Chapter 3 Linear Regression

```
library(MASS)
library(ISLR2)
```

```
##
## Attaching package: 'ISLR2'

## The following object is masked from 'package:MASS':
##
##     Boston
```

Exercise 1

Statistically, TV and radio are significant predictors but newspaper. This means newspaper has no effect (no association) with Sales. We can reject the null hypothesis coefficients of TV and radio being 0 and we fail to reject the null hypothesis coefficient of newspaper being 0.

Exercise 2

KNN classifier outputs discrete values (labels) whereas KNN regression outputs continuous values.

KNN classifier makes decisions based on the conditional probability for a specific class from K training observations. KNN regression, however, averages the K nearest training observations.

Exercise 3

Starting salary after graduation = 50 + 20 * GPA + 0.07 * IQ + 35 * Level + 0.01 * GPA * IQ - 10 * GPA * Level (1)

a

Since IQ and GPA are fixed, we can let them be 100 and 3. Plug these numbers into (1), we have:

Salary = 120 + 35 * Level - 10 * GPA * Level = 120 + Level * (35 - 10 * GPA)

High school graduate salary - College graduate salary = 0 - 1 * (35 - 10 * GPA) = 10 * GPA - 35

Therefore, we cannot say whether statement i, ii is true or false.

For statement iii, if GPA is high enough (greater than 3.5), it is true.

For statement iv, if GPA is high enough (greater than 3.5), it is false.

b

Predicted salary = 50 + 20 * 4.0 + 0.07 * 110 + 35 * 1 + 0.01 * 4 * 110 - 10 * 4 * 1 = $ 137.1K

c

False since the coefficient does not tell us how large the effect is. (It is the p-value of the coefficient).

Exercise 4

a

We would expect the cubic regression to have a lower training RSS than the linear regression because it could make a tighter fit

b

The cubic regression is likely to overfit the test data so we expect that the RSS of linear regression to be lower.

c

We expect that the RSS of linear regression to be lower.

d

It depends on how far the true relationship between X and Y is from linear. The further the it is from linear, the lower the RSS of the cubic regression (relative to the RSS of the linear regression).

Exercise 5

If we just plug

$$\widehat{\beta}$$

into the fitted value equation, we get:

$$\widehat{y}_i = x_i \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i'=1}^{n} x_{i'}^2}$$

However, this could be problematic due to the "i" inside and outside the sum symbol. Instead, we can rewrite:

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

and,

$$\widehat{\beta} = \frac{\sum_{i=1}^{n} x_{i'} y_{i'}}{\sum_{j=1}^{n} x_j^2}$$

Plugging

$$\widehat{\beta}$$

again into the fitted value equation:

$$\widehat{y_i} = x_i \frac{\sum_{i'=1}^{n} x_{i'} y_{i'}}{\sum_{j=1}^{n} x_j^2}$$

$$\Leftrightarrow \sum_{i'=1}^{n} \frac{x_{i'} x_i}{\sum_{j=1}^{n} x_j} y_{i'}$$

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

with

$$a_{i'} = \frac{x_{i'} x_i}{\sum_{j=1}^{n} x_j}$$

Exercise 6

From (3.2), we have the least square line equation:

$$\widehat{y} = \widehat{\beta_0} + \widehat{\beta_1} x$$

And from (3.4), we have:

$$\widehat{\beta_0} = \overline{y} - \widehat{\beta_1} \overline{x}$$

Plug

$$\widehat{\beta_0}$$

into (3.2):

$$\widehat{y} = \widehat{\beta_0} + \widehat{\beta_1} \overline{x}$$

$$\Leftrightarrow \widehat{y} = \widehat{\beta_0} + \overline{y} - \widehat{\beta_0} = \overline{y}$$

Hence, in the case of simple linear regression, the least squares line always passes through the point (

$$\overline{x}, \overline{y}$$

)

Exercise 7

On the one hand,

$$R^2 = \frac{\sum_{i=1}^{n} \left(\hat{y}_i - \bar{y}\right)^2}{\sum_{i=1}^{n} \left(y_i - \bar{y}\right)^2}$$

$$= \frac{\sum_{i=1}^{n} \left(\hat{\beta}_0 + \hat{\beta}_1 x_i - \bar{y}\right)^2}{\sum_{i=1}^{n} \left(y_i - \bar{y}\right)^2}$$

$$= \frac{\sum_{i=1}^{n} \left(\bar{y} - \hat{\beta}_1 \bar{x} + \hat{\beta}_1 x_i - \bar{y}\right)^2}{\sum_{i=1}^{n} \left(y_i - \bar{y}\right)^2}$$

$$= \frac{\sum_{i=1}^{n} \left(\hat{\beta}_1 \left(x_i - \bar{x}\right)\right)^2}{\sum_{i=1}^{n} \left(y_i - \bar{y}\right)^2}$$

$$= \hat{\beta}_1^2 \frac{\frac{1}{n-1} \sum_{i=1}^{n} \left(x_i - \bar{x}\right)^2}{\frac{1}{n-1} \sum_{i=1}^{n} \left(y_i - \bar{y}\right)^2}$$

$$= \hat{\beta}_1^2 \frac{\sigma_x^2}{\sigma_y^2}$$

On the other hand,

$$Cor(X, Y) = \frac{Cov(X, Y)}{\sigma_x \sigma_y}$$

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

Also,

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} \left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)}{\sum_{i=1}^{n} \left(x_i - \bar{x}\right)^2}$$

$$\hat{\beta}_1 = \frac{\dfrac{1}{n-1} \sum_{i=1}^{n} \left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)}{\dfrac{1}{n-1} \sum_{i=1}^{n} \left(x_i - \bar{x}\right)^2}$$

$$\hat{\beta}_1 = \frac{\sigma_{xy}}{\sigma_x^2}$$

Therefore, we can write the correlation as

$$\rho_{xy} = \frac{\hat{\beta}_1 \sigma_x}{\sigma_y}$$

$$\rho_{xy}^2 = \hat{\beta}_1^2 \frac{\sigma_x^2}{\sigma_y^2} = R^2$$

Exercise 8

a

```
head(Auto)
```

```
##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307        130   3504         12.0   70      1
## 2  15         8          350        165   3693         11.5   70      1
## 3  18         8          318        150   3436         11.0   70      1
## 4  16         8          304        150   3433         12.0   70      1
## 5  17         8          302        140   3449         10.5   70      1
## 6  15         8          429        198   4341         10.0   70      1
##                        name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500
```

The result suggests that there is a relationship between horsepower and mpg. The relationship is moderately strong and negative.

```
attach(Auto)
eighta_model = lm(mpg ~ horsepower, data = Auto)
summary(eighta_model)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

```
newdata = data.frame(horsepower = 98)
predict(eighta_model, newdata = newdata, interval = 'confidence')
```

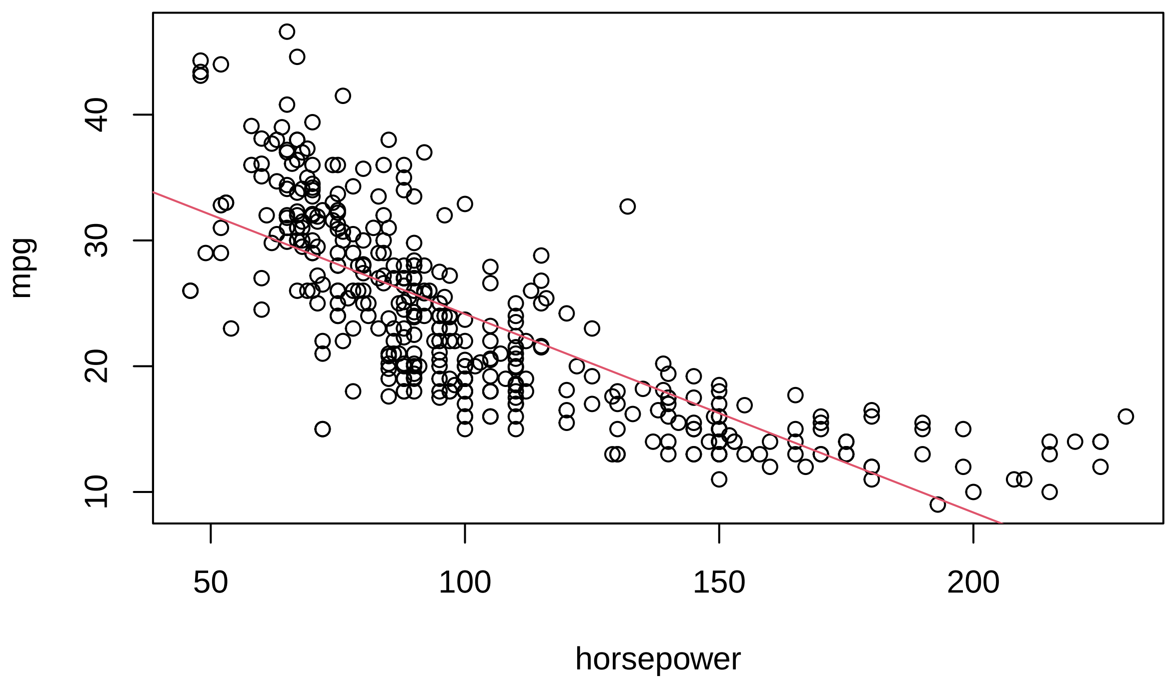
```
##        fit      lwr      upr
## 1 24.46708 23.97308 24.96108
```

```r
predict(eighta_model, newdata = newdata, interval = 'prediction')
```

```
##        fit     lwr      upr
## 1 24.46708 14.8094 34.12476
```

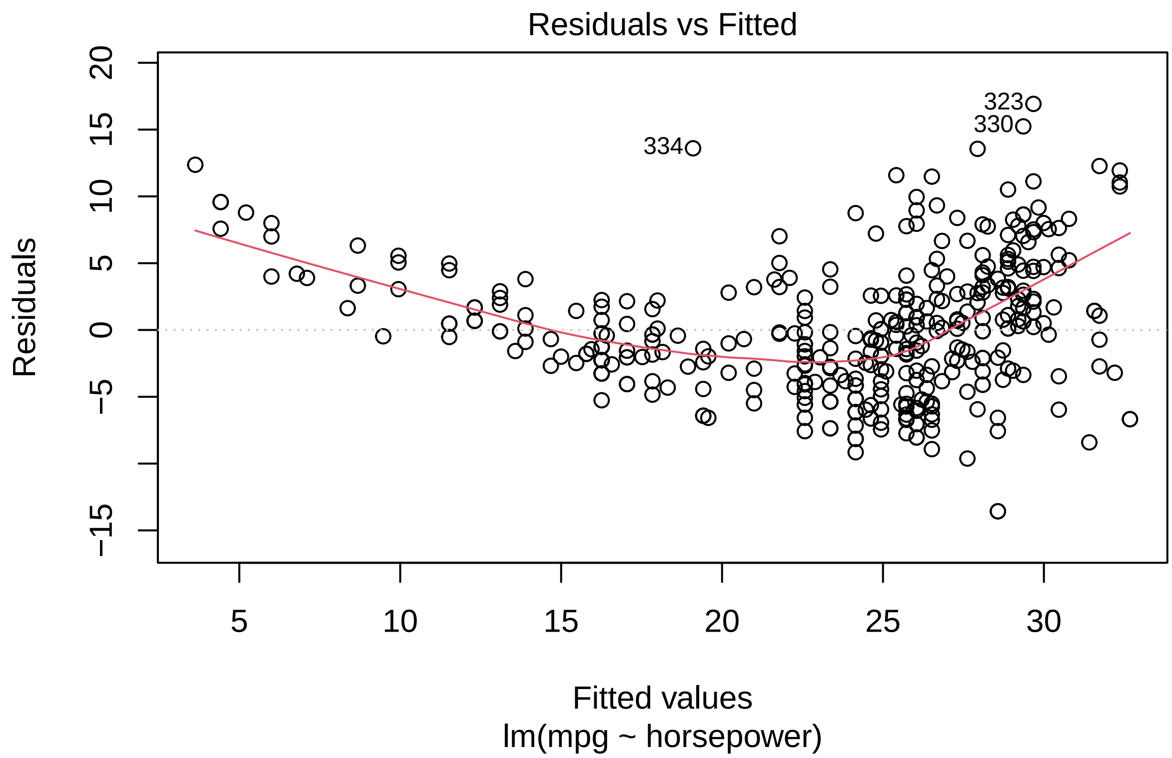
b

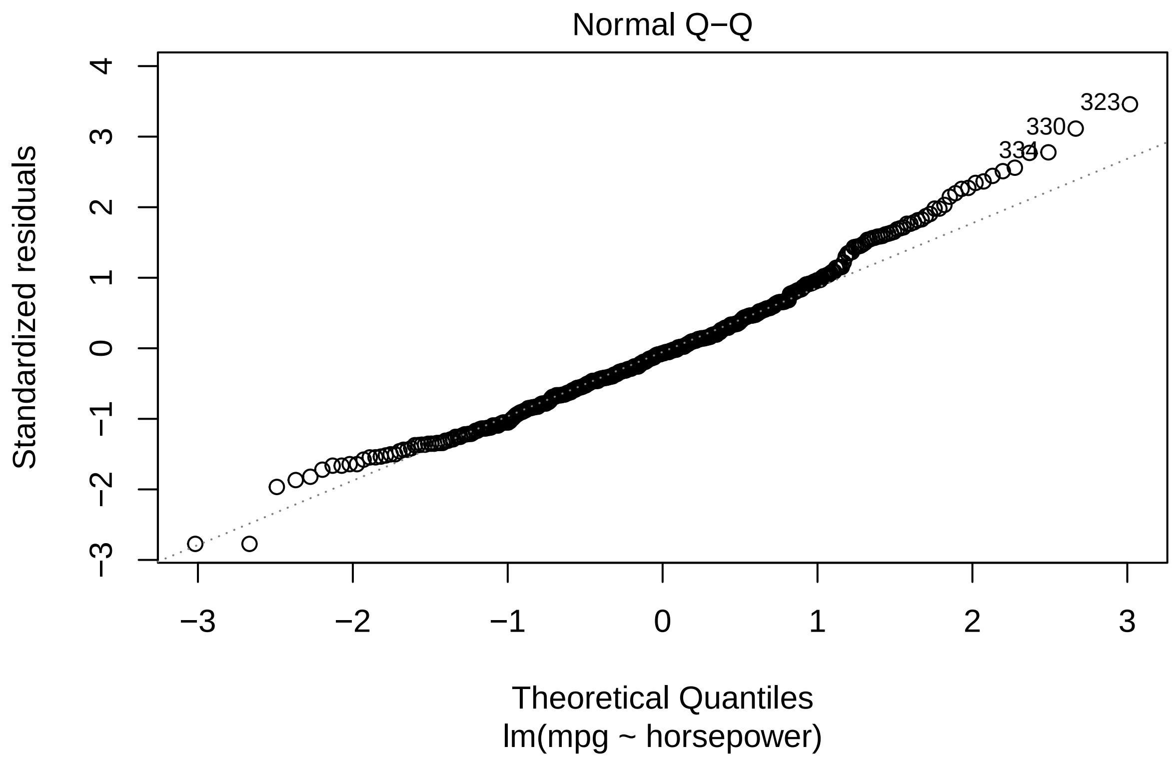
```r
plot(horsepower, mpg)
abline(eighta_model, col = 2)
```

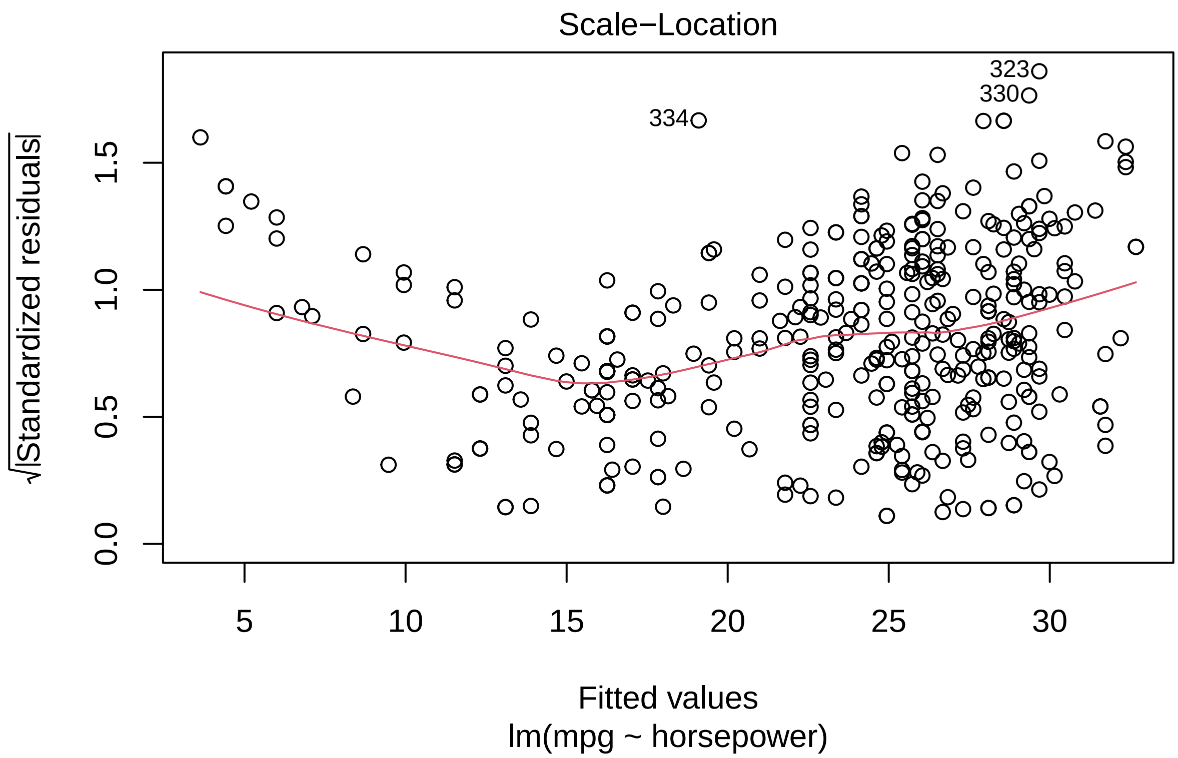


c

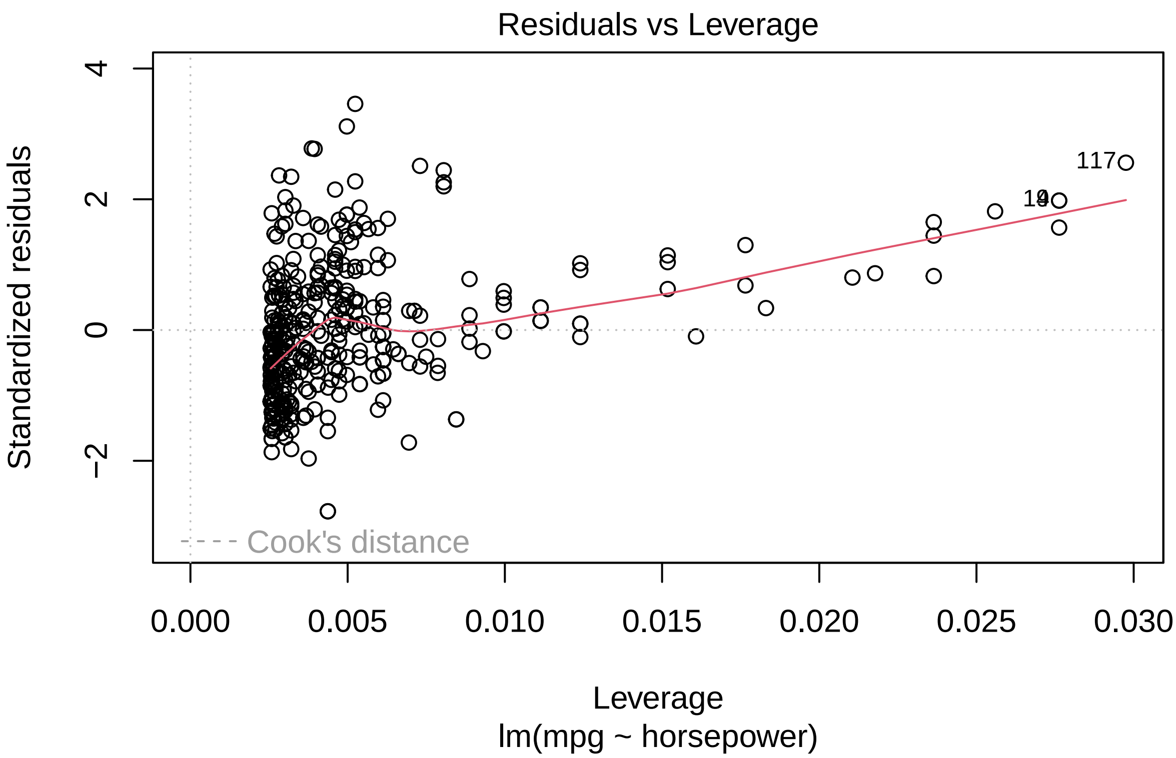
```r
plot(eighta_model)
```


Residuals vs Fitted
Residuals
Fitted values
lm(mpg ~ horsepower)
334
323
330
−15
−5
0
5
10
15
20
25
30


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(mpg ~ horsepower)
323
330
334
−3
−2
−1
0
1
2
3
4



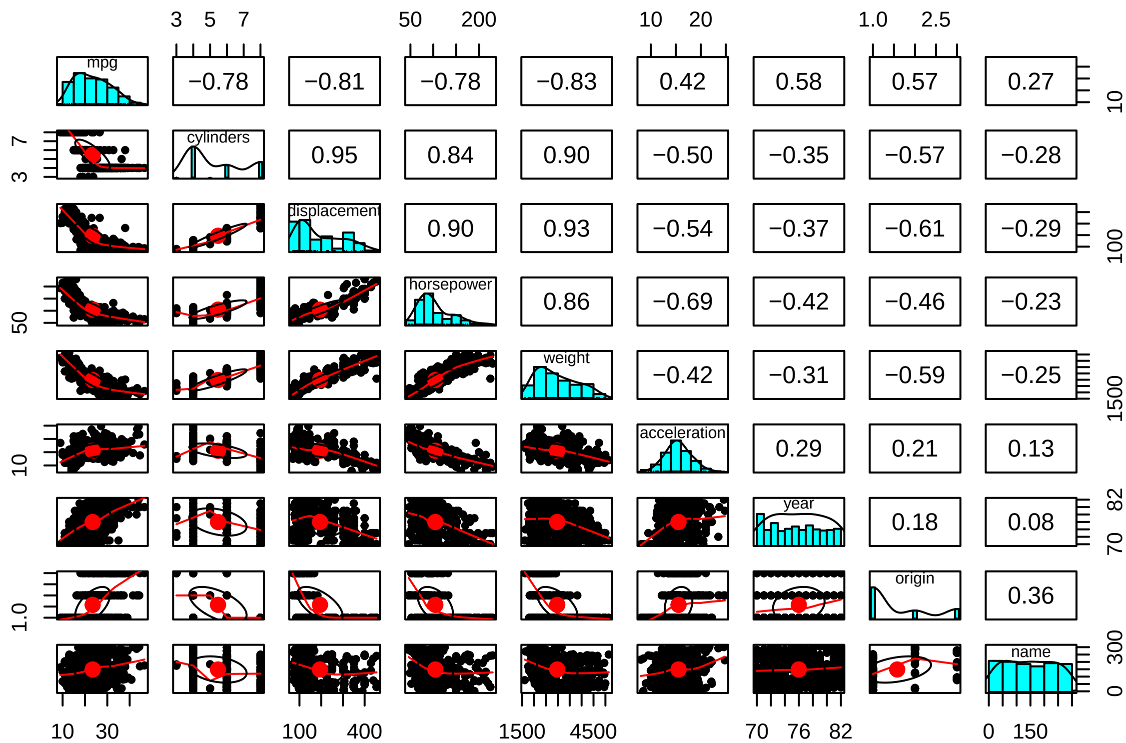
Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
5
10
15
20
25
30
334
323
330
Fitted values
lm(mpg ~ horsepower)



Exercise 9

a

