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

Arbeitsblatt 3

1) Als Datensatz verwenden wir die sog. 'Cross National Equivalent Files' (CNEF) der 'Panel Study of Income Dynamics' (PSID). Datensatz und Codebücher sind über `http://www.human.cornell.edu/che/PAM/Research/Centers-Programs/German-Panel/Cross-National-Equivalent-File_CNEF.cfm` zugänglich.

Wir benötigen das 'PSID-CNEF codebook', Teil 1 und 2; sowie den Datensatz: `praweqv.exe`. Speichern Sie alle drei Dateien in ein separates Verzeichnis. Der Datensatz muss nun (zweimal) entpackt werden, das Ergebnis sind 20 `.csv`-Dateien.

2) Öffnen Sie eine der `.csv`-Dateien im Editor. Der Datensatz ist wie eine Matrix (vgl. Arbeitsblatt 2) organisiert; die Spalten sind durch Kommata voneinander getrennt.

In R lässt sich der Datensatz folgendermaßen einlesen:

```
X <- read.csv("D:/Rechentech/daten/pequiv99.csv", header=TRUE)
```

Die Variablennamen aus der ersten Zeile der `.csv`-Datei stehen zur Verfügung:

```
names(X)
X$X11101LL
```

Das Objekt `X` ist ein `data.frame`. Diese Datenstruktur ist einer Matrix ähnlich und lässt sich wie eine Liste interpretieren, deren Elemente die nebeneinanderstehenden Spalten der Matrix sind. Zur Illustration:

```
Y <- data.frame(Var1=c("a","b","c"), Var2=c(4,5,6))
Y
Y$Var2
```

3) Deskriptive Statistiken: alle im Arbeitsblatt 2 Punkt 5 aufgeführten Operationen lassen sich natürlich auch hier anwenden.

```
length(X)
length(X$X11101LL)
sum(X$X11101LL)/length(X$X11101LL)
summary(X$D1110799)
```

Häufigkeitstabellen (contingency tables) lassen sich mit `table()` erstellen:

```
table(X$D11102LL)
plot(table(X$D1110199))
```

Relative Häufigkeiten:

```
table(X$D1110799)/length(X$D1110799)
```

Berücksichtigung fehlender Werte bei der Auszählung der Häufigkeiten:

```
table(X$D1110199, exclude=NULL)
```

Vergröberter Merkmalsraum:

```
table(cut(X$D1110199, seq(0,100,by=10)))
```

4) Die Variablennamen in `X` sind nach einem `attach(X)` auch direkt zugreifbar. Diese Bindung lässt sich mit `detach(X)` wieder rückgängig machen.

```
attach(X)
summary(D1110799)
detach(X)
```

Genaugenommen macht sich R, nachdem es auf einen Variablen- oder Funktionsnamen gestoßen ist, nach diesem in den durch `search()` einsehbaren Paketen und Objekten auf die Suche. Mit `attach(X)` wird die Variable `X` (hier: der eingelesene Datensatz) zu dieser Liste hinzugefügt, mit `detach(X)` wieder entfernt.

```
search()
attach(X)
search()
table(D11102LL)
detach(X)
search()
```

Eine Auflistung der selbstdefinierten Objekte und Funktionen erhalten Sie mit `ls()`. Um Objekte/Variablen aus dem Speicher zu entfernen, benutzen Sie `rm()` (gleichbedeutend mit `remove()`):

```
a <- c(1,2,3)
ls()
rm(a)
ls()
```

```
attach(X)
D11102LL <- 2-D11102LL
table(D11102LL)
table(X$D11102LL)
ls()
```

4) Der ungefähre Speicherverbrauch eines R-Objekts kann mit `object.size()` ausgegeben werden, bspw. `object.size(X)`. Der momentane und der bisherige maximale Speicherverbrauch der R-Sitzung wird mit `gc()` angezeigt.

5) Zur einfacheren Handhabung kann es sinnvoll sein, die Variablen umzubenennen.

```
detach(X)
names(X)
names(X)[1]
names(X)[1] <- "PID"
names(X)

names(X)[2] <- "HHID"
names(X)[3] <- "GENDER"
names(X)[31] <- "AGE99"

attach(X)
table(AGE99)
```

Ein derart modifizierter Datensatz kann bspw. mit
`write.table(X, "D:/Rechentech/daten/p99.dat", row.names=FALSE)`

in eine Datei geschrieben und mit
`rm(X)`
`X <- read.table("D:/Rechentech/daten/p99.dat", header=TRUE)`
wieder eingelesen werden.

6) Schlagen Sie im Codebuch die Definition der Variable 'Individual Labour Earnings' nach. Berechnen Sie Durchschnitt und Varianz und lassen Sie einen Kern-Dichte-Schätzer plotten. Fehlende Werte lassen sich in allen drei Fällen durch den zusätzlichen Parameter `na.rm=TRUE` von der Berechnung ausschließen.

Bei der Berechnung des Dichte-Schätzers lässt sich das zu berücksichtigende Intervall einschränken:
`plot(density(I1111099, na.rm=TRUE, to=100000))`

7) Die Berechnungen lassen sich auch auf Fälle beschränken, bei denen der Wert einer anderen und/oder derselben Variablen gewisse Bedingungen erfüllt:

```
mean(I1111099[GENDER==1], na.rm=TRUE)
mean(I1111099[I1111099>0], na.rm=TRUE)
mean(I1111099[I1111099>0 & GENDER==2], na.rm=TRUE)
```

Das ist automatisierbar:

```
tapply(I1111099, GENDER, mean, na.rm=TRUE)
```

führt `mean(I1111099, na.rm=TRUE)` einzeln für die durch die unterschiedlichen Werte in `GENDER` definierten Gruppen aus.

8) Mit

```
Z <- subset(X, I1111099>0, select=c(PID, AGE99, GENDER, I1111099))
```

wird `Z` eine Untermenge der in `X` enthaltenen Daten zugewiesen. In diesem Beispiel besteht die Untermenge aus den Spalten `PID, AGE99, GENDER, I1111099` und denjenigen Zeilen aus `X`, auf die die Bedingung

```
I1111099>0 & !is.na(I1111099)
```

zutrifft (d.h. fehlende Werte bzw. `NA`s werden von `subset()` automatisch ausgeschlossen).