Prof. Dr. J. Althammer
Dr. U. Pötter
Empirische Gesundheitsforschung

Ruhr-Universität Bochum
Sommersemester 2007

Einführung in R: Arbeitsblatt 1

1) Installieren Sie das Programm R. Es gibt Versionen für alle gängigen Betriebssysteme (Linux, Windows, Mac) auf `http://cran.r-project.org/`. R ist ein Interpreter, d.h. es führt Befehle nach Eingabe aus und präsentiert das Ergebnis. Daher kann man es wie einen Taschenrechner benutzen: Tippen Sie

```
> 2+2
```

antwortet R mit

```
[1] 4
```

Hilfe erhält man mit

```
> ?"+"
> ?matrix
> help(log)
```

etc. Die HTML-Version kann mit

```
> options(htmlhelp=TRUE)
> help.start()
```

gestartet werden. Alternativ kann die HTML-Hilfe direkt in einem Browser geöffnet werden. Sie können R mit

```
> q()
```

verlassen. Sie werden dann gefragt, ob Sie den sogenannten "workspace" speichern wollen. I.d.R. sollten Sie mit `n` (no) antworten.

2) Die meisten Statistikprogramme benutzen zur Berechnung numerischer Ausdrücke "double precision reals". Berechnungen in diesem "Zahlensystem" führen immer zu Rundungs- und Abschneidefehlern. Die maximale Genauigkeit dieser Arithmetik entspricht etwa 15 Dezimalstellen. Die Druckdarstellung der Zahlen kann in R durch `options(digits=5)` verändert werden (7 ist die Voreinstellung). Dies verändert nicht die Rechengenauigkeit!

a) Berechnen Sie $2 + 3 * (4 + 5 * (6 + 7 * (8 + 9)))$, 2^{11}, $\log_e(2)$, und $\sqrt{333}$.

3) Variable in R können beliebige Namen haben, die aber mit einem Buchstaben beginnen müssen. Groß- und Kleinschreibung wird unterschieden. Die Zuweisung von Werten zu Variablen erfolgt durch `<-`:

```
a <- 3; a
```

Die wichtigste Datenstruktur in R ist ein "Vektor", eine Liste von Elementen. Vektoren können durch den Befehl `c` erzeugt werden:

```
a <- c(1,3,5)
b <- c(a,1,a)
```

Die Länge eines Vektors (die Anzahl seiner Elemente) wird durch `length(b)` berechnet. Fast alle Befehle in R sind "vektorisiert" und operieren auf allen Elementen eines Vektors.

```
sin(a)+1
```

Wenn Vektoren unterschiedlicher Länge in einem Ausdruck verwandt werden, dann werden in den Berechnungen die kürzeren Vektoren durch Wiederholung ihrer Werte verlängert, bis sie die Länge des längsten Vektors erreicht haben.

```
d <- 2*a + b +1
```

4) Um mit R zu arbeiten, wird ein Editor benötigt, in dem man Programme korrigieren und editieren kann. Wir schlagen vor, mit `TINN-R` zu arbeiten (`http://www.sciviews.org/Tinn-R/`). Installieren Sie `TINN-R`. Benutzen Sie bitte *nicht* die neueste Version, sondern die vorletzte (1.17.2.4). Legen Sie eine Datei mit dem Namen „test.R" in einem Arbeitsverzeichnis an und schreiben Sie die folgenden Kommandos in die Datei:

```
a <- c(1,3,5); b <- c(a,1,a)
d <- 2*a+b+1
sin(d)
```

Speichern Sie die Datei und versuchen Sie, den Inhalt der Datei aus `TINN-R` heraus an R zu senden.

5) Installieren Sie das Zusatzpaket `Rcmdr` (Menue: Pakete -> Installieren..). `Rcmdr` ist ein GUI (graphical User Interface), mit dem sich einfache Analysen durchführen lassen. Es aber aber nur begrenzt als Ersatz für das Arbeiten mit dem Editor und mit Skripten geeignet.

Erstellen Sie einen Datensatz mit zwei Variablen und den Daten (1,2,3) und (2,4,5): Data → New Data. Berechnen Sie einige deskriptive Statistiken: Statistics → Summaries → Active Data Set. Was ist der entsprechende R-Befehl?

6) Eine $n \times m$ Matrix A besteht aus nm rechteckig angeordneten Zahlen mit n Zeilen und m Spalten:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix}$$

Analog kann man Felder beliebiger Dimension definieren.

