Programmierung und Deskriptive Statistik

BSc Psychologie WiSe 2022/23

Belinda Fleischmann

(4) Matrizen

Matrizen

Übungen und Selbstkontrollfragen

Matrizen

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.

Matrizen

Erzeugung

Die `matrix()` Funktion befüllt Matrizen mit Vektorelementen

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

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

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

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

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

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

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

Matrizen

Erzeugung Die Funktion `cbind()` konkateniert passende Matrizen spaltenweise

```
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 Die Funktion `rbind()` konkateniert passende Matrizen reihenweise

```
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
```

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.

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

Matrizen

Arithmetik

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

B = A^2                          # B[i,j]  = A[i,j]^2,  1 <= i,j <= 2
     [,1] [,2]
[1,]    1    9                   # 1^2, 3^2
[2,]    4   16                   # 2^2, 4^2

C = sqrt(B)                      # C[i,j]  = sqrt(A[i,j]^2),  1 <= i,j <= 2
     [,1] [,2]
[1,]    1    3                   # sqrt(1^2), sqrt(3^2)
[2,]    2    4                   # sqrt(2^2), sqrt(4^2)

D = exp(A)                       # D[i,j] = exp(A[i,j]),  1 <= i,j <= 2
     [,1] [,2]
[1,]  2.7 20.0                   # exp(1), exp(3)
[2,]  7.4 54.6                   # exp(2), exp(4)
```

Matrizen

Arithmetik

Matrizen passender Größe 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
     [,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
     [,1] [,2]
[1,]    5    7
[2,]    6    8

C = A + B                        # C[i,j] = A[i,j] + B[i,j], 1 <= i,j <= 2
     [,1] [,2]
[1,]    6   10                   # 1 + 5, 3 + 7
[2,]    8   12                   # 2 + 6, 4 + 8

D = A * B
     [,1] [,2]
[1,]    5   21                   # 1 * 5, 3 * 7
[2,]   12   32                   # 2 * 6, 4 * 8
```

Matrizen

Arithmetik

Mit R Matrizen kann Lineare Algebra betrieben werden

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

```
C = A  % * % B                    # 2 x 2 Matrixprodukt
      [,1] [,2]
[1,]    23   31                   # 1*5 + 3*6, 1*7+3*8
[2,]    34   46                   # 2*5 + 4*6, 2*7+4*8

A_T = t(A)                        # Transposition von A
      [,1] [,2]
[1,]    1    2                    # A[1,1], A[2,1]
[2,]    3    4                    # A[1,2], A[2,2]

A_inv = solve(A)                  # Inverse von A
     [,1] [,2]
[1,]   -2  1.5
[2,]    1 -0.5

A_det = det(A)                    # Determinante von A
[1] -2                            # 1*4 - 2*3
```

Matrizen

Attribute

Formal sind Matrizen atomare Vektoren mit einem `dim` Attribut

```
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
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"
```

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

Matrizen

Übungen und Selbstkontrollfragen

Übungen und Selbstkontrollfragen

1. Dokumentieren Sie alle in dieser Einheit eingeführten Befehle in einem R Skript.
2. Erzeugen Sie 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. Kopieren Sie die zweite Zeile von A in einen Vektor.
4. Kopieren Sie die erste und dritte Spalte von B in eine 3 $\times$ 2 Matrix
5. Setzen Sie alle Nullen in B auf -1.
6. Setzen Sie die zweite Zeile von A auf $(1\,2\,3\,4)$.
7. Addieren Sie die Matrizen A und B.
8. Multiplizieren Matrix A mit 3.
9. Konkatenieren Sie die Matrizen A und B zeilenweise.
10. Konkatenieren Sie die Matrizen A und B spaltenweise.