Programmierung und Deskriptive Statistik

BSc Psychologie WiSe 2024/25

Belinda Fleischmann

Datum	Einheit	Thema	Form
15.10.24	R Grundlagen	(1) Einführung	Seminar
22.10.24	R Grundlagen	(2) R und Visual Studio Code	Seminar
29.10.24	R Grundlagen	(2) R und Visual Studio Code	Übung
05.11.24	**R Grundlagen**	**(3) Vektoren**, **(4) Matrizen**	**Seminar**
12.11.24	R Grundlagen	(5) Listen und Dataframes	Seminar
	Leistungsnachweis 1		
19.11.24	R Grundlagen	(6) Datenmanagement	Seminar
26.11.24	R Grundlagen	(2)-(6) R Grundlagen	Übung
03.12.24	Deskriptive Statistik	(7) Häufigkeitsverteilungen	Seminar
10.12.24	Deskriptive Statistik	(8) Verteilungsfunktionen und Quantile	Seminar
	Leistungsnachweis 2		
17.12.24	Deskriptive Statistik	(9) Maße der zentralen Tendenz und Datenvariabilität	Seminar
	Weihnachtspause		
07.01.25	R Grundlagen	(10) Strukturiertes Programmieren: Kontrollfluss, Debugging	Seminar
14.01.25	Deskriptive Statistik	(11) Anwendungsbeispiel	Übung
	Leistungsnachweis 3		
21.01.25	Deskriptive Statistik	(11) Anwendungsbeispiel	Seminar
28.01.25	Deskriptive Statistik	(11) Anwendungsbeispiel, Q&A	Seminar

(4) Matrizen

Erzeugung

Charakterisierung

Indizierung

Arithmetik

Attribute

Programmierübungen und Selbstkontrollfragen

Übersicht und Erzeugung

Charakterisierung

Indizierung

Arithmetik

Attribute

Programmierübungen und Selbstkontrollfragen

Übersicht

Matrizen sind zweidimensionale, rechteckige Datenstrukturen der Form

$$M = \begin{pmatrix} m_{11} & m_{12} & \cdots & m_{1n_c} \\ m_{21} & m_{22} & \cdots & m_{2n_c} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n_r1} & m_{n_r2} & \cdots & m_{n_rn_c} \end{pmatrix} \tag{1}$$

- Die Elemente $m_{ij}, i = 1, ..., n_r, j = 1, ..., n_c$ sind vom gleichen Typ.
- n_r ist die Anzahl der Zeilen (rows), n_c ist die Anzahl der Spalten (columns).
- Jedes Element einer Matrix hat einen Zeilenindex i und einen Spaltenindex j.
- Intuitiv sind Matrizen numerisch indizierte Tabellen.
- Formal sind Matrizen in R zweidimensional interpretierte atomare Vektoren.
- Matrizen in R sind nicht identisch mit dem mathematischen Matrixbegriff.
- Matrizen in R können allerdings für Lineare Algebra verwendet werden.
- Lineare Algebra ist die Sprache (linearer) statistischer Modelle.

Erzeugung mit `matrix()`

Die `matrix()` Funktion befüllt Matrizen mit Vektorelementen

`matrix(data, nrow, ncol, byrow)`

```r
matrix(c(1:12), nrow = 3)               # 3 x 4 Matrix der Zahlen 1,...,12, byrow = FALSE
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
```

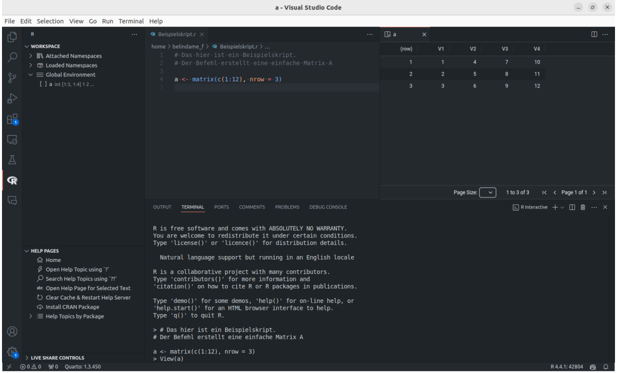
```r
matrix(c(1:12), ncol = 4)               # 3 x 4 Matrix der Zahlen 1,...,12, byrow = FALSE
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
```

```r
matrix(c(1:12), nrow = 3, byrow = TRUE) # 3 x 4 Matrix der Zahlen 1,...,12, byrow = TRUE
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
```

Table Viewer



Mit dem Befehl `View()` oder im R $\boxed{\text{WORKSPACE}}$ $\rightarrow$ $\boxed{\text{Global Environment}}$ über das View Symbol  neben entsprechendem Objekt öffnen.

