Beispiel in R: Prognose im multiplen Regressionsmodell

Regina Tüchler

2006-10-09

```
> load("statlab.rda")
> attach(statlab)
```

Wir betrachten das Modell, in dem die Größe des Kindes durch die Größe von Vater und Mutter erklärt wird:

$$CTHGHT_i = \beta_1 + \beta_2 MTHGHT_i + \beta_3 FTHGHT_i + \varepsilon_i$$

(vgl. Übung 2). Wir bezeichen die Daten mit $y, x = (x2, x3)$:

```
> y <- CTHGHT
> x2 <- MTHGHT
> x3 <- FTHGHT
```

Wir erstellen für drei Szenarien eine Prognose für die Größe von Kindern: mit Eltern, die $x_{1297} = (58, 70); x_{1298} = (62, 65); x_{1299} = (66, 75)$ inch groß sind. Die Angabe für die zukünftigen Szenarien erfolgt durch die Angabe der geordneten Paare "(Größe d. Mutter, Größe d. Vaters)". Da wir 1296 Werte in der ursprünglichen Stichprobe haben, vergeben wir für diese 3 Szenarien die Nummern $1297, 1298, 1299$.

Punktprognose:

Wir rechnen zunächst das Regressionsmodell:

```
> fm <- lm(y ~ x2 + x3)

Call:
lm(formula = y ~ x2 + x3)

Coefficients:
(Intercept)           x2           x3
    16.6082       0.2936       0.2558
```

Die Punktprognose kann man durch Einsetzen der neuen Werte, für die die Prognose erstellt werden soll, berechnen:
$\hat{y}_{1297} = 16.61 + 0.29 \cdot 58 + 0.26 \cdot 70 = 51.55$
$\hat{y}_{1298} = 16.61 + 0.29 \cdot 62 + 0.26 \cdot 65 = 51.44$
$\hat{y}_{1299} = 16.61 + 0.29 \cdot 66 + 0.26 \cdot 75 = 55.17$

Dieses Einsetzen kann mit Hilfe der `R`-Funktion `predict` automatisch ausgeführt werden. Für die Punktprognose müssen folgende Argumente angegeben werden:

Das Objekt vom Typ `lm`: hier unter `fm` abgespeichert .

Die Daten der neuen Szenarien, für die die Prognose erstellt werden soll. Dabei ist es wichtig, dass diese Daten in einem `R-data frame` sind. Daher werden wir die Daten der Prädiktorvariablen, für die die Prognose erstellt werden soll, mit Hilfe des Befehle `data.frame` zusammensetzen. Es müssen auch die Variablennamen `x2` und `x3` richtig vergeben werden. (Vgl. dazu die "Einf. in R" zu Beginn der LV, wo wir verschiedene Datenstrukturen kennengelernt haben. `data.frame` definiert eine Liste von Datenpunkten und stellt einen weiteren Typ von Datenstruktur in R dar.)

```
> new <- data.frame(x2 = c(58, 62, 66), x3 = c(70, 65, 75))

  x2 x3
1 58 70
2 62 65
3 66 75
```

Die Punktprognosen stimmen mit den obigen Werten überein:

```
> pred.point <- predict(fm, newdata = new)

       1        2        3
51.54592 51.44120 55.17397
```

Schwankungsbreite der Punktprognose:

Wir schätzen nur die Schwankungbreite mit 95 % Sicherheit. Es geht hier darum jene statistische Unsicherheit abzuschätzen, die durch die Unsicherheit der Parameterschätzungen von β_j zustande kommt. Es wird also die Abweichung der Punktschätzung, die ja auf den Schätzwerten $\hat{\beta}_j$ beruht, vom erwarteten Prognosewert, wenn man die wahren β_j zur Verfügung hätte, abgeschätzt.

Auch dazu kann man die Funktion `predict` verwenden, diesmal mit der zusätzlichen Option `interval="confidence"`. Wenn man sich auch noch die verwendete Standardabweichung $SD_{\hat{y}}$ ausgeben lassen will, setzt man auch die Option `se.fit=TRUE`.

```
> pred.conf <- predict(fm, newdata = new, se.fit = TRUE, interval = "confidence")

$fit
       fit      lwr      upr
1 51.54592 51.18968 51.90215
2 51.44120 51.18031 51.70209
3 55.17397 54.92725 55.42069

$se.fit
        1         2         3
0.1815837 0.1329836 0.1257621

$df
[1] 1293

$residual.scale
[1] 2.268084
```

Es wird nun unter `$fit` in der Spalte `fit` die Punktschätzung ausgegeben und in den Spalten `lwr` bzw. `uwr` stehen die untere bzw. obere Grenze der Schwankung. In `$se.fit` steht $SD_{\hat{y}}$ für die 3 Schätzungen. Weiters werden in `$df` die Anzahl der Freiheitsgrade $T - k$ ausgegeben, und in `$residual.scale` die geschätzte Varianz der Residuen $\hat{\sigma}^2$. Diese beiden letzten Werte werden erst bei der nun folgenden Schätzung der gesamten Unsicherheit für die Punktprognose verwendet:

Intervallprognose für unbekannte y_{1297}, y_{1298}, y_{1299}:

Es wird nun das Intervall geschätzt, das die gesamte Modellunsicherheit berücksichtigt. Auch dies kann mit Hilfe der Funktion `predict` gemacht werden (Signifikanzniveau wieder 95%):

Die Option `interval="prediction"` muss nun verwendet werden:

```
> pred.pred <- predict(fm, newdata = new, se.fit = TRUE, interval = "prediction")

$fit
       fit      lwr      upr
1 51.54592 47.08215 56.00968
2 51.44120 46.98403 55.89837
3 55.17397 50.71760 59.63033

$se.fit
        1         2         3
0.1815837 0.1329836 0.1257621

$df
[1] 1293

$residual.scale
[1] 2.268084
```

Die Interpretation der Ausgaben erfolgt ganz analog wie oben.