```
library(psych)
pairs.panels(Auto)
```




```
colnames(Auto)
```

```
## [1] "mpg"          "cylinders"    "displacement" "horsepower"   "weight"
## [6] "acceleration" "year"         "origin"       "name"
```

b

```
data.frame(cor(Auto[, 1:8]))
```

```
##                     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

Predictors displacement, weight, year and origin appear to have statistically significant relationship to the response.

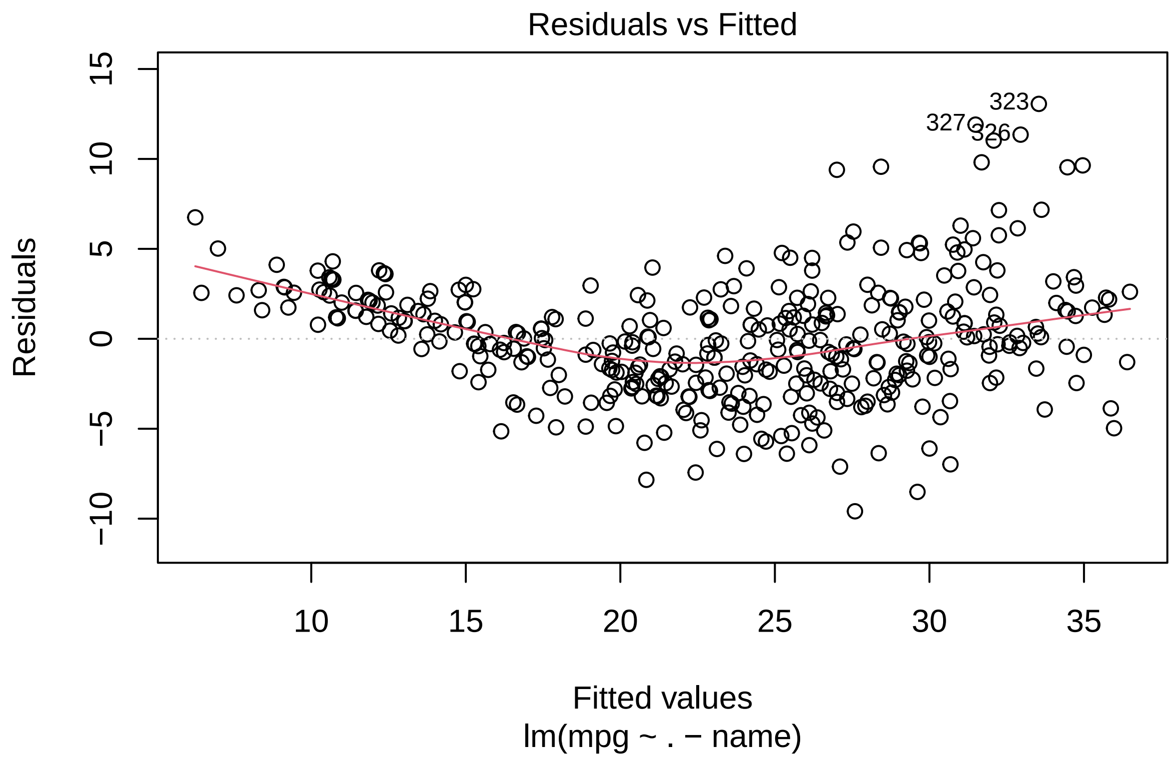
```r
ninec_model = lm(mpg ~ . - name, data = Auto)
summary(ninec_model)
```

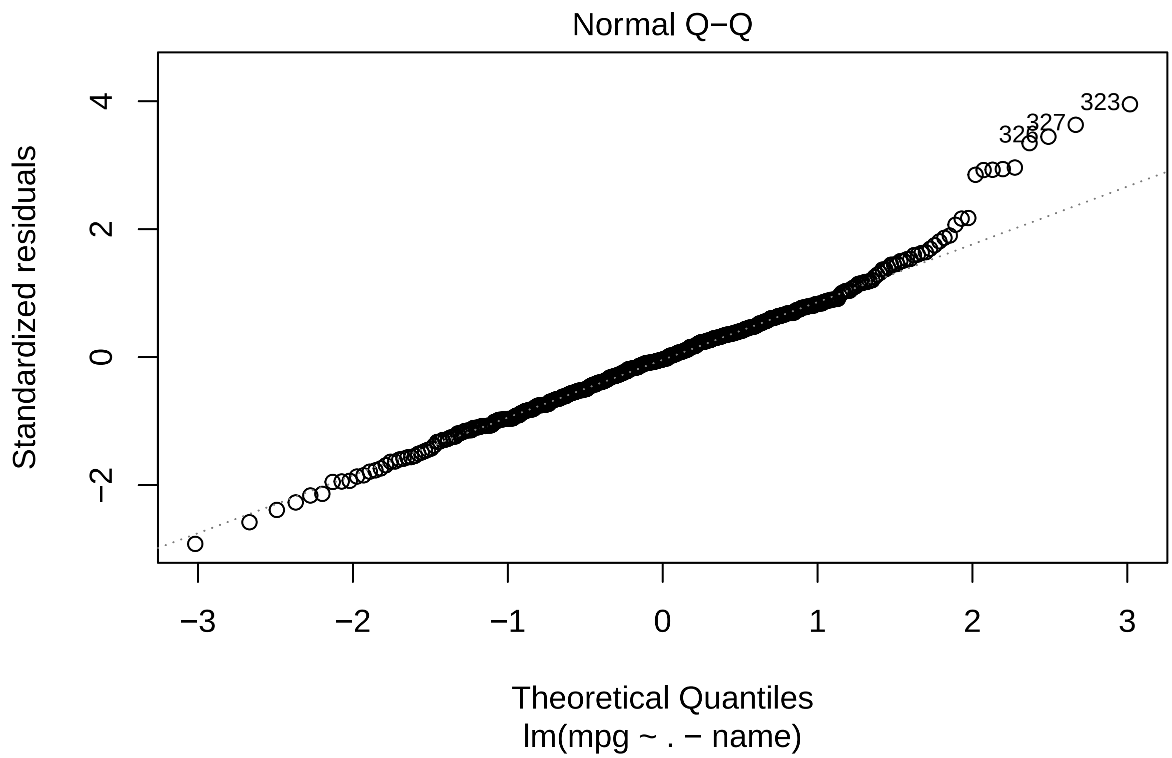
```
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## 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
```

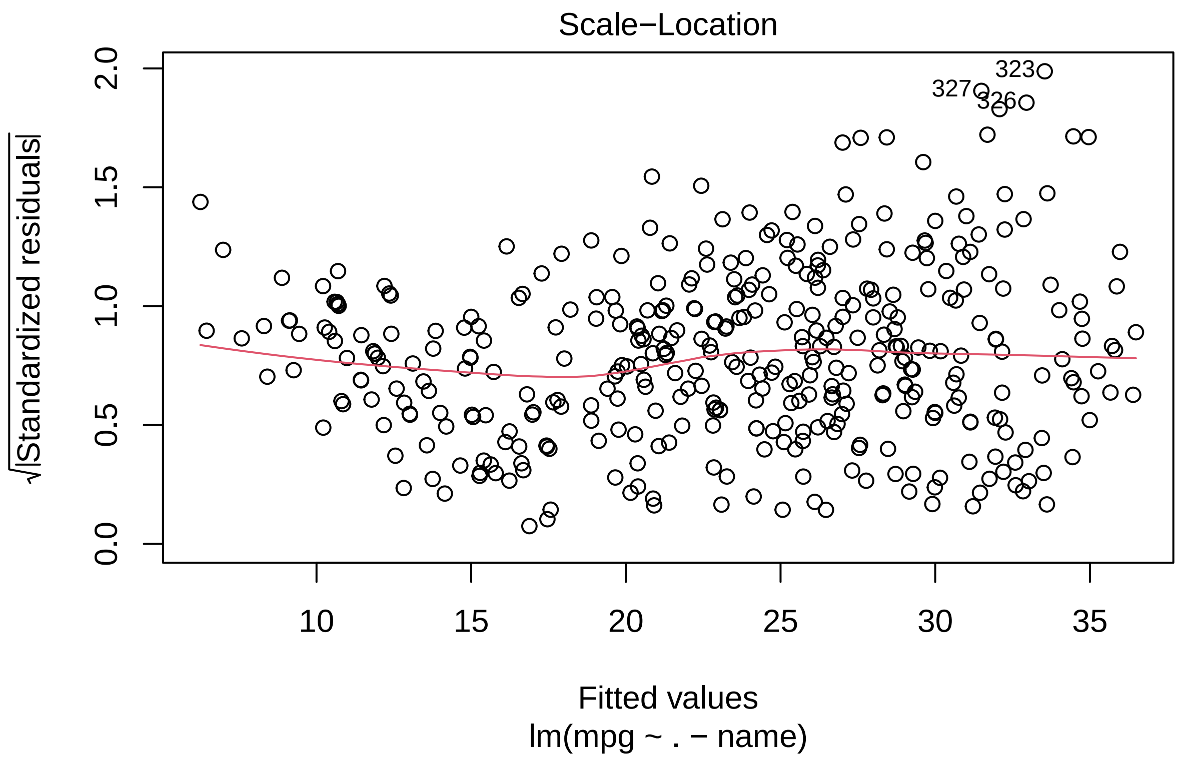
d

Observations 323, 326 and 327 are large outliers. The plot also indentifies the 14th observation as a high leverage.

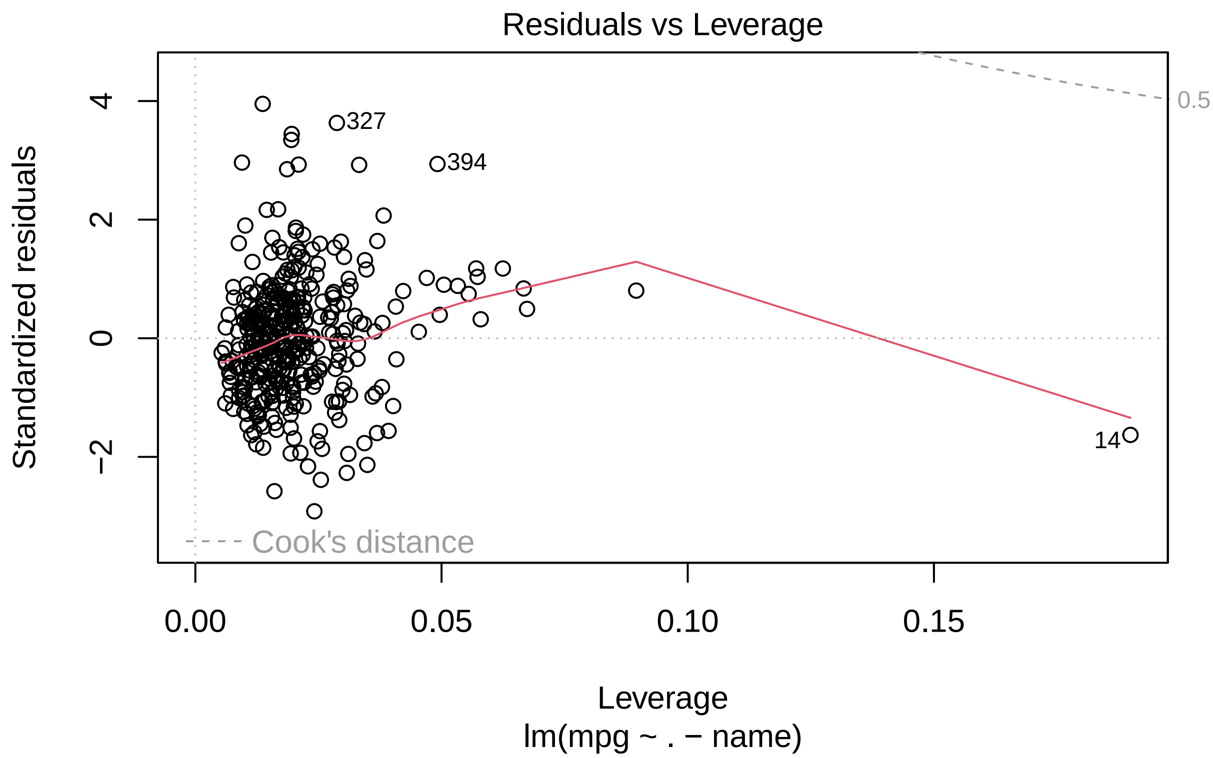
```r
plot(ninec_model)
```


Residuals vs Fitted
Residuals
15
10
5
0
−5
−10
323
327
326
10
15
20
25
30
35
Fitted values
lm(mpg ~ . − name)


