(2) Matrizen

Ziel dieser Sitzung ist es, die Matrizenrechnung in **R** anhand der in der Vorlesung diskutierten Beispiele nachzuvollziehen und optional weitere Übungsaufgaben mithilfe von **R** zu lösen.

Grundlegende Matrixoperationen

Wir betrachten zunächst die spaltenweise und zeilenweise Definition von Matrizen in **R** anhand zweier in der Vorlesung gegebener Beispiele zur Matrixaddition und Matrixsubtraktion. Man beachte, dass die zeilenweise Definition einer Matrix, die mithilfe des Arguments `byrow = TRUE` (non-default) erreicht wird, eine höhere Korrespondenz zwischen **R**-Codebild und **R** Repräsentation ermöglicht.

```r
# spaltenweise Definition von A (R-Default)
A = matrix(c( 2, 1,-3, 6, 0, 5), nrow = 2)
print(A)
```

```
     [,1] [,2] [,3]
[1,]    2   -3    0
[2,]    1    6    5
```

```r
# zeilenweise Definition von B
B = matrix(c( 4, 1, 0,
             -4, 2, 0),
           nrow  = 2,
           byrow = TRUE)
print(B)
```

```
     [,1] [,2] [,3]
[1,]    4    1    0
[2,]   -4    2    0
```

Die Addition und Subtraktion von Matrizen werden in **R** dann mit den Operatoren `+` und `-` implementiert. Entsprechend der in der Vorlesung eingeführten Rechenregeln ergeben sich

```r
# Addition
C = A + B
print(C)
```

```
     [,1] [,2] [,3]
[1,]    6   -2    0
[2,]   -3    8    5
```

und

```
# Subtraktion
D = A - B
print(D)
```

```
     [,1] [,2] [,3]
[1,]   -2   -4    0
[2,]    5    4    5
```

Das in der Vorlesung betrachtete Beispiel zur Skalarmultiplikation ergibt sich mit dem Skalarmultiplikationsoperator `*` wie folgt.

```
# Definitionen
A = matrix(c(3,1,1,
             5,2,5,
             2,7,1,
             3,4,2),
           nrow = 4,
           byrow = TRUE)
c = -3

# Skalarmultiplikation
B = c*A
print(B)
```

```
     [,1] [,2] [,3]
[1,]   -9   -3   -3
[2,]  -15   -6  -15
[3,]   -6  -21   -3
[4,]   -9  -12   -6
```

Das in der Vorlesung betrachtete Beispiel zur Matrixtransposition schließlich ergibt sich mit dem Transpositionsoperator `t()` wie folgt.

```
# Definition
A = matrix(c(2,3,0,
             1,6,5),
           nrow = 2,
           byrow = TRUE)

# Transposition
AT = t(A)
print(A)
```

```
     [,1] [,2] [,3]
[1,]    2    3    0
[2,]    1    6    5
```

```r
print(AT)
```

```
     [,1] [,2]
[1,]    2    1
[2,]    3    6
[3,]    0    5
```

Multiplikation von Matrizen

Das erste in der Vorlesung betrachtete Beispiel zur Matrixmultiplikation implementiert man in **R** mithilfe des Matrixmultiplikationsoperators `%*%` wie folgt.

```r
# Definitionen
A = matrix(c( 2,-3, 0,
              1, 6, 5),
           nrow  = 2,
           byrow = TRUE)
B = matrix(c( 4, 2,
             -1, 0,
              1, 3),
           nrow  = 3,
           byrow = TRUE)

# Matrixmultiplikation
C = A %*% B
print(C)
```

```
     [,1] [,2]
[1,]   11    4
[2,]    3   17
```

Das zweite Beispiel zur Matrixmultiplikation ergibt demzufolge so:

```
# Definitionen
A = matrix(c( 2,-3, 0,
              1, 6, 5),
           nrow  = 2,
           byrow = TRUE)
B = matrix(c( 4, 2,
             -1, 0,
              1, 3),
           nrow  = 3,
           byrow = TRUE)

# Matrixmultiplikation
D = B %*% A
print(D)
```

```
     [,1] [,2] [,3]
[1,]   10    0   10
[2,]   -2    3    0
[3,]    5   15   15
```

