

Statistische Software (R)

Paul Fink, M.Sc.

Institut für Statistik

Ludwig-Maximilians-Universität München

Lineare Algebra



- Vorbelegung aller Matrixelemente mit einer bestimmten Zahl

```
> matrix(nrow = 4, ncol = 2, data = 1)
      [,1] [,2]
[1,]  1   1
[2,]  1   1
[3,]  1   1
[4,]  1   1
```

- Konstruktion einer Diagonalmatrix, hier der Einheitsmatrix der Dimension 2

```
> diag(1, nrow = 2, ncol = 2)
      [,1] [,2]
[1,]  1   0
[2,]  0   1
```

Paul Fink: Statistische Software (R) SoSe 2015

2

Matrixoperationen

- Transponieren einer Matrix X : X'

```
> (x <- matrix(nrow = 4, ncol = 2, data = 1:8, byrow = TRUE))
      [,1] [,2]
[1,]  1   2
[2,]  3   4
[3,]  5   6
[4,]  7   8
> (xt <- t(x))
      [,1] [,2] [,3] [,4]
[1,]  1   3   5   7
[2,]  2   4   6   8
```

Matrixoperationen

- Matrixmultiplikation: Operator `%*%`, hier am Beispiel der Multiplikation von X' mit X

```
> xtx <- t(x) %*% x
> xtx
      [,1] [,2]
[1,]  84  100
[2,]  100  120
```

- Kreuzprodukt einer Matrix X , $X'X$, mittels der Funktion `crossprod()`

```
> xtx2 <- crossprod(x)
> xtx2
      [,1] [,2]
[1,]  84  100
[2,]  100  120
```

Bemerkung: dieser Befehl führt die Multiplikation schneller aus als die herkömmliche Methode mit `t(x) %*% x`

- Multiplikation einer Matrix mit einer Konstanten

```
> 4 * x
      [,1] [,2]
[1,]    4    8
[2,]   12   16
[3,]   20   24
[4,]   28   32
```

- Addition und Subtraktion von Matrizen (gleicher Dimension!) kann mittels der üblichen Operatoren + und - durchgeführt werden

- Zugriff auf komplette Zeilen, Spalten oder Blöcke einer Matrix

```
> x <- matrix(nrow = 5, ncol = 3, byrow = TRUE, data = 1:15)
> x
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
[4,]   10   11   12
[5,]   13   14   15
> x[3,]
[1] 7 8 9
> x[,2]
[1] 2 5 8 11 14
> x[4:5, c(1,3)]
      [,1] [,2]
[1,]   10   12
[2,]   13   15
```

Übersicht

<code>matrix()</code>	Erstellen einer Matrix
<code>t()</code>	Transponieren einer Matrix
<code>%*%</code>	Matrixmultiplikation
<code>%o%</code> , <code>outer()</code>	Äußeres Produkt
<code>crossprod()</code>	Kreuzprodukt
<code>solve()</code>	Invertieren
<code>det()</code>	Determinante
<code>backsolve()</code> , <code>forwardsolve()</code>	Lösen von Gleichungssystemen
<code>eigen()</code>	Eigenwerte und Eigenvektoren
<code>nrow()</code> , <code>ncol()</code>	Anzahl Zeilen und Spalte
<code>dim()</code>	Dimension

Aufgaben

1. Erstellen Sie die Matrix

$$X = \begin{pmatrix} 17 & 25 & 32 \\ 23 & 18 & 12 \\ 10 & 12 & 16 \\ 28 & 156 & 167 \end{pmatrix}$$

- a) Berechnen Sie $X'X$ auf 2 Arten! Wie ist die Dimension der Ergebnismatrix?
- b) Berechnen Sie XX' ! Wie ist die Dimension der Ergebnismatrix?

2. Erstellen Sie eine 10×1 Matrix **a**, welche die Zahlen 1, 3, 5, usw. enthält und eine 1×10 Matrix **b** mit den Zahlen 2, 4, 6, usw.
 - a) Was liefert $a * a$? (Gemeint ist: unter Verwendung der herkömmlichen Multiplikation)
 - b) Was liefert $b' * b'$?
 - c) Berechnen Sie das Matrixprodukt ab . Wie ist die Dimension der Ergebnismatrix?
 - d) Berechnen Sie das Matrixprodukt $a' b'$ auf 2 Arten! Welche Dimension hat dieses Ergebnis? Ist das Ergebnis vom Typ Matrix?

3. Mittels des Befehls `rnorm()` (siehe Hilfe) lassen sich eine oder mehrere normalverteilte (Pseudo-)Zufallszahlen erzeugen.
 - a) Erzeugen Sie eine Liste **l1** mit zwei (3×3) Matrizen **A1**, **A2**, die mit standardnormalverteilten Zufallsvariablen vorbelegt sind.
 - b) Erzeugen Sie eine weitere Liste **l2**, die als Elemente jeweils $A_i' A_i$, $i = 1, 2$ enthält. Mit welchem Befehl erhalten Sie das Element $A_{2,3}$?
 - c) Eine weitere Liste **l3** soll jetzt die Inversen der Matrizen in **l2** enthalten. Hinweis: `solve()`. Testen Sie ihre Ergebnisliste.