Introduction to R: Simple linear regression

Ziv Shkedy, Hasselt University (July 2020)

```
## Warning: package 'mvtnorm' was built under R version 3.6.2
```

Introduction

Slides, code and tutorials

This chapter of the interactive book contains all R code that was used to produce the results and output presented in chapter 3 (modeling: simple linear regression) in the course's slides. We include YouTube tutorials as a part of the chapter and links to the relevant tutorials are provided. Note that these tutorials were not developed especially for this book, they cover the same topics using different examples.

R ?

No previous knowledge about R is required. We use the R function lm() to a fit simple linear regression model R and the cars dataset cars is used for illustraions. The model we discussed in this chapter can be fitted using the function glm() as well.

Slides

Slide for this part of the course are avilable online in the >eR-BioStat website. See RcourseModeling.

The cars data

The cars data gives the speed of cars (the response) and the distances taken to stop (the predictor). Note that the data were recorded in the 1920s. The data are given as a data frame (cars) in R.

```
x<-cars$speed
y<-cars$dist
head(cars)
```

```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

Fitting Simple Linear regression Model in R

YouTube tutorials: Simple linear regression in R

R - Simple Linear Regression (part 1)

For a YouTube tutorial about simple linear regression in R by Jalayer Academy see YTREgression1a.

R - Simple Linear Regression (part 2)

For a YouTube tutorial about simple linear regression by Jalayer Academy in R see YTREgression1b.

Simple linear regression in R | R Tutorial 5.1

For a YouTube tutorial about simple linear regression in R by MarinStatsLectures see YTREgression2.

The lm() Function

The R function which we use to fit a linear regression model is lm() . A General call of the function has the form of lm(dependent variable~predictor(s)) .

Scatterplot

Figure @ref(fig:fig1) shows the scatterplot of Y (stopping distance) versus X (car's speed).

```
plot(x, y)
title("Y vs. X")
```

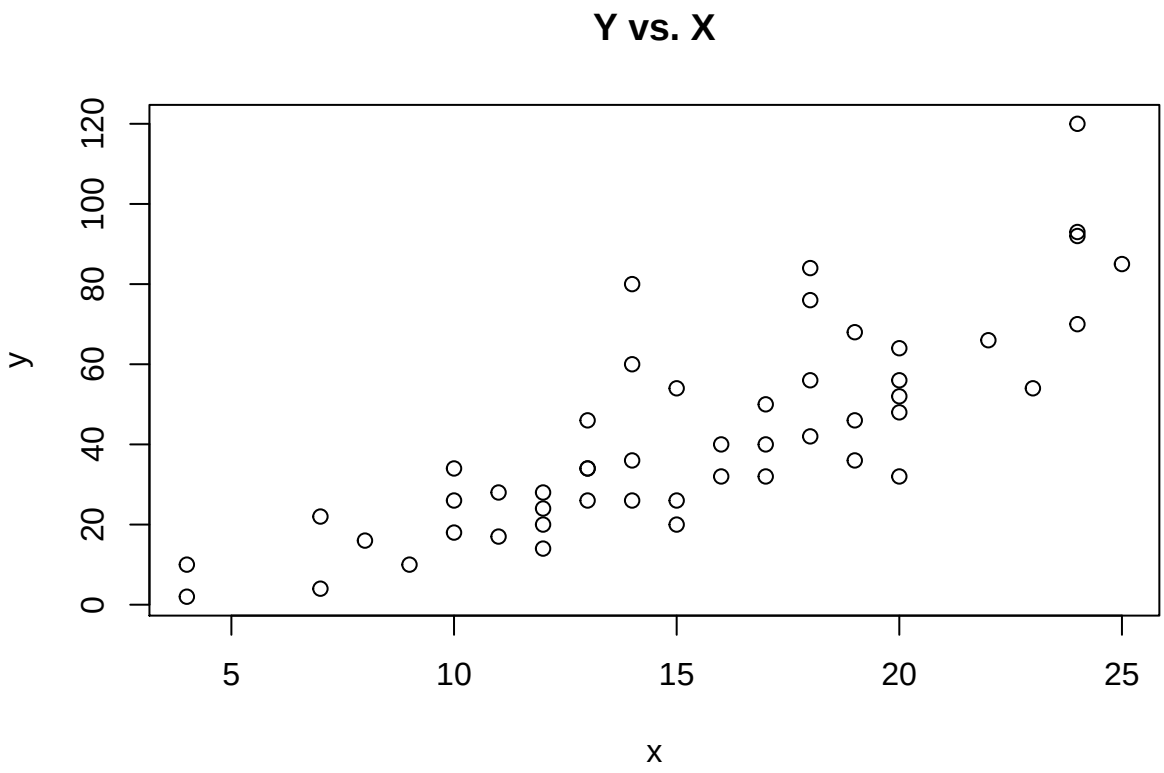


Figure 1: Speed (X) versus stopping ditance (Y).

Fitting the regression model in R

In order to fit a simplie linear regression model,

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i, \;\; i = 1, \ldots, n,$$

we use the lm() function in the following way:

```
fit.LM <- lm(y ~ x)
```

The R object fit.LM contains the results of the estimated model.

```
fit.LM
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##     -17.579        3.932
```

Parameter estimates: inference

Parameter estimates, standard errors, t-tests (and p-values) are obtained using the function summary()

```
summary(fit.LM)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## x             3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

We use the function anova() in order to obtain the ANOVA table for the model and F-test.

```
anova(fit.LM)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value   Pr(>F)
## x          1  21186 21185.5  89.567 1.49e-12 ***
## Residuals 48  11354   236.5
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residuals and model diagnostic

Data and fitted model

The object fit.LM$fit contains the fitted values.

```
fit.LM$fit
```

```
##         1         2         3         4         5         6         7         8
## -1.849460 -1.849460  9.947766  9.947766 13.880175 17.812584 21.744993 21.744993
##         9        10        11        12        13        14        15        16
## 21.744993 25.677401 25.677401 29.609810 29.609810 29.609810 29.609810 33.542219
##        17        18        19        20        21        22        23        24
## 33.542219 33.542219 33.542219 37.474628 37.474628 37.474628 37.474628 41.407036
##        25        26        27        28        29        30        31        32
## 41.407036 41.407036 45.339445 45.339445 49.271854 49.271854 49.271854 53.204263
##        33        34        35        36        37        38        39        40
## 53.204263 53.204263 53.204263 57.136672 57.136672 57.136672 61.069080 61.069080
##        41        42        43        44        45        46        47        48
## 61.069080 61.069080 61.069080 68.933898 72.866307 76.798715 76.798715 76.798715
##        49        50
## 76.798715 80.731124
```

Figure @ref(fig:fig2)shows the scaterplot of Y versus X and the fitted values.

```
plot(x, y)
lines(x, fit.LM$fit)
title("data and fitted model")
```

Note that the straight line is the estimated model given by

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i.$$

Distribution of the residuals

The object fit.LM$resid contains the residuals: $e_i = Y_i - \hat{Y}_i$.