Ist die Multiplikation zweier in **R** definierter Matrizen nicht definiert, d.h. stimmt die Anzahl der Spalten der ersten Matrix nicht mit der Anzahl der Zeilen der zweiten Matrix überein, so ergibt sich ein Fehler.

```
# Beispiel für eine undefinierte Matrixmultipliation
E = t(A) %*% B      # (3 x 2)(3 x 2)
```

```
Error in t(A) %*% B: non-conformable arguments
```

Inverse Matrizen

Inverse Matrizen berechnet man in **R** für gewöhnlich mit dem Befehl `solve()`. Für das in der Vorlesung betrachtete Beispiel einer invertierbaren 2×2 Matrix ergibt sich entsprechend folgender **R**-Code. (Man beachte, dass sich bei der rechtsseitigen Multiplikation von A mit ihrer inversen minimale Rundungsfehler ergeben.)

```
# Definition
A = matrix(c(2,1,
             3,4),
           nrow  = 2,
           byrow = TRUE)

# Berechnen von A^{-1}
print(solve(A))
```

```
     [,1] [,2]
[1,]  0.8 -0.2
[2,] -0.6  0.4
```

```
# Probe: Multiplikation von A^{-1} mit A
print(solve(A) %*% A)
```

```
     [,1] [,2]
[1,]    1    0
[2,]    0    1
```

```
print(A %*% solve(A))
```

```
     [,1]          [,2]
[1,]    1 -5.551115e-17
[2,]    0  1.000000e+00
```

Das in der Vorlesung betrachtete Beispiel für eine nicht-invertierbare Matrix ist auch numerisch nicht invertierbar, wie folgender Beispielcode in **R** demonstriert.

```
# nicht-invertierbare Matrizen sind auch numerisch nicht-invertierbar (= singulär)
B = matrix(c(1,0,
             0,0),
           nrow  = 2,
           byrow = TRUE)
solve(B)
```

```
Error in solve.default(B): Lapack routine dgesv: system is exactly singular: U[2,2] = 0
```

Determinanten

Determinanten berechnet man in **R** mithilfe des Befehls `det()`. Für die in der Vorlesung betrachteten Beispiele ergibt sich folgender **R**-Code.

```
# Beispiel 1
A = matrix(c(2,1,                    # Matrixdefinition
             3,4),
           nrow = 2,
           byrow = TRUE)
det(A)                               # Determinantenberechnung
```

```
[1] 5
```

```
B = matrix(c(1,0,                    # Matrixdefinition
             0,0),
           nrow = 2,
           byrow = TRUE)
det(B)                               # Determinantenberechnung
```

```
[1] 0
```

```
# Beispiel 2
C = matrix(c(2,0,0,                  # Matrixdefinition
             0,1,0,
             0,0,3),
           nrow = 3,
           byrow = TRUE)
det(C)                               # Determinantenberechnung
```

```
[1] 6
```

Weitere Aufgaben zur Übung

(1) Es seien

$$A := \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}, \quad B := \begin{pmatrix} 3 & 0 \\ 1 & 2 \end{pmatrix}, \quad \text{und} \quad c := 2 \tag{1}$$

Berechnen Sie

$$D := c\left(A - B^T\right) \quad \text{und} \quad E := (cA)^T + B. \tag{2}$$

mit **R**.

(2) Es seien

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 3 & 2 & 0 \end{pmatrix}, \quad B := \begin{pmatrix} 1 & 2 & 2 \\ 1 & 3 & 1 \\ 2 & 0 & 0 \end{pmatrix} \quad \text{und} \quad C := \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix}. \tag{3}$$

Berechnen Sie die Matrixprodukte

$$AB, \qquad B^T A^T, \qquad \left(B^T A^T\right)^T \quad \text{und} \quad AC \tag{4}$$

mit **R**.

(3) Invertieren Sie die Matrizen A und B aus der vorherigen Aufgabe in **R** mithilfe von `solve()` und überprüfen Sie, dass es sich bei den erzielten Ergebnissen tatsächlich um die Inversen von A und B handelt.

(4) Berechnen Sie die Determinanten von

$$A := \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \quad B := \begin{pmatrix} 3 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 3 \end{pmatrix} \quad \text{und} \quad C := \operatorname{diag}(1, 2, 3) \tag{5}$$

mit **R**.