Beispiel in R: Einfache lineare Regression

Regina Tüchler

2006-10-09

Die einfache lineare Regression erklärt eine Responsevariable durch eine lineare Funktion einer Prädiktorvariable. Wir führen eine lineare Regression an einem einfachen Beispiel durch und definieren 2 Variable `x` und `y`:

```
> x <- c(-2, -1, -0.8, -0.3, 0, 0.5, 0.6, 0.7, 1, 1.2)
> y <- c(1.9, 0.6, 0.5, 0.8, -0.4, -0.9, -0.7, -0.1, -1.7, -0.2)
```

Streudiagramm:

Wir zeichnen diese beiden Variable in ein Streudiagramm. Der Befehl dazu ist wieder `plot`. (Wenn wir in einen schon bestehenden Plot zusätzlich Punkte einzeichnen wollten, müssten wir den Befehl `points` verwenden.)

```
> plot(x, y, xlim = c(-3, 3), ylim = c(-3, 3), pch = 19)
```

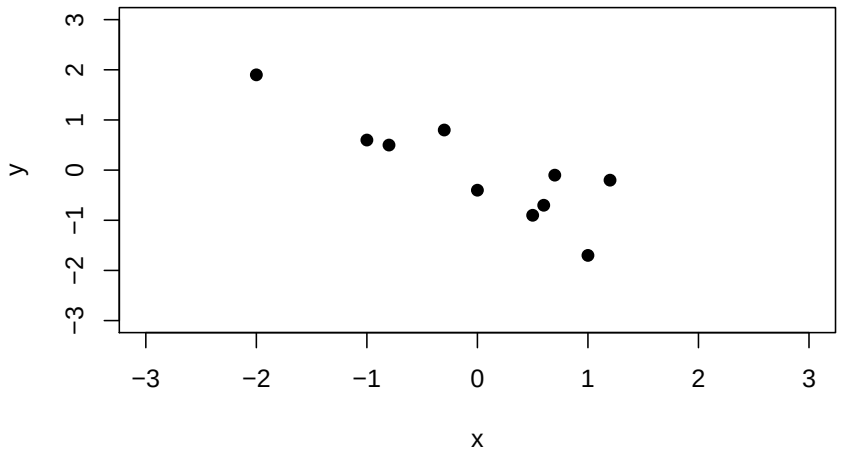

Figure 1: Streudiagramm von x gegen y.

Es würde auch einfach der Befehl `plot(x,y)` ausreichen. Dann würde `R` das Layout des Plots automatisch wählen. In Fig. 1 haben wir aber zusätzlich die Option `xlim=c(-3,3)` verwendet, sodass die x-Achse von -3 bis 3 gezeichnet wurde. Analog y-Achse. Die Option `pch=19` zeichnet

volle Punkte. Mit der Option `cex` könnte man die Punktgröße steuern.

KQ-Schätzung:

Zur Modellschätzung in linearen Modellen steht in `R` die Funktion `lm` ("linear model") zur Verfügung. Diese kann auch für allgemeinere (und daher kompliziertere Modelle als wir sie in dieser LV kennenlernen werden) verwendet werden. Daher gibt es viele Optionen, die wir hier nicht brauchen werden (s. help-Menü).

In dieser LV werden wir die Funktion wie folgt verwenden: Man muss in der Klammer des Funktionsaufrufs angeben nach welcher Formel die Modellschätzung durchgeführt werden soll. In unserem einfachen Regressionsbeispiel ist das die Formel:

$$\hat{y}_i = \beta_1 + \beta_2 x_i$$

Daher schaut die Option so aus:

$$y \sim x.$$

Es wird automatisch auch der Parameter β_1 mitgeschätzt. Man kann diese Konstante auch esplizit in der Formel berücksichtigen und erhält dasselbe Ergebnis mit der Option:

$$y \sim 1 + x.$$

Falls man ein Modell ohne Konstante rechnen wollte, muss man sie ausschließen. Dazu gibt es die folgenden beiden Möglichkeiten: $y \sim x - 1$ oder $y \sim x + 0$.

Wir speichern das Ergebnis unter dem Namen `fm` ("fitted model") ab:

```
> fm <- lm(y ~ x)

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
   -0.02855     -0.85468
```

Die folgenden Ergebnisse aus `fm` werden wir verwenden:

In `coefficients(fm)` stehen die Regressionskoeffizienten β_1 unter `(Intercept)` zur Konstanten gehörend, und β_2 zur Variablen x gehörend. Die Regressionsgerade hat daher die folgende Gestelt: y_i = -0.03 + -0.85 $x + \varepsilon_i$.

Die Prognosewerte für die Responsevariable $\hat{y}_i$ stehen in

```
> fitted.values(fm)

          1           2           3           4           5           6
 1.68080699  0.82613011  0.65519474  0.22785630 -0.02854677 -0.45588521
          7           8           9          10
-0.54135290 -0.62682058 -0.88322365 -1.05415902
```

Die Residuen ε_i stehen unter

```
> residuals(fm)

         1          2          3          4          5          6          7
 0.2191930 -0.2261301 -0.1551947  0.5721437 -0.3714532 -0.4441148 -0.1586471
         8          9         10
 0.5268206 -0.8167764  0.8541590
```

```
> plot(x, y, xlim = c(-3, 3), ylim = c(-3, 3), pch = 19)
> abline(fm, col = "red")
```

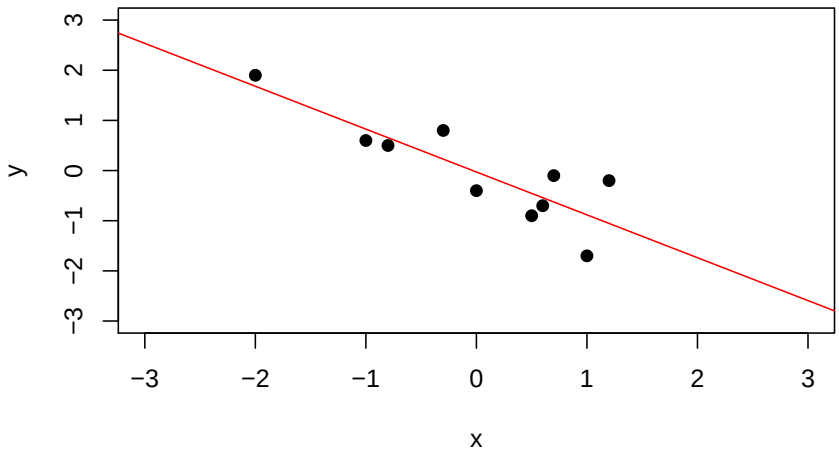

Figure 2: Regressionsgerade.

In Fig. 2 wurde in das schon bestehende Streudiagramm die Regressionsgerade mit `abline` als rote Linie einzeichnen.

Prognose:

Aus der Regressionsgeraden kann man Prognosen für neue Untersuchungseinheiten machen. Z.B. Wird der Prognosewert $\hat{y}$ an der Stelle -1.5 berechnet als:
$\hat{y} =$ -0.03 + -0.85 $\cdot(-1.5) = 1.25$.