```
fit.LM$resid
```

```
##          1          2          3          4          5          6          7
##   3.849460  11.849460  -5.947766  12.052234   2.119825  -7.812584  -3.744993
##          8          9         10         11         12         13         14
##   4.255007  12.255007  -8.677401   2.322599 -15.609810  -9.609810  -5.609810
##         15         16         17         18         19         20         21
##  -1.609810  -7.542219   0.457781   0.457781  12.457781 -11.474628  -1.474628
##         22         23         24         25         26         27         28
##  22.525372  42.525372 -21.407036 -15.407036  12.592964 -13.339445  -5.339445
##         29         30         31         32         33         34         35
## -17.271854  -9.271854   0.728146 -11.204263   2.795737  22.795737  30.795737
##         36         37         38         39         40         41         42
## -21.136672 -11.136672  10.863328 -29.069080 -13.069080  -9.069080  -5.069080
```

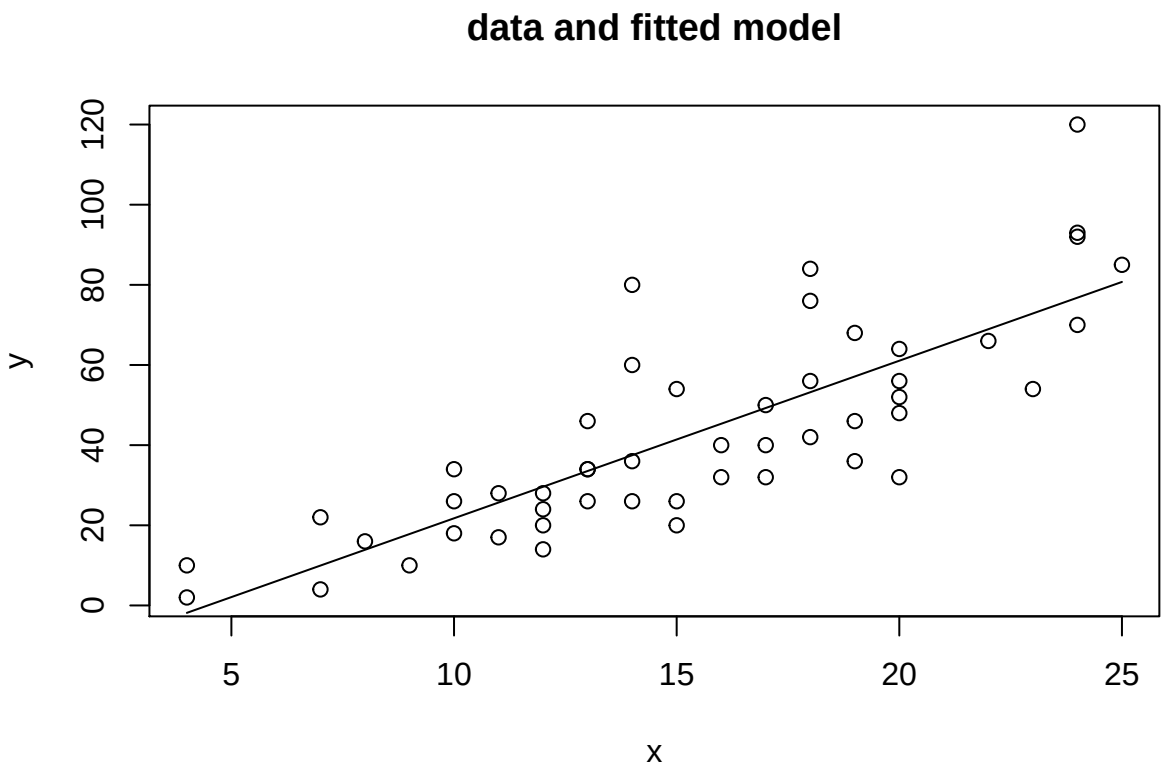


Figure 2: Data and fitted model.

```
##         43         44         45         46         47         48         49
##   2.930920  -2.933898 -18.866307  -6.798715  15.201285  16.201285  43.201285
##         50
##   4.268876
```

Histogram and normal probability plot for the residuals are shown in Figure @ref(fig:fig3).

