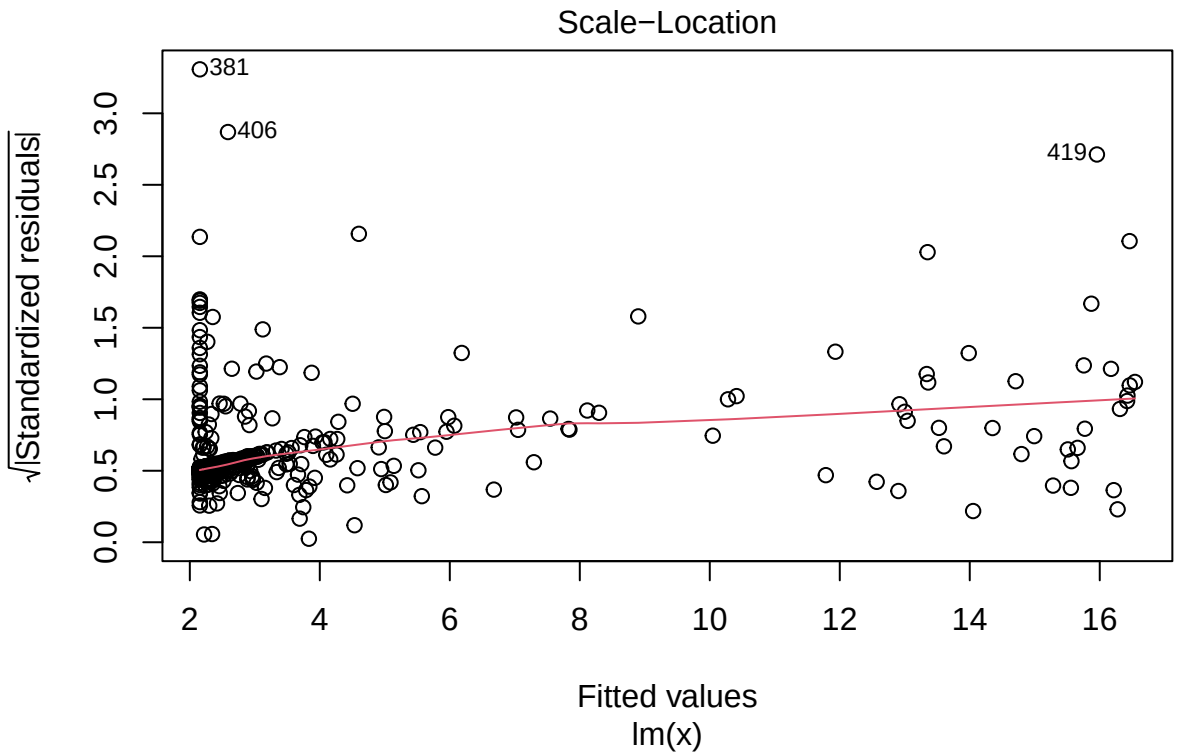

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
381
419
406
−2
0
2
4
6
8
Fitted values
lm(x)

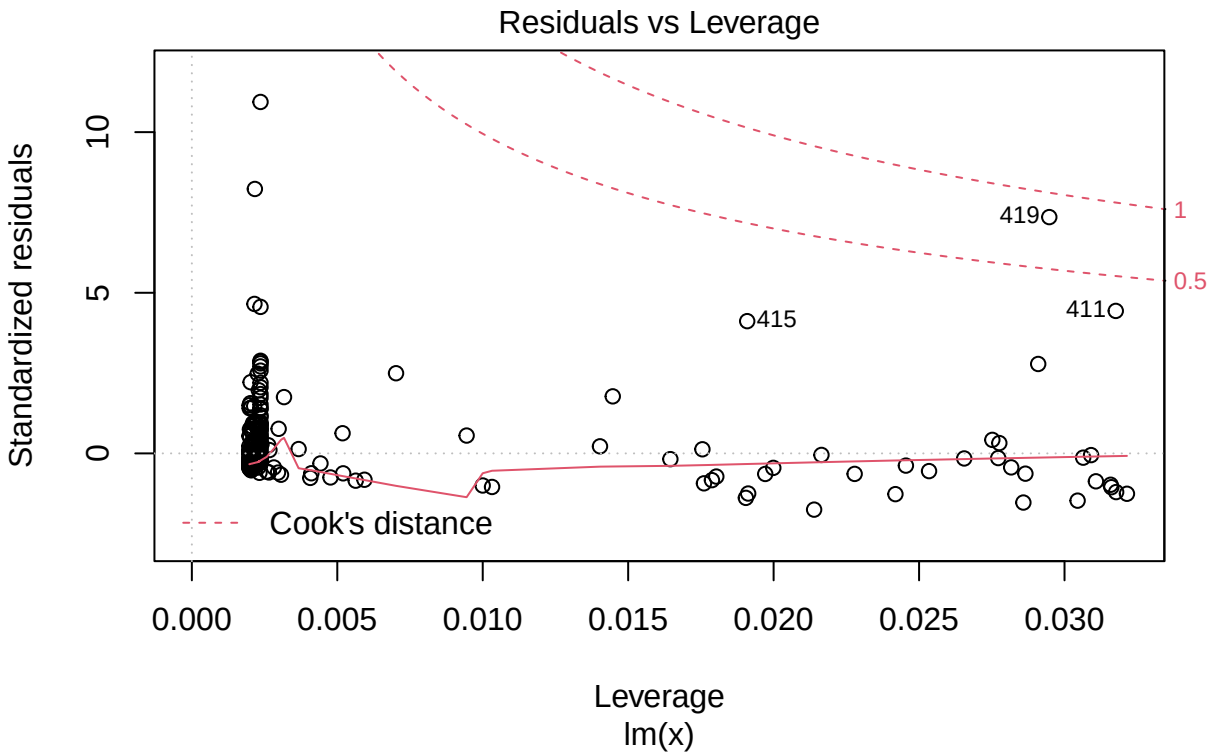



```
##
## Call:
## lm(formula = x, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.756  -2.299  -2.095  -1.296  86.822
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.553529   1.425903  11.609   <2e-16 ***
## black       -0.036280   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
##
## plots for  black
```

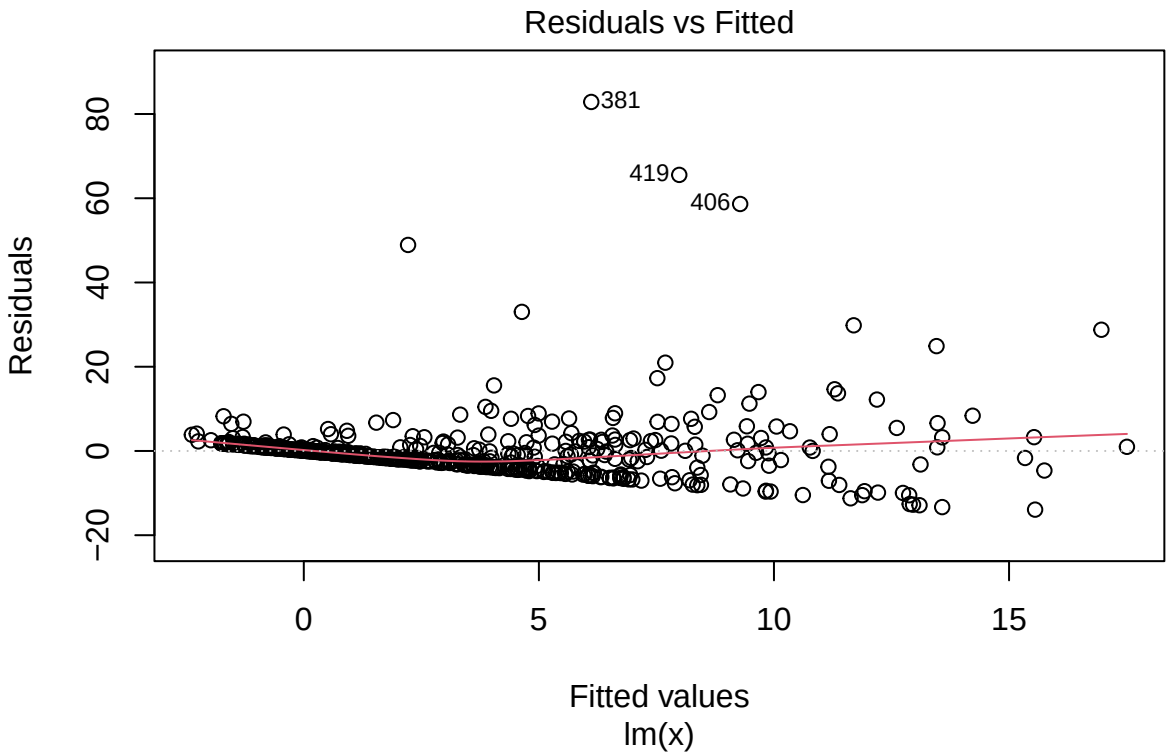

Residuals vs Fitted
381
406
419
Residuals
−20
0
20
40
60
80
2
4
6
8
10
12
14
16
Fitted values
lm(x)

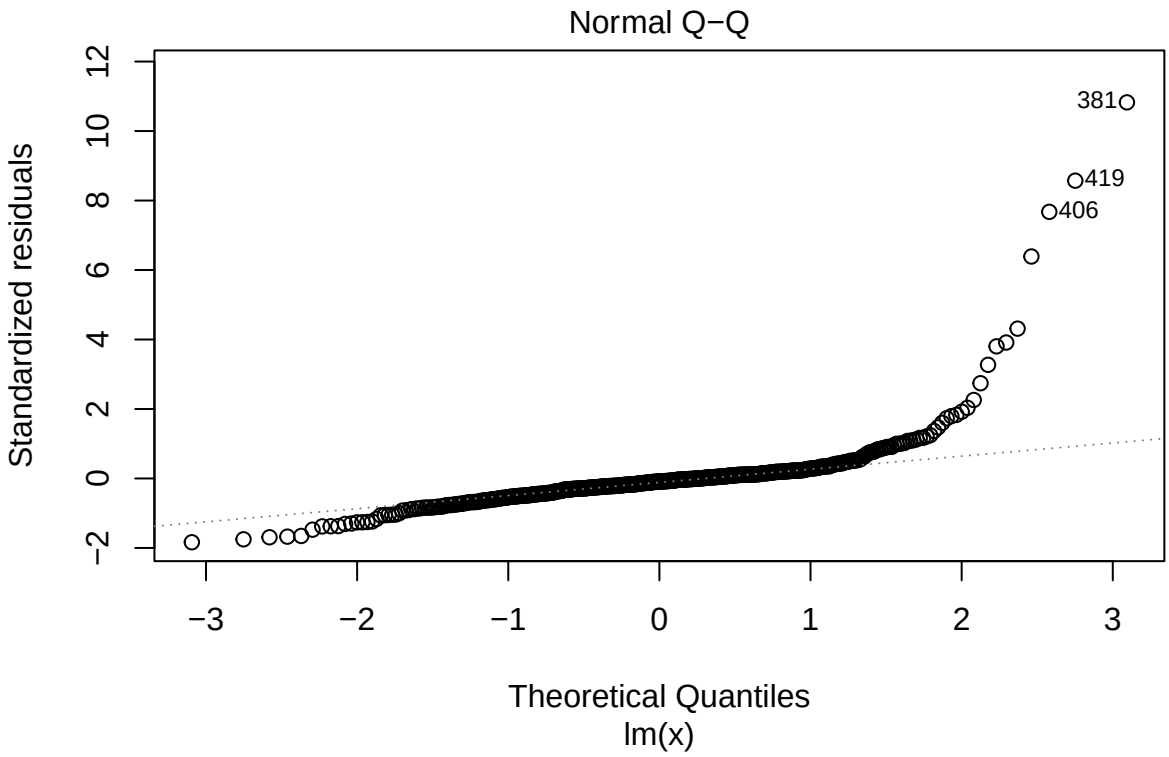

Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(x)
381
406
419
−2
0
2
4
6
8
10
12
−3
−2
−1
0
1
2
3