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



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



e

Using the output from (c), we can choose displacement, weight and year to fit a model with interaction effects. Displacement * weight appears to be statistically significant as a result.

```
nined_model = lm(mpg ~ displacement * weight + displacement * year +
                   year * weight, data = Auto[, 1:8])
summary(nined_model)
```

```
##
## Call:
## lm(formula = mpg ~ displacement * weight + displacement * year +
##     year * weight, data = Auto[, 1:8])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7299 -1.6773 -0.0834  1.2071 13.5557
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -4.250e+01  1.889e+01  -2.250    0.025 *
## displacement         4.021e-02  7.820e-02   0.514    0.607
## weight              -4.230e-03  1.086e-02  -0.390    0.697
## year                 1.269e+00  2.438e-01   5.205 3.17e-07 ***
```

```
## displacement:weight  1.880e-05  2.319e-06   8.107 7.00e-15 ***
## displacement:year   -1.455e-03  1.070e-03  -1.359    0.175
## weight:year         -7.481e-05  1.429e-04  -0.524    0.601
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.958 on 385 degrees of freedom
## Multiple R-squared:  0.8586, Adjusted R-squared:  0.8564
## F-statistic: 389.6 on 6 and 385 DF,  p-value: < 2.2e-16
```

f

In general, the more features are added to the model, the higher the performance.

```
ninef_model_one = lm(mpg ~ . + log(horsepower) - name - horsepower, data = Auto)
summary(ninef_model_one)
```

```
##
## Call:
## lm(formula = mpg ~ . + log(horsepower) - name - horsepower, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3115 -2.0041 -0.1726  1.8393 12.6579
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     27.254005   8.589614   3.173  0.00163 **
## cylinders       -0.486206   0.306692  -1.585  0.11372
## displacement     0.019456   0.006876   2.830  0.00491 **
## weight          -0.004266   0.000694  -6.148 1.97e-09 ***
## acceleration    -0.292088   0.103804  -2.814  0.00515 **
## year             0.705329   0.048456  14.556  < 2e-16 ***
## origin           1.482435   0.259347   5.716 2.19e-08 ***
## log(horsepower) -9.506436   1.539619  -6.175 1.69e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.18 on 384 degrees of freedom
## Multiple R-squared:  0.837,  Adjusted R-squared:  0.834
## F-statistic: 281.6 on 7 and 384 DF,  p-value: < 2.2e-16
```

```
ninef_model_two = lm(mpg ~ . + log(horsepower) + log(weight) + log(displacement)-
                       name - horsepower, data = Auto)
summary(ninef_model_two)
```

```
##
## Call:
## lm(formula = mpg ~ . + log(horsepower) + log(weight) + log(displacement) -
##     name - horsepower, data = Auto)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -10.4364  -1.5591  -0.1519   1.5100  12.1084
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       191.862207  40.191389   4.774 2.58e-06 ***
## cylinders          -0.236825   0.300269  -0.789  0.43077
## displacement        0.036134   0.012025   3.005  0.00283 **
## weight              0.003112   0.002195   1.418  0.15700
## acceleration       -0.141636   0.098865  -1.433  0.15279
## year                0.774565   0.045793  16.915  < 2e-16 ***
## origin              0.594202   0.271587   2.188  0.02928 *
## log(horsepower)    -6.386029   1.530397  -4.173 3.73e-05 ***
## log(weight)       -22.626049   7.111200  -3.182  0.00158 **
## log(displacement)  -6.165190   2.639448  -2.336  0.02002 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.949 on 382 degrees of freedom
## Multiple R-squared:  0.8605, Adjusted R-squared:  0.8572
## F-statistic: 261.8 on 9 and 382 DF,  p-value: < 2.2e-16
```

```r
ninef_model_three = lm(mpg ~ . + poly(horsepower, 4) - name, data = Auto)
summary(ninef_model_three)
```

```
##
## Call:
## lm(formula = mpg ~ . + poly(horsepower, 4) - name, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.4965 -1.7419 -0.0375  1.4713 11.8465
##
## Coefficients: (1 not defined because of singularities)
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -1.254e+01  4.170e+00  -3.006 0.002819 **
## cylinders            -6.449e-02  3.361e-01  -0.192 0.847945
## displacement         -3.861e-03  7.333e-03  -0.527 0.598821
## horsepower           -5.406e-02  1.374e-02  -3.935 9.89e-05 ***
## weight               -3.579e-03  6.737e-04  -5.312 1.85e-07 ***
## acceleration         -2.906e-01  9.843e-02  -2.953 0.003347 **
## year                  7.438e-01  4.525e-02  16.438  < 2e-16 ***
## origin                8.817e-01  2.533e-01   3.480 0.000559 ***
## poly(horsepower, 4)1         NA         NA      NA       NA
## poly(horsepower, 4)2  3.363e+01  3.804e+00   8.842  < 2e-16 ***
## poly(horsepower, 4)3 -1.270e+01  3.401e+00  -3.736 0.000216 ***
## poly(horsepower, 4)4 -2.450e+00  3.119e+00  -0.786 0.432595
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.948 on 381 degrees of freedom
## Multiple R-squared:  0.8609, Adjusted R-squared:  0.8573
## F-statistic: 235.9 on 10 and 381 DF,  p-value: < 2.2e-16
```

Exercise 10

a

```r
head(Carseats)
```

```
##   Sales CompPrice Income Advertising Population Price ShelveLoc Age Education
## 1  9.50       138     73          11        276   120       Bad  42        17
## 2 11.22       111     48          16        260    83      Good  65        10
## 3 10.06       113     35          10        269    80    Medium  59        12
## 4  7.40       117    100           4        466    97    Medium  55        14
## 5  4.15       141     64           3        340   128       Bad  38        13
## 6 10.81       124    113          13        501    72       Bad  78        16
##   Urban  US
## 1   Yes Yes
## 2   Yes Yes
## 3   Yes Yes
## 4   Yes Yes
## 5   Yes  No
## 6    No Yes
```

```r
ex_ten_model = lm(Sales ~ Price + Urban + US, data = Carseats)
```

b

If the Price increases by 1 (unit), Sales decreases by 0.05 on average, given the other predictors remain unchanged.

If the person is from urban area, the Sales decreases by 0.05 on average, given the other predictors remain unchanged.

If the person is from the US, the Sales decreases by 0.05 on average, given the other predictors remain unchanged.

c

Sales = 13.04 - 0.05 * Price - 0.02 * UrbanYes + 1.2 * USYes

```r
summary(ex_ten_model)
```

```
##
## Call:
## lm(formula = Sales ~ Price + Urban + US, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9206 -1.6220 -0.0564  1.5786  7.0581
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 13.043469   0.651012  20.036  < 2e-16 ***
## Price       -0.054459   0.005242 -10.389  < 2e-16 ***
## UrbanYes    -0.021916   0.271650  -0.081    0.936
## USYes        1.200573   0.259042   4.635 4.86e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.472 on 396 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2335
## F-statistic: 41.52 on 3 and 396 DF,  p-value: < 2.2e-16
```

d

We can reject the null hypothesis H0: coefficients

$$\beta$$

= 0 for predictors Price and US

e

```r
smaller_model = lm(Sales ~ Price + US, data = Carseats)
```

f

Using RSE (Residual Standard Error) and R-squared as metrics, we can see that the smaller model performs better although this is not significant. Both models explain 23% - 24% the variability in Sales. And the lack of fit is about 2.4 - 2.5

```r
summary(ex_ten_model)
```

```
##
## Call:
## lm(formula = Sales ~ Price + Urban + US, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9206 -1.6220 -0.0564  1.5786  7.0581
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.043469   0.651012  20.036  < 2e-16 ***
## Price       -0.054459   0.005242 -10.389  < 2e-16 ***
## UrbanYes    -0.021916   0.271650  -0.081    0.936
## USYes        1.200573   0.259042   4.635 4.86e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.472 on 396 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2335
## F-statistic: 41.52 on 3 and 396 DF,  p-value: < 2.2e-16
```

```
summary(smaller_model)
```

```
##
## Call:
## lm(formula = Sales ~ Price + US, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9269 -1.6286 -0.0574  1.5766  7.0515
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.03079    0.63098  20.652  < 2e-16 ***
## Price       -0.05448    0.00523 -10.416  < 2e-16 ***
## USYes        1.19964    0.25846   4.641 4.71e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.469 on 397 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2354
## F-statistic: 62.43 on 2 and 397 DF,  p-value: < 2.2e-16
```

g

```
smaller_model_summary = data.frame(summary(smaller_model)$coefficients)
smaller_model_summary$Estimate - 2 * smaller_model_summary$Std..Error
```

```
## [1] 11.76884014 -0.06493788  0.68272089
```

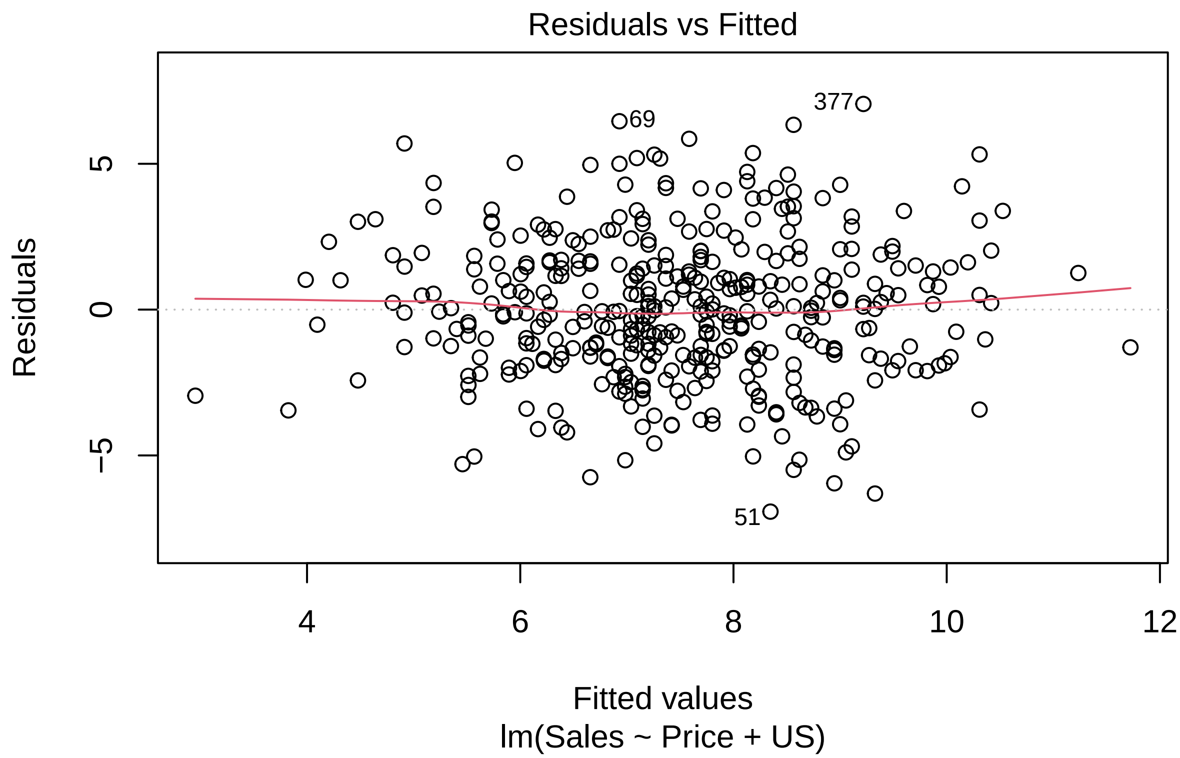
```
confint(smaller_model)
```

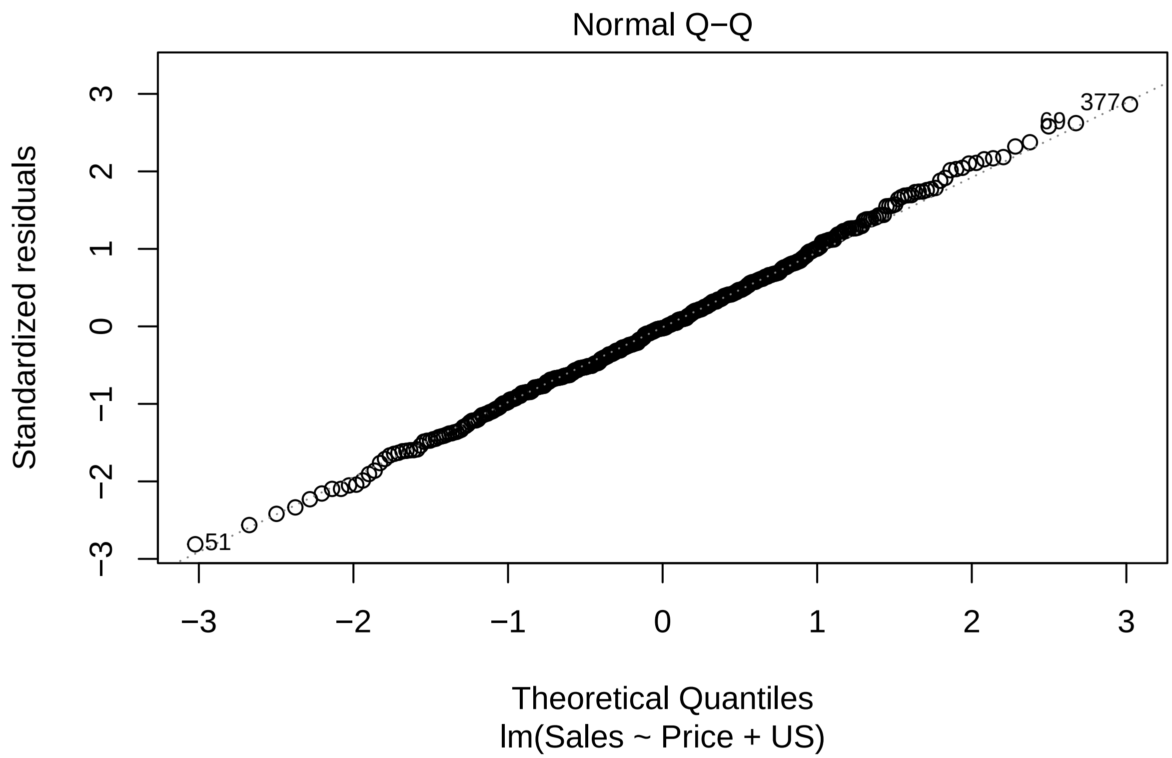
```
##                   2.5 %      97.5 %
## (Intercept) 11.79032020 14.27126531
## Price       -0.06475984 -0.04419543
## USYes        0.69151957  1.70776632
```

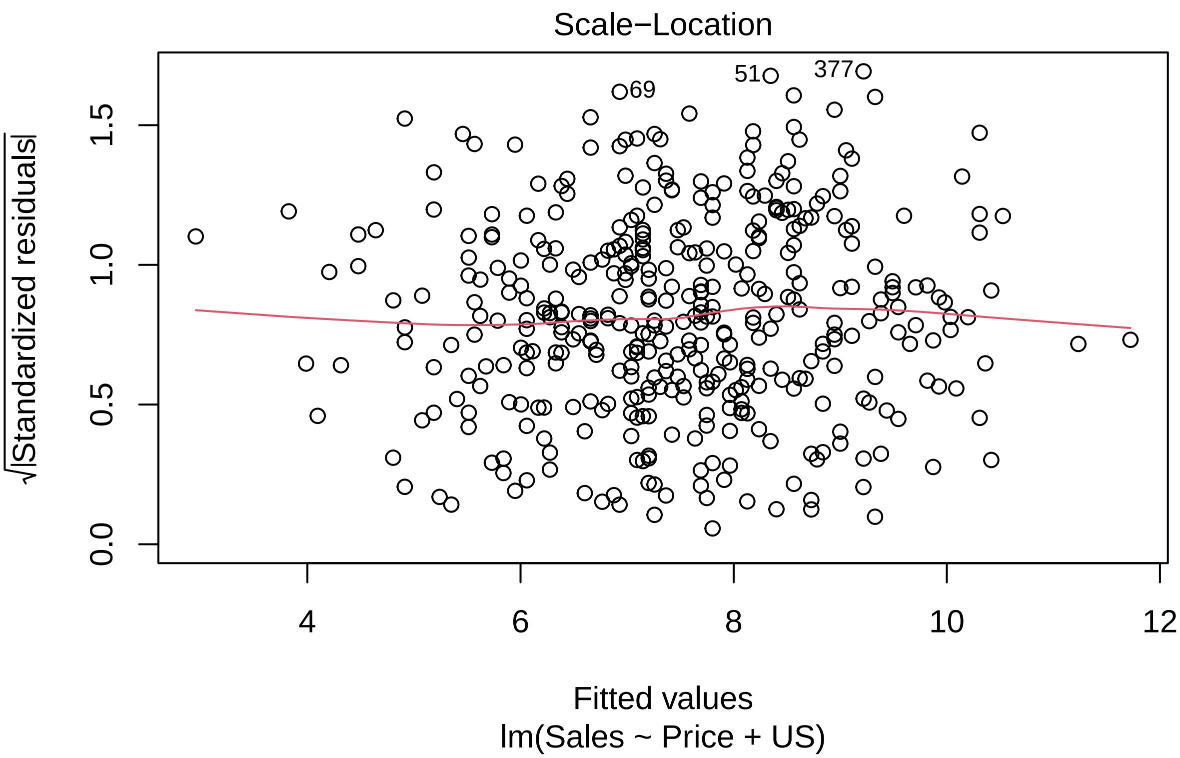
h

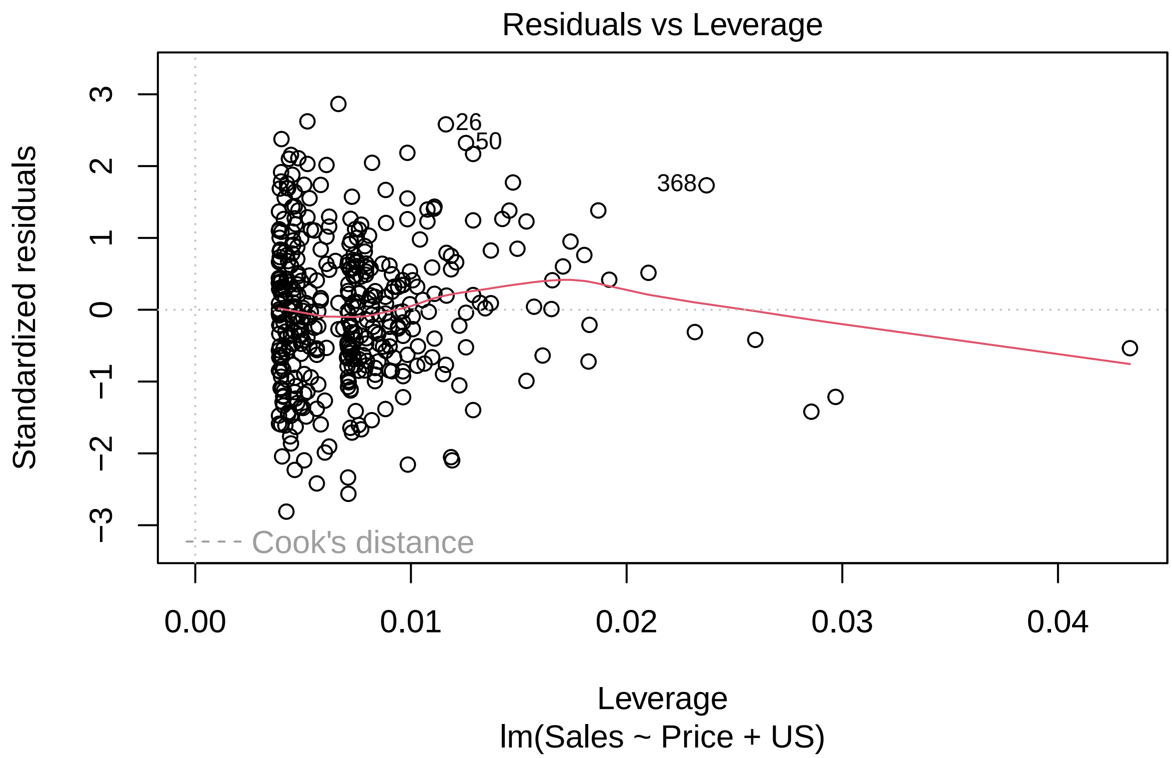
Observation 43 appears to be the high leverage.

```
plot(smaller_model)
```


