Thilo Ernst
Ulrich Pötter
Ruhr-Universität Bochum
Sommersemester 2006
Rechentechniken der Datenanalyse

Arbeitsblatt 5

1) Der Zusammenhang zwischen individuellem (Arbeits-) Einkommen (`I1111099`) und Alter (`D1110199`), Geschlecht (`D11102LL`), Schuljahren (`D1110999`) und Arbeitszeit (Stunden je Jahr) (`E1110199`) soll mit der 1999 Welle der PSID-Studie untersucht werden. Es sollen nur Beschäftigte `E1110299==1` der Hauptstichprobe (`X11104LL==11`) betrachtet werden, die auch zur Stichprobe gerechnet werden (`X1110399==1`).

```
X <- read.csv("../../cnef/psid/pequiv99.csv",head=T)
X <- subset(X, E1110299==1&X11104LL==11&X1110399==1,
            c(X11101LL,X1110299,D11102LL,
              D1110199,D1110999,E1110199,I1111099))
```

Wieviele Fälle bleiben übrig? In wievielen Fällen fehlen Angaben? Sind die Bereiche der Variablen plausibel? Wieviele Beschäftigte gibt es je Haushalt? Wieviele Haushalte sind in dieser Teilstichprobe?

2) Fehlende Werte:

`complete.cases(X)` erzeugt einen logischen Vektor der Länge `dim(X)[1]`, dessen Elemente den Wert `TRUE` genau dann haben, wenn die entsprechende Zeile eines `data.frame` (oder einer Matrix oder eines Vektors) keine fehlenden Werte (`NA` oder `NaN`) enthält.

`na.omit(X)` entfernt alle Zeilen aus einem `data.frame` oder einer Matrix (und aus Vektoren), die fehlende Werte enthalten.

```
X1 <- na.omit(X)
dim(X1)
```

Die Zeilennummern (und deren Namen), die durch `na.omit` ausgeschlossen werden, werden als Attribut des erzeugten Objekts mit dem Namen `na.action` gespeichert. Sie können durch

```
om1 <- attr(X1,"na.action")
```

angesprochen werden.

`na.exclude(X)` entfernt ebenfalls alle Zeilen mit fehlenden Werten. Beide Befehle ergeben numerisch identische Ergebnisse und halten beide die Zeilennummern der ausgeschlossenen Zeilen fest:

```
X2 <- na.exclude(X)
all(X1-X2==0)  # TRUE
om2 <- attr(X2,"na.action")
class(om1); class(om2)
```

Alle multivariaten statistischen Prozeduren verwenden zunächst `na.omit` oder `na.exclude`, bevor sie ihre Ergebnisse berechnen.

Der Unterschied besteht in der Behandlung fehlender Werte bei der Berechnung von Residuen und vorhergesagten Werten insbesondere in Regressionsmodellen: Bei `na.exclude` werden die ausgeschlossenen Fälle an der „richtigen“ Stelle wieder in die Vektoren der Residuen bzw. der vorhergesagten Werte aufgenommen.

Die Voreinstellung zur Behandlung fehlender Werte ist durch den Wert von `options("na.action")` gegeben und kann durch `options(na.action="na.exlude")` bzw. durch `options(na.action="na.omit")` verändert werden.

3) Lineare Regression:

Ein lineares Modell (multiple Regression mit der Methode der kleinsten Quadrate) wird mit der Funktion `lm()` berechnet:

```
attach(X1)
erg <- lm(I1111099 ~ D1110199 + D11102LL + D1110999 + E1110199)
summary(erg)
```

Die `lm()`-Funktion muss mindestens eine Formel enthalten, die das zu schätzende Modell beschreibt: Links von dem Zeichen `~` steht der Name der abhängigen Variablen, rechts davon die Namen der unabhängigen („erklärenden“) Variablen, die durch `+` verbunden werden.

`summary(erg)` erzeugt eine Zusammenfassung der Ergebnisse der Regression mit den geschätzten Koeffizienten, deren Standardfehler, t-Werten und deren beobachtetes Signifikanzniveau. Außerdem werden R^2 Werte und weitere Fitstatistiken ausgegeben.

Das Ergebnis eines Aufrufes von `lm()` ist eine Liste, die u.a. die folgenden Elemente enthält: `coefficients` (die Regressionskoeffizienten), `residuals` (die Residuen des Modells), `fitted.values` (die vorhergesagten Werte). Die

entsprechenden Elemente können also etwa durch `erg$coefficients` angesprochen werden.

Oft ist es aber einfacher, anstelle der Listenelemente entsprechende Funktionen zu verwenden, um auf Teilergebnisse zuzugreifen. Für alle Regressionsmodelle gibt es die folgenden Funktionen: `coef(erg)`, `resid(erg)`, `fitted(erg)`, `summary(erg)`, `vcov(erg)` (die geschätzte Kovarianzmatrix der geschätzten Parameter. Sie ist nicht direkt in der Liste der Ergebnisse (`erg`) enthalten), `predict(erg,newdata=ndat)`:

```
coef(erg)
kov <- vcov(erg)
stdfehler <- sqrt(diag(kov));stdfehler
```

4) Diagnostische Plots:

`plot(erg)` erzeugt vier diagnostische Plots, die es erlauben sollen, Probleme des Modells aufzuzeigen. Das erste Bild zeigt ein Scatterplot der Residuen geordnet nach den vorhergesagten Werten der abhängigen Variablen. Insbesondere sollten sich Hinweise auf Heteroskedastie sowie Ausreißer und Abweichungen von der linearen Form des Kovariableneinflusses finden lassen.

Das nächste Bild zeigt ein Q-Q Diagramm (Quantil-Quantil Diagramm) der Residuen im Vergleich mit den Quantilen einer Normalverteilung.

Bild 3 zeichnet die Wurzel der standardisierten Residuen gegen die vorhergesagten Werte. Das sollte die Form einer möglichen Heteroskedastie zeigen.

Im letzten Bild werden die standardisierten Residuen gegen den „Leverage“ der Beobachtungen (den „Abstand“ der unabhängigen Variablen der Beobachtungen von ihrem Mittelwert) abgetragen. Damit sollten sich auch Ausreißer in den Kovariablen identifizieren lassen.

5) Formeln und Transformationen:

Transformationen der Variablen können direkt in der Formel benutzt werden, die das Modell beschreibt:

```
erg2 <- lm(log(I1111099) ~ D1110199 + D11102LL +
           D1110999 + E1110199)
summary(erg2)
```

Allerdings haben einige arithmetische Operatoren in den Formeln, die Modelle beschreiben, besondere Bedeutung: `var1:var2` erzeugt einen Interaktionsterm zwischen `var1` und einem Faktor `var2`. `var1*var2` ist eine Abkürzung für `var1 + var2 + var1:var2`. Entsprechend kann auch `(var1 + var2)^2` geschrieben werden. Die letzte Form kann auch für beliebig viele Terme benutzt werden.

Um diese speziellen Formeloperatoren von den üblichen arithmetischen Operatoren zu unterscheiden, müssen die arithmetischen Operationen in `I()` eingeschlossen werden.

```
erg3 <- lm(log(I1111099) ~ D1110199*D11102LL + D1110999 +
           I(D1110999^2)+ E1110199)
summary(erg3)
```

`erg3` enthält also die Haupteffekte zwischen Alter und Geschlecht sowie eine quadratische Funktion der Schuljahre.

6) Weitere Diagnostiken:

`einfl <- influence(erg)` liefert eine Liste mit den Elementen: `hat` (der Einfluss der Kovariablenvektoren jedes Falls), `coefficients` (eine Matrix, die die Änderung in den geschätzten Koeffizienten angibt, falls die i-te Beobachtung aus dem Modell ausgeschlossen wird. Diese Größe wird auch Einflussfunktion genannt) und `sigma` (der Schätzer der Resualvarianz, falls die i-te Beobachtung ausgeschlossen wird).

Insbesondere ist

```
einfl <- influence(erg)
robkov <- var(einfl$coef)*(dim(einfl$coef)[1]-1)
robstdfehler <- sqrt(diag(robkov))
```

ein „robuster“ Schätzer der Standardfehler der Koeffizienten.

Ist `na.action` als `na.exclude` definiert, dann werden auch die Zeilen mit fehlenden Werten wieder in die Ergebnisse aufgenommen und haben den Wert `NA`.

Die Ergenisse können auch einzelnen mit `dfbeta`, `rstandard`, `rstudent` und `hatvalues` berechnet oder aus `influence()` extrahiert werden.