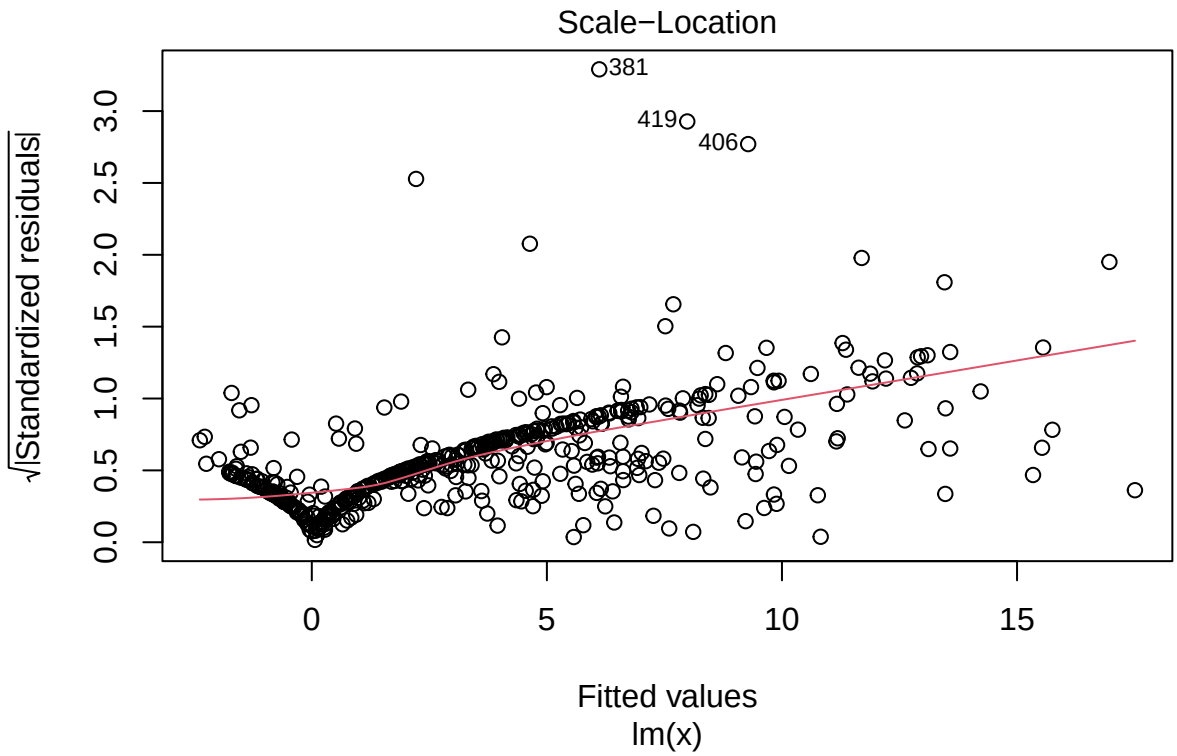

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
381
406
419
2
4
6
8
10
12
14
16
Fitted values
lm(x)

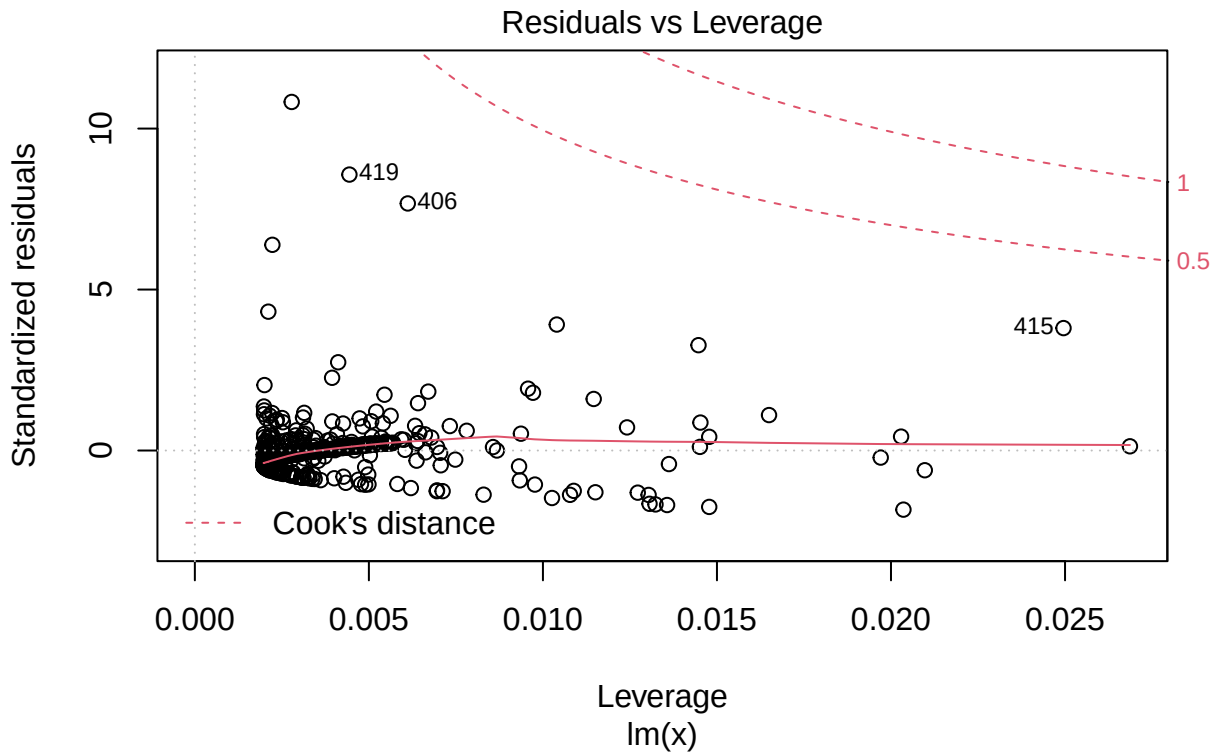



```
##
## Call:
## lm(formula = x, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.925  -2.822  -0.664   1.079  82.862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.33054    0.69376  -4.801 2.09e-06 ***
## lstat        0.54880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

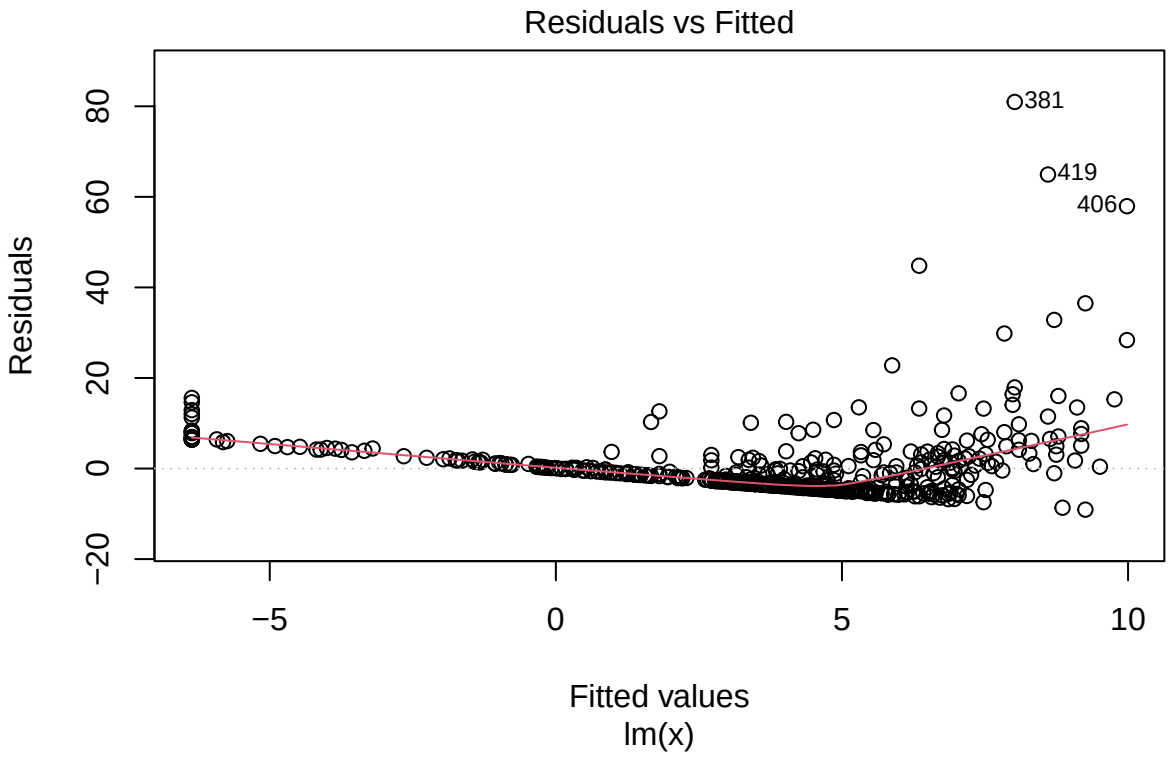

Residuals vs Fitted
381
419
406
Residuals
−20
0
20
40
60
80
0
5
10
15
Fitted values
lm(x)

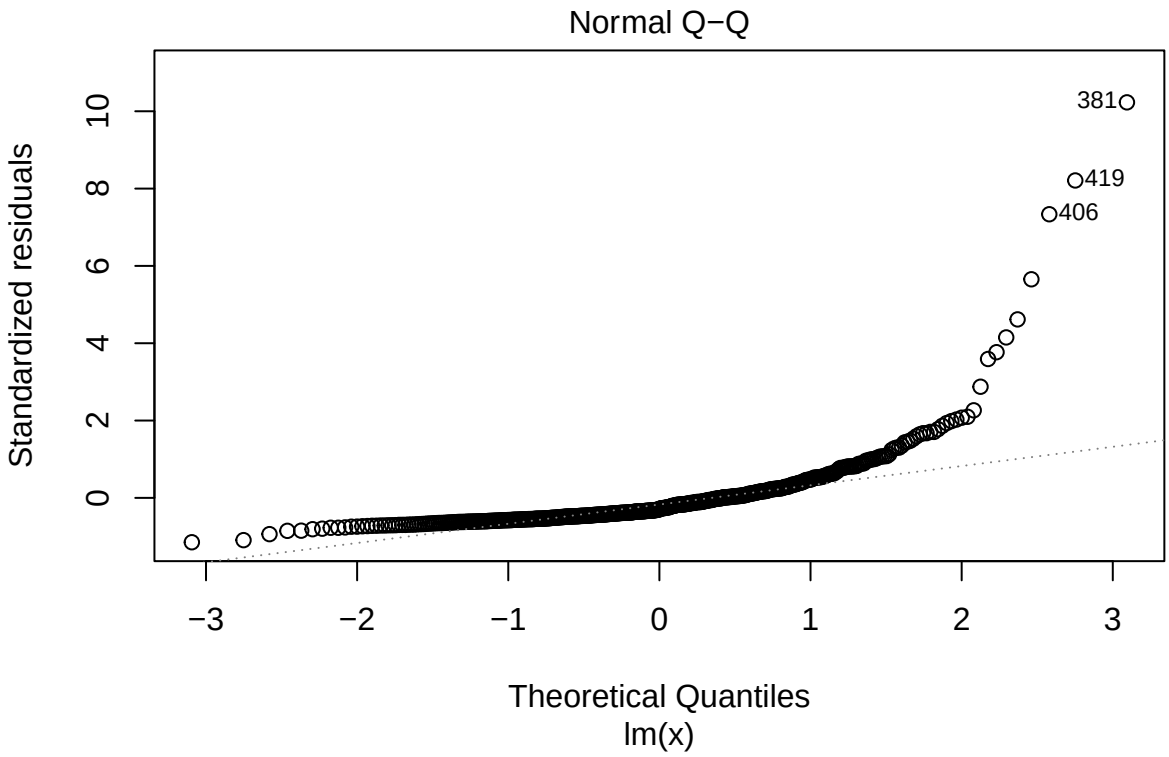

Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(x)
381
419
406
−2
0
2
4
6
8
10
12
−3
−2
−1
0
1
2
3

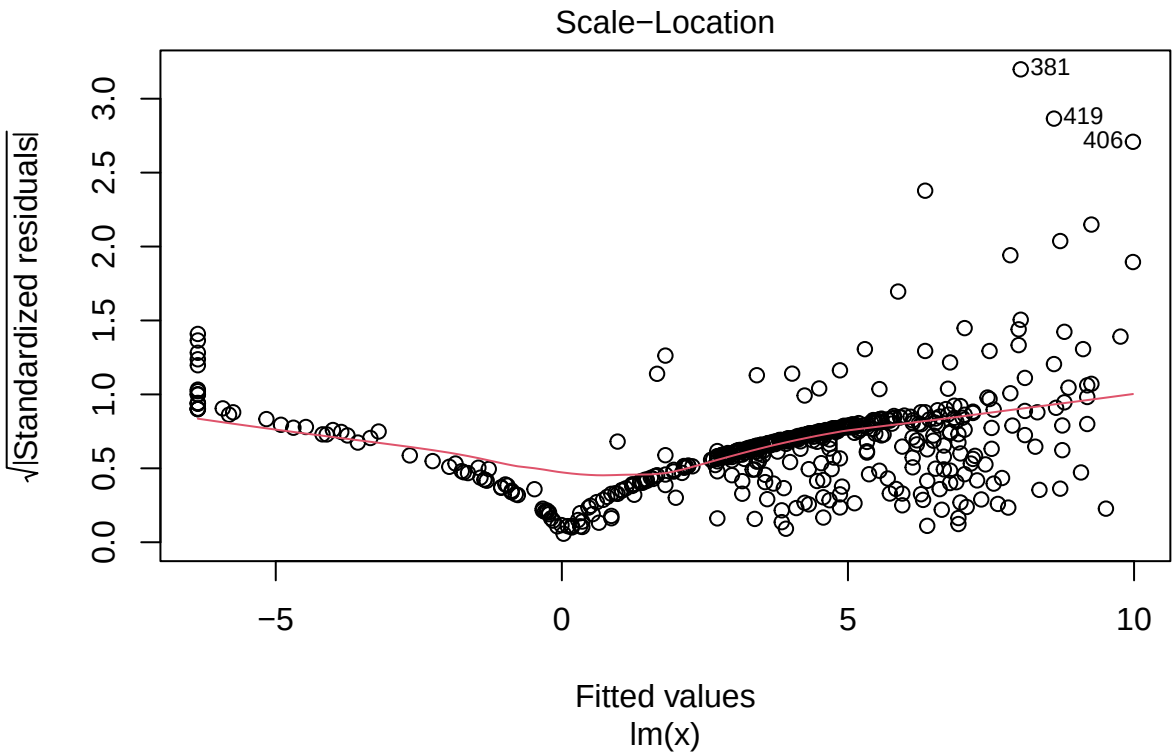

Scale−Location
381
419
406
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
5
10
15
Fitted values
lm(x)

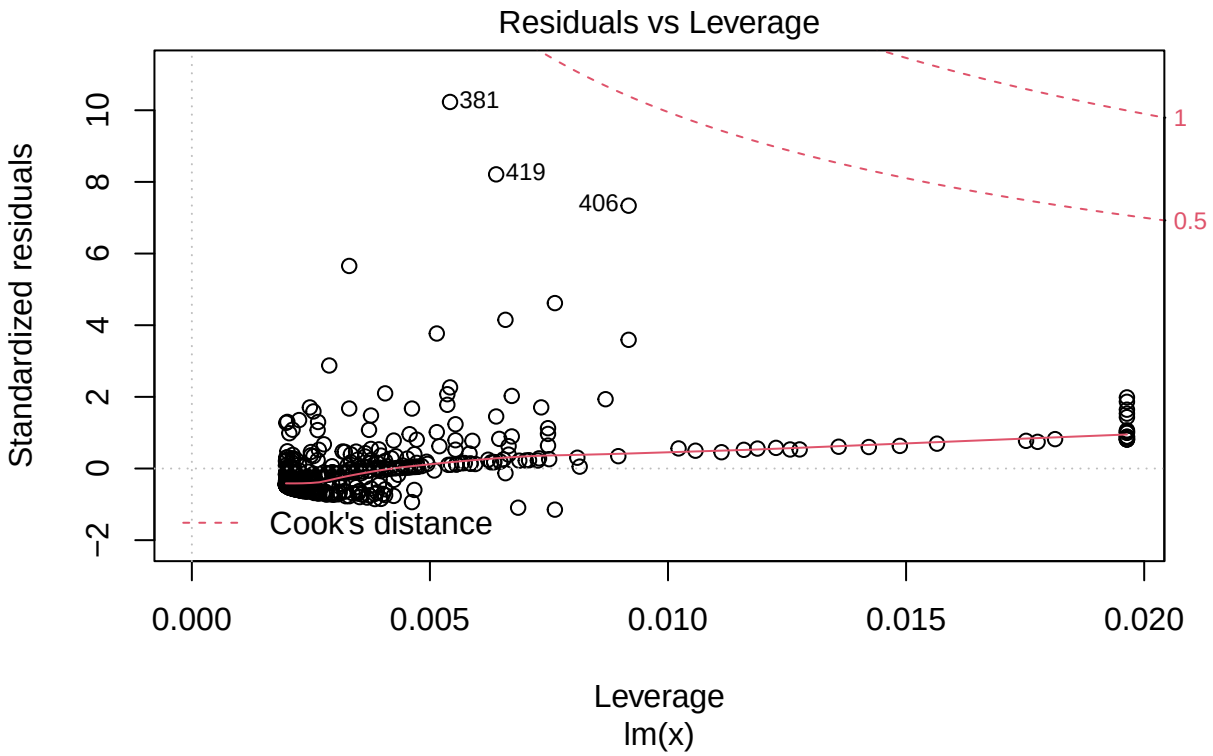