Residuals vs Fitted
Residuals
Fitted values
lm(Sales ~ Price + US)
69
377
51
5
0
−5
4
6
8
10
12


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Sales ~ Price + US)
377
69
51
−3
−2
−1
0
1
2
3


Scale−Location
69
51
377
√|Standardized residuals|
0.0
0.5
1.0
1.5
4
6
8
10
12
Fitted values
lm(Sales ~ Price + US)



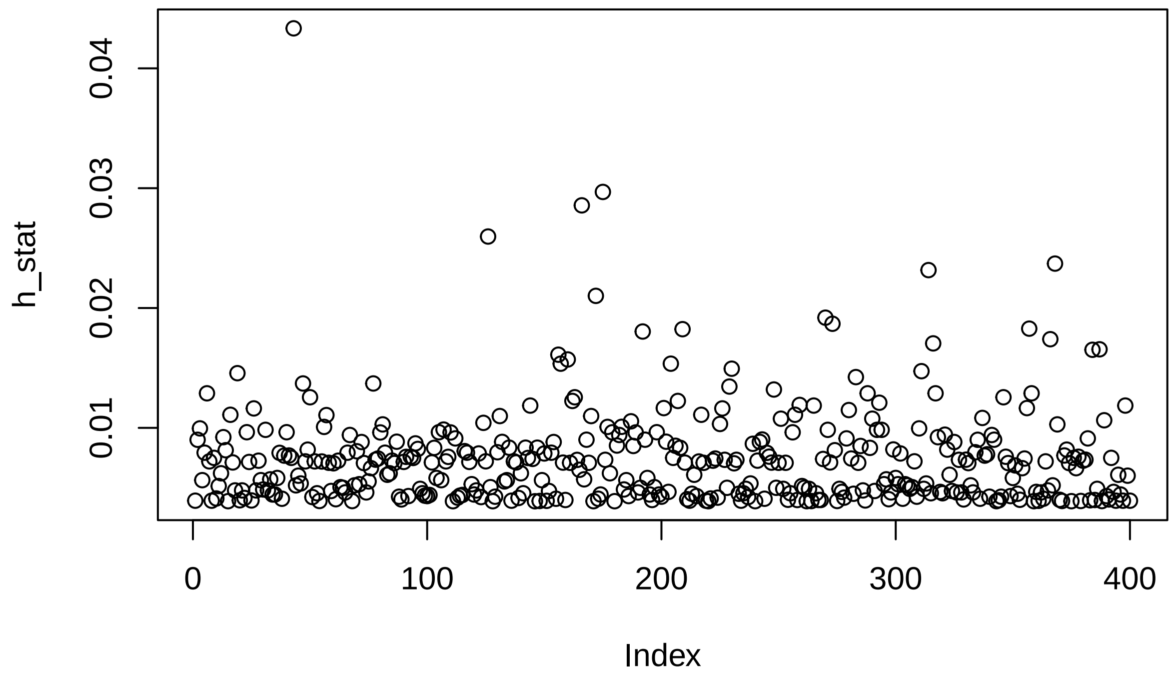

```r
prices = Carseats$Price
mean_price = mean(prices)
numerator = (prices - mean_price)^2
denominator = sum(numerator)
n = length(prices)

high_leverage_stat = 1/n + numerator / denominator
high_leverage_stat[1:10]
```

```
##  [1] 0.002579053 0.007308408 0.008228367 0.004079322 0.003165981 0.011075020
##  [7] 0.002771655 0.002579053 0.002800984 0.002800984
```

```r
hatvalues(lm(Sales ~ Price, data = Carseats))[1:10]
```

```
##           1           2           3           4           5           6
## 0.002579053 0.007308408 0.008228367 0.004079322 0.003165981 0.011075020
##           7           8           9          10
## 0.002771655 0.002579053 0.002800984 0.002800984
```

```r
h_stat = hatvalues(smaller_model)
plot(h_stat)
```

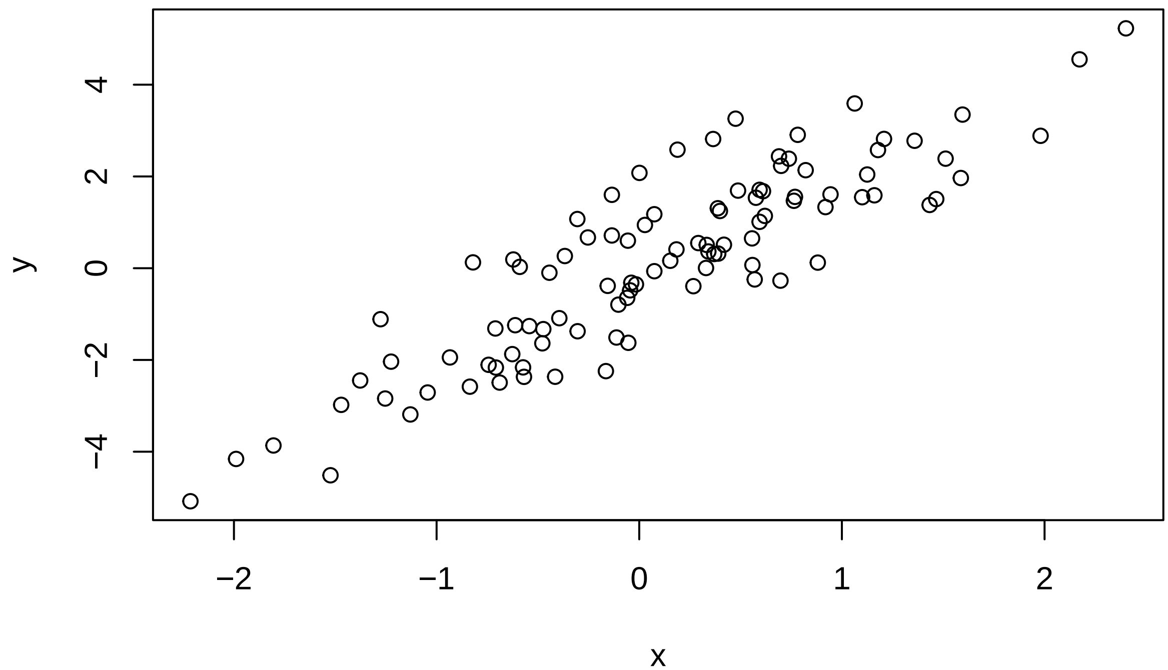



```
which(h_stat > 0.04)
```

```
## 43
## 43
```

Exercise 11

```
set.seed(1)
x = rnorm(100)
y = 2 * x + rnorm(100)
plot(x, y)
```



a

```
no_intercept_y_onto_x = lm(y ~ x + 0)
summary(no_intercept_y_onto_x)
```

```
##
## Call:
## lm(formula = y ~ x + 0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9154 -0.6472 -0.1771  0.5056  2.3109
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## x   1.9939     0.1065   18.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9586 on 99 degrees of freedom
## Multiple R-squared:  0.7798, Adjusted R-squared:  0.7776
## F-statistic: 350.7 on 1 and 99 DF,  p-value: < 2.2e-16
```

```
# summary(lm(y ~ x))
```

b

```
no_intercept_x_onto_y = lm(x ~ y + 0)
summary(no_intercept_x_onto_y)
```

```
##
## Call:
## lm(formula = x ~ y + 0)
##
## Residuals:
##      Min       1Q  Median      3Q     Max
## -0.8699 -0.2368  0.1030  0.2858  0.8938
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## y  0.39111    0.02089   18.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4246 on 99 degrees of freedom
## Multiple R-squared:  0.7798, Adjusted R-squared:  0.7776
## F-statistic: 350.7 on 1 and 99 DF,  p-value: < 2.2e-16
```

```
# summary(lm(x ~ y))
```

c

The t-statistic from both models are the same, hence, p-values are equal also.

From the first model, we can write y = 2x +

$$\epsilon$$

From the first model, we can write x = 0.5x +

$$\epsilon$$

d

$$t = \frac{\widehat{\beta} - \beta_0}{SE(\widehat{\beta})}$$

$$t = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i'=1}^{n} x_{i'}^2} \Bigg/ \sqrt{\frac{\sum_{i=1}^{n} \left(y - x_i \widehat{\beta}\right)^2}{(n-1)\sum_{i'=1}^{n} x_{i'}^2}}$$

$$= \frac{\sqrt{n-1}\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i'=1}^{n} x_{i'}^2}\sqrt{\sum_{i=1}^{n} \left(y_i - x_i \widehat{\beta}\right)^2}}$$

$$= \frac{\sqrt{n-1}\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i'=1}^{n} x_{i'}^2 \left(\sum_{i=1}^{n} y_i^2 - \sum_{i=1}^{n} 2y_i x_i \widehat{\beta} + \sum_{i=1}^{n} x_i^2 \widehat{\beta}^2\right)}}$$

Just keep the numerator as that way and work out the denominator,

$$denominator^2 = \sum_{i'=1}^{n} x_{i'}^2 \sum_{i=1}^{n} y_i^2 - 2\sum_{i'=1}^{n} x_{i'}^2 \sum_{i=1}^{n} y_i x_i \widehat{\beta} + \sum_{i'=1}^{n} x_{i'}^2 \sum_{i=1}^{n} x_i^2 \widehat{\beta}^2$$

$$= \sum_{i'=1}^{n} x_{i'}^2 \sum_{i=1}^{n} y_i^2 - 2\sum_{i'=1}^{n} x_{i'}^2 \sum_{i=1}^{n} y_i x_i \frac{\sum_{j=1}^{n} x_j y_j}{\sum_{k=1}^{n} x_k^2} + \sum_{i'=1}^{n} x_{i'}^2 \sum_{i=1}^{n} x_i^2 \left(\frac{\sum_{j=1}^{n} x_j y_j}{\sum_{k=1}^{n} x_k^2}\right)^2$$

Note that these terms are the same:

$$\sum_{i'=1}^{n} x_{i'}^2 = \sum_{k=1}^{n} x_k^2 = \sum_{i=1}^{n} x_i^2$$

And,

$$\sum_{j=1}^{n} x_j y_j = \sum_{i=1}^{n} x_i y_i$$

Hence,

$$= \sum_{i'=1}^{n} x_{i'}^2 \sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} x_i y_i\right)^2$$

$$= \sum_{i=1}^{n} x_i^2 \sum_{i'=1}^{n} y_{i'}^2 - \left(\sum_{i'=1}^{n} x_{i'} y_{i'}\right)^2$$

Finally, the t-statistic can be written as:

$$\frac{\sqrt{n-1}\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \sum_{i'=1}^{n} y_{i'}^2 - \left(\sum_{i'=1}^{n} x_{i'} y_{i'}\right)^2}}$$

e

Obviously, if there is only one variable and one response, the role of the variable and the response in the t-statistic are interchangeable.

$$\frac{\sqrt{n-1}\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \sum_{i'=1}^{n} y_{i'}^2 - \left(\sum_{i'=1}^{n} x_{i'} y_{i'}\right)^2}} = \frac{\sqrt{n-1}\sum_{i=1}^{n} y_i x_i}{\sqrt{\sum_{i=1}^{n} y_i^2 \sum_{i'=1}^{n} x_{i'}^2 - \left(\sum_{i'=1}^{n} y_{i'} x_{i'}\right)^2}}$$

f

$$x_i$$

$$y_i$$

```r
summary(lm(y ~ x))
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8768 -0.6138 -0.1395  0.5394  2.3462
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.03769    0.09699  -0.389    0.698
## x            1.99894    0.10773  18.556   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9628 on 98 degrees of freedom
## Multiple R-squared:  0.7784, Adjusted R-squared:  0.7762
## F-statistic: 344.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

```r
summary(lm(x ~ y))
```

```
##
## Call:
## lm(formula = x ~ y)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.90848 -0.28101  0.06274  0.24570  0.85736
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.03880    0.04266    0.91    0.365
## y            0.38942    0.02099   18.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4249 on 98 degrees of freedom
## Multiple R-squared:  0.7784, Adjusted R-squared:  0.7762
## F-statistic: 344.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

Exercise 12

a

The coefficient estimate for the regression of X onto Y will be the same as the coefficient for the regression from Y onto X when sum squared of

$$x_i$$

equals to sum squared of

$$y_i$$

b

```r
set.seed(1)
xx = rnorm(100)
yy = 2 * xx
```

```r
summary(lm(yy ~ xx + 0))
```

```
## Warning in summary.lm(lm(yy ~ xx + 0)): essentially perfect fit: summary may be
## unreliable
```

```
##
## Call:
## lm(formula = yy ~ xx + 0)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -3.776e-16 -3.378e-17  2.680e-18  6.113e-17  5.105e-16
##
## Coefficients:
##     Estimate Std. Error   t value Pr(>|t|)
## xx 2.000e+00  1.296e-17 1.543e+17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.167e-16 on 99 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.382e+34 on 1 and 99 DF,  p-value: < 2.2e-16
```

```r
summary(lm(xx ~ yy + 0))
```

```
## Warning in summary.lm(lm(xx ~ yy + 0)): essentially perfect fit: summary may be
## unreliable
```

```
##
## Call:
## lm(formula = xx ~ yy + 0)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.888e-16 -1.689e-17  1.339e-18  3.057e-17  2.552e-16
##
## Coefficients:
##    Estimate Std. Error   t value Pr(>|t|)
## yy 5.00e-01   3.24e-18 1.543e+17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.833e-17 on 99 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.382e+34 on 1 and 99 DF,  p-value: < 2.2e-16
```

c

```
set.seed(1)
xxx = rnorm(100)
yyy = sample(xxx)

summary(lm(yyy ~ xxx + 0))
```

```
##
## Call:
## lm(formula = yyy ~ xxx + 0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1665 -0.4995  0.1140  0.6945  2.2833
##
## Coefficients:
##     Estimate Std. Error t value Pr(>|t|)
## xxx -0.07768    0.10020  -0.775     0.44
##
## Residual standard error: 0.9021 on 99 degrees of freedom
## Multiple R-squared:  0.006034,   Adjusted R-squared:  -0.004006
## F-statistic: 0.601 on 1 and 99 DF,  p-value: 0.4401
```

```
summary(lm(xxx ~ yyy + 0))
```

```
##
## Call:
## lm(formula = xxx ~ yyy + 0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2182 -0.4969  0.1595  0.6782  2.4017
##
## Coefficients:
##     Estimate Std. Error t value Pr(>|t|)
## yyy -0.07768    0.10020  -0.775     0.44
##
## Residual standard error: 0.9021 on 99 degrees of freedom
## Multiple R-squared:  0.006034,   Adjusted R-squared:  -0.004006
## F-statistic: 0.601 on 1 and 99 DF,  p-value: 0.4401
```

Exercise 13

a

```
set.seed(1)
x = rnorm(100, 0, 1)
```

b

```r
set.seed(1)
eps = rnorm(100, 0, 0.25)
```

c

y length is 100,

$$\beta_0$$

= -1,

$$\beta_1$$

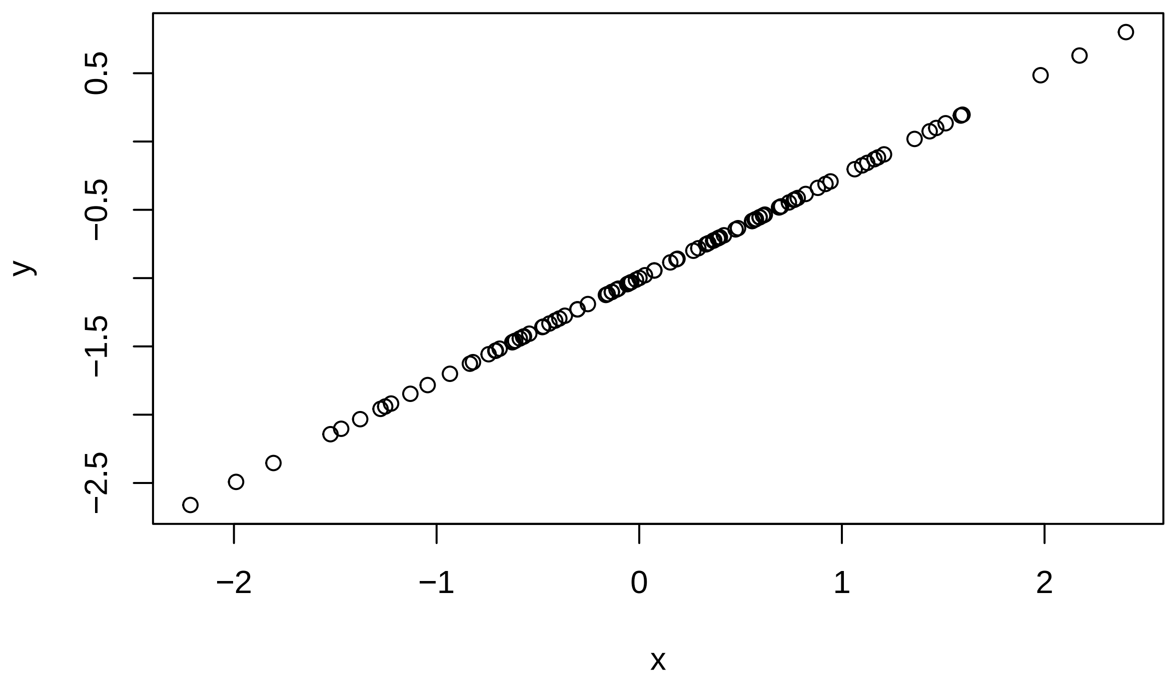
= 0.5.

