A Handbook of Statistical Analyses Using R — 2nd Edition

Brian S. Everitt and Torsten Hothorn

CHAPTER 6

Simple and Multiple Linear Regression: How Old is the Universe and Cloud Seeding

6.1 Introduction

6.2 Simple Linear Regression

6.3 Multiple Linear Regression

6.3.1 Regression Diagnostics

6.4 Analysis Using R

Both the boxplots (Figure 6.1) and the scatterplots (Figure 6.2) show some evidence of outliers. The row names of the extreme observations in the `clouds` *data.frame* can be identified via

```
R> rownames(clouds)[clouds$rainfall %in% c(bxpseeding$out,
+                                           bxpecho$out)]
```

```
[1] "1"  "15"
```

where `bxpseeding` and `bxpecho` are variables created by `boxplot` in Figure 6.1. Now we shall not remove these observations but bear in mind during the modelling process that they may cause problems.

In this example it is sensible to assume that the effect that some of the other explanatory variables is modified by seeding and therefore consider a model that includes seeding as covariate and, furthermore, allows interaction terms for `seeding` with each of the covariates except `time`. This model can be described by the *formula*

```
R> clouds_formula <- rainfall ~ seeding +
+       seeding:(sne + cloudcover + prewetness + echomotion) +
+       time
```

and the design matrix $\mathbf{X}^\star$ can be computed via

```
R> Xstar <- model.matrix(clouds_formula, data = clouds)
```

By default, treatment contrasts have been applied to the dummy codings of the factors `seeding` and `echomotion` as can be seen from the inspection of the `contrasts` attribute of the model matrix

```
R> attr(Xstar, "contrasts")
```

```
R> data("clouds", package = "HSAUR2")
R> layout(matrix(1:2, nrow = 2))
R> bxpseeding <- boxplot(rainfall ~ seeding, data = clouds,
+      ylab = "Rainfall", xlab = "Seeding")
R> bxpecho <- boxplot(rainfall ~ echomotion, data = clouds,
+      ylab = "Rainfall", xlab = "Echo Motion")
```