```
##
## Call:
## lm(formula = x, data = boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.071 -4.022 -2.343  1.298 80.957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.79654    0.93419   12.63   <2e-16 ***
## medv        -0.36316    0.03839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


Residuals vs Fitted
Residuals
Fitted values
lm(x)
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(x)


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(x)



Answer:

By fitting the models, all predictors were significantly associated with response except for chas variable.

(b) Fit a multiple regression model to predict the response using all of the predictors. Describe your results. For which predictors can we reject the null hypothesis $H_0 : \beta_j = 0$?

```
func = "crim ~ zn"
var_names2 = var_names[2:13]
for (name in var_names2) {
  func = paste(func, '+ ', name)
}
func = as.formula(func)
reg3 <- lm(func, data = boston)
summary(reg3)
```

```
##
## Call:
## lm(formula = func, 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
```

Answer

Based on the model results, for zn, dis, rad, black, medv we can reject the null hypothesis of $\beta_j = 0$

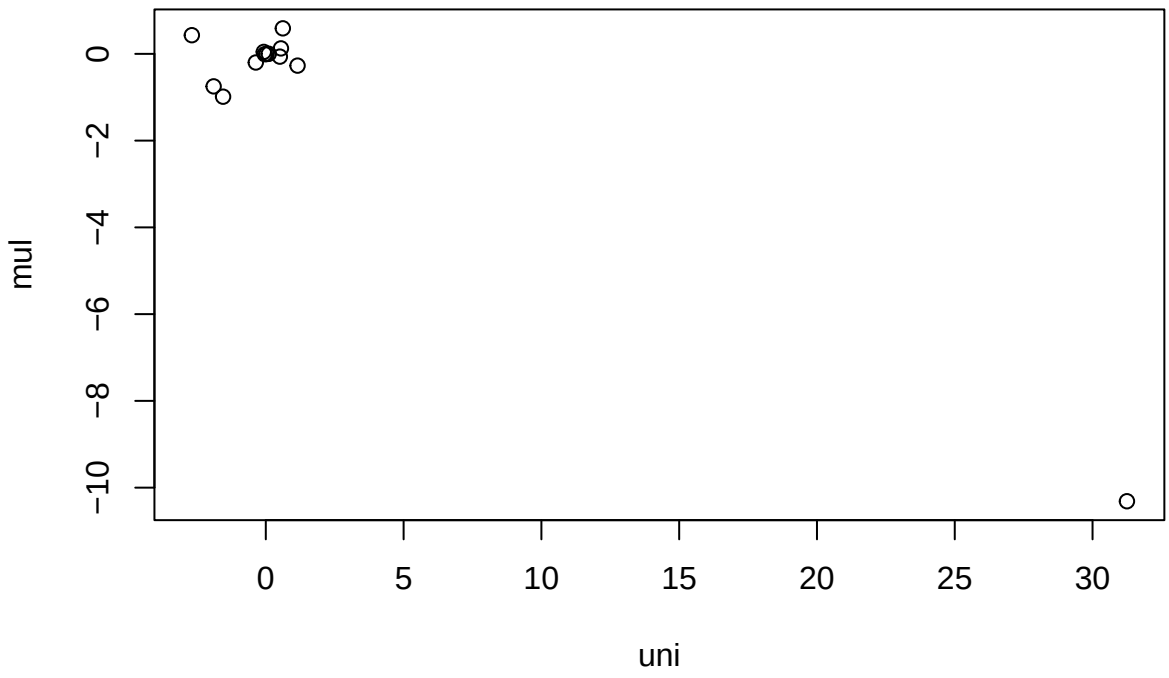
(c) How do your results from (a) compare to your results from (b)? Create a plot displaying the univariate regression coefficients from (a) on the x-axis, and the multiple regression coefficients from (b) on the y-axis. That is, each single point in the plot. Its coefficient in a simple linear regression model is shown on the x-axis, and its coefficient estimate in the multiple linear regression model is shown on the y-axis.

Answer Part 1

As compared to question (a), question (b) suggested that zn, dis, rad, black, medv which were significantly associated with response variable crim in (a) are now not significantly associated with crim in (b).

Plot

```
uni <- vector("numeric", 0)
for (n in 1:13) {
  uni <- c(uni, allModelsResults[[n]]$coefficient[2])
}
mul <- vector("numeric", 0)
mul <- c(mul, reg3$coefficients)
mul <- mul[-1]
plot(uni, mul)
```



(d) Is there evidence of non-linear association between any of the predictors and the response? To answer this question, for each predictor X, fit a model of the form $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$

```
var_names3 <- var_names[-3]
allModelsListPoly <- lapply(paste("crim ~", " ", "poly(", var_names3, ", 3)"), as.formula)
allModelsPoly <- lapply(allModelsListPoly, function(x) lm(x, data= boston))
for (n in 1:12) {
  print(summary(allModelsPoly[[n]]))
}
```

```
##
## Call:
## lm(formula = x, 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)    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 = x, 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)        3.614      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 = x, 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)     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 = x, 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)    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 = x, 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)     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 = x, 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)     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 = x, 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)     3.6135     0.2971  12.164  < 2e-16 ***
## poly(rad, 3)1 120.9074     6.6824  18.093  < 2e-16 ***
## poly(rad, 3)2  17.4923     6.6824   2.618  0.00912 **
## poly(rad, 3)3   4.6985     6.6824   0.703  0.48231
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.682 on 502 degrees of freedom
## Multiple R-squared:    0.4,  Adjusted R-squared:  0.3965
## F-statistic: 111.6 on 3 and 502 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = x, 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)     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 = x, 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)          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 = x, 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)       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 = x, 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)       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 = x, 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)       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
```

Answer

Based on the p-values associated with each coefficient, we can see that for predictors of: indus, nox, age, dis, ptratio, and medv, all 3 coefficients (linear, quadratic, cubic) were significant. For predictors of :zn, rm, rad, tax, and lstat, only linear and quadratic terms were significant. For predictor of: black, only linear term coefficient was significant. Therefore, for indus, nox, age, dis, ptratio, medv, zn, rm, rad, tax, and lstat, there were evidences of non-linear relationship, but for black there was no non-linear relationship observed.

Clissfication - Question 1, 8, 10, 11

Question 1

Using a little bit of algebra, prove that (4.2) is equivalent to (4.3). In other words, the logistic function representation and logit representation for the logistic regression model are equivalent.

Answer

Equation 4.2:

$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$

Equation 4.3:

$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X}$

$$
\begin{aligned}
P(x) &= \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}} \\
P(x)(1+^{\beta_0+\beta_1 x}) &= e^{\beta_0+\beta_1 x} \\
\frac{P(x)}{\frac{1}{1+e^{\beta_0+\beta_1 x}}} &= e^{\beta_0+\beta_1 x} \\
\frac{P(x)}{1-\frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}} &= e^{\beta_0+\beta_1 x} \\
\frac{P(x)}{1-P(x)} &= e^{\beta_0+\beta_1 x}
\end{aligned}
$$

Question 8

Suppose that we take a data set, divide it into equally-sized training and test sets, and then try out two different classification procedures. First we use logistic regression and get an error rate of 20% on the training data and 30 % on the test data. Next we use 1-nearest neighbors (i.e. K = 1) and get an average error rate (averaged over both test and training data sets) of 18 %. Based on these results, which method should we prefer to use for classification of new observations? Why?

Answer

We should prefer to use logistic regression.

