

Lineare Algebra mit dem Statistikprogramm „R“

1. Verwendung von Variablen

Variablen werden in R definiert, indem man einem Variablennamen einen Wert zuweist. Bei Variablennamen wird zwischen Groß- und Kleinbuchstaben unterschieden. Die Zeichenfolