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

Arbeitsblatt 6

1) Die Daten vom letzten Arbeitsblatt sollen wieder eingelesen werden. Als Variablen wurden betrachtet: individuelles (Arbeits-) Einkommen (`I1111099`), Alter (`D1110199`), Geschlecht (`D11102LL`), Schuljahre (`D1110999`) und Arbeitszeit (Stunden je Jahr) (`E1110199`) der 1999 Welle der PSID-Studie. 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))
### fehlende Werte ausschliessen:
X <- na.omit(X)
attach(X)
```

Außerdem soll wieder die lineare Regression von Einkommen auf Alter, Geschlecht, Schuljahre und Arbeitszeit betrachtet werden.

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

2) Matrix-Operationen: Die Lösung eines Kleinst-Quadrate-Problems läßt sich als

$$\hat{\beta} = (X'X)^{-1}X'Y$$

schreiben, wobei $'$ die Transponierte einer Matrix angibt. Diese Formel kann mit den Matrix-Operationen von `R` gelöst werden:

```
x <- cbind(1,D1110199,D11102LL,D1110999,E1110199)
naiv1 <- solve(t(x)%*%x) %*% t(x)%*%I1111099
```

Dabei ist `%*%` die Matrixmultiplikation, `t(x)` die Transponierte von `x` und `solve()` berechnet die inverse Matrix. Die Matrix `x` muss durch `cbind(1,...)` gebildet werden, wenn eine Konstante (`1`) einbezogen werden soll.

Die Rechnung ist in dieser Formulierung allerdings recht ineffizient. Zum einen kann ausgenutzt werden, dass $X'X$ symmetrisch ist, so dass nicht alle Elemente explizit berechnet werden müssen. Das geht mit dem Befehl `crossprod()`, der `t(x)%*%x` und `t(x)%*%I1111099` ersetzt. Zum anderen ist die angegebene Formel äquivalent zu der Gleichung

$$(X'X)\hat{\beta} = X'Y$$

die direkt gelöst werden kann. Das ergibt:

```
naiv2 <- solve(crossprod(x), crossprod(x, I1111099))
cbind(erg,naiv1,naiv2)
```

Der Aufruf der Funktion `lm()` benutzt allerdings die sogenannte QR Zerlegung und liefert auch nur diese als Ergebnis zurück. Man kann also nicht direkt etwa auf $(X'X)^{-1}$ zugreifen. Z.B. ist

$$\mathrm{Var}(\hat{\beta}) = (X'X)^{-1}\sigma^2$$

was duch

```
varb <- solve(crossprod(x))*var(erg$residuals)*
          (dim(x)[1]-1)/(dim(x)[1]-dim(x)[2])
sqrt(diag(varb))
```

geschätzt werden kann.

Die QR-Zerlegung der Matrix `x` ist

$$x = Q\begin{pmatrix} R \\ 0 \end{pmatrix} = Q_f R$$

wobei Q eine orthogonale $n \times n$ Matrix ist, Q_f ihre ersten p Spalten bezeichnet und R eine obere Dreiecksmatrix der Dimension $p \times p$ ist. Dabei heißt Q orthogonal, wenn $Q'Q = QQ' = I$ ist. Die QR-Zerlegung von `erg` ist in `erg$qr`. Die R Matrix der Zerlegung erhält man mit `r <- qr.R(erg$qr)` aus der QR-Zerlegung.

Nun ist

$$(x'x)^{-1} = ((Q\begin{pmatrix} R \\ 0 \end{pmatrix})'(Q\begin{pmatrix} R \\ 0 \end{pmatrix}))^{-1} = (R'R)^{-1} = R^{-1}R'^{-1}$$