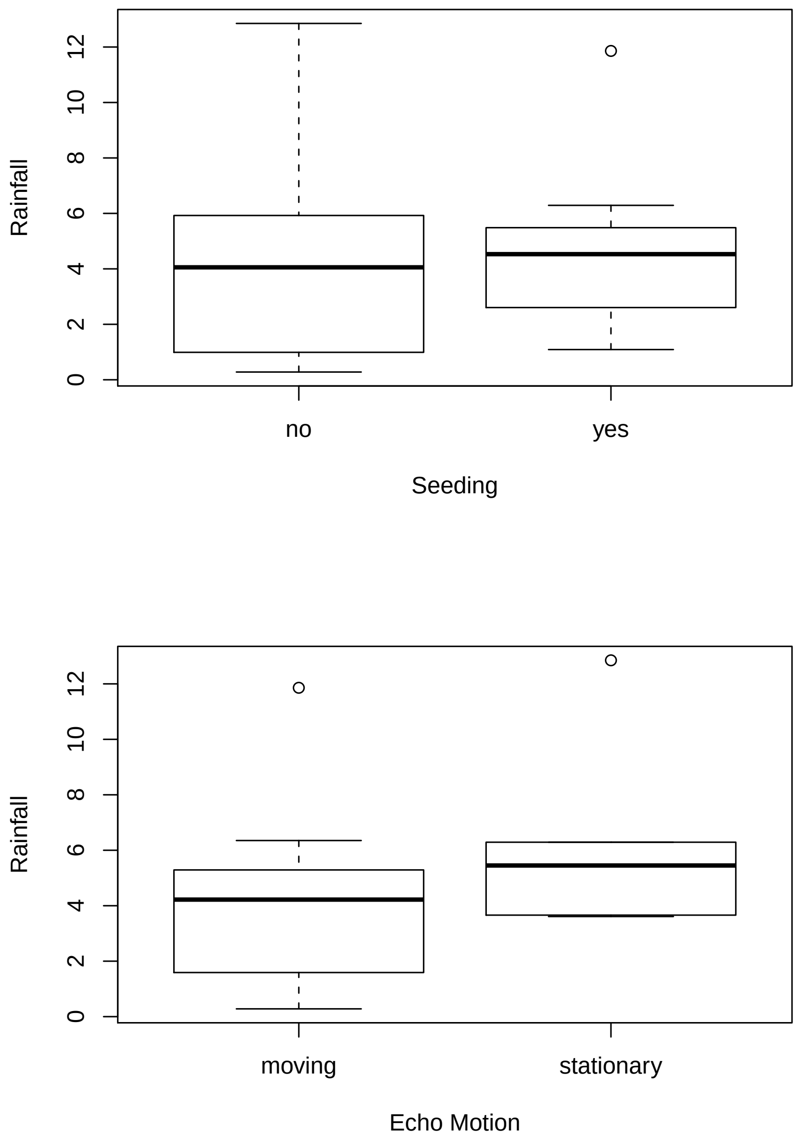

Figure 6.1 Boxplots of `rainfall`.

```
R> layout(matrix(1:4, nrow = 2))
R> plot(rainfall ~ time, data = clouds)
R> plot(rainfall ~ cloudcover, data = clouds)
R> plot(rainfall ~ sne, data = clouds, xlab="S-Ne criterion")
R> plot(rainfall ~ prewetness, data = clouds)
```

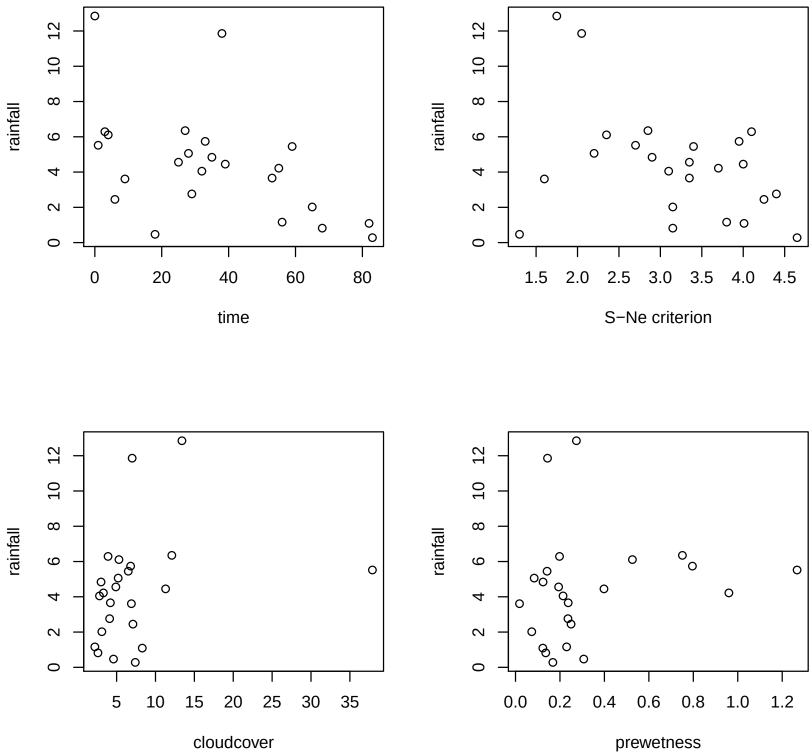

Figure 6.2 Scatterplots of `rainfall` against the continuous covariates.

```
$seeding
[1] "contr.treatment"

$echomotion
[1] "contr.treatment"
```

The default contrasts can be changed via the `contrasts.arg` argument to `model.matrix` or the `contrasts` argument to the fitting function, for example `lm` or `aov` as shown in Chapter 5.

However, such internals are hidden and performed by high-level model-fitting functions such as `lm` which will be used to fit the linear model defined by the *formula* `clouds_formula`:

```
R> clouds_lm <- lm(clouds_formula, data = clouds)
R> class(clouds_lm)
```

```
[1] "lm"
```

The results of the model fitting is an object of class *lm* for which a `summary` method showing the conventional regression analysis output is available. The output in Figure 6.3 shows the estimates $\hat{\beta}^\star$ with corresponding standard errors and t-statistics as well as the F-statistic with associated p-value.