x and y is linearly related. The relationship is strong and positive.

```r
y = -1 + 0.5*x + eps
```

d

```r
plot(x, y)
```



e

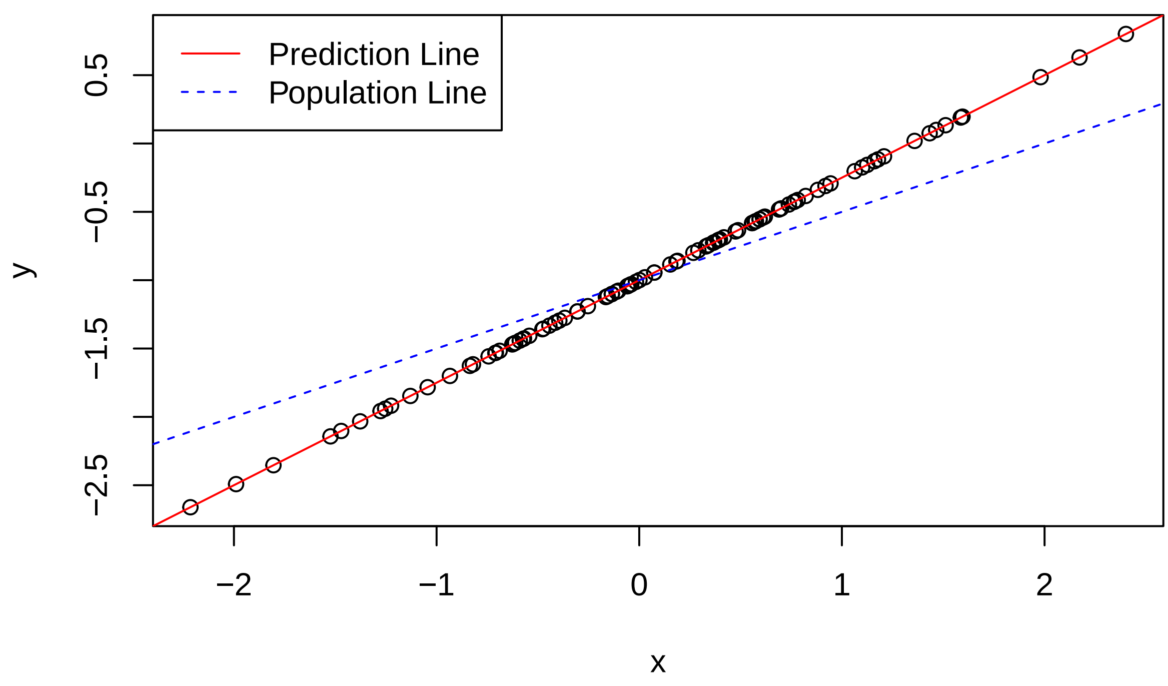
The obtained coefficients (beta hat) are close to the actual coefficients.

```
first_model = lm(y ~ x)
first_model$coefficients
```

```
## (Intercept)           x
##       -1.00        0.75
```

f

```
plot(x, y)
abline(first_model, col = 'red', lty = 1)
abline(a = -1, b = 0.5, col = 'blue', lty = 2)
legend(x = 'topleft',legend = c('Prediction Line', 'Population Line'),
       col = c('red', 'blue'), lty = c(1, 2))
```



g

Using RSE and R-squared, the polynomial regression model slightly improved the model fit.

```r
second_model = lm(y ~ poly(x, 2))
summary(first_model)
```

```
## Warning in summary.lm(first_model): essentially perfect fit: summary may be
## unreliable
```

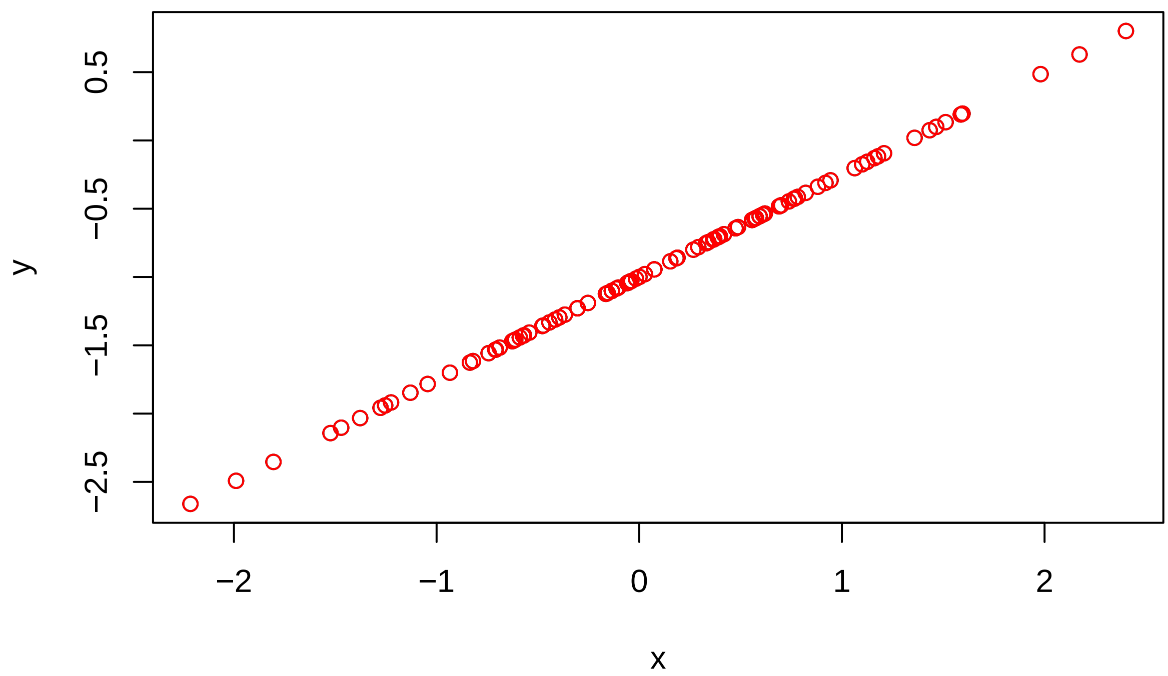
```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.638e-16 -9.452e-17 -1.566e-17  2.395e-17  2.419e-15
##
## Coefficients:
##               Estimate Std. Error    t value Pr(>|t|)
## (Intercept) -1.000e+00  2.651e-17 -3.772e+16   <2e-16 ***
## x            7.500e-01  2.945e-17  2.547e+16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.632e-16 on 98 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 6.487e+32 on 1 and 98 DF,  p-value: < 2.2e-16
```

```r
summary(second_model)
```

```
## Warning in summary.lm(second_model): essentially perfect fit: summary may be
## unreliable
```

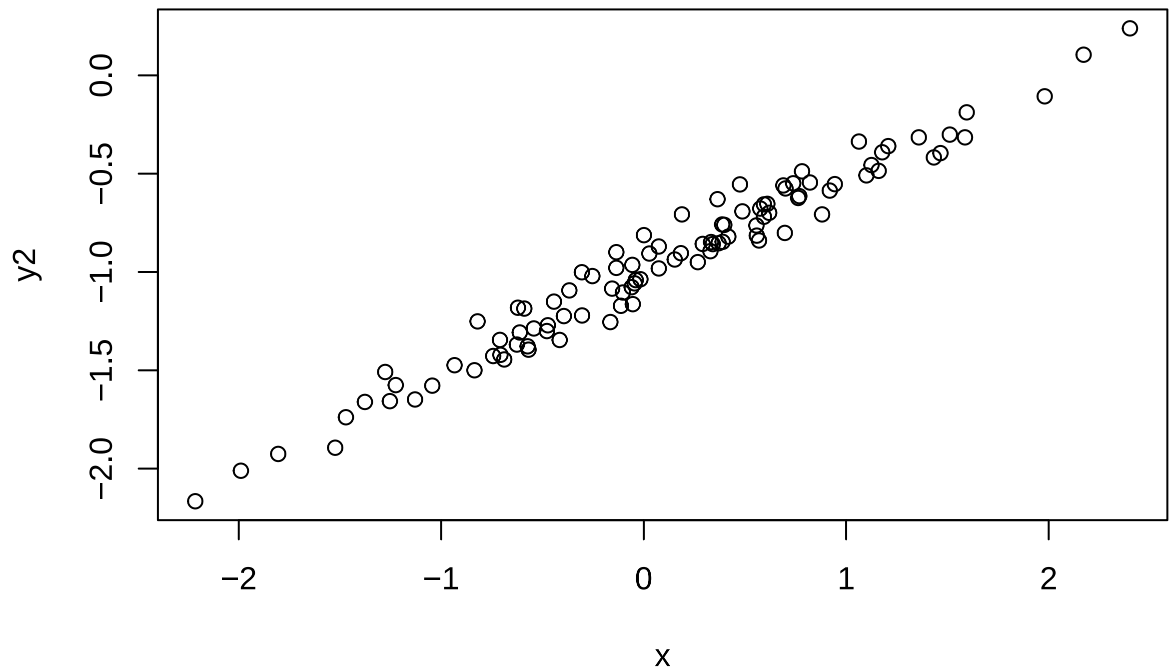
```
##
## Call:
## lm(formula = y ~ poly(x, 2))
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -5.871e-16 -9.873e-17 -2.238e-17  4.522e-17  2.582e-15
##
## Coefficients:
##               Estimate Std. Error    t value Pr(>|t|)
## (Intercept) -9.183e-01  2.905e-17 -3.161e+16   <2e-16 ***
## poly(x, 2)1  6.703e+00  2.905e-16  2.307e+16   <2e-16 ***
## poly(x, 2)2 -7.562e-16  2.905e-16 -2.603e+00   0.0107 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.905e-16 on 97 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.661e+32 on 2 and 97 DF,  p-value: < 2.2e-16
```

```
plot(x, y)
points(x, second_model$fitted.values, col = 'red')
```

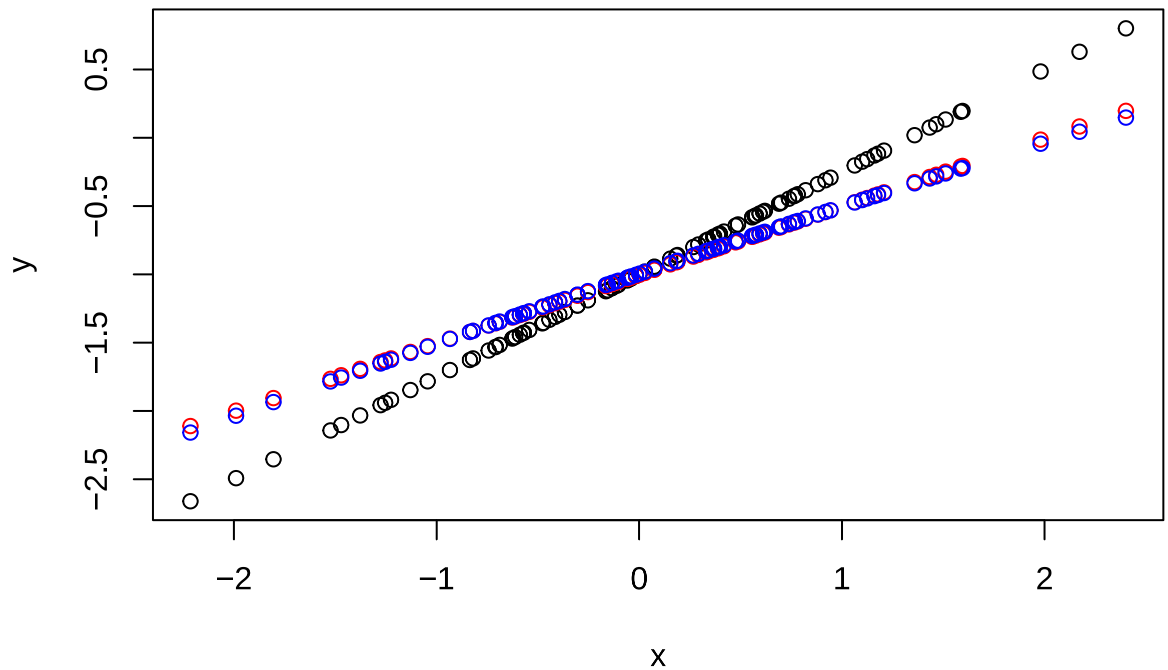


h

```
set.seed(1)

x = rnorm(100, 0, 1)
eps2 = rnorm(100, 0, 0.09)
y2 = -1 + 0.5*x + eps2

plot(x, y2)
```

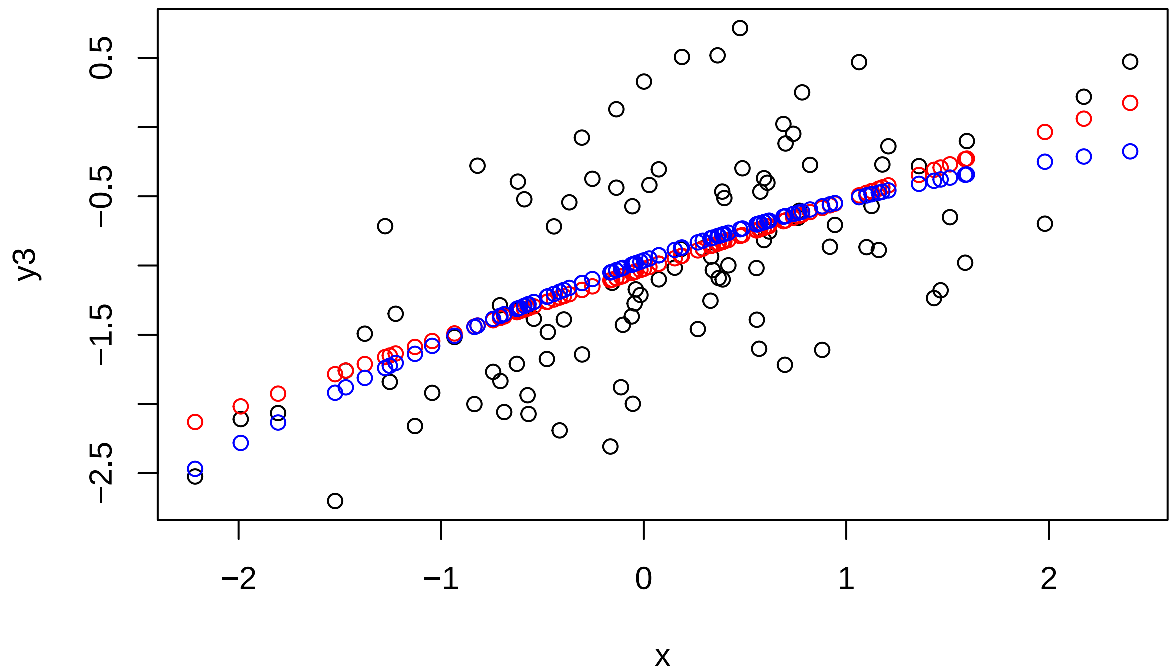



```
third_model = lm(y2 ~ x)
fourth_model = lm(y2 ~ poly(x, 2))

plot(x, y)
points(x, third_model$fitted.values, col = 'red')
points(x, fourth_model$fitted.values, col = 'blue')
```



i

```
set.seed(1)

x = rnorm(100, 0, 1)
eps3 = rnorm(100, 0, 0.64)
y3 = -1 + 0.5*x + eps3

fifth_model = lm(y3 ~ x)
sixth_model = lm(y3 ~ x + I(x^2))

plot(x, y3)
points(x, fifth_model$fitted.values, col = 'red')
points(x, sixth_model$fitted.values, col = 'blue')
```



j

Obviously, the model is more confident when there is less noise and vice versa.

```
confint(first_model)
```

```
## Warning in summary.lm(object, ...): essentially perfect fit: summary may be
## unreliable
```

```
##              2.5 % 97.5 %
## (Intercept) -1.00  -1.00
## x            0.75   0.75
```

```
confint(third_model)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.0207145 -0.9860702
## x            0.4806643  0.5191448
```

```
confint(fifth_model)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.1473029 -0.9009437
## x            0.3625016  0.6361411
```

```r
summary(lm(y ~ x + I(x^2)))
```

```
## Warning in summary.lm(lm(y ~ x + I(x^2))): essentially perfect fit: summary may
## be unreliable
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2))
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.192e-16 -8.955e-17 -1.631e-17  2.841e-17  2.413e-15
##
## Coefficients:
##               Estimate Std. Error    t value Pr(>|t|)
## (Intercept) -1.000e+00  3.238e-17 -3.088e+16   <2e-16 ***
## x            7.500e-01  2.972e-17  2.524e+16   <2e-16 ***
## I(x^2)      -1.846e-17  2.333e-17 -7.910e-01    0.431
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.637e-16 on 97 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 3.231e+32 on 2 and 97 DF,  p-value: < 2.2e-16
```

```r
summary(lm(y ~ poly(x, 2)))
```

```
## Warning in summary.lm(lm(y ~ poly(x, 2))): essentially perfect fit: summary may
## be unreliable
```

```
##
## Call:
## lm(formula = y ~ poly(x, 2))
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -5.871e-16 -9.873e-17 -2.238e-17  4.522e-17  2.582e-15
##
## Coefficients:
##               Estimate Std. Error    t value Pr(>|t|)
## (Intercept) -9.183e-01  2.905e-17 -3.161e+16   <2e-16 ***
## poly(x, 2)1  6.703e+00  2.905e-16  2.307e+16   <2e-16 ***
## poly(x, 2)2 -7.562e-16  2.905e-16 -2.603e+00   0.0107 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.905e-16 on 97 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.661e+32 on 2 and 97 DF,  p-value: < 2.2e-16
```

Exercise 14

a

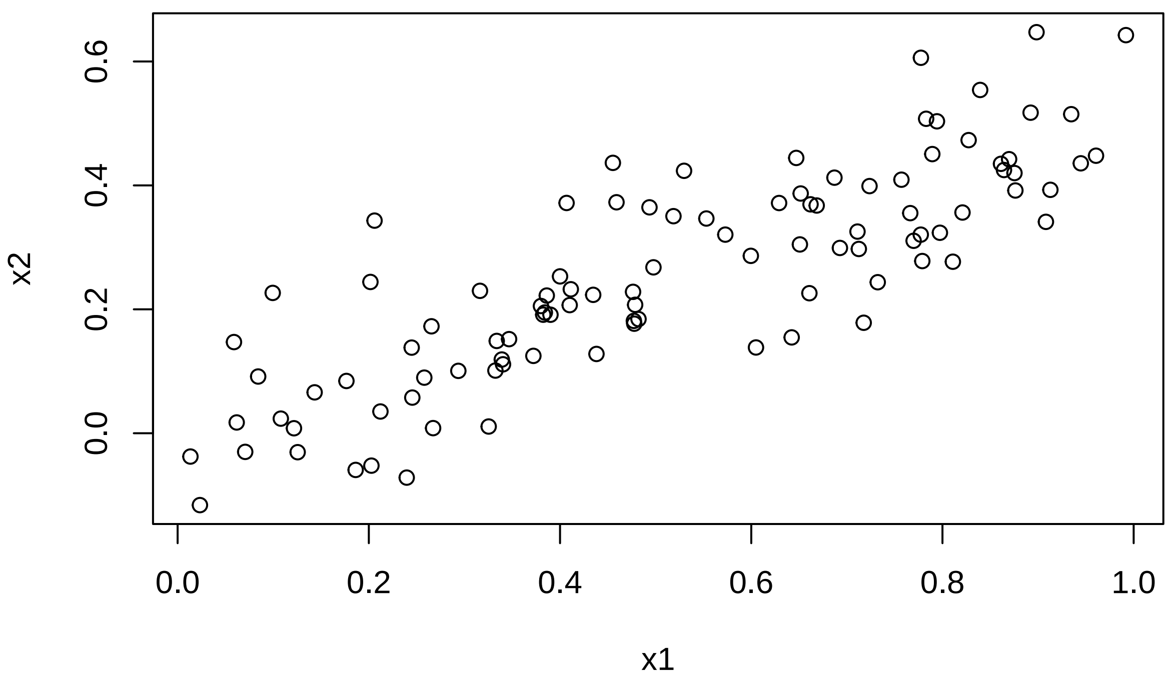
$$Y = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2$$

```
set.seed(1)
x1 = runif(100)
x2 = 0.5 * x1 + rnorm(100)/10
y = 2 + 2*x1 + 0.3*x2 + rnorm(100)
```

b

Variables x1 and x2 linearly correlated.

```
plot(x1, x2)
```



c

$$\widehat{\beta_0}, \widehat{\beta_1}, \widehat{\beta_2}$$

are 2.1, 1.4 and 1 respectively. Recall that the "true"

$$\beta_0, \beta_1, \beta_2$$

are 2, 2 and 0.3 respectively.