The average error rate over both test and training data sets for 1-nearest neighbors was 18%, which means the error rate in the test dataset will be 36%, since when using only one neighbor, the estimation of any observation from the training dataset will be its response, so that the error rate under 1-nearest neighbor for the training dataset will be 0%. After getting this 0%, we can infer that the error rate for test dataset is $18\% * 2 - 0\% = 36\%$. As 36% was higher than the test set error rate under logistic regression of 30%, we should not use the 1-nearest neighbor method thus prefer the logistic regression method.

Question 10

This question should be answered using the Weekly data set, which is part of the ISLR package. This data is similar in nature to the Smarket data from this chapter's lab, except that it contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

(a) Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?

```
library(ISLR)
names(Weekly)
```

```
## [1] "Year"      "Lag1"      "Lag2"      "Lag3"      "Lag4"      "Lag5"
## [7] "Volume"    "Today"     "Direction"
```

```
dim(Weekly)
```

```
## [1] 1089    9
```

```
summary(Weekly)
```

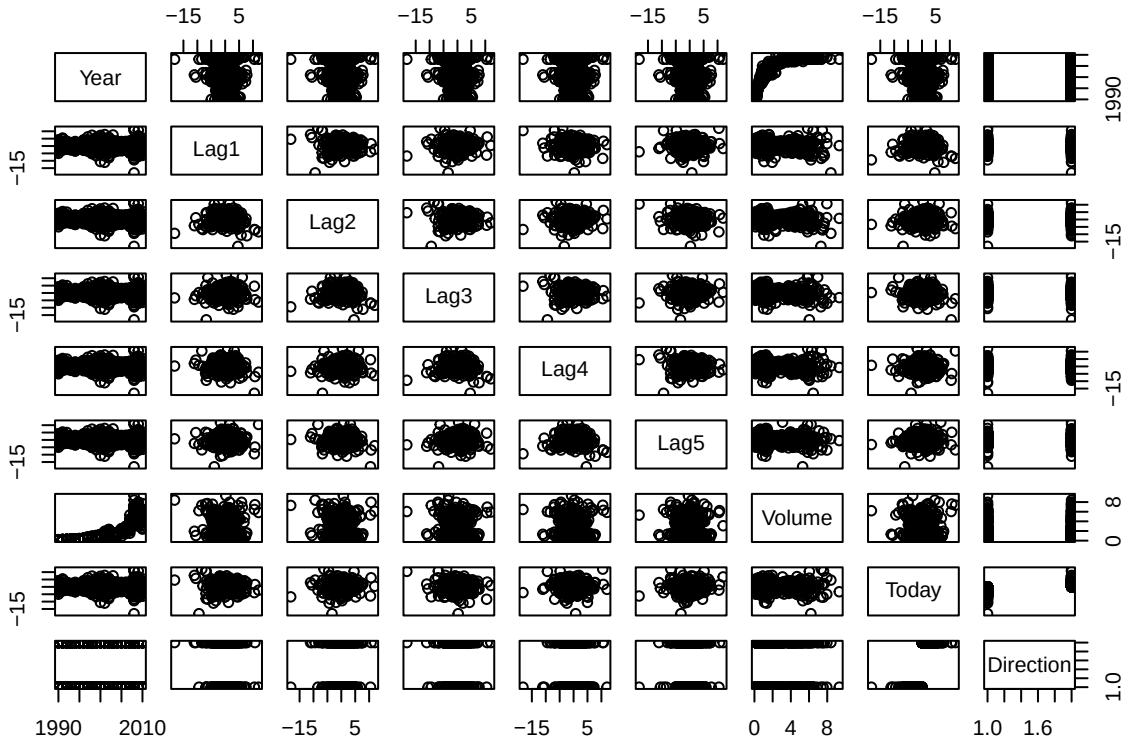
```
##       Year           Lag1               Lag2               Lag3
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410
##  Mean   :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
##
##
##
##
```

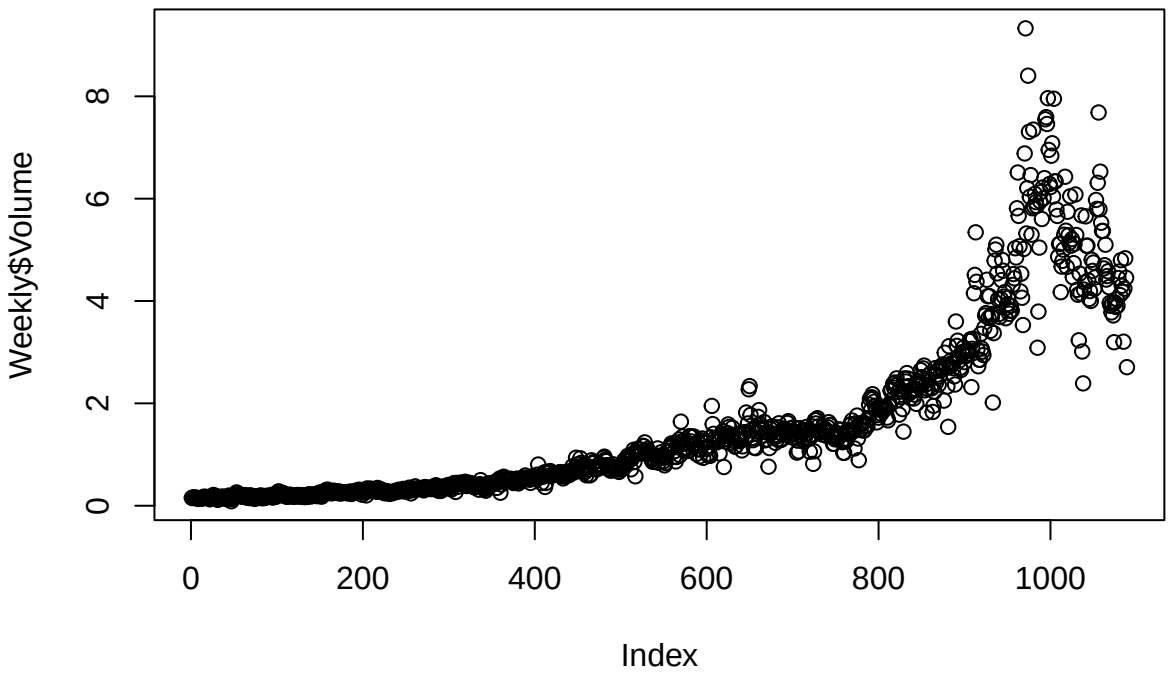
```
cor(Weekly[,1:8])
```

```
##               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
```

```
pairs(Weekly)
```




```
plot(Weekly$Volume)
```



Answer As suggested by cor(Weekly), the correlations between lag variables and today's return were weak. The only strong correlations were between Year and Volume, and the volume tend to increase over time.

(b) Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

```
weekly <- Weekly
reg4 <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = weekly, family = binomial)
summary(reg4)
```

```
##
## 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
```

Answer The only statistically significant predictor was Lag2.

(c) Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

```
reg4.probs <- predict(reg4, type = "response")
reg4.pred <- rep("Down", length(reg4.probs))
reg4.pred[reg4.probs > 0.5] <- "Up"
table(reg4.pred, weekly$Direction)
```

```
##
## reg4.pred Down  Up
##      Down   54  48
##      Up    430 557
```

Answer

The confusion matrix is able to show the true positives, false positives, true negatives, false negatives thus will be able to tell us the percentage of correct and wrong predictions. The overall correction rate was:

$\frac{54+557}{54+48+430+557} * 100\% = 56.11\%$

The correction rate among weeks that the market goes up:

$\frac{557}{557+48} * 100\% = 92.06\%$

The correction rate among weeks that the market goes down:

$\frac{54}{54+430} * 100\% = 11.16\%$

(d) Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).

```
training <- weekly %>%
            filter(Year < 2009)
test <- weekly %>%
        filter(Year > 2008)
reg5 <- glm(Direction ~ Lag2, data=training, family=binomial)
summary(reg5)
```

```
##
## Call:
## glm(formula = Direction ~ Lag2, family = binomial, data = training)
##
## 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
```

```
reg5.probs = predict(reg5, test, type="response")
reg5.pred = rep("Down", 104)
reg5.pred[reg5.probs > 0.5] = "Up"

cmatrix <- table(reg5.pred, test$Direction)
print(cmatrix)
```

```
##
## reg5.pred Down Up
##      Down    9  5
##      Up     34 56
```

```
mean(reg5.pred == test$Direction)
```

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

Answer The precentage of correct prediction on test set is:

$\frac{9+56}{9+5+34+56} * 100\% = 62.5\%$

(e) Repeat (d) using LDA.

```
library(MASS)
reg6 <- lda(Direction ~ Lag2, data = training)
reg6
```

```
## Call:
## lda(Direction ~ Lag2, data = training)
##
```

```
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##             Lag2
## Down -0.03568254
## Up    0.26036581
##
## Coefficients of linear discriminants:
##            LD1
## Lag2 0.4414162
```

```
reg6.pred <- predict(reg6, test)
table(reg6.pred$class, test$Direction)
```

```
##
##        Down Up
##   Down    9  5
##   Up     34 56
```

```
mean(reg6.pred$class == test$Direction)
```

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

Answer

The precentage of correct prediction on test set is:

$\frac{9+56}{9+5+34+56} * 100\% = 62.5\%$

(f) Repeat (d) using QDA.

```
reg7 <-qda(Direction ~ Lag2, data = training)
reg7
```

```
## Call:
## qda(Direction ~ Lag2, data = training)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##             Lag2
## Down -0.03568254
## Up    0.26036581
```

```
reg7.pred <- predict(reg7,test)
table(reg7.pred$class,test$Direc)
```

```
##
##        Down Up
##   Down    0  0
##   Up     43 61
```

```
mean(reg7.pred$class == test$Direction)
```

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

Answer

The precentage of correct prediction on test set is:

$\frac{0+61}{0+61+43+61} * 100\% = 58.7\%$

(g) Repeat (d) using KNN with K = 1.

```
library(class)
set.seed(1)
knn1.training <- as.matrix(training$Lag2)
knn1.test <- as.matrix(test$Lag2)
knn1.pred <- knn(knn1.training, knn1.test, training$Direction, k=1)
table(knn1.pred, test$Direction)
```

```
##
## knn1.pred Down Up
##      Down   21 30
##      Up     22 31
```

```
mean(knn1.pred == test$Direction)
```

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

Answer

The precentage of correct prediction on test set is:

$\frac{31+21}{21+31+22+30} * 100\% = 50.0\%$

(h) Which of these methods appears to provide the best results on this data?

Answer

Comparing the overall correction rate in test sets, we can see that logistic and linear discriminitive analysis were the two best methods and provided the best results, followed by quadratic discriminative analysis, and the one provided wrost result was KNN-1.

(i) Experiment with different combinations of predictors, including possible transformations and interactions, for each of the methods. Report the variables, method, and associated confusion matrix that appears to provide the best results on the held out data. Note that you should also experiment with values for K in the KNN classifier.

```r
# k = 3
set.seed(1)
knn3.pred <- knn(knn1.training, knn1.test, training$Direction, k = 3)
cat(paste('Method: knn with k = 3', 'associated confusion matrix', sep = '\n'))
```

```
## Method: knn with k = 3
## associated confusion matrix
```

```r
table(knn3.pred, test$Direction)
```

```
##
## knn3.pred Down Up
##      Down   16 20
##      Up     27 41
```

```r
cat('the test correction rate is: \n')
```

```
## the test correction rate is:
```

```r
mean(knn3.pred == test$Direction)
```

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

```r
# k = 10
set.seed(1)
knn10.pred <- knn(knn1.training, knn1.test, training$Direction, k = 10)
cat(paste('Method: knn with k = 10', 'associated confusion matrix', sep = '\n'))
```

```
## Method: knn with k = 10
## associated confusion matrix
```

```r
table(knn10.pred, test$Direction)
```

```
##
## knn10.pred Down Up
##       Down   17 21
##       Up     26 40
```

```r
cat('the test correction rate is: \n')
```

```
## the test correction rate is:
```

```r
mean(knn10.pred == test$Direction)
```

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

```r
# k = 100
set.seed(1)
knn100.pred <- knn(knn1.training, knn1.test, training$Direction, k=100)
cat(paste('Method: knn with k = 100', 'associated confusion matrix', sep = '\n'))
```

```
## Method: knn with k = 100
## associated confusion matrix
```

```r
table(knn100.pred, test$Direction)
```

```
##
## knn100.pred Down Up
##        Down   10 11
##        Up     33 50
```

```r
cat('the test correction rate is: \n')
```

```
## the test correction rate is:
```

```r
mean(knn100.pred == test$Direction)
```

```
## [1] 0.5769231
```

```r
# LDA, all Lag values
reg8 <- lda(Direction ~ Lag2:Lag1, data=training)
reg8.pred <- predict(reg8,test)
cat(paste('Method: Linear Discriminative Analysis', 'Variables: Lag1, Lag2, interaction term between Lag
```

```
## Method: Linear Discriminative Analysis
## Variables: Lag1, Lag2, interaction term between Lag1 and Lag2
## associated confusion matrix
```

```r
table(reg8.pred$class,test$Direction)
```

```
##
##        Down Up
##   Down    0  1
##   Up     43 60
```

```r
cat('the test correction rate is: \n')
```

```
## the test correction rate is:
```

```r
mean(reg8.pred$class == test$Direction)
```

```
## [1] 0.5769231
```

```r
# Logistic, Lag2 Lag1 interaction
reg9 <- glm(Direction ~ Lag2:Lag1, data=training, family=binomial)

reg9.probs <- predict(reg9, test, type="response")
reg9.pred <- rep("Down", 104)
reg9.pred[reg9.probs > 0.5] = "Up"
cat(paste('Method: Logistic Regression Analysis', 'Variables: Lag1, Lag2, interaction term between Lag1
```

```
## Method: Logistic Regression Analysis
## Variables: Lag1, Lag2, interaction term between Lag1 and Lag2
## associated confusion matrix
```

```r
table(reg9.pred, test$Direction)
```

```
##
## reg9.pred Down Up
##      Down    1  1
##      Up     42 60
```

```r
cat('the test correction rate is: \n')
```

```
## the test correction rate is:
```

```r
mean(reg9.pred == test$Direction)
```

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

```r
# QDA, Lag2 lag1 interaction
reg10 <- qda(Direction ~ Lag2:Lag1, data=training)
reg10.pred <- predict(reg10, test)$class
cat(paste('Method: Quadratic Discriminative Analysis', 'Variables: Lag1, Lag2, interaction term between
```

```
## Method: Quadratic Discriminative Analysis
## Variables: Lag1, Lag2, interaction term between Lag1 and Lag2
## associated confusion matrix
```

```r
table(reg10.pred, test$Direction)
```

```
##
## reg10.pred Down Up
##       Down   16 32
##       Up     27 29
```

```r
cat('the test correction rate is: \n')
```

```
## the test correction rate is:
```

```
mean(reg10.pred == test$Direction)
```

```
## [1] 0.4326923
```

Answer

The results suggested logistic regression methods provided the best results of correction rate on the test (held out) data. Information about confusion matrix and correction rate is below:

```
cat(paste('Method: Linear Discriminative Analysis', 'Variables: Lag1, Lag2, interaction term between Lag
```

```
## Method: Linear Discriminative Analysis
## Variables: Lag1, Lag2, interaction term between Lag1 and Lag2
## associated confusion matrix
```

```
table(reg8.pred$class,test$Direction)
```

```
##
##        Down Up
##   Down    0  1
##   Up     43 60
```

```
cat('the test correction rate is: \n')
```

```
## the test correction rate is:
```

```
mean(reg8.pred$class == test$Direction)
```

```
## [1] 0.5769231
```

Question 11

(a) Create a binary variable, mpg01, that contains a 1 if mpg contains a value above its median, and a 0 if mpg contains a value below its median. You can compute the median using the median() function. Note you may find it helpful to use the data.frame() function to create a single data set containing both mpg01 and the other Auto variables.

```
library(ISLR)
attach(Auto)
```

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

```
## The following object is masked from package:ggplot2:
##
##     mpg
```

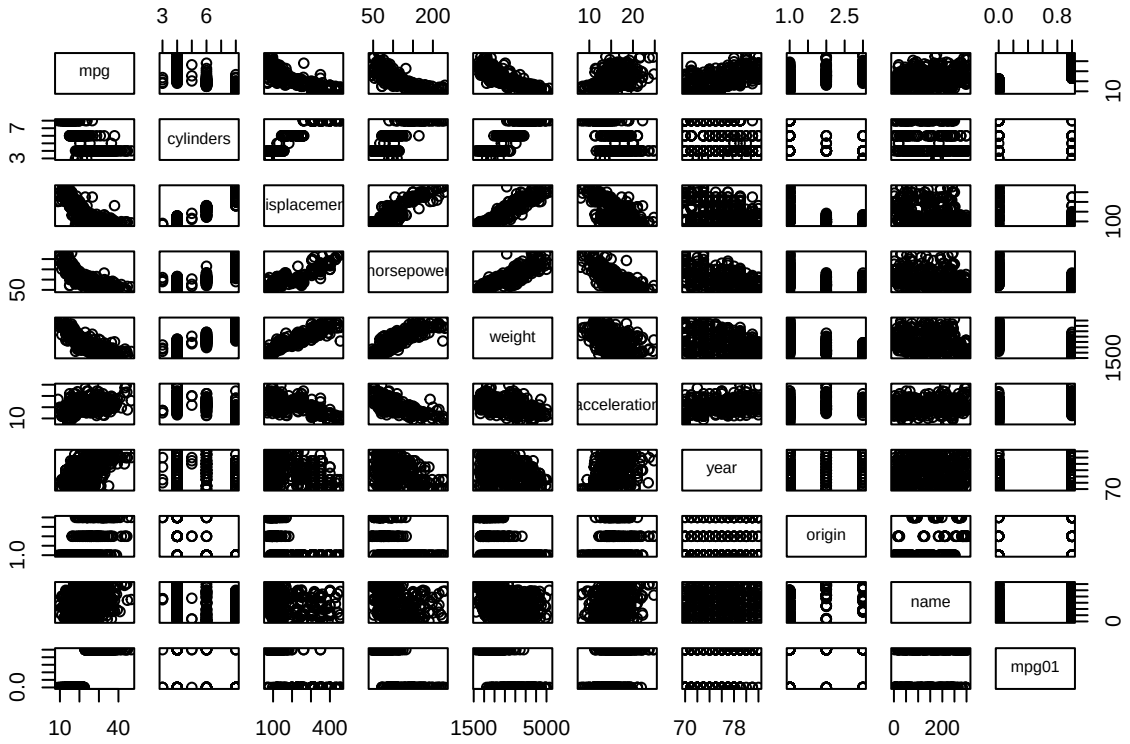
```
mpg01 = rep(0, length(mpg))
mpg01[mpg > median(mpg)] = 1
Auto <- data.frame(Auto, mpg01)
```

(b) Explore the data graphically in order to investigate the association between mpg01 and the other features. Which of the other features seem most likely to be useful in predicting mpg01? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings.

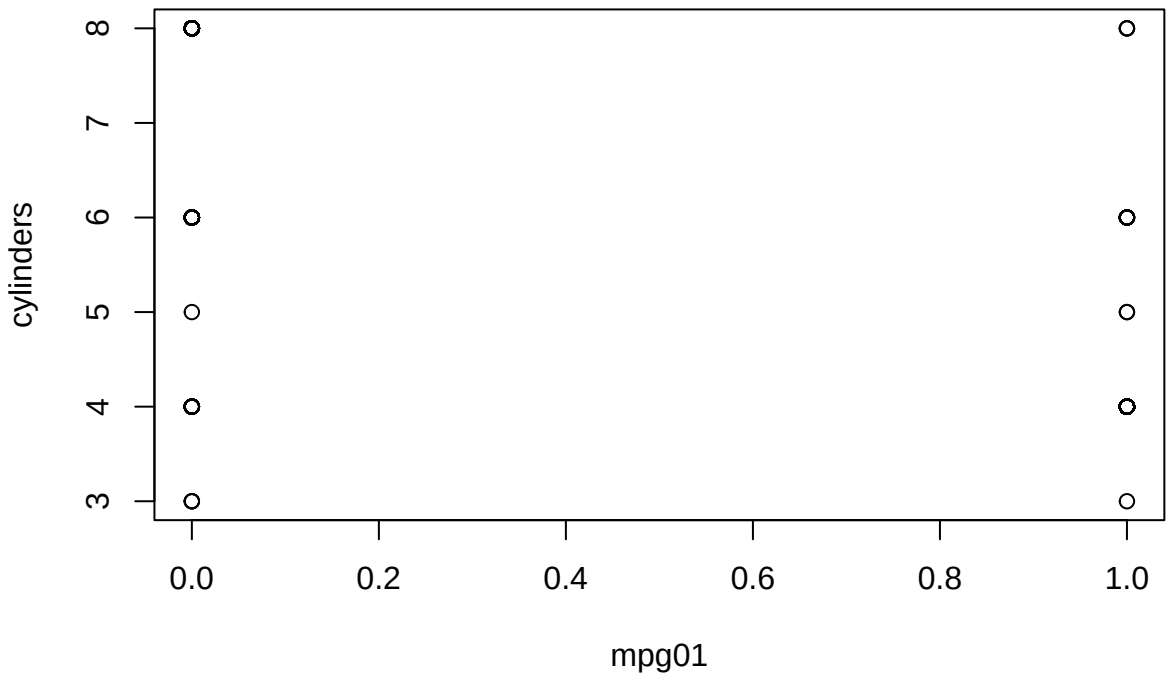
```
summary(Auto [1:8])
```

```
##       mpg          cylinders      displacement     horsepower        weight
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613
##  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225
##  Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Median :2804
##  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   :2978
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140
##   acceleration        year           origin
##  Min.   : 8.00   Min.   :70.00   Min.   :1.000
##  1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000
##  Median :15.50   Median :76.00   Median :1.000
##  Mean   :15.54   Mean   :75.98   Mean   :1.577
##  3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000
##  Max.   :24.80   Max.   :82.00   Max.   :3.000
```

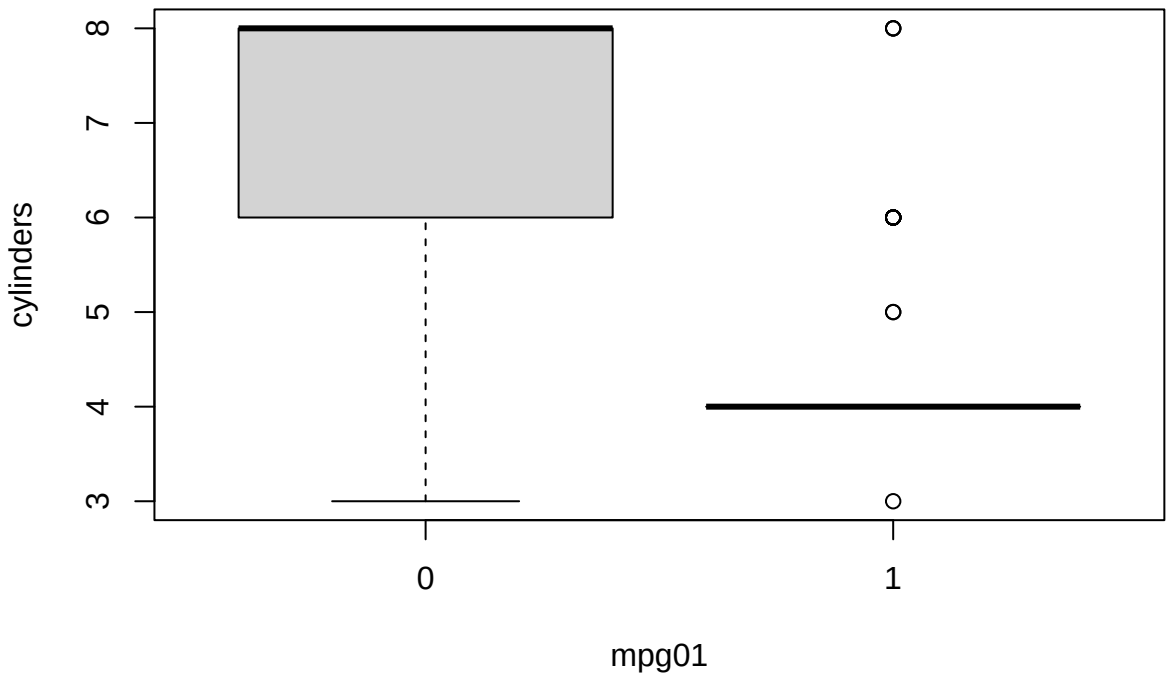
```
pairs(Auto)
```



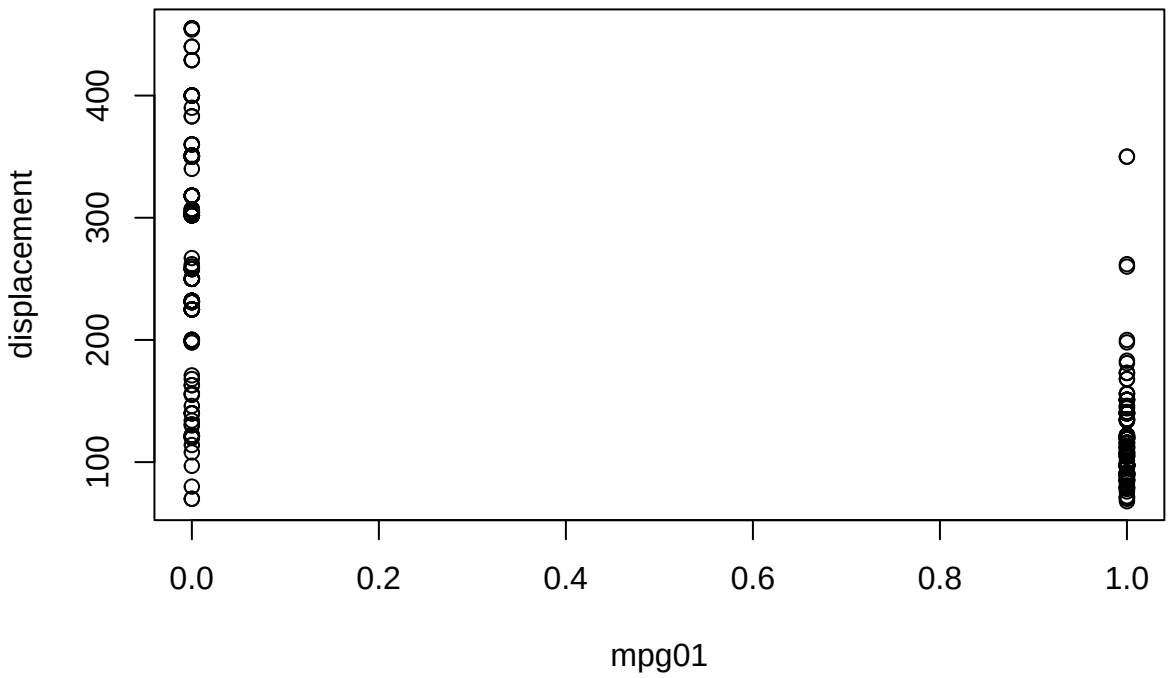

```
varlist <- names(Auto)[2:8]

plot(cylinders~mpg01)
```