Many methods are available for extracting components of the fitted model. The estimates $\hat{\beta}^\star$ can be assessed via

```
R> betastar <- coef(clouds_lm)
R> betastar
```

```
                      (Intercept) 
                          -0.3462 
                      seedingyes 
                          15.6829 
                             time 
                          -0.0450 
                   seedingno:sne 
                           0.4198 
                  seedingyes:sne 
                          -2.7774 
            seedingno:cloudcover 
                           0.3879 
           seedingyes:cloudcover 
                          -0.0984 
            seedingno:prewetness 
                           4.1083 
           seedingyes:prewetness 
                           1.5513 
 seedingno:echomotionstationary 
                           3.1528 
seedingyes:echomotionstationary 
                           2.5906 
```

and the corresponding covariance matrix $\mathsf{Cov}(\hat{\beta}^\star)$ is available from the `vcov` method

```
R> Vbetastar <- vcov(clouds_lm)
```

where the square roots of the diagonal elements are the standard errors as shown in Figure 6.3

```
R> sqrt(diag(Vbetastar))
```

```
                      (Intercept) 
                           2.7877 
                      seedingyes 
```

```
R> summary(clouds_lm)

Call:
lm(formula = clouds_formula, data = clouds)

Residuals:
   Min     1Q Median     3Q    Max 
-2.526 -1.149 -0.270  1.040  4.391 

Coefficients:
                                Estimate Std. Error t value
(Intercept)                      -0.3462     2.7877   -0.12
seedingyes                       15.6829     4.4463    3.53
time                             -0.0450     0.0251   -1.80
seedingno:sne                     0.4198     0.8445    0.50
seedingyes:sne                   -2.7774     0.9284   -2.99
seedingno:cloudcover              0.3879     0.2179    1.78
seedingyes:cloudcover            -0.0984     0.1103   -0.89
seedingno:prewetness              4.1083     3.6010    1.14
seedingyes:prewetness             1.5513     2.6929    0.58
seedingno:echomotionstationary    3.1528     1.9325    1.63
seedingyes:echomotionstationary   2.5906     1.8173    1.43
                                Pr(>|t|)
(Intercept)                       0.9031
seedingyes                        0.0037
time                              0.0959
seedingno:sne                     0.6274
seedingyes:sne                    0.0104
seedingno:cloudcover              0.0984
seedingyes:cloudcover             0.3885
seedingno:prewetness              0.2745
seedingyes:prewetness             0.5744
seedingno:echomotionstationary    0.1268
seedingyes:echomotionstationary   0.1776

Residual standard error: 2.2 on 13 degrees of freedom
Multiple R-squared: 0.716,	Adjusted R-squared: 0.497 
F-statistic: 3.27 on 10 and 13 DF,  p-value: 0.0243
```

Figure 6.3 R output of the linear model fit for the `clouds` data.

```
         4.4463
           time
         0.0251
  seedingno:sne
         0.8445
 seedingyes:sne
         0.9284
```

```
            seedingno:cloudcover
                          0.2179
           seedingyes:cloudcover
                          0.1103
            seedingno:prewetness
                          3.6010
           seedingyes:prewetness
                          2.6929
 seedingno:echomotionstationary
                          1.9325
seedingyes:echomotionstationary
                          1.8173
```

In order to investigate the quality of the model fit, we need access to the residuals and the fitted values. The residuals can be found by the `residuals` method and the fitted values of the response from the `fitted` (or `predict`) method

```
R> clouds_resid <- residuals(clouds_lm)
R> clouds_fitted <- fitted(clouds_lm)
```

Now the residuals and the fitted values can be used to construct diagnostic plots; for example the residual plot in Figure 6.5 where each observation is labelled by its number. Observations 1 and 15 give rather large residual values and the data should perhaps be reanalysed after these two observations are removed. The normal probability plot of the residuals shown in Figure 6.6 shows a reasonable agreement between theoretical and sample quantiles, however, observations 1 and 15 are extreme again.

An index plot of the Cook's distances for each observation (and many other plots including those constructed above from using the basic functions) can be found from applying the `plot` method to the object that results from the application of the `lm` function. Figure 6.7 suggests that observations 2 and 18 have undue influence on the estimated regression coefficients, but the two outliers identified previously do not. Again it may be useful to look at the results after these two observations have been removed (see Exercise 6.2).

```
R> psymb <- as.numeric(clouds$seeding)
R> plot(rainfall ~ sne, data = clouds, pch = psymb,
+       xlab = "S-Ne criterion")
R> abline(lm(rainfall ~ sne, data = clouds,
+            subset = seeding == "no"))
R> abline(lm(rainfall ~ sne, data = clouds,
+            subset = seeding == "yes"), lty = 2)
R> legend("topright", legend = c("No seeding", "Seeding"),
+         pch = 1:2, lty = 1:2, bty = "n")
```