Matrizen und Felder beliebiger Dimension werden durch

```
A <- matrix(c(1,2,3,4), nrow=2, ncol=2)
```

```
B <- array(c(1,2,3,4,1,2,3,4), dim=c(2,2,2))
```
erzeugt.

Im Gegensatz zur mathematischen Sprechweise haben in `R` Vektoren keine Dimension, wohl aber Matrizen und Felder.
```
dim(a); dim(B)
```
Elementweise Operationen werden auch bei Matrizen und Feldern für alle Elemente gleichzeitig durchgeführt:
```
2*A + 1
C <- matrix(c(3,4,1,2),nrow=2)
A + C
```
Elemente von Vektoren, Matrizen und Feldern können durch die Angabe ihrer Indizes angesprochen werden.
```
a[2]
a[c(1,2)]
A[1,1]
A[,1]
C[C>2]
B[1,2,1]
```
Negative Indizes entfernen die entsprechenden Elemente.
```
a[-2]
```

7) Listen sind geordnete Mengen von verschiedenen Objekten:
```
d <- list(a, c(-1,3,5), "Gesundheit"); d
```
`unlist()` macht aus einer Liste einen Vektor. Elemente einer Liste können durch Angabe ihres Indexes ausgewählt werden.
```
d[1]
d[[1]]
```
Die erste Form ergibt eine Liste mit den ausgewählten Elementen, die zweite Form die ausgewählten Elemente. Da das erste Element von `d` ein Vektor ist, ist
```
d[[1]][2]
```
das zweite Element des Vektors `a`. Elemente einer Liste können Namen haben. Einzelne Elemente einer Liste können auch über ihren Namen ausgewählt werden:
```
d <- list(eins=a, zwei=c(-1,3,5), drei="Gesundheit")
names(d)
names(d) <- c("alpha", "beta", "Wort")
names(d); d$Wort
```

8) Die Funktion `runif()` liefert gleichverteilte Pseudozufallszahlen auf dem 0-1 Intervall. Die Funktion `rnorm()` berechnet normalverteilte Pseudozufallszahlen. Insbesondere liefert
```
X <- runif(100)
```
100 gleichverteilte Pseudozufallszahlen, die der Variablen mit dem Namen `X` zugewiesen werden. Man kann die Startwerte des Zufallszahlengenerators durch `set.seed(n)` festlegen, wobei `n` eine ganze Zahl ist.

a) Erzeugen Sie 100 gleichverteilte Pseudozufallszahlen auf dem Intervall $(-1, 1)$.

b) Erzeugen Sie 100 Pseudozufallszahlen mit Werten in $\{1, 2\}$ und den Wahrscheinlichkeiten $\Pr(\{1\}) = \Pr(\{2\}) = 0.5$. Hinweis: Der Operator `X < Y` erzeugt den Wert `TRUE`, falls `X < Y` ist, `FALSE` sonst. `as.numeric(X)` verwandelt diese logischen Werte in die Zahlen 1 bzw. 0.

9) `sum(), mean(), var(), sd()` berechnen die Summe, den Mittelwert, die Varianz und die Standardabweichung ihrer Argumente. `cov()` und `cor()` berechnen Kovarianzen und Korrelationen. `summary()` berechnet die Quartile, den Mittelwert und die Extrema, wenn das Argument der Funktion ein numerischer Vektor ist. `plot()` interpretiert die Elemente seiner Argumente als Koordinaten und malt sie in einem Graphikfenster. `hist()` malt ein Histogram. `density()` berechnet einen Kern-Dichte-Schätzer der Daten.
```
X <- runif(100)
mean(X)
sum(X)/length(X)
var(X)
Y <- X/2 + runif(100)
cov(X,Y)
plot(X,Y)
summary(X)
hist(X)
plot(density(X))
```

a) Erzeugen Sie zwei Variable wie in den Aufgaben 8 a-b mit je 1000 Beobachtungen und berechnen Sie deskriptive Statistiken.

b) Berechnen Sie einen Kern-Dichte-Schätzer für die Pseudozufallszahlen mit der Verteilung $F(x) = 1 - \exp(-0.5x)$ und zeichnen Sie ihn.

c) Bilden Sie neue Variable durch

```
X <- rnorm(1000)
Y <- rnorm(1000) + X
Z <- rnorm(1000) + X
```

Was ist die Korrelation zwischen `X` und `Y`? Zwischen `Y` und `Z`?