```
par(mfrow = c(1, 2))
hist(fit.LM$resid)
qqnorm(fit.LM$resid)
```

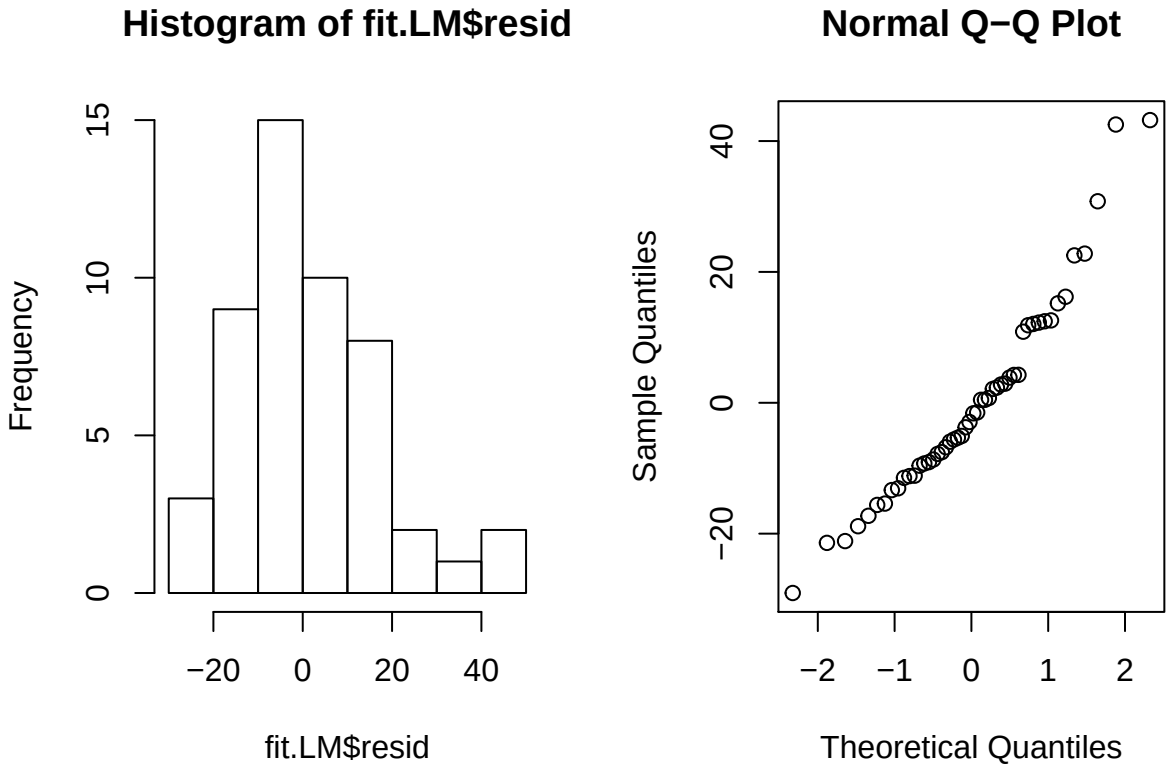
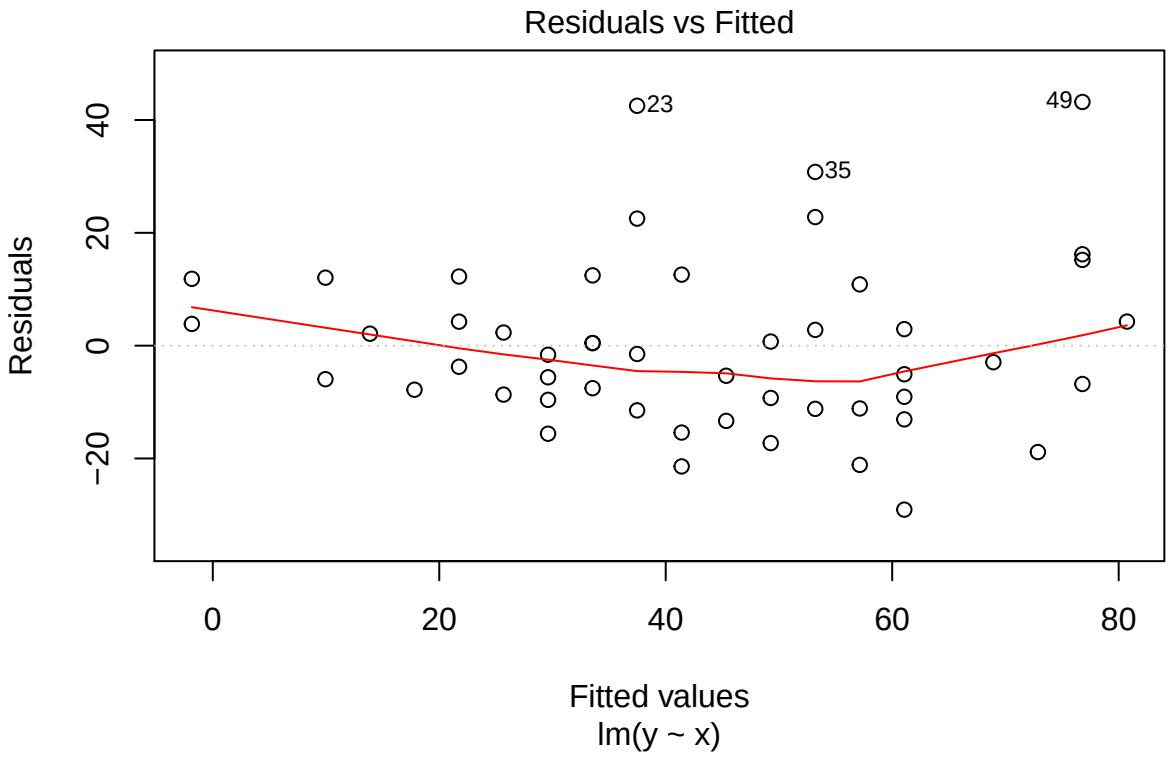
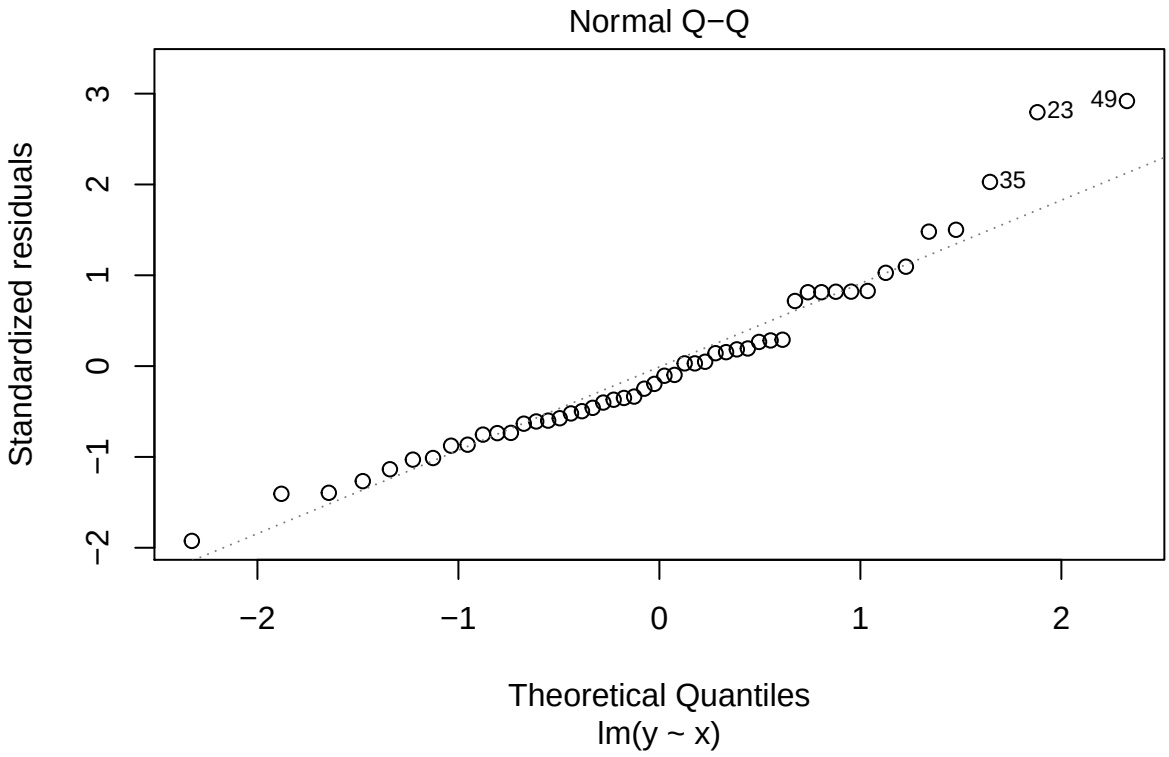


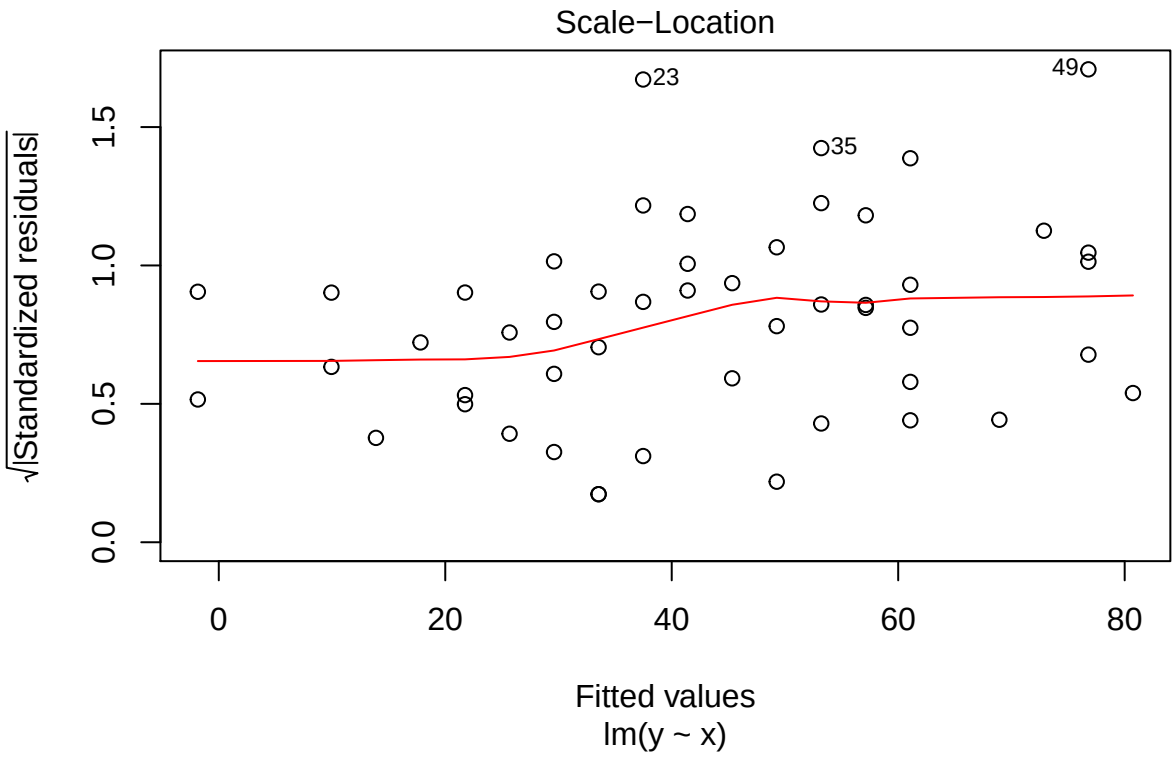