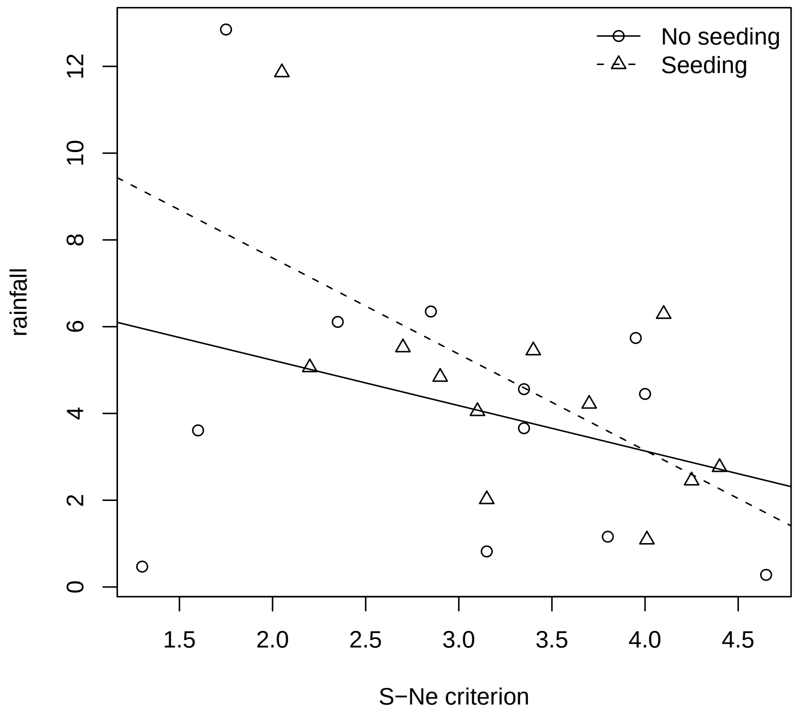

Figure 6.4 Regression relationship between S-Ne criterion and rainfall with and without seeding.

```
R> plot(clouds_fitted, clouds_resid, xlab = "Fitted values",
+       ylab = "Residuals", type = "n",
+       ylim = max(abs(clouds_resid)) * c(-1, 1))
R> abline(h = 0, lty = 2)
R> text(clouds_fitted, clouds_resid, labels = rownames(clouds))
```

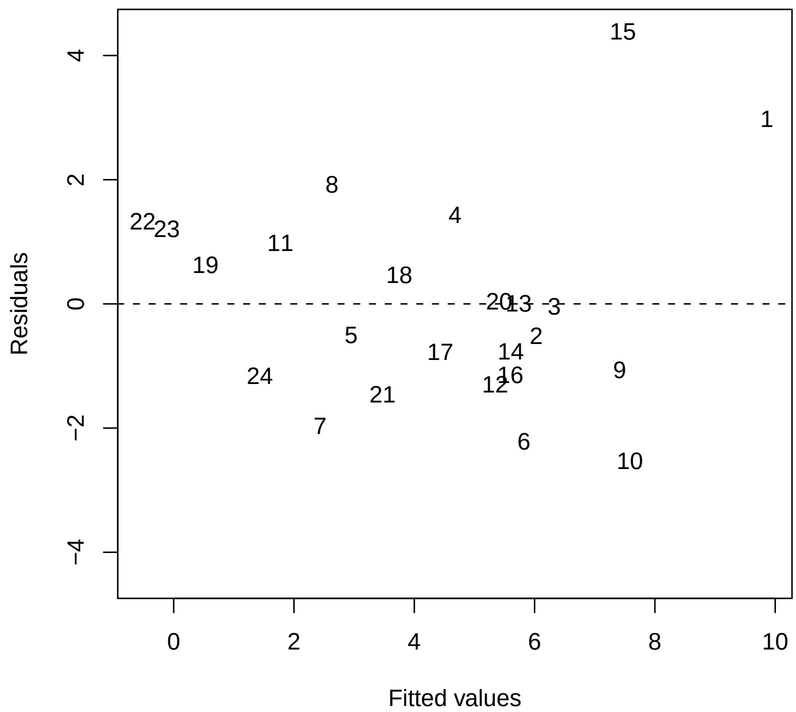

Figure 6.5 Plot of residuals against fitted values for `clouds` seeding data.

```
R> qqnorm(clouds_resid, ylab = "Residuals")
R> qqline(clouds_resid)
```