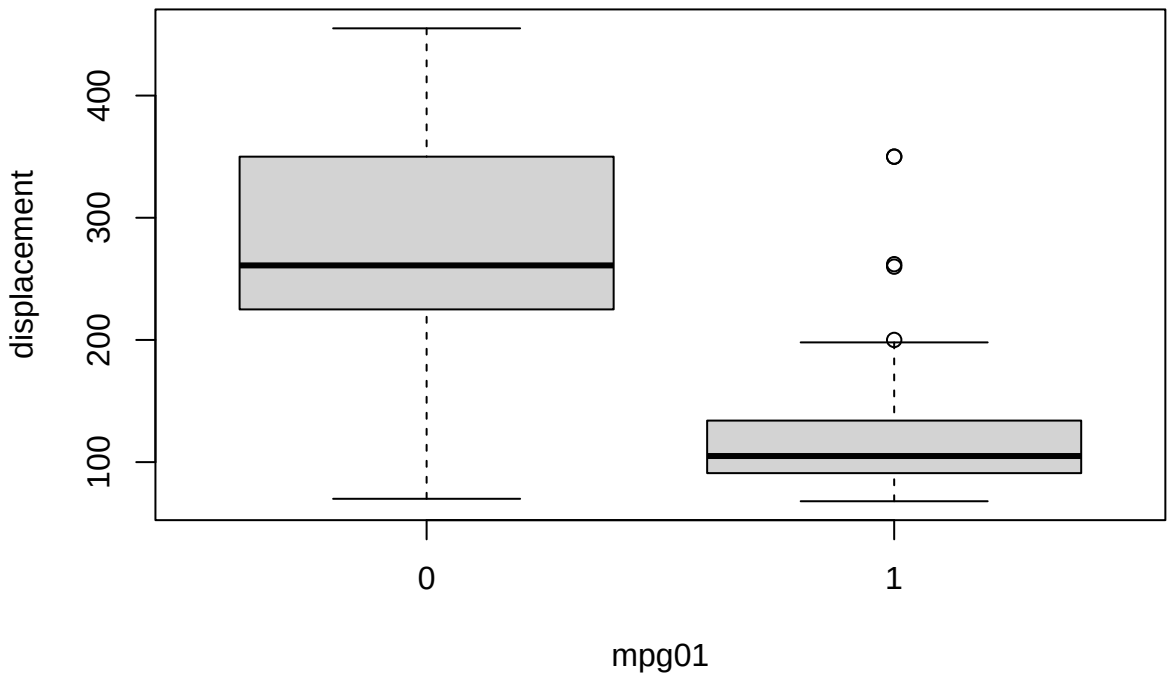

```
boxplot(cylinders~mpg01)
```



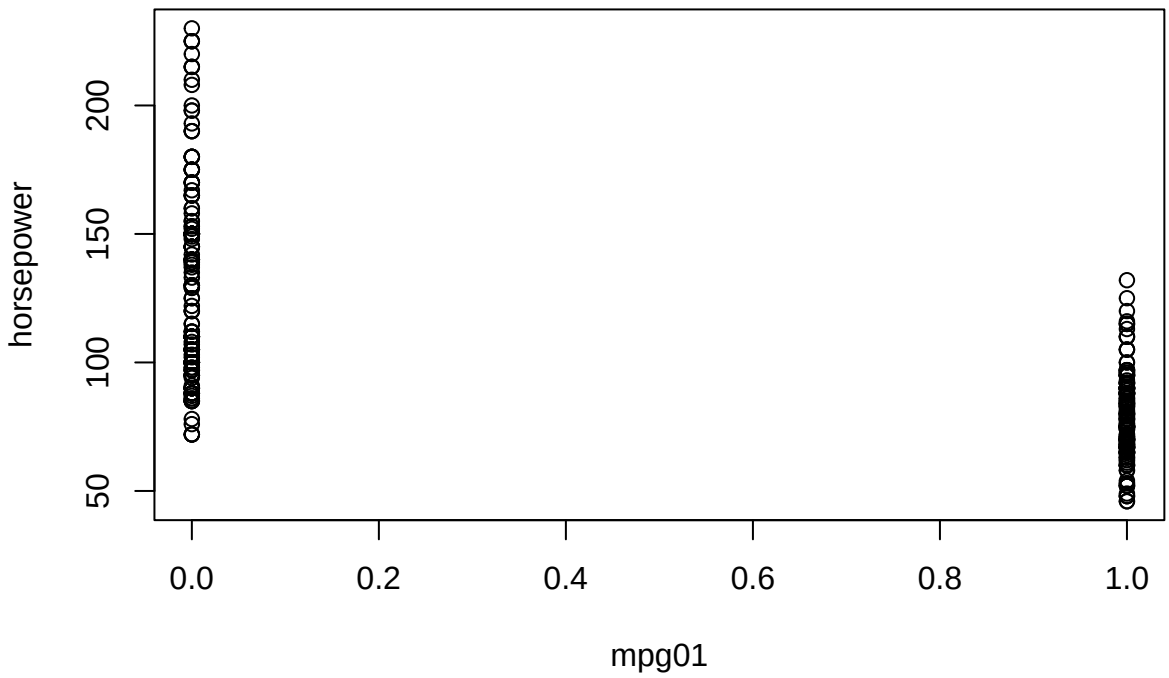

```
plot(displacement~mpg01)
```



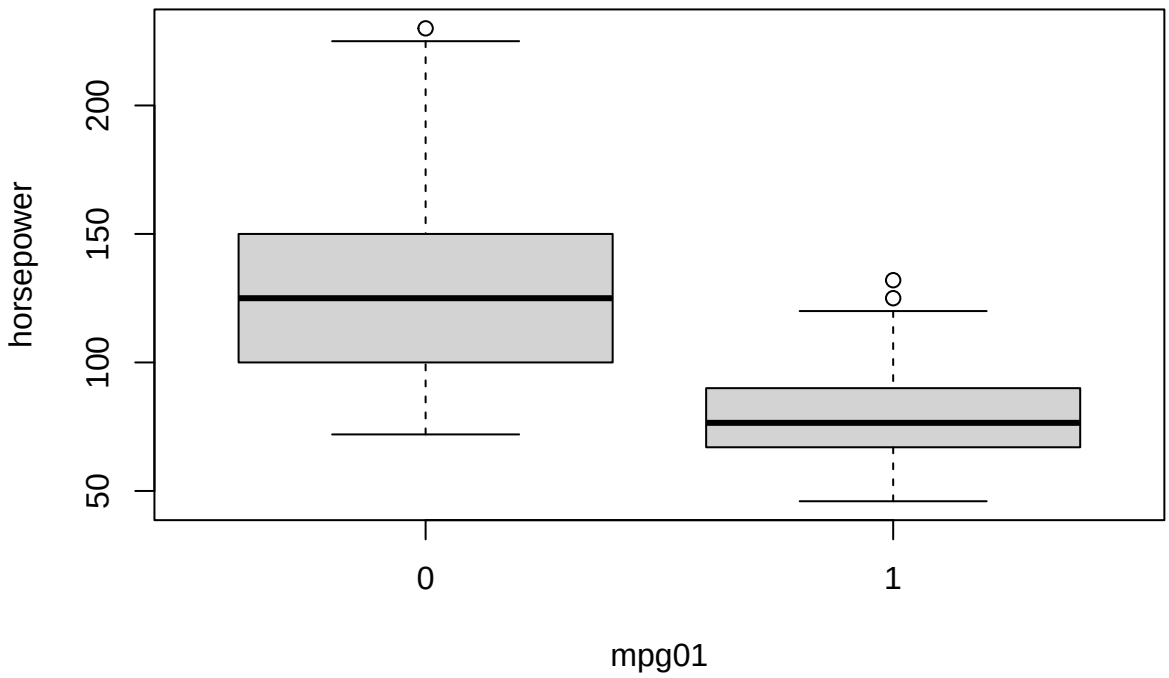

```
boxplot(displacement~mpg01)
```



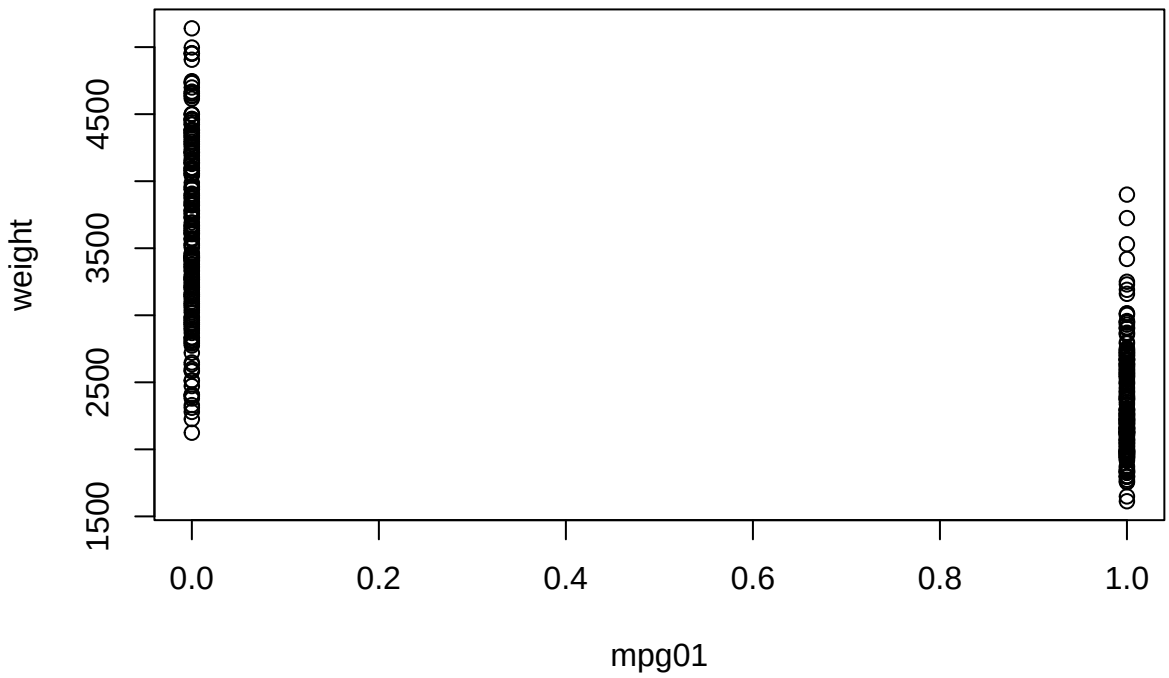

```
plot(horsepower~mpg01)
```



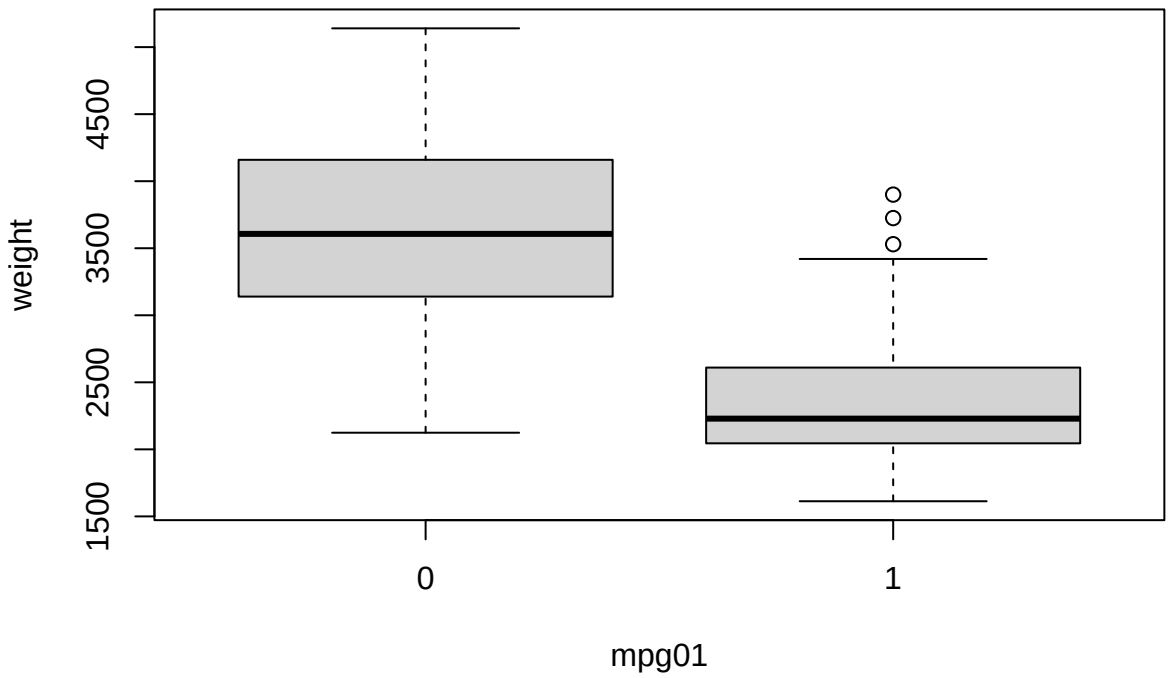

```
boxplot(horsepower~mpg01)
```



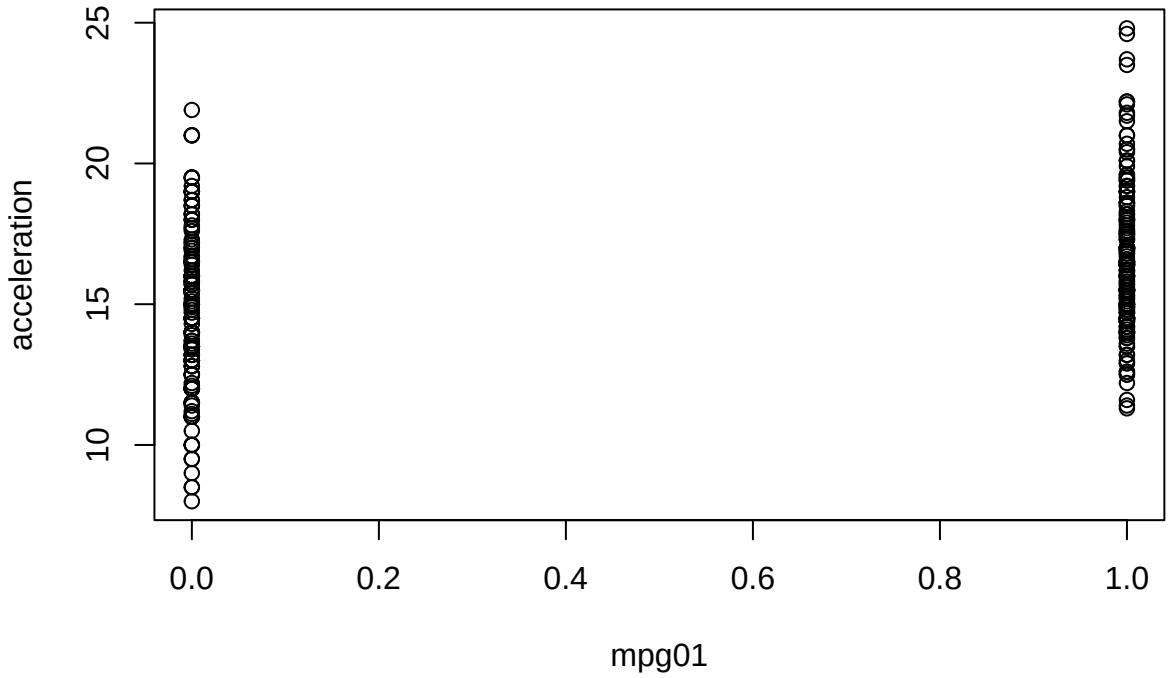

```
plot(weight~mpg01)
```



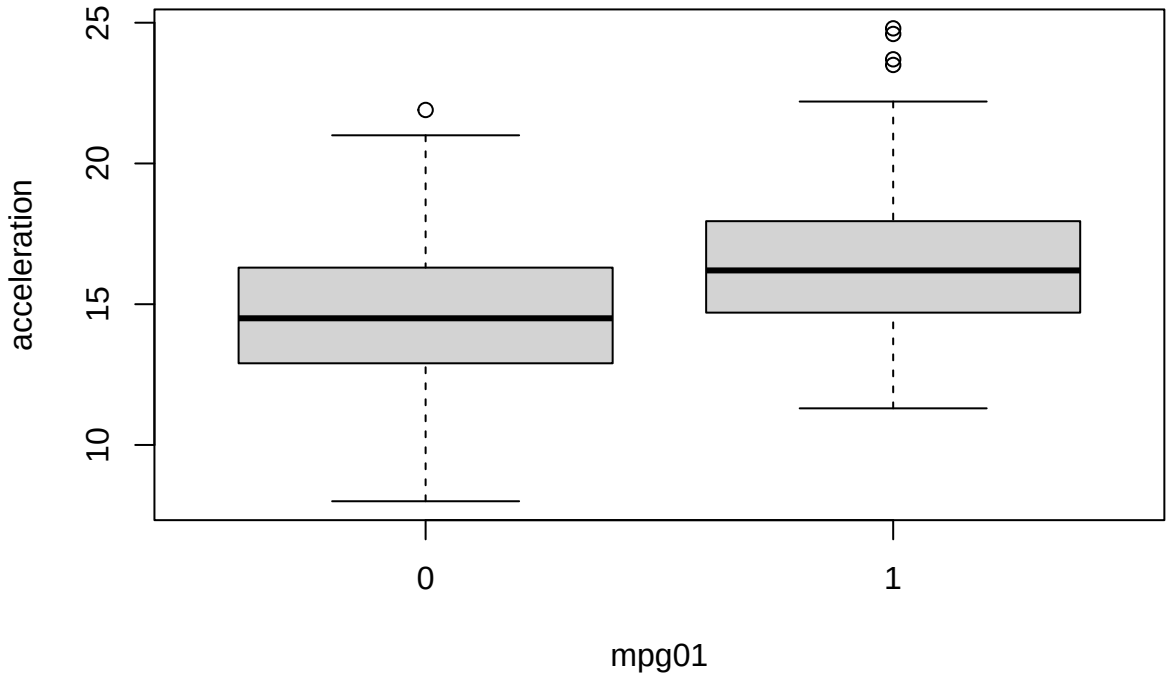

```
boxplot(weight~mpg01)
```



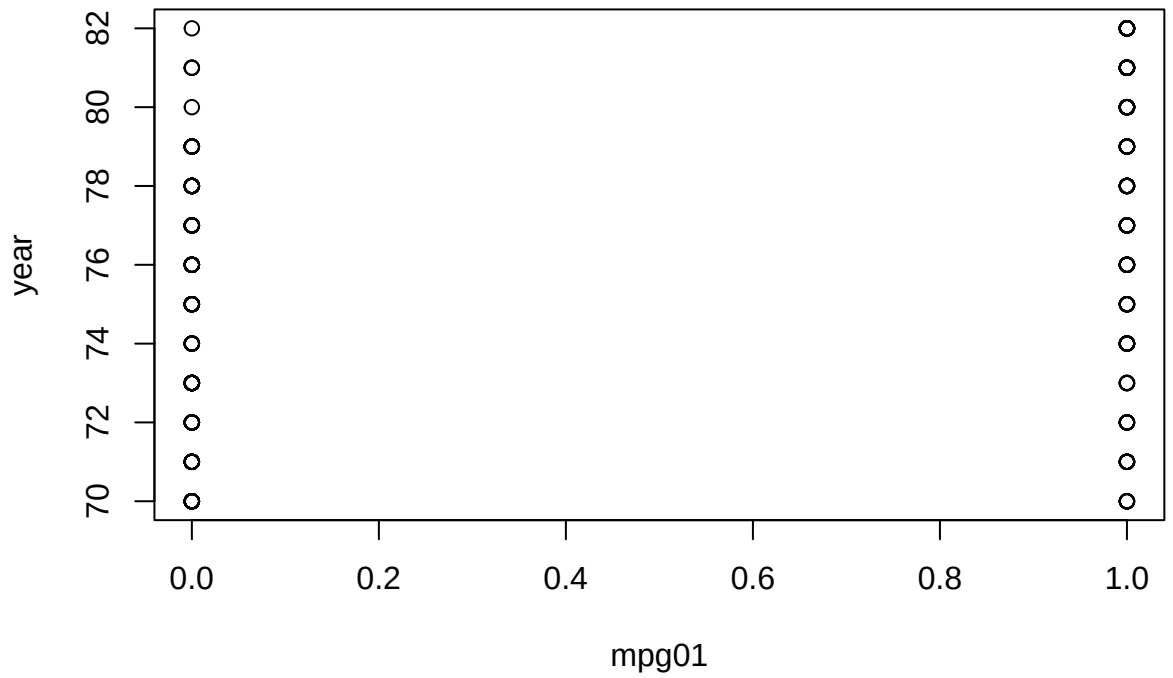

```
plot(acceleration~mpg01)
```



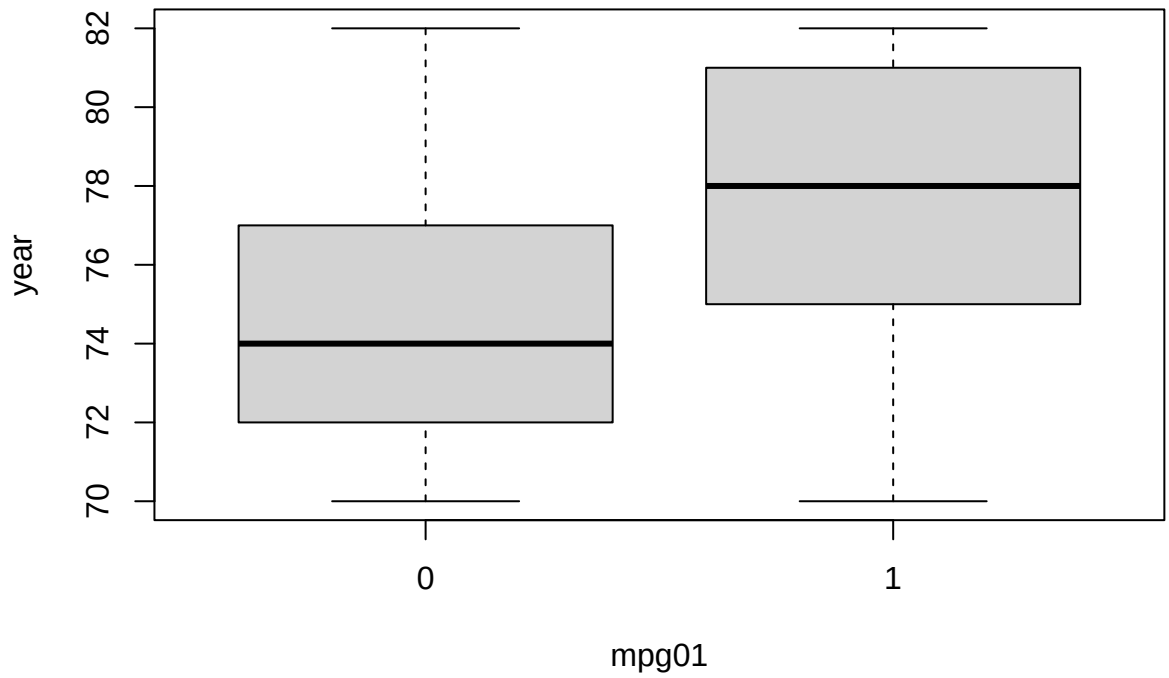

```
boxplot(acceleration~mpg01)
```



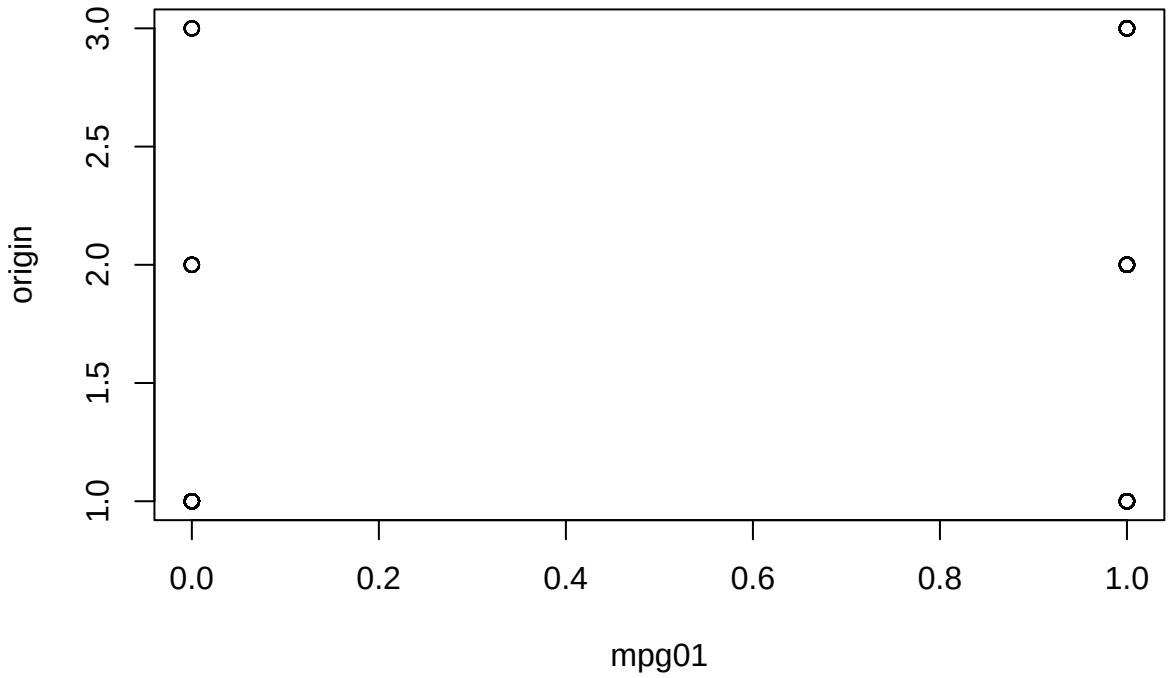

```
plot(year~mpg01)
```



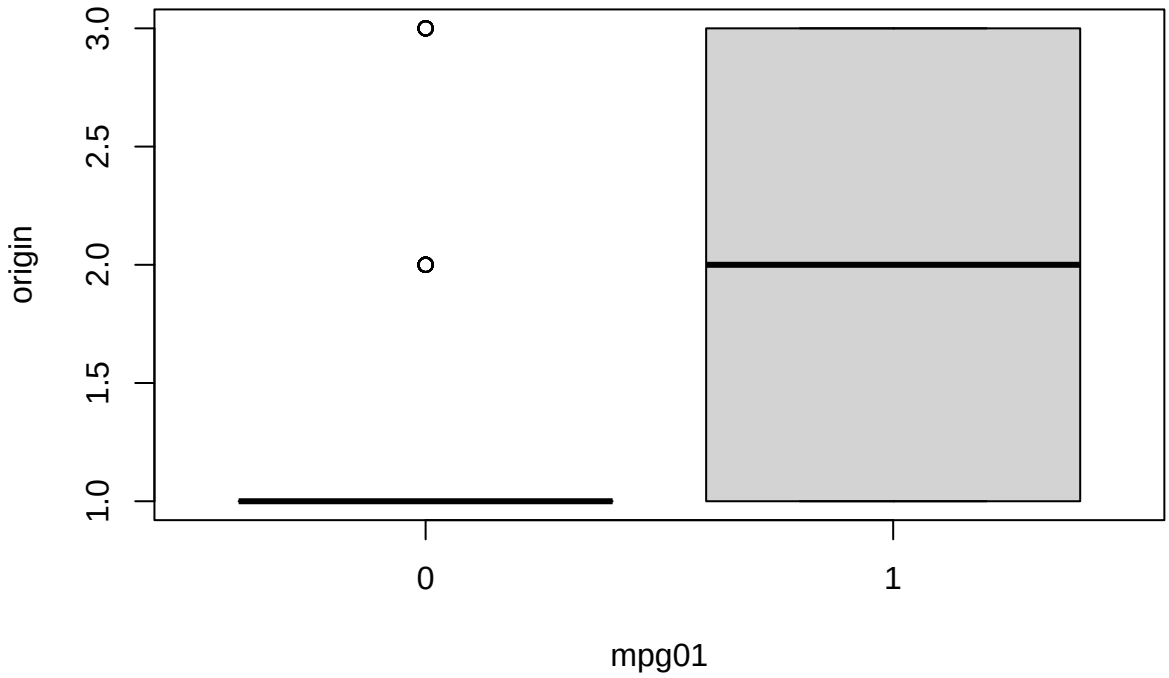

```
boxplot(year~mpg01)
```




```
plot(origin~mpg01)
```




```
boxplot(origin~mpg01)
```



Answer

The graphs showed that cylinders, weight, displacement, and horsepower may be associated with mpg01 thus likely to be useful in predicting mpg01.

(c) Split the data into a training set and a test set

```
training1 <- (year%%2 == 0)
training2 <- Auto[training1,]
test2 <- Auto[!training1,]
test.mpg01 <- mpg01[!training1]
```

(d) Perform LDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
lda.auto <- lda(mpg01~cylinders+weight+displacement+horsepower, subset = training1)
lda.auto
```

```
## Call:
## lda(mpg01 ~ cylinders + weight + displacement + horsepower, subset = training1)
##
## Prior probabilities of groups:
##         0         1
## 0.4571429 0.5428571
##
```

```
## Group means:
##   cylinders   weight displacement horsepower
## 0  6.812500 3604.823     271.7396  133.14583
## 1  4.070175 2314.763     111.6623   77.92105
##
## Coefficients of linear discriminants:
##                        LD1
## cylinders    -0.6741402638
## weight       -0.0011465750
## displacement  0.0004481325
## horsepower    0.0059035377
```

```
lda.predauto <- predict(lda.auto, test2)
table(lda.predauto$class, test.mpg01)
```

```
##    test.mpg01
##      0  1
##   0 86  9
##   1 14 73
```

```
mean(lda.predauto$class != test.mpg01)
```

```
## [1] 0.1263736
```

Answer

The test error of the model was 12.64%.

(e) Perform QDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
qda.auto <- qda(mpg01~cylinders+weight+displacement+horsepower, subset = training1)
qda.auto
```

```
## Call:
## qda(mpg01 ~ cylinders + weight + displacement + horsepower, subset = training1)
##
## Prior probabilities of groups:
##         0         1
## 0.4571429 0.5428571
##
## Group means:
##   cylinders   weight displacement horsepower
## 0  6.812500 3604.823     271.7396  133.14583
## 1  4.070175 2314.763     111.6623   77.92105
```

```
qda.predauto <-predict(qda.auto, test2)
table(qda.predauto$class, test.mpg01)
```

```
##    test.mpg01
##      0  1
##   0 89 13
##   1 11 69
```

```
mean(qda.predauto$class != test.mpg01)
```

```
## [1] 0.1318681
```

Answer

The test error of the model was 13.19%.

(f) Perform logistic regression on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
