Datentypen

Jan-Philipp Kolb

05 Mai, 2019

Verschiedene Datentypen

Datentyp	Beschreibung	Beispiel
numeric	ganze und reele Zahlen	`5, 3.462`
logical	logische Werte	`FALSE, TRUE`
character	Buchstaben und Zeichenfolgen	`"Hallo"`

Quelle: R. Münnich und M. Knobelspieß (2007): Einführung in das statistische Programmpaket R

Verschiedene Datentypen

```
b <- c(1,2) # numeric
log <- c(T,F) # logical
char <-c("A","b") # character
fac <- as.factor(c(1,2)) # factor
```

Mit `str()` bekommt man den Objekttyp.

```
str(fac)
```

```
##  Factor w/ 2 levels "1","2": 1 2
```

Indizieren eines Vektors:

```
A1 <- c(1,2,3,4)
A1
```

```
## [1] 1 2 3 4
```

```
A1[1]
```

```
## [1] 1
```

```
A1[4]
```

```
## [1] 4
```

```
A1[1:3]
```

```
## [1] 1 2 3
```

```
A1[-4]
```

```
## [1] 1 2 3
```

data.frames

Beispieldaten generieren:

```
AGE <- c(20,35,48,12)
SEX <- c("m","w","w","m")
```

Diese beiden Vektoren zu einem data.frame verbinden:

```
Daten <- data.frame(Alter=AGE,Geschlecht=SEX)
```

Anzahl der Zeilen/Spalten herausfinden

```
nrow(Daten) # Zeilen
```

```
## [1] 4
```

```
ncol(Daten) # Spalten
```

```
## [1] 2
```

Indizieren

Indizieren eines dataframe:

```
AA <- 4:1
A2 <- cbind(A1,AA)
A2[1,1]
```

```
## A1
##  1
```

```
A2[2,]
```

```
## A1 AA
##  2  3
```

```
A2[,1]
```

```
## [1] 1 2 3 4
```

```
A2[,1:2]
```

Matrizen und Arrays

- In Matrizen und Arrays stehen meist nur numerische Werte.
- Dadurch wird beispielsweise Matrix Multiplikation möglich.
- Anders als beim data.frame sind mehr als zwei Dimensionen möglich.

```
A <- matrix(seq(1,100), nrow = 4)
dim(A)
```

```
## [1]  4 25
```

Ein Array erzeugen

```
A3 <- array(1:8,c(2,2,2))
A3
```

```
## , , 1
##
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
##
## , , 2
##
##      [,1] [,2]
## [1,]    5    7
## [2,]    6    8
```

Indizieren eines Array

```
A3[,,2]
```

```
##      [,1] [,2]
## [1,]    5    7
## [2,]    6    8
```

Listen

- Eine Liste in R entspricht einem geschachtelten Array in anderen Programmiersprachen
- Listen können alles enthalten
- Listen können geschachtelt sein
- Listen sollte man sehr bedacht verwenden

Indizieren einer Liste

```
A4 <- list(A1,1)
A4
```

```
## [[1]]
## [1] 1 2 3 4
##
## [[2]]
## [1] 1
```

```
A4[[2]]
```

```
## [1] 1
```

Logische Operatoren

```
# Ist 1 größer als 2?
1>2
```

```
## [1] FALSE
```

```
1<2
```

```
## [1] TRUE
```

```
1==2
```

```
## [1] FALSE
```

Operatoren um Subset für Datensatz zu bekommen

Diese Operatoren eignen sich gut um Datensätze einzuschränken

```
Daten
```

```
##   Alter Geschlecht
## 1    20           m
## 2    35           w
## 3    48           w
## 4    12           m
```

```
Daten[AGE>20,]
```

```
##   Alter Geschlecht
## 2    35           w
## 3    48           w
```

Datensätze einschränken

```
Daten[SEX=="w",]
```

```
##   Alter Geschlecht
## 2    35          w
## 3    48          w
```

```
# gleiches Ergebnis:
Daten[SEX!="m",]
```

```
##   Alter Geschlecht
## 2    35          w
## 3    48          w
```

Weitere wichtige Optionen

```
# Ergebnis in ein Objekt speichern
subDat <- Daten[AGE>20,]

# mehrere Bedingeungen können mit
# & verknüpft werden:
Daten[AGE>18 & SEX=="w",]

##   Alter Geschlecht
## 2    35          w
## 3    48          w
```

Sequenzen

```
# Sequenz von 1 bis 10
1:10
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
Daten[1:3,]
```

```
##   Alter Geschlecht
## 1    20          m
## 2    35          w
## 3    48          w
```

Weitere Sequenzen

```
seq(-2,8,by=1.5)
```

```
## [1] -2.0 -0.5  1.0  2.5  4.0  5.5  7.0
```

```
a <-seq(3,12,length=12)
```

```
b <- seq(to=5,length=12,by=0.2)
```

```
d <- 1:10
d <- seq(1,10,1)
d <- seq(length=10,from=1,by=1)
```

Wiederholungen

```
# wiederhole 1 10 mal
rep(1,10)
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1
```

```
rep("A",10)
```

```
##  [1] "A" "A" "A" "A" "A" "A" "A" "A" "A" "A"
```

Die Funktion paste

```
?paste
```

```
paste(1:4)
```

```
## [1] "1" "2" "3" "4"
```

```
paste("A", 1:6, sep = "")
```

```
## [1] "A1" "A2" "A3" "A4" "A5" "A6"
```

- Ein weiteres Beispiel:

```
paste0("A", 1:6)
```

```
## [1] "A1" "A2" "A3" "A4" "A5" "A6"
```