In this case, only

$$\widehat{\beta_0}$$

is close to its "true" value. Also, we can reject to the null hypothesis that

$$\beta_1 = 0$$

, but we fail to reject to the null hypothesis that

$$\beta_2 = 0$$

.

```
first_model = lm(y ~ x1 + x2)
summary(first_model)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8311 -0.7273 -0.0537  0.6338  2.3359
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1305     0.2319   9.188 7.61e-15 ***
## x1            1.4396     0.7212   1.996   0.0487 *
## x2            1.0097     1.1337   0.891   0.3754
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.056 on 97 degrees of freedom
## Multiple R-squared:  0.2088, Adjusted R-squared:  0.1925
## F-statistic:  12.8 on 2 and 97 DF,  p-value: 1.164e-05
```

d

$$\widehat{\beta_0}, \widehat{\beta_1}$$

are 2.1, 2.0 respectively. Recall that the "true"

$$\beta_0, \beta_1, \beta_2$$

are 2, 2 and 0.3 respectively.

In this case,

$$\widehat{\beta_0}, \widehat{\beta_1}$$

are close to its true values. Also, we can reject to the null hypothesis that

$$\beta_1$$

= 0.

RSE and Adjusted R-squared are almost the same as in the first model.

```r
second_model = lm(y ~ x1)
summary(second_model)
```

```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.89495 -0.66874 -0.07785  0.59221  2.45560
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1124     0.2307   9.155 8.27e-15 ***
## x1            1.9759     0.3963   4.986 2.66e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.055 on 98 degrees of freedom
## Multiple R-squared:  0.2024, Adjusted R-squared:  0.1942
## F-statistic: 24.86 on 1 and 98 DF,  p-value: 2.661e-06
```

e

In this case, we can reject to the null hypothesis that

$$\beta_2$$

$= 0$.

```r
third_model = lm(y ~ x2)
summary(third_model)
```

```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.62687 -0.75156 -0.03598  0.72383  2.44890
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3899     0.1949   12.26  < 2e-16 ***
## x2            2.8996     0.6330    4.58 1.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.072 on 98 degrees of freedom
## Multiple R-squared:  0.1763, Adjusted R-squared:  0.1679
## F-statistic: 20.98 on 1 and 98 DF,  p-value: 1.366e-05
```

f

The results from (c) - (e) contradict each other but this makes sense since collinearity is in presence. The response can be predicted using x1 or x2 only.

```
summary(first_model)$coefficients
```

```
##             Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 2.130500  0.2318817 9.1878742 7.606713e-15
## x1          1.439555  0.7211795 1.9961126 4.872517e-02
## x2          1.009674  1.1337225 0.8905831 3.753565e-01
```

```
summary(second_model)$coefficients
```

```
##             Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 2.112394  0.2307448 9.154676 8.269388e-15
## x1          1.975929  0.3962774 4.986227 2.660579e-06
```

```
summary(third_model)$coefficients
```

```
##             Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 2.389949  0.1949307 12.260508 1.682395e-21
## x2          2.899585  0.6330467  4.580365 1.366430e-05
```

g

```
set.seed(1)

x1 = runif(100)
x2 = 0.5 * x1 + rnorm(100)/10
y = 2 + 2*x1 + 0.3*x2 + rnorm(100)

x1 = c(x1, 0.1)
x2 = c(x2, 0.8)
y = c(y, 6)
```

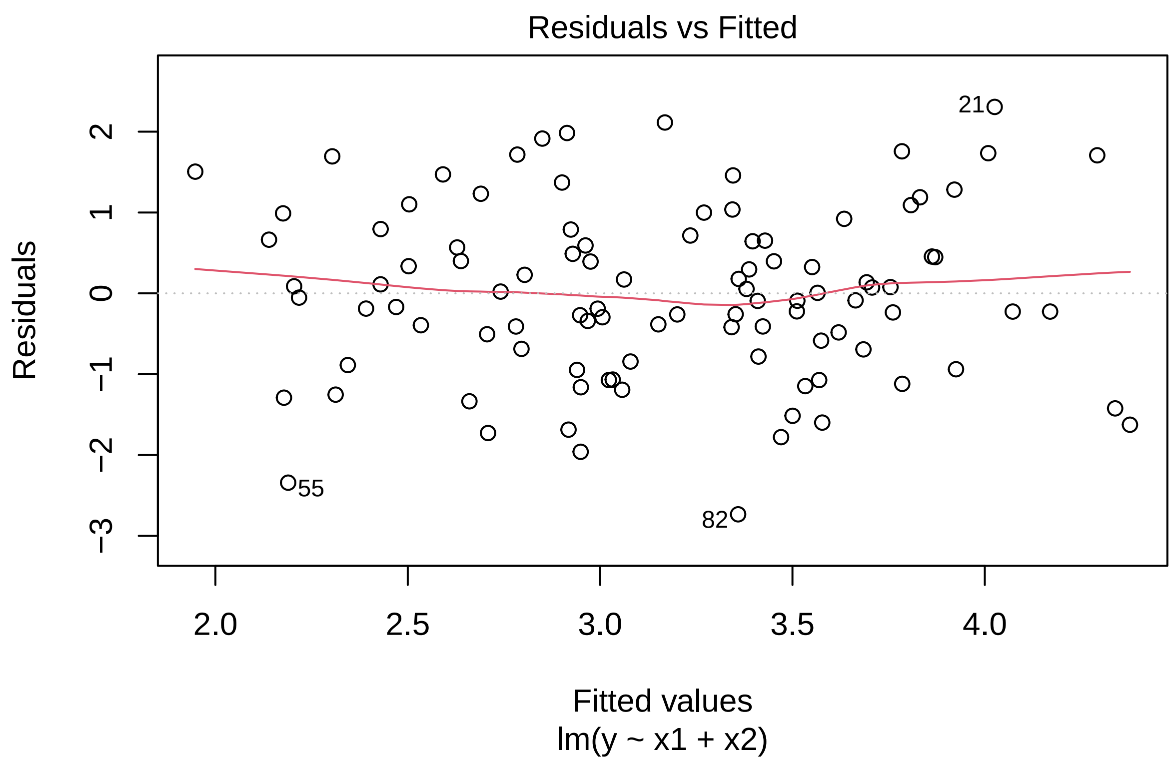
In this case (y ~ x1 + x2), the additional observation is a high-leverage point.

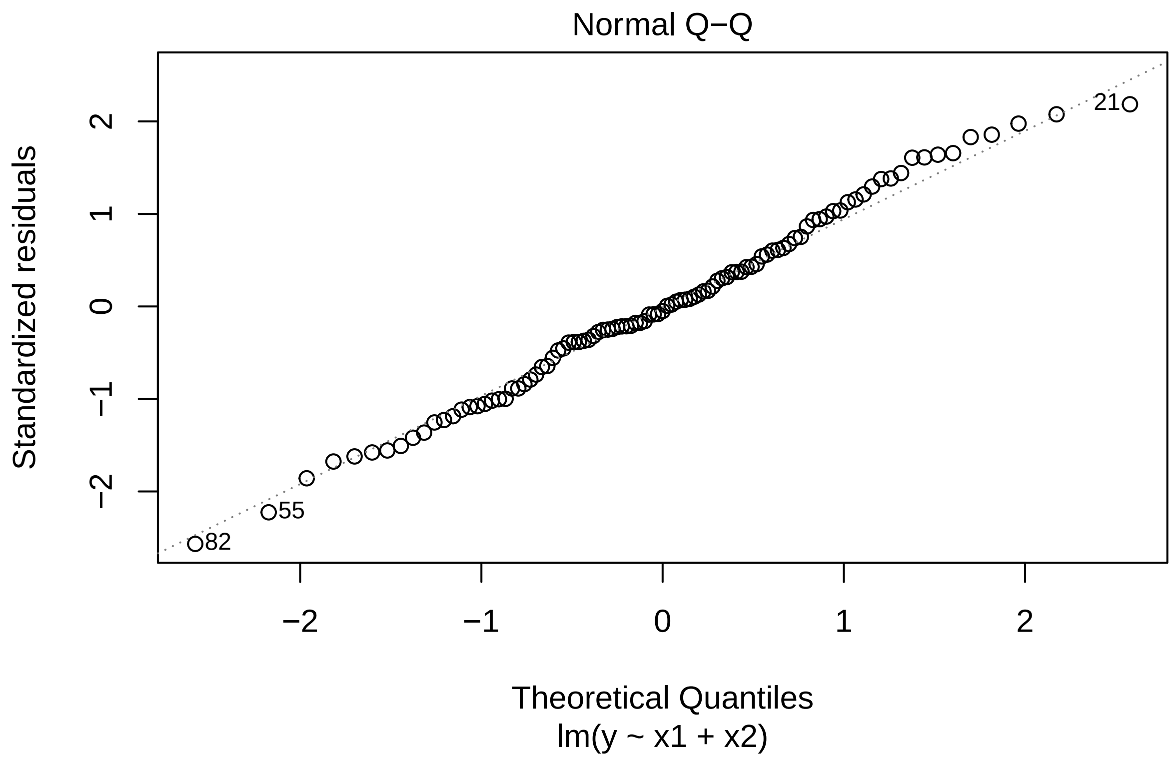
```
fourth_model = lm(y ~ x1 + x2)
summary(fourth_model)
```

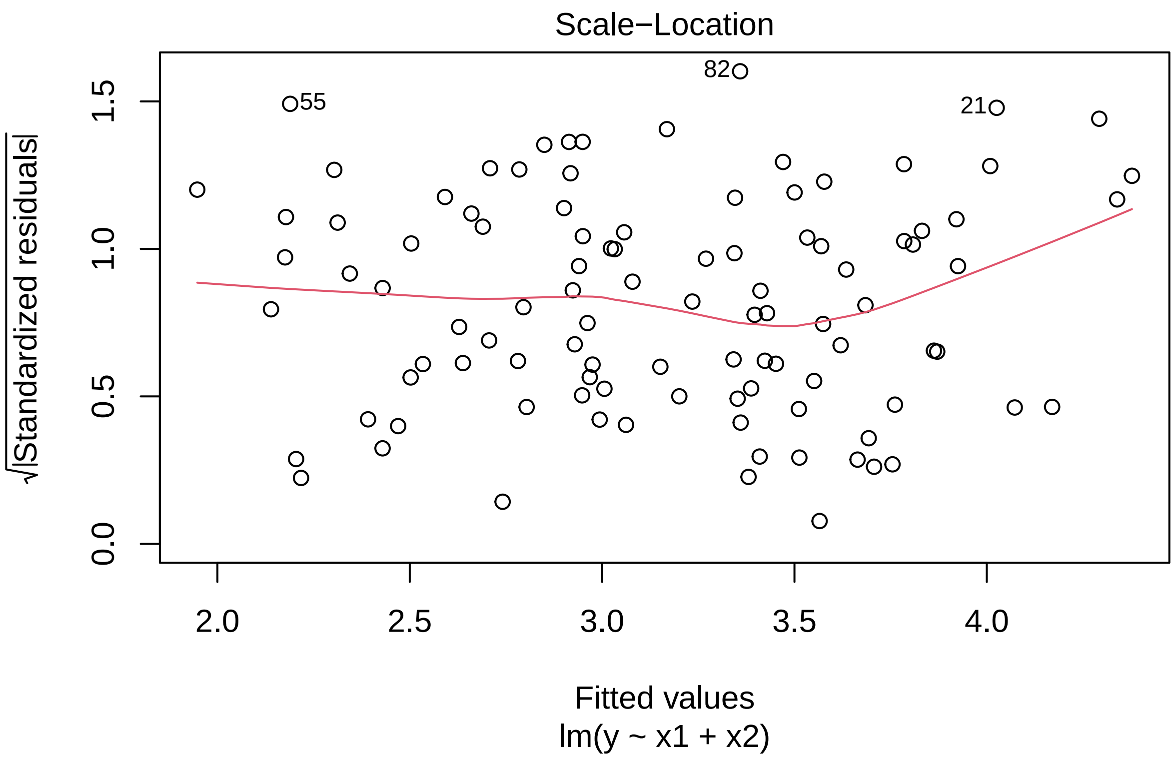
```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.73348 -0.69318 -0.05263  0.66385  2.30619
##
```

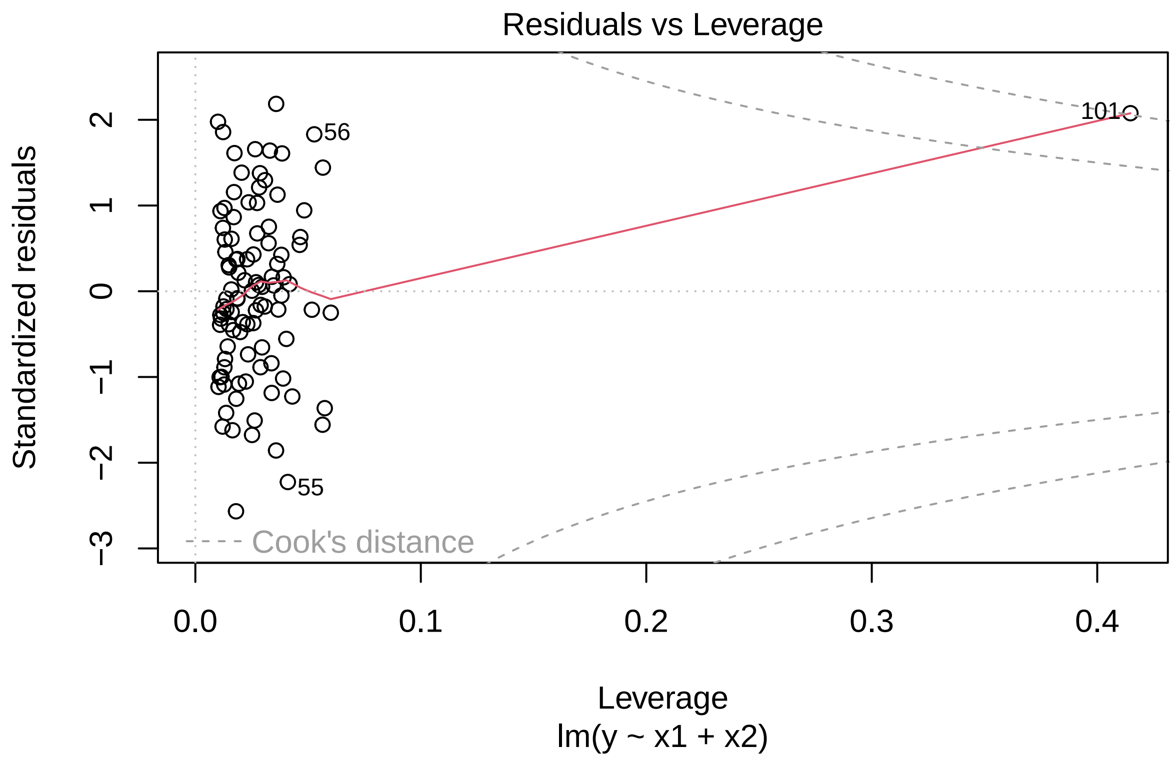
```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2267     0.2314   9.624 7.91e-16 ***
## x1            0.5394     0.5922   0.911  0.36458
## x2            2.5146     0.8977   2.801  0.00614 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.075 on 98 degrees of freedom
## Multiple R-squared:  0.2188, Adjusted R-squared:  0.2029
## F-statistic: 13.72 on 2 and 98 DF,  p-value: 5.564e-06
```

```
plot(fourth_model)
```