logit.auto <-glm(mpg01~cylinders+weight+displacement+horsepower, family = binomial, subset = training1)
summary(logit.auto)
```

```
##
## Call:
## glm(formula = mpg01 ~ cylinders + weight + displacement + horsepower,
##     family = binomial, subset = training1)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.48027  -0.03413   0.10583   0.29634   2.57584
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  17.658730   3.409012   5.180 2.22e-07 ***
## cylinders    -1.028032   0.653607  -1.573   0.1158
## weight       -0.002922   0.001137  -2.569   0.0102 *
## displacement  0.002462   0.015030   0.164   0.8699
## horsepower   -0.050611   0.025209  -2.008   0.0447 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 289.58  on 209  degrees of freedom
## Residual deviance:  83.24  on 205  degrees of freedom
## AIC: 93.24
##
## Number of Fisher Scoring iterations: 7
```

```
probs3 <- predict(logit.auto, test2, type = "response")
pred.auto <- rep(0, length(probs3))
pred.auto[probs3 > 0.5] <- 1
table(pred.auto, test.mpg01)
```

```
##          test.mpg01
## pred.auto  0  1
##         0 89 11
##         1 11 71
```

```
mean(pred.auto != test.mpg01)
```

```
## [1] 0.1208791
```

Answer

The test error of the model was 12.09%.

(g) Perform KNN on the training data, with several values of K, in order to predict "mpg01" using the variables that seemed most associated with "mpg01" in (b). What test errors do you obtain? Which value of K seems to perform the best on this data set ?

```
# k=1
set.seed(1)
training.x1 <- cbind(cylinders, weight, displacement, horsepower)[training1, ]
test.x1 <- cbind(cylinders, weight, displacement, horsepower)[!training1, ]
training.mpg <- mpg01[training1]
pred.knn1 <- knn(training.x1, test.x1, training.mpg, k=1)
table(pred.knn1, test.mpg01)
```

```
##          test.mpg01
## pred.knn1  0  1
##         0 83 11
##         1 17 71
```

```
knn1 = mean(pred.knn1 != test.mpg01)

# k = 3
set.seed(1)
pred.knn.3 <- knn(training.x1, test.x1, training.mpg, k=3)
table(pred.knn.3, test.mpg01)
```

```
##           test.mpg01
## pred.knn.3  0  1
##          0 84  9
##          1 16 73
```

```
knn3 = mean(pred.knn.3 != test.mpg01)

# k = 10
set.seed(1)
pred.knn.10 <- knn(training.x1, test.x1, training.mpg, k=10)
table(pred.knn.10, test.mpg01)
```

```
##            test.mpg01
## pred.knn.10  0  1
##           0 79  7
##           1 21 75
```

```r
knn10 = mean(pred.knn.10 != test.mpg01)

# k = 100
set.seed(1)
pred.knn.100 <- knn(training.x1, test.x1, training.mpg, k=100)
table(pred.knn.100, test.mpg01)
```

```
##              test.mpg01
## pred.knn.100  0  1
##             0 81  7
##             1 19 75
```

```r
knn100 = mean(pred.knn.100 != test.mpg01)

print(c(knn1, knn3, knn10, knn100))
```

```
## [1] 0.1538462 0.1373626 0.1538462 0.1428571
```

Answer

The test error rates for k = 1,3,10,100 were 15.38%, 13.74%, 15.38%, 14.29% respectively. Therefore, k value of 3 seems to perform the best on this data set as it has the lowest test error rate.