Figure 3: Distribution of the residuals.

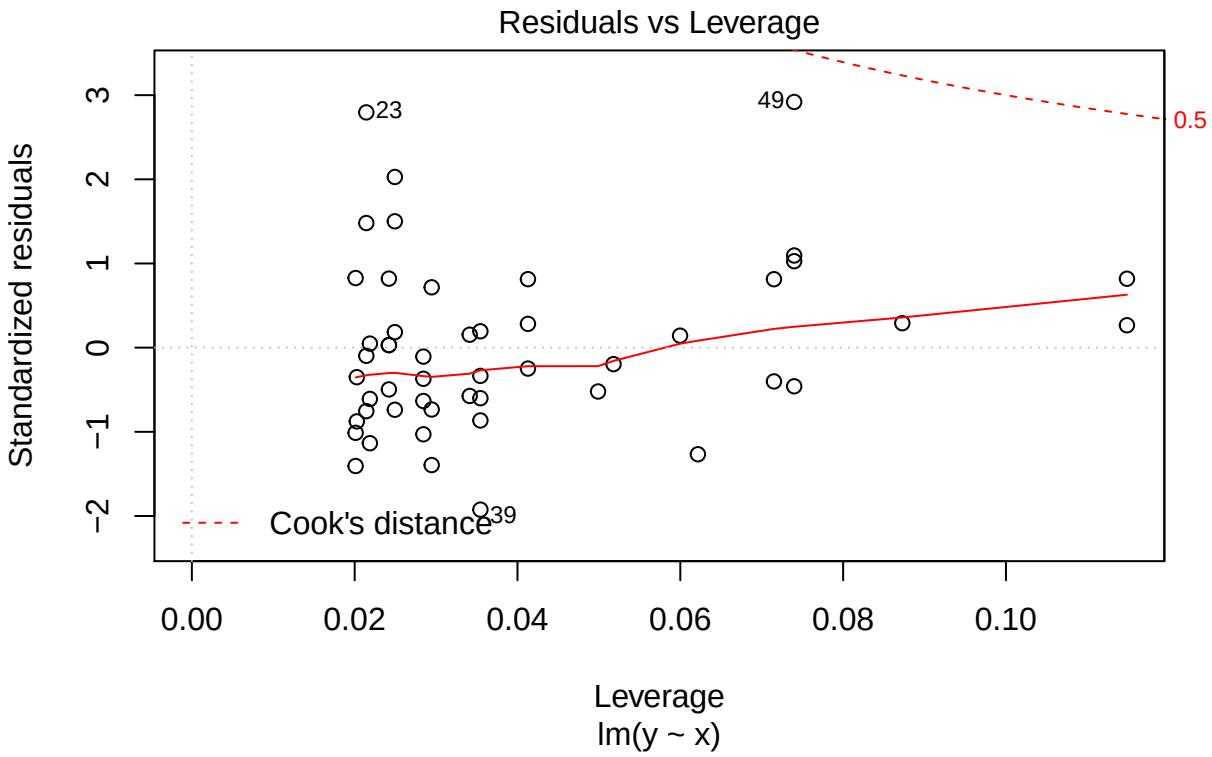
A set of diagnostic plots can be produced using the function plot() with the object fit.LM .

```
plot(fit.LM)
```


Residuals vs Fitted
Residuals
Fitted values
lm(y ~ x)
23
49
35
40
20
0
−20
0
20
40
60
80


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(y ~ x)
23
49
35
3
2
1
0
−1
−2
−2
−1
0
1
2


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
23
35
49
0
20
40
60
80
Fitted values
lm(y ~ x)


Residuals vs Leverage
Standardized residuals
3
2
1
0
−1
−2
23
49
39
0.5
Cook's distance
0.00
0.02
0.04
0.06
0.08
0.10
Leverage
lm(y ~ x)