

Statistische Software (R)

Paul Fink, M.Sc.

Institut für Statistik
Ludwig-Maximilians-Universität München

Listen und Data Frames



Vektor vom Typ **list** kann auch komplexere Objekte unterschiedlicher Klassen als Elemente enthalten.

```
> mat <- matrix(seq(from = 2, by = 3, length = 12), nrow = 3)
> list1 <- list(numeric = numvec, character = charvec,
+             matrix = mat)
> list1
$numeric
[1] 2.54 4.22 2.99 3.14 3.44

$character
[1] "Statistische" "Software"

$matrix
  [,1] [,2] [,3] [,4]
[1,]  2  11  20  29
[2,]  5  14  23  32
[3,]  8  17  26  35
```

Paul Fink: Statistische Software (R) SoSe 2015

2

Listen

Rekursive Datenstruktur: Liste kann auch Listen enthalten

```
> list2 <- list(logicvec, list = list1)
> list2
[[1]]
[1] TRUE FALSE FALSE TRUE

$list
$list$numeric
[1] 2.54 4.22 2.99 3.14 3.44

$list$character
[1] "Statistische" "Software"

$list$matrix
  [,1] [,2] [,3] [,4]
[1,]  2  11  20  29
[2,]  5  14  23  32
[3,]  8  17  26  35
```

Paul Fink: Statistische Software (R) SoSe 2015

3

Zugriff auf Listenelemente

- Der Zugriff auf die Elemente einer Liste sollte über den `[[]]` Operator erfolgen.

```
> list1[[1]][2]
[1] 4.22
```

- Auf Element mit Namen kann man auch über `$` zugreifen.

```
> list2$list$matrix
  [,1] [,2] [,3] [,4]
[1,]  2  11  20  29
[2,]  5  14  23  32
[3,]  8  17  26  35
```

Paul Fink: Statistische Software (R) SoSe 2015

4

- `x[1]` liefert das Objekt an der ersten Stelle vom selben Datentyp wie `x` zurück
- `x[[1]]` liefert das Objekt an der ersten Stelle mit dessen Datentyp zurück
- `x[z]` liefert `NA` zurück, wenn `z > length(x)`
- `x[[z]]` gibt Fehler aus, wenn `z > length(x)`

Datenmatrix

Die wohl wichtigste Struktur zur Haltung von Daten im üblichen Rechteckschema, wo die Beobachtungen in den Zeilen und die Variablen in den Spalten dargestellt werden, ist die Datenmatrix. In R wird diese `data.frame` genannt.

Data.frames sind spezielle Listen, deren Elemente wiederum Vektoren gleicher Länge sind. Data.frames sind **DIE** typische Datenstruktur in R. Data.frames können komplette Datensätze aufnehmen, die (meist) mit anderen Programmen erstellt wurden (Spreadsheet- Dateien, SPSS-Dateien, tab-delimited ASCII Dateien, etc.).

Jedes Vektoren/Listen-Element *kann* einen Namen haben

Abfragen über Funktion `names()`

```
> names(list1)
[1] "numeric" "character" "matrix"
> names(list2)
[1] "" "list"
> names(1:8)
NULL
```

Datenmatrix - Beispiel

Beispiel `mtcars` (10 Kennzahlen zu 32 Autos im Jahr 1974)

```
> mtcars
      mpg  cyl  disp  hp drat   wt  qsec vs  am  gear  carb
Mazda RX4    21.0   6  160.0  110 3.90  2.620 16.46  0   1    4    4
Mazda RX4 Wag 21.0   6  160.0  110 3.90  2.875 17.02  0   1    4    4
Datsun 710    22.8   4  108.0   93 3.85  2.320 18.61  1   1    4    1
Hornet 4 Drive 21.4   6  258.0  110 3.08  3.215 19.44  1   0    3    1
Hornet Sportabout 18.7   8  360.0  175 3.15  3.440 17.02  0   0    3    2
.....
```

Überblick verschaffen mit Funktion `head()`:

```
> head(mtcars)
      mpg cyl disp  hp drat   wt  qsec vs am gear carb
Mazda RX4           21.0   6  160 110 3.90 2.620 16.46 0  1   4   4
Mazda RX4 Wag       21.0   6  160 110 3.90 2.875 17.02 0  1   4   4
Datsun 710          22.8   4  108  93 3.85 2.320 18.61 1  1   4   1
Hornet 4 Drive      21.4   6  258 110 3.08 3.215 19.44 1  0   3   1
Hornet Sportabout  18.7   8  360 175 3.15 3.440 17.02 0  0   3   2
Valiant            18.1   6  225 105 2.76 3.460 20.22 1  0   3   1
```

Man kann die Zeilennamen und Variablenamen bekommen mit

```
> rownames(mtcars)
 [1] "Mazda RX4"           "Mazda RX4 Wag"       "Datsun 710"
 [4] "Hornet 4 Drive"      "Hornet Sportabout"   "Valiant"
 [7] "Duster 360"         "Merc 240D"           "Merc 230"
[10] "Merc 280"           "Merc 280C"           "Merc 450SE"
[13] "Merc 450SL"         "Merc 450SLC"         "Cadillac Fleetwood"
[16] "Lincoln Continental" "Chrysler Imperial"   "Fiat 128"
[19] "Honda Civic"         "Toyota Corolla"      "Toyota Corona"
[22] "Dodge Challenger"   "AMC Javelin"         "Camaro Z28"
[25] "Pontiac Firebird"    "Fiat X1-9"           "Porsche 914-2"
[28] "Lotus Europa"        "Ford Pantera L"      "Ferrari Dino"
[31] "Maserati Bora"       "Volvo 142E"

> colnames(mtcars) # oder 'names(mtcars)
 [1] "mpg" "cyl" "disp" "hp" "drat" "wt" "qsec" "vs" "am" "gear"
[11] "carb"
```

Achtung, die Zeilennamen sind keine eigene Variable!

Zugriff auf Elemente in Data.frame

- Man kann es wie eine Matrix behandeln, also über `[]`

```
> mtcars[1:4,]
      mpg cyl disp  hp drat   wt  qsec vs am gear carb
Mazda RX4           21.0   6  160 110 3.90 2.620 16.46 0  1   4   4
Mazda RX4 Wag       21.0   6  160 110 3.90 2.875 17.02 0  1   4   4
Datsun 710          22.8   4  108  93 3.85 2.320 18.61 1  1   4   1
Hornet 4 Drive      21.4   6  258 110 3.08 3.215 19.44 1  0   3   1
```

- Einzelne Variablen auch über `$` wie bei Listen

```
> mtcars$cyl
 [1] 6 6 4 6 8 6 8 4 4 6 6 8 8 8 8 8 8 4 4 4 4 8 8 8 8 4 4 4 8 6 8 4
```