VS Code Wiki - Interactive viewers

Erzeugung mit cbind()

Die Funktion cbind() konkateniert passende Matrizen spaltenweise (*column-bind*)

```
A <- matrix(c(1:4) , nrow = 2)          # 2 x 2 Matrix der Zahlen 1,...,4
print(A)
```

```
     [,1] [,2]
[1,]    1    3
[2,]    2    4
```

```
B <- matrix(c(5:10), nrow = 2)          # 2 x 3 Matrix der Zahlen 5,...,10
print(B)
```

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

```
C <- cbind(A, B)                        # Spaltenweise Konkatenierung von A und B
print(C)
```

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

Erzeugung mit rbind()

Die Funktion `rbind()` konkateniert passende Matrizen reihenweise (*row-bind*)

```
A <- matrix(c(1:6) , nrow = 2, byrow = T)  # 2 x 3 Matrix der Zahlen 1,...,6
print(A)
```

```
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
```

```
B <- matrix(c(7:9), nrow = 1)              # 1 x 3 Matrix der Zahlen 5,...,10
print(B)
```

```
     [,1] [,2] [,3]
[1,]    7    8    9
```

```
C <- rbind(A, B)                           # reihenweise Konkatenierung von A und B
print(C)
```

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

Übersicht und Erzeugung

Charakterisierung

Indizierung

Arithmetik

Attribute

Programmierübungen und Selbstkontrollfragen

Charakterisierung

`typeof()`gibt den elementaren Datentyp einer Matrix aus

```
A <- matrix(c(T, T, F, F), nrow = 2)        # 2 x 2 Matrix von Elementen vom Typ logical
typeof(A)
```

```
[1] "logical"
```

```
B <- matrix(c("a", "b", "c"), nrow = 1)     # 1 x 3 Matrix von Elementen vom Typ character
typeof(B)
```

```
[1] "character"
```

`nrow()` und `ncol()` geben die Zeilen- bzw. Spaltenanzahl aus

```
C <- matrix(1:12, nrow = 3)                 # 3 x 4 Matrix
nrow(C)                                     # Anzahl Zeilen
```

```
[1] 3
```

```
ncol(C)                                     # Anzahl Spalten
```

```
[1] 4
```

Indizierung

Generell gilt

- Matrixelemente werden mit einen Zeilenindex und einem Spaltenindex indiziert.
- Die Indexreihenfolge ist immer 1. Zeile, 2. Spalte.
- Die Prinzipien der Indizierung entsprechen der Vektorindizierung.
- Indizes verschiedener Dimensionen können unterschiedlich indiziert werden.
- Eindimensionale Resultate liegen als Vektor, nicht als Matrix vor.

Beispiele

```r
A <- matrix(c(2:7)^2, nrow = 2)      # 2 x 3 Matrix der Zahlen 2^2,...,7^2
print(A)
```

```
     [,1] [,2] [,3]
[1,]    4   16   36
[2,]    9   25   49
```

```r
a_13 <- A[1, 3]                      # Element in 1. Zeile, 3. Spalte von A [36]
a_22 <- A[2, 2]                      # Element in 2. Zeile, 2. Spalte von A [35]
a_2. <- A[2,]                        # Alle Elemente der 2. Zeile [9,25,49]
a_.3 <- A[,3]                        # Alle Elemente der 3. Spalte [36,49]
A_12 <- A[1:2, 1:2]                  # Submatrix der ersten zwei Zeilen und Spalten
A10  <- A[A>10]                      # Elemente von A groesser 10 [16,25,36,49]
A_13 <- A[1, c(F, F, T)]             # Element in 1. Zeile, 3. Spalte von A [36]
```

Übersicht und Erzeugung

Charakterisierung

Indizierung

Arithmetik

Attribute

Programmierübungen und Selbstkontrollfragen

Unitäre arithmetische Operationen

Unitäre arithmetische Operatoren und Funktionen werden elementweise ausgewertet.

```
A <- matrix(c(1:4), nrow = 2)  # 2 x 2 Matrix der Zahlen 1,2,3,4
print(A)
```

```
     [,1] [,2]
[1,]    1    3
[2,]    2    4
```

```
B <- A^2                       # B[i,j]  = A[i,j]^2,  1 <= i,j <= 2
print(B)
```

```
     [,1] [,2]
[1,]    1    9
[2,]    4   16
```

```
C <- sqrt(B)                   # C[i,j]  = sqrt(A[i,j]^2),  1 <= i,j <= 2
print(C)
```

```
     [,1] [,2]
[1,]    1    3
[2,]    2    4
```

```
D <- exp(A)                    # D[i,j] = exp(A[i,j]),  1 <= i,j <= 2
print(D)
```

```
         [,1]     [,2]
[1,] 2.718282 20.08554
[2,] 7.389056 54.59815
```

Binäre arithmetische Funktionen

Matrizen **passender Größen** können mit binären arithmetischen Operatoren verknüpft werden.