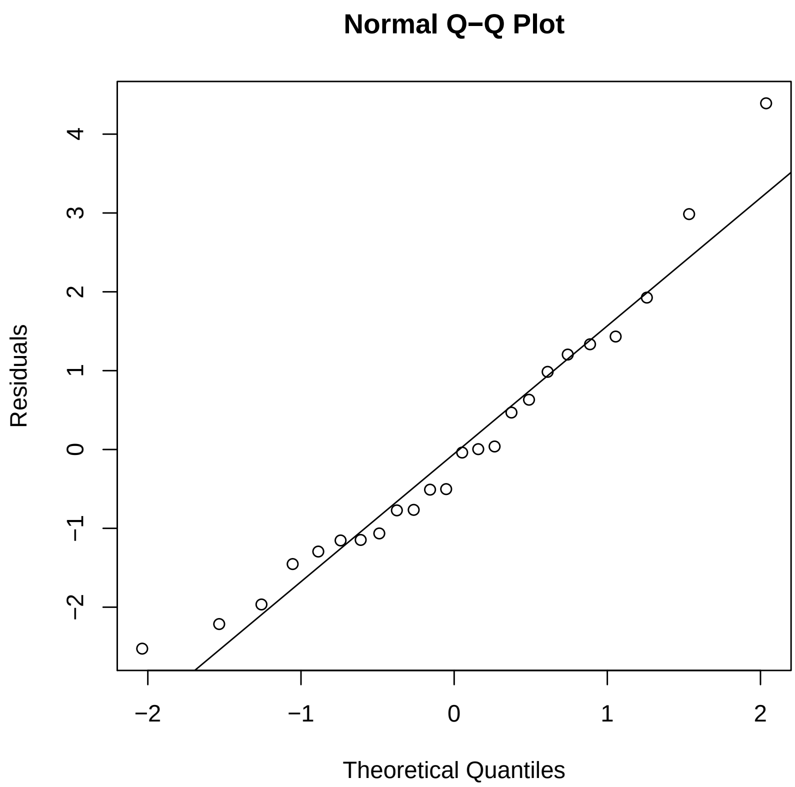

Figure 6.6 Normal probability plot of residuals from cloud seeding model `clouds_lm`.

```
R> plot(clouds_lm)
```

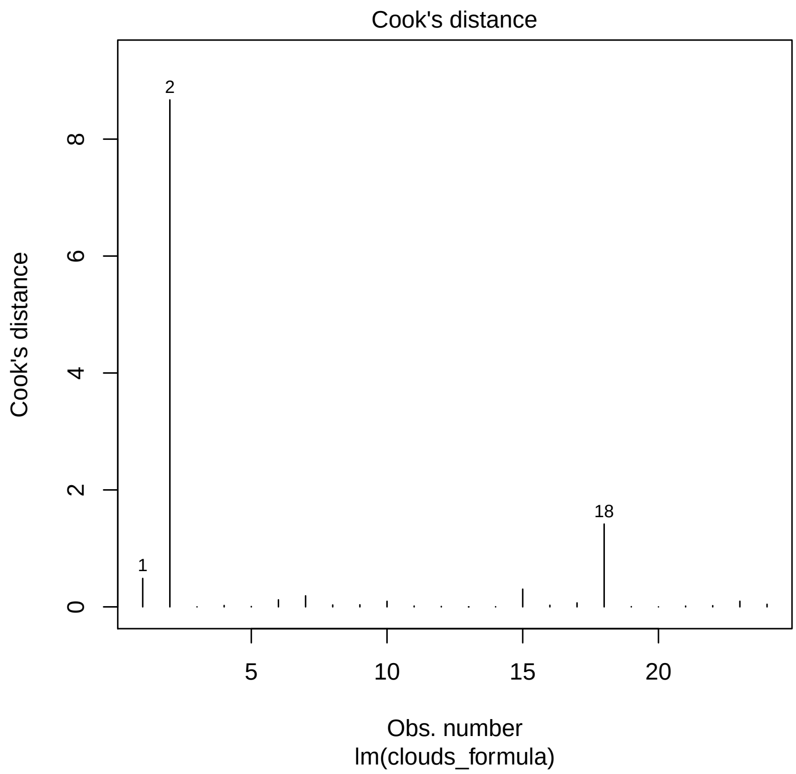

Figure 6.7 Index plot of Cook's distances for cloud seeding data.