```r
# '1. Vektoren'

# 1.1 'Die c() Funktion'

# Numerisch
vec  = 20
vec

# Schriftzeichen
vec = "ips"

# Numerischer Vektor 'x' mit 6 Elementen
x  = c(3, 4, 6, 89.3, 0, -10)

# Schriftzeichen-Vektor mit 4 Elementen
labs  =  c("E", "A", "E", "i:")

# Elemente 2 bis 4 von y mit 0 überschreiben
y[2:4] = 0

# '1.2 Indizierung'

x[2]               # Element 2 von  x
x[2:4]             # Elemente 2 bis  4
x[c(1,4)]          # Elemente 1 und 4
x[-2]              # Alle Elemente außer Element 2
x[2:4] = 0         # Elemente 2 bis 4 auf  0 setzen

# Alle Elemente von 'x' ohne den 2en und 5en?

# Elemente von 'x' in der anderen Reihenfolge (also von 6 bis 1)?

# '1.3 Arithmetische Manipulationen von Vektoren (*, /, +, -)'

a = c(10, 4, 20)
a * 10

b = c(5, 2, 7)
a + b
# 15  6 27

sum(a)             # Summe aller Elemente
sqrt(a)            # Wurzel pro Element
a^3                # Jedes Element hoch 3
range(a)           # Bereich
max(a)             # Maximum
mean(a)            # Mittelwert

# '1.4 Einige nützliche Funktionen, um Vektoren zu manipulieren'
length()
y = c("i", "i", "a", "a", "E", "E", "E", "E", "U")
length(y)
```

```r
# Wie könnte man das letzte Element von 'y' listen?
# Das vorletzte? Das Vorletzte und Letzte zusammen?

# Intervalle: 'seq()'
seq(10, 20, length=5)   # 5 Intervalle zwischen 10 und 20
seq(10, 20, by=1.5)     # In Intervallen von 1.5

# Wiederholdung: 'rep()'
a = c(10, 4, 20)
rep(a, 4)
rep(a, each=2)

# 'unique()'
y = c("i", "i", "a", "a", "E", "E", "E", "E", "U")
unique(y)
#  "i" "a" "E" "U"

# Tabelle, 'table()'
table(y)
y
# a E i U
# 2 4 2 1

# 'paste()'
paste(y, "V", sep=".")
#  "i.V" "i.V" "a.V" "a.V" "E.V" "E.V" "E.V" "E.V" "U.V"

# String  manipulation:  'nchar()', 'substring()'
cities = c("London", "Paris", "Hamburg", "Verona", "New York")

# Anzahl der Schriftzeichen pro Element
nchar(cities)
 6 5 7 6 8

# Die ersten 3 Schriftzeichen pro Element
substring(cities, 1, 3)

# '2. Matrizen, Data-Frames, Segmentlisten'
# lassen sich in R alle auf eine sehr ähnliche Weise behanden

# '2.1 Matrizen'
a = c(10, 3, 8, 7)
b = c(11, 45, 20, -1)
x = rbind(a, b)         # 2 x 4 Matrix
y = cbind(a, b)         # 4 x 2 Matrix
nrow(x)                 # Reihenanzahl
ncol(x)                 # Spaltenanzahl
dim(x)                  # Dimensionen
rownames(x) = c("S1", "S2")     # Namen der Reihen
colnames(x) = c("A", "B", "C", "D")
class(x)                # Eigenschaften überprüfen

# '2.2 Data-Frame'
```

```
# Ein Data-Frame ist wie eine Matrix; sie kann aber auch
# aus Schriftzeichen-Vektoren bestehen, z.B.
vec = c("A", "B", "C", "D")
d = data.frame(a, b, vec)
class(d)

# Ein bereits vorhandener Data-Frame
ai = read.table(file.path(pfadu, "ai.txt"))
class(ai)               # Eigenschaften

# die ersten paar Reihen (oder Beobachtungen)
head(ai)

# Reihen und Spaltenanzahl
nrow(ai)
ncol(ai)
dim(ai)

# Die Spaltennamen oder Variablen
names(ai)
# das gleiche
colnames(ai)

# Die Namen der Reihen (Beobachtungen)
rownames(ai)

# '2.3 Segmentliste'
# Eine Emu-Segmentliste ist ein Data-Frame und wird immer mit
# mit emu.query() aus einer vorhandenen Sprachdatenbank erstellt.
# Eine Segmentliste lässt sich auf fast die selbe Weise wie ein
# Data-Frame oder Matrix behandeln:
library(emu)
# Alle H-Segmente in der Sprachdatenbanke
d.s = emu.query("xyz", "*", "phonetic=H")
class(d.s)
nrow(d.s)
ncol(d.s)
dim(d.s)

# Die Funktionen label(), start(), end(), utt(), emu.requery()
# können auf eine Segmentliste angewandt werden

v.s = emu.query("xyz", "*", "phonetic = a|U")
# Die Etikettierung (Schriftzeichen-Vektor)
v.l = label(v.s)
# Die Start- und Endzeiten (Numerische Vektoren)
start(v.s)
end(v.s)
# Die Äußerungen, aus denen die Segmente stammen (Schriftzeichen-Vektor)
u = utt(v.s)
u[1:4]
# Die davorkommenden Segmente (Segmentliste)
davor = emu.requery(v.s, "phonetic", "phonetic", seq=-1)
```

```
# Die Segmente danach (Segmentliste)
danach = emu.requery(v.s, "phonetic", "phonetic", seq=1)
# Die damit verbundenen Wörter
wort = emu.requery(v.s, "phonetic", "word")

# '3. Indizierung'
# 'x[m,]' = Reihe m     # x ist eine Matrix, Data-Frame, oder Segmentliste
# 'x[,m]' = Spalte m

# Reihe (Segment) 2
d.s[2,]

# Segmente 2 bis 10
d.s[2:10,]

# Segmente 2, 3, und 8
ai[c(2, 3, 8),]

# oder
vec = c(2, 3, 8)
ai[vec,]

# Spalte 2 (Variable 2)
ai[,2]
d.s[,2]

# Reihen 2 bis 10 von Spalte 2
ai[2:10,2]

# Reihen 2, 5, 8 von Spalten 2 und 3?

# Reihen 12 bis 18 von Spalten 1 und 3?

# Ein Minuszeichen in '[-n, ]' oder '[,-n]': alle außer n
# Alle Spalten außer Spalte 1
ai[,2:3]

# oder
ai[,-1]

# Anzahl der Beobachtungen
n = nrow(d.s)

# Letzte Beobachtung (indem 'n' verwendet wird)?

# Vorletzte Beobachtung (indem 'n' verwendet wird)?

# Letzte 3 Beobachtungen (indem 'n' verwendet wird)?
```