Binäre arithmetische Operatoren +,-,*,\werden bei gleicher Größe elementweise ausgewertet.

```
A <- matrix(c(1:4), nrow = 2)      # 2 x 2 Matrix der Zahlen 1,2,3,4
print(A)
```

```
     [,1] [,2]
[1,]    1    3
[2,]    2    4
```

```
B <- matrix(c(5:8), nrow = 2)      # 2 x 2 Matrix der Zahlen 5,6,7,8
print(B)
```

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

```
print(A + B)                       # C[i,j] = A[i,j] + B[i,j], 1 <= i,j <= 2
```

```
     [,1] [,2]
[1,]    6   10
[2,]    8   12
```

```
print(A * B)                       # C[i,j] = A[i,j] * B[i,j], 1 <= i,j <= 2
```

```
     [,1] [,2]
[1,]    5   21
[2,]   12   32
```

Lineare Algebra mit R Matrizen

- Addition, Subtraktion, Hadamardprodukt elementweise definiert wie oben
- Matrixmultiplikation, Transposition, Inversion, Determinante

```
C <- A  %*% B           # 2 x 2 Matrixprodukt
print(C)
```

```
     [,1] [,2]
[1,]   23   31
[2,]   34   46
```

```
A_T <- t(A)             # Transposition von A
print(A_T)
```

```
     [,1] [,2]
[1,]    1    2
[2,]    3    4
```

```
A_inv <- solve(A)       # Inverse von A
print(A_inv)
```

```
     [,1] [,2]
[1,]   -2  1.5
[2,]    1 -0.5
```

```
A_det <- det(A)         # Determinante von A
print(A_det)
```

```
[1] -2
```

Attribute

Formal sind Matrizen atomare Vektoren mit einem Attribut namens "`dim`".

```
A <- matrix(1:12, nrow = 4 )              #  4 x 3 Matrix
attributes(A)                             # Aufrufen der Attribute von A
```

```
$dim
[1] 4 3
```

`rownames()` und `colnames()` spezifizieren das Attribut "`dimnames`".

```
rownames(A) <- c("P1", "P2", "P3", "P4") # Benennung der Zeilen von A
colnames(A) <- c("Age", "Hgt", "Wgt")    # Benennung der Spalten von A
print(A)                                  # A mit Attribut dimnames
```

```
   Age Hgt Wgt
P1   1   5   9
P2   2   6  10
P3   3   7  11
P4   4   8  12
```

```
attr(A, "dimnames")                       # Aufrufen des Attributs dimnames
```

```
[[1]]
[1] "P1" "P2" "P3" "P4"

[[2]]
[1] "Age" "Hgt" "Wgt"
```

Anmerkung: Bei Matrizen ist die Benennung von Zeilen und Spalten eher ungewöhnlich.

Programmierübungen

1. Dokumentiere alle in dieser Einheit eingeführten Befehle in einem R Skript.

2. Erzeuge in R die Matrizen

$$A = \begin{pmatrix} 4 & 3 & 2 & 1 \\ 3 & 2 & 1 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix} \text{ und } B = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

3. Kopiere die zweite Zeile von A in einen Vektor.

4. Kopiere die erste und dritte Spalte von B in eine 3×2 Matrix.

5. Setze alle Nullen in B auf -1.

6. Setze die zweite Zeile von A auf $(1\,2\,3\,4)$.

7. Addiere die Matrizen A und B.

8. Multipliziere Matrix A mit 3.

9. Konkateniere die Matrizen A und B zeilenweise.

10. Konkatenieren die Matrizen A und B spaltenweise.

11. Öffne beide Matrizen im VSCode table Viewer.

Selbstkontrollfragen

1. Nenne drei Operationen der linearen Algebra, die für Matrizen elementweise durchgeführt werden mit entsprechenden R Befehlen.
2. Nenne vier Operationen der linearen Algebra, die für Matrizen *nicht* elementweise durchgeführt werden mit entsprechenden R Befehlen.
3. Was ist der Modus von Matrizen in R?