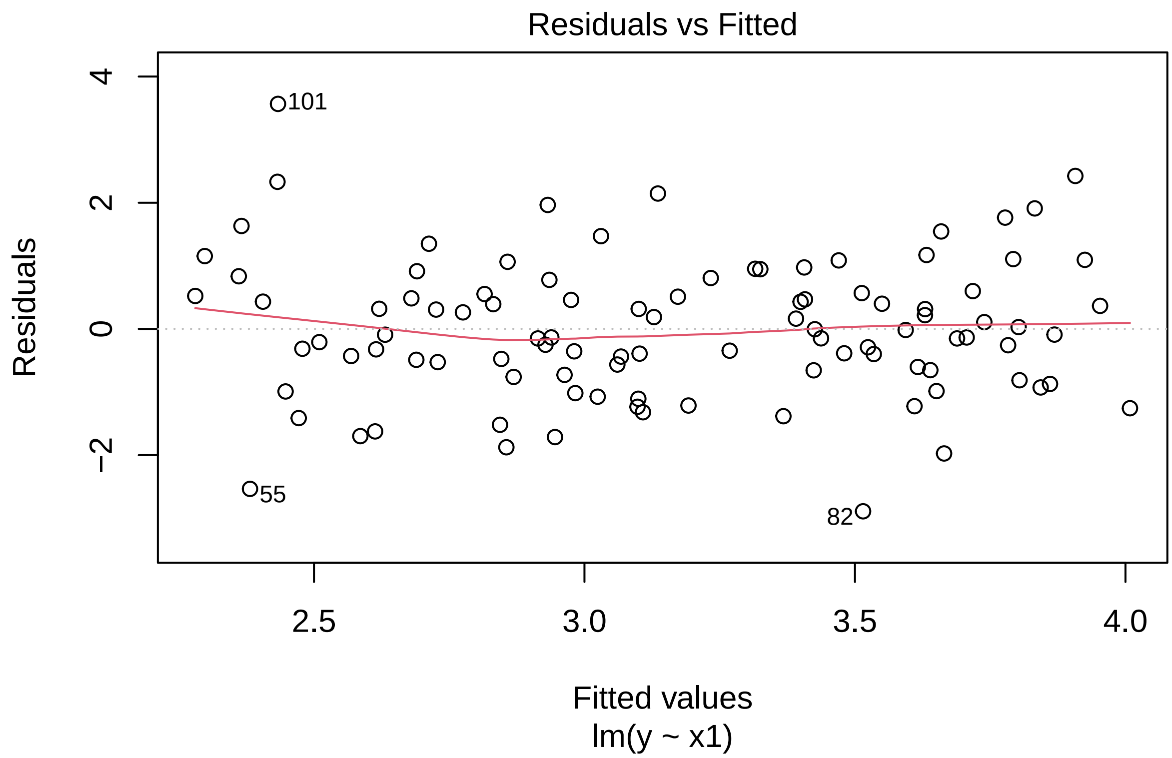

Normal Q−Q
Standardized residuals
2
1
0
−1
−2
21
55
82
−2
−1
0
1
2
Theoretical Quantiles
lm(y ~ x1 + x2)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
55
82
21
Fitted values
lm(y ~ x1 + x2)

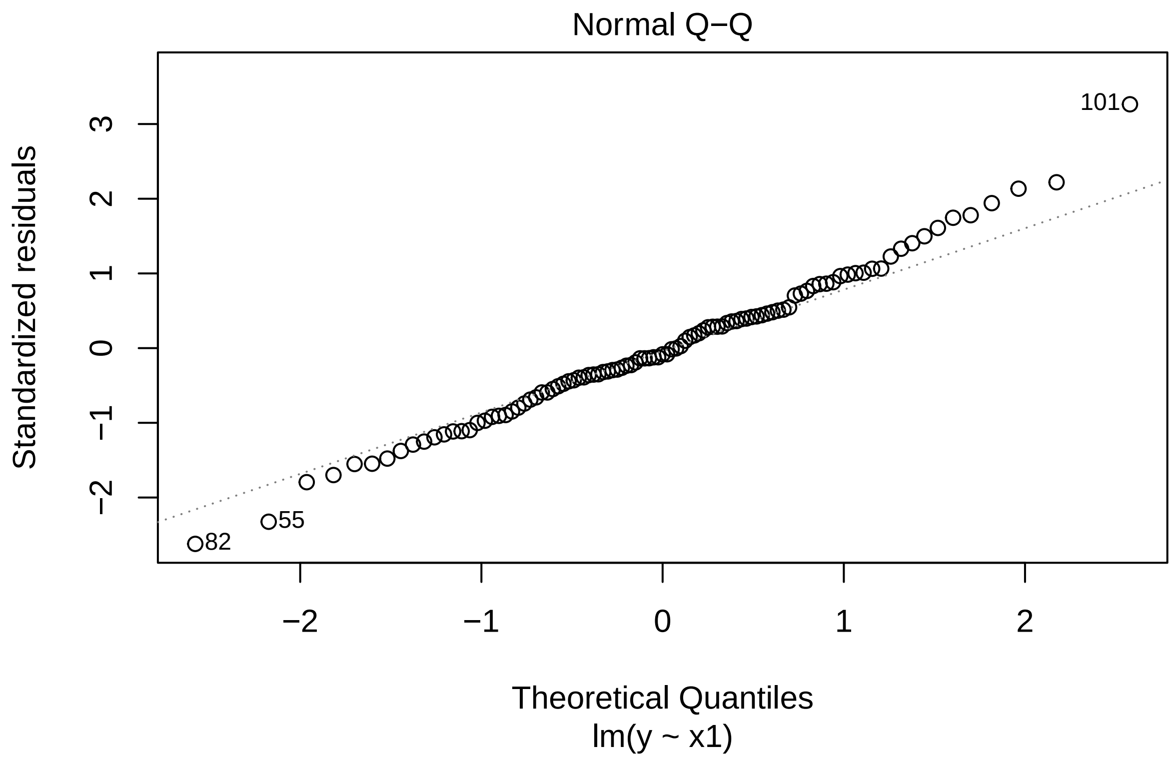


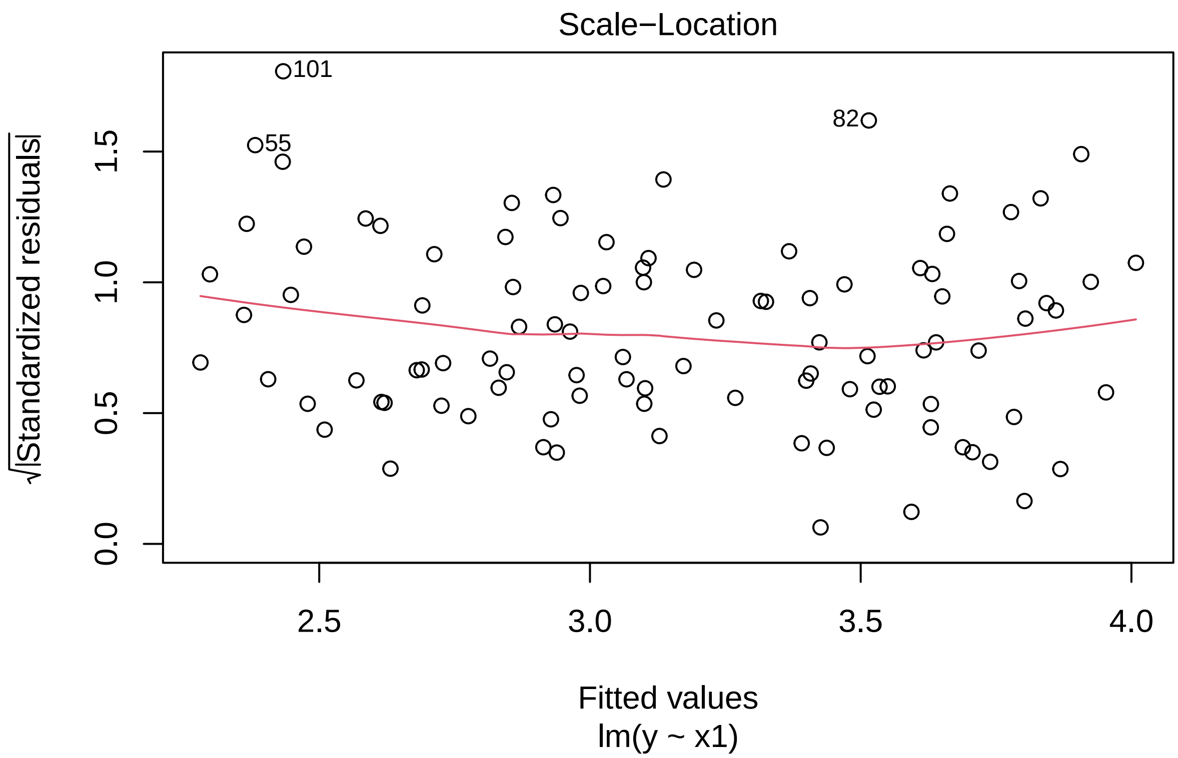
In this case (y ~ x1), the additional observation is an outlier.

```
fifth_model = lm(y ~ x1)
plot(fifth_model)
```

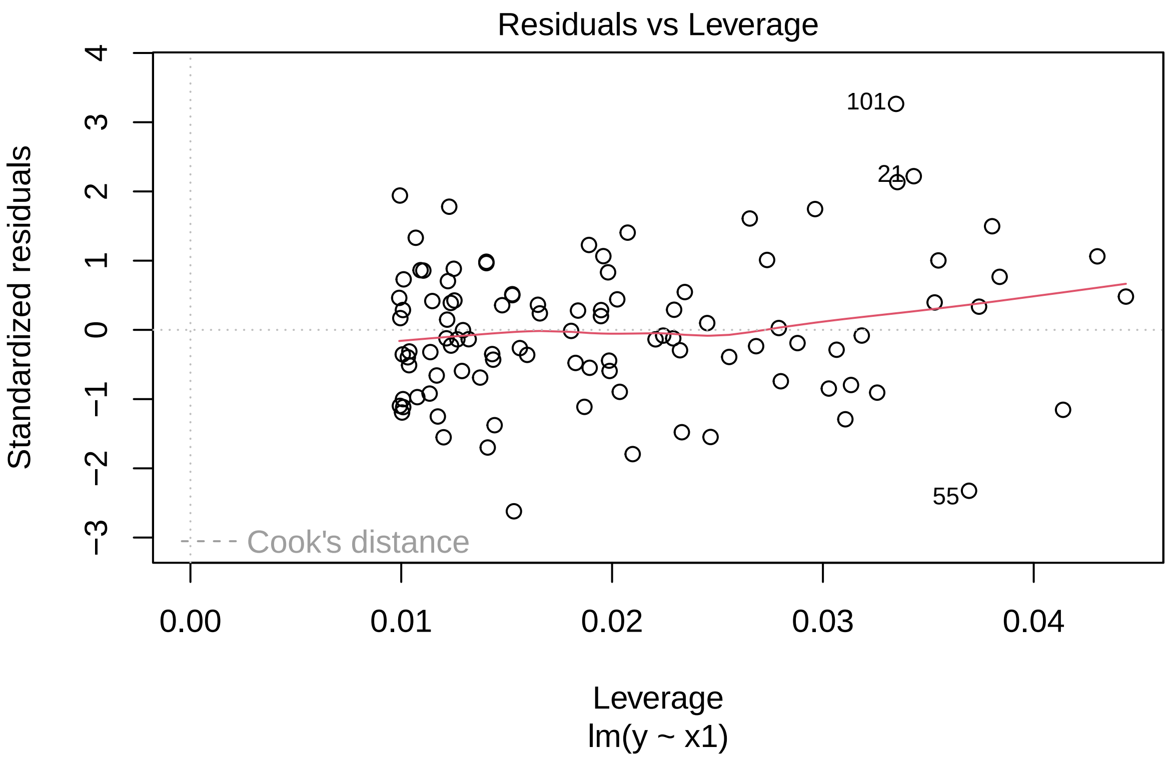


Residuals vs Fitted
Residuals
4
2
0
−2
101
55
82
2.5
3.0
3.5
4.0
Fitted values
lm(y ~ x1)


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(y ~ x1)
101
55
82
−2
−1
0
1
2
3

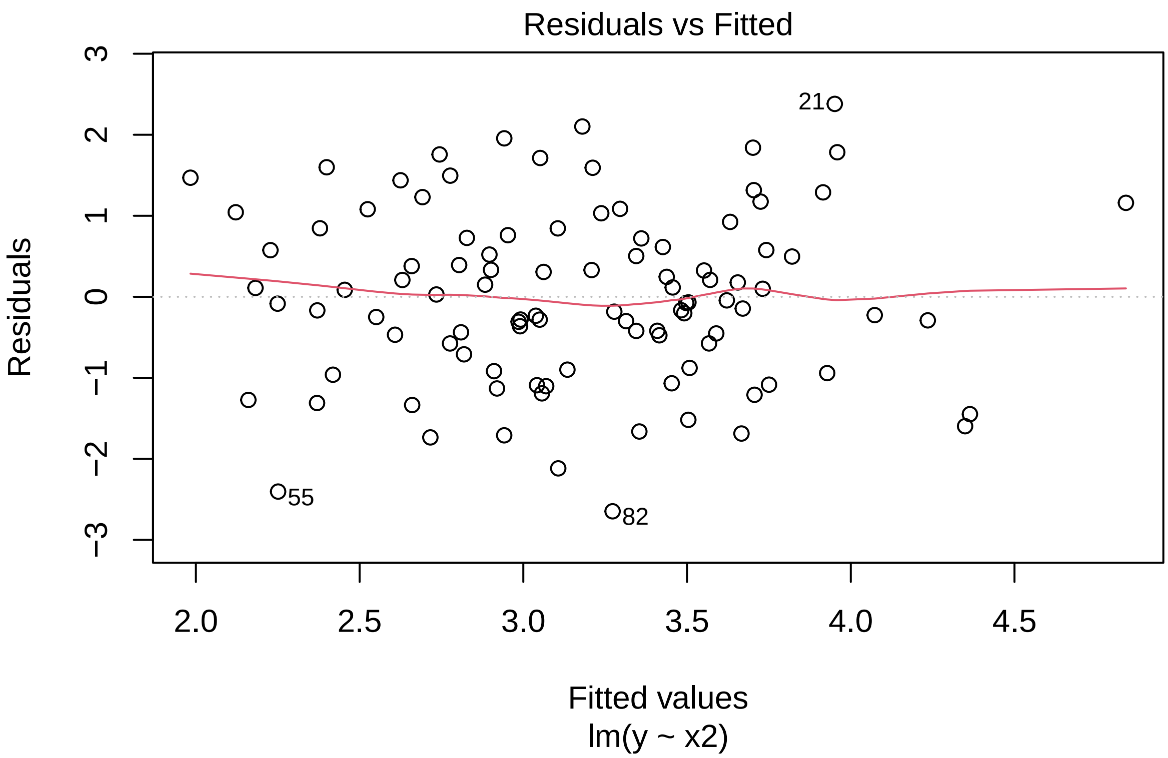


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
101
55
82
2.5
3.0
3.5
4.0
Fitted values
lm(y ~ x1)

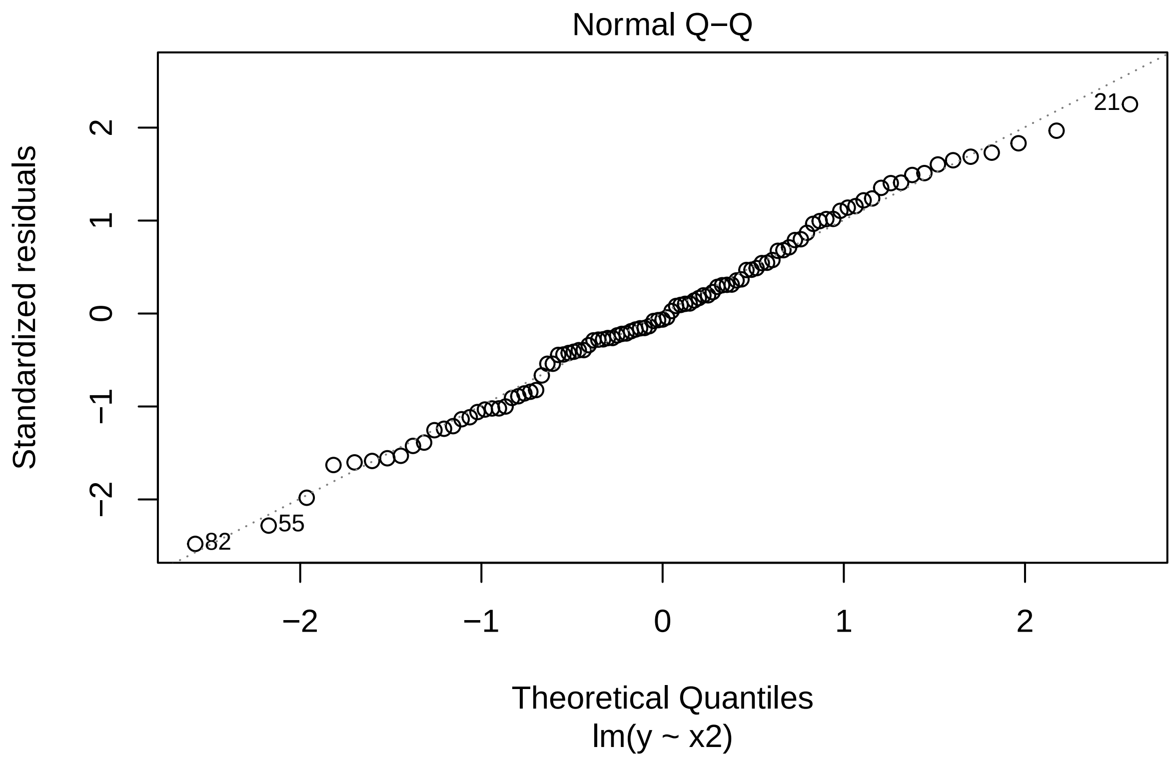


In this case (y ~ x2), the additional observation is a high-leverage point.

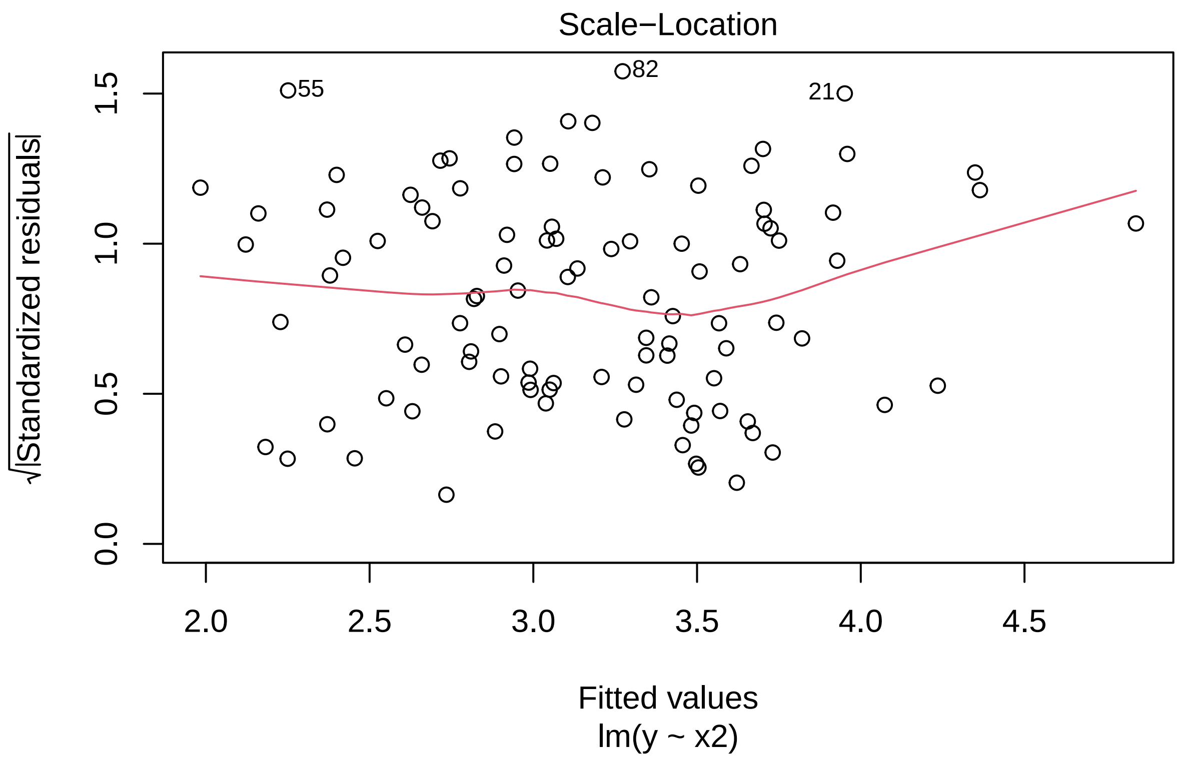
```
sixth_model = lm(y ~ x2)
plot(sixth_model)
```



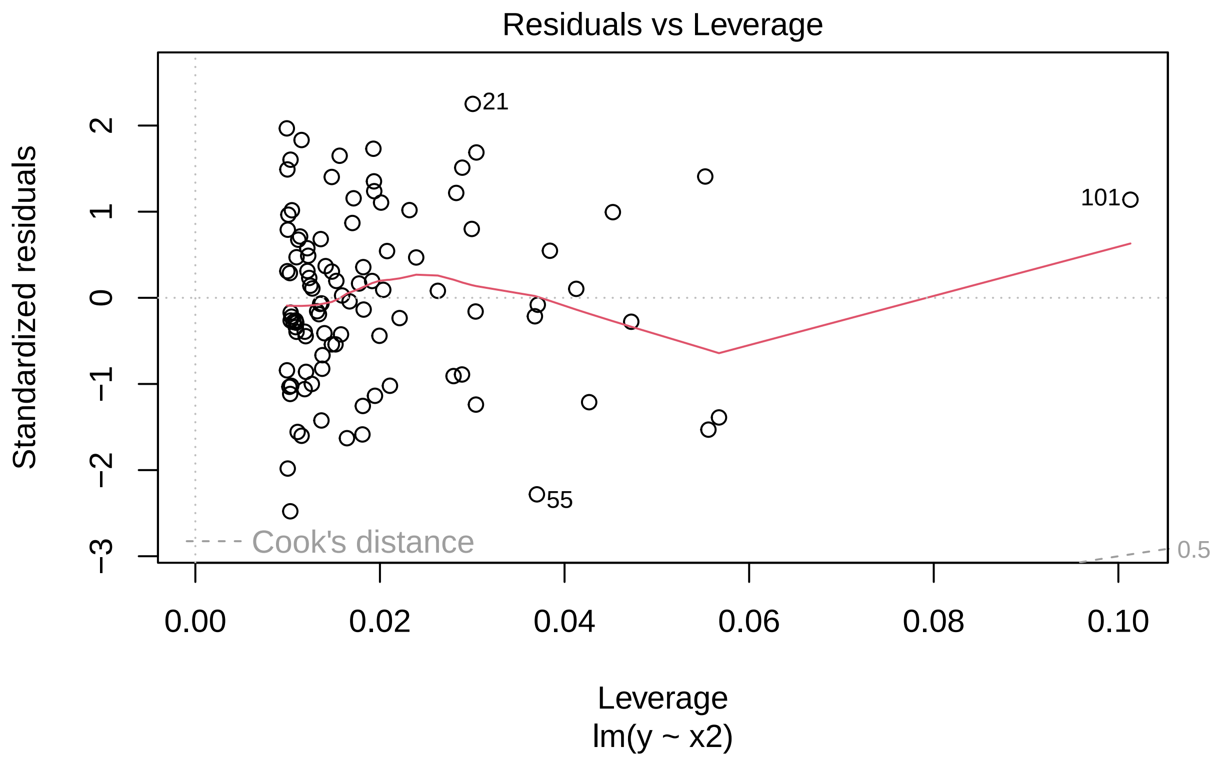
Residuals vs Fitted
Residuals
3
2
1
0
−1
−2
−3
21
55
82
2.0
2.5
3.0
3.5
4.0
4.5
Fitted values
lm(y ~ x2)



Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(y ~ x2)
−2
−1
0
1
2
82
55
21



Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
55
82
21
Fitted values
lm(y ~ x2)



Exercise 15

```
head(Boston)
```

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

a

Except "chas", all other predictors appear to be statistically significant.

```
all_predictors = colnames(Boston)
individual_predictor_result = data.frame()

for (i in 2:13){

  x = Boston[, i]
```

```r
  model = lm(crim ~ x, data = Boston)
  df = round(data.frame(summary(model)$coefficients), 3)[2,]
  individual_predictor_result = rbind(individual_predictor_result, df)
  individual_predictor_result = individual_predictor_result
}

rownames(individual_predictor_result) = all_predictors[2:13]
individual_predictor_result
```

```
##         Estimate Std..Error t.value Pr...t..
## zn        -0.074      0.016  -4.594    0.000
## indus      0.510      0.051   9.991    0.000
## chas      -1.893      1.506  -1.257    0.209
## nox       31.249      2.999  10.419    0.000
## rm        -2.684      0.532  -5.045    0.000
## age        0.108      0.013   8.463    0.000
## dis       -1.551      0.168  -9.213    0.000
## rad        0.618      0.034  17.998    0.000
## tax        0.030      0.002  16.099    0.000
## ptratio    1.152      0.169   6.801    0.000
## lstat      0.549      0.048  11.491    0.000
## medv      -0.363      0.038  -9.460    0.000
```

```r
# View(individual_predictor_result)
```

b

We reject the null hypothesis H0:

$$\beta$$

= 0 for predictors zn, dis, rad and medv

```r
full_model = lm(crim ~ ., data = Boston)
all_predictors_result = round(data.frame(summary(full_model)$coefficients), 3)
all_predictors_result
```

```
##             Estimate Std..Error t.value Pr...t..
## (Intercept)   13.778      7.082   1.946    0.052
## zn             0.046      0.019   2.433    0.015
## indus         -0.058      0.084  -0.698    0.486
## chas          -0.825      1.183  -0.697    0.486
## nox           -9.958      5.290  -1.882    0.060
## rm             0.629      0.607   1.036    0.301
## age           -0.001      0.018  -0.047    0.962
## dis           -1.012      0.282  -3.584    0.000
## rad            0.612      0.088   6.997    0.000
## tax           -0.004      0.005  -0.730    0.466
## ptratio       -0.304      0.186  -1.632    0.103
## lstat          0.139      0.076   1.833    0.067
## medv          -0.220      0.060  -3.678    0.000
```

```
# View(all_predictors_result)
```

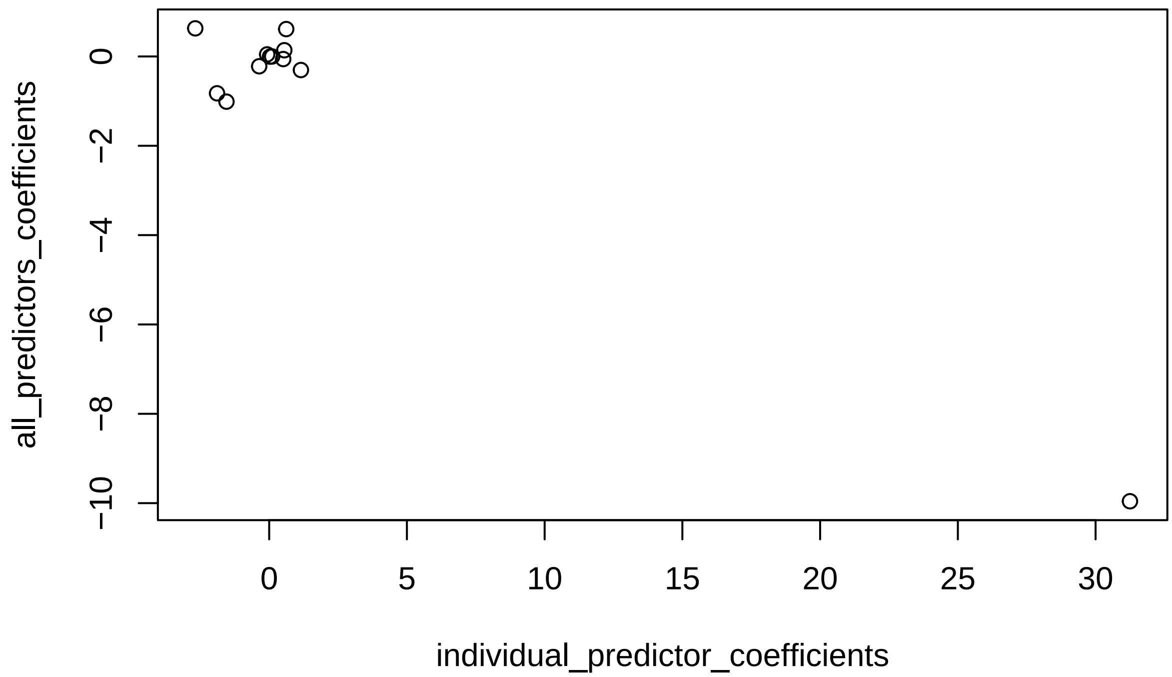
c

```
individual_predictor_coefficients = data.frame()

for (i in 1:13){

  x = Boston[, i]
  model = lm(crim ~ x, data = Boston)
  df = round(data.frame(model$coefficients), 3)[2,]
  individual_predictor_coefficients = rbind(individual_predictor_coefficients, df)
  individual_predictor_coefficients = individual_predictor_coefficients
}

individual_predictor_coefficients = individual_predictor_coefficients[2:13,]
```

```
all_predictors_coefficients = full_model$coefficients
all_predictors_coefficients = data.frame(all_predictors_coefficients)[2:13, 1]
all_predictors_coefficients
```

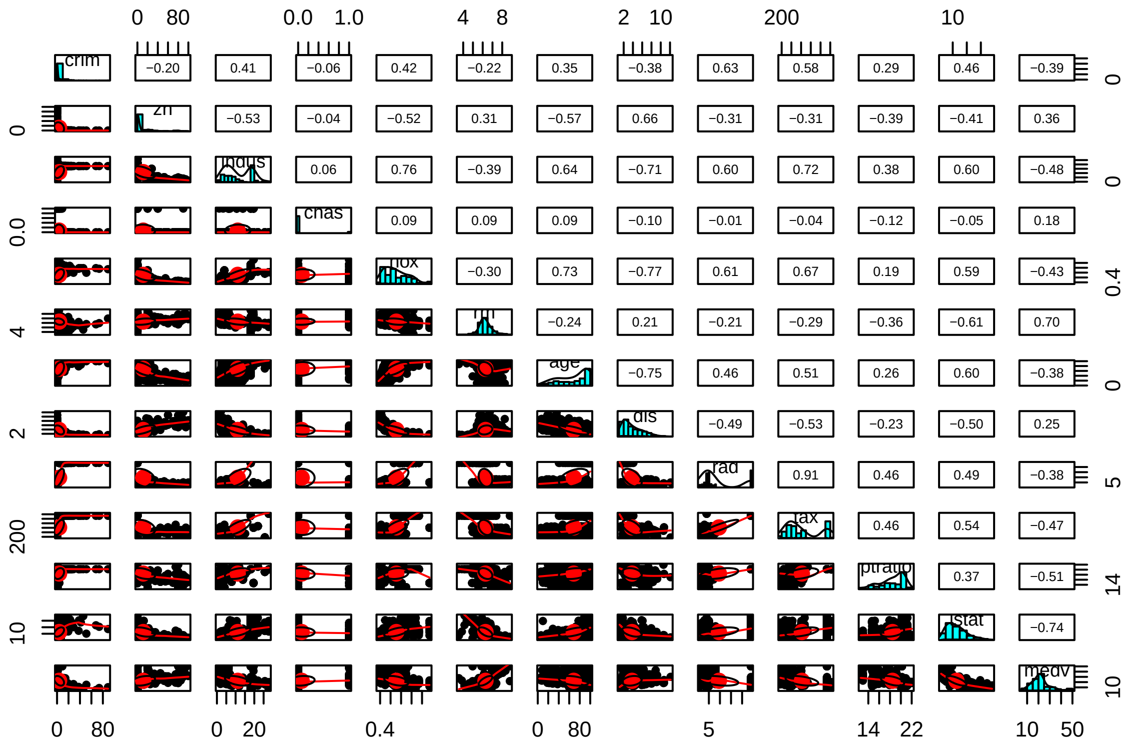
```
##  [1]  0.0457100386 -0.0583501107 -0.8253775522 -9.9575865471  0.6289106622
##  [6] -0.0008482791 -1.0122467382  0.6124653115 -0.0037756465 -0.3040727572
## [11]  0.1388005968 -0.2200563590
```

```
plot(individual_predictor_coefficients, all_predictors_coefficients)
```



d

```
library(psych)
pairs.panels(Boston)
```



There is evidence of non-linear association between all of these predictors and the response.

```
summary(lm(crim ~ poly(zn, 3), data = Boston))
```

```
##
## Call:
## lm(formula = crim ~ poly(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)    3.6135     0.3722   9.709  < 2e-16 ***
## poly(zn, 3)1 -38.7498     8.3722  -4.628  4.7e-06 ***
## poly(zn, 3)2  23.9398     8.3722   2.859  0.00442 **
## poly(zn, 3)3 -10.0719     8.3722  -1.203  0.22954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.372 on 502 degrees of freedom
## Multiple R-squared:  0.05824,    Adjusted R-squared:  0.05261
## F-statistic: 10.35 on 3 and 502 DF,  p-value: 1.281e-06
```

```r
summary(lm(crim ~ poly(indus, 3), data = Boston))
```

```
##
## Call:
## lm(formula = crim ~ poly(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)        3.614      0.330  10.950  < 2e-16 ***
## poly(indus, 3)1   78.591      7.423  10.587  < 2e-16 ***
## poly(indus, 3)2  -24.395      7.423  -3.286  0.00109 **
## poly(indus, 3)3  -54.130      7.423  -7.292  1.2e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.423 on 502 degrees of freedom
## Multiple R-squared:  0.2597, Adjusted R-squared:  0.2552
## F-statistic: 58.69 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
summary(lm(crim ~ poly(nox, 3), data = Boston))
```

```
##
## Call:
## lm(formula = crim ~ poly(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)     3.6135     0.3216  11.237  < 2e-16 ***
## poly(nox, 3)1  81.3720     7.2336  11.249  < 2e-16 ***
## poly(nox, 3)2 -28.8286     7.2336  -3.985 7.74e-05 ***
## poly(nox, 3)3 -60.3619     7.2336  -8.345 6.96e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.234 on 502 degrees of freedom
## Multiple R-squared:  0.297,  Adjusted R-squared:  0.2928
## F-statistic: 70.69 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
summary(lm(crim ~ poly(rm, 3), data = Boston))
```

```
##
## Call:
## lm(formula = crim ~ poly(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)    3.6135     0.3703   9.758  < 2e-16 ***
## poly(rm, 3)1 -42.3794     8.3297  -5.088 5.13e-07 ***
## poly(rm, 3)2  26.5768     8.3297   3.191  0.00151 **
## poly(rm, 3)3  -5.5103     8.3297  -0.662  0.50858
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.33 on 502 degrees of freedom
## Multiple R-squared:  0.06779,    Adjusted R-squared:  0.06222
## F-statistic: 12.17 on 3 and 502 DF,  p-value: 1.067e-07
```

```r
summary(lm(crim ~ poly(age, 3), data = Boston))
```

```
##
## Call:
## lm(formula = crim ~ poly(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)     3.6135     0.3485  10.368  < 2e-16 ***
## poly(age, 3)1  68.1820     7.8397   8.697  < 2e-16 ***
## poly(age, 3)2  37.4845     7.8397   4.781 2.29e-06 ***
## poly(age, 3)3  21.3532     7.8397   2.724  0.00668 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.84 on 502 degrees of freedom
## Multiple R-squared:  0.1742, Adjusted R-squared:  0.1693
## F-statistic: 35.31 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
summary(lm(crim ~ poly(dis, 3), data = Boston))
```

```
##
## Call:
## lm(formula = crim ~ poly(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)     3.6135     0.3259  11.087  < 2e-16 ***
## poly(dis, 3)1 -73.3886     7.3315 -10.010  < 2e-16 ***
## poly(dis, 3)2  56.3730     7.3315   7.689 7.87e-14 ***
## poly(dis, 3)3 -42.6219     7.3315  -5.814 1.09e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.331 on 502 degrees of freedom
## Multiple R-squared:  0.2778, Adjusted R-squared:  0.2735
## F-statistic: 64.37 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
summary(lm(crim ~ poly(rad, 3), data = Boston))
```

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

```r
summary(lm(crim ~ poly(tax, 3), data = Boston))
```

```
##
## Call:
## lm(formula = crim ~ poly(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)     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
```

```
summary(lm(crim ~ poly(ptratio, 3), data = Boston))
```

```
##
## Call:
## lm(formula = crim ~ poly(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)          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
```

```
summary(lm(crim ~ poly(lstat, 3), data = Boston))
```

```
##
## Call:
## lm(formula = crim ~ poly(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)       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
```

```
summary(lm(crim ~ poly(medv, 3), data = Boston))
```

```
##
## Call:
## lm(formula = crim ~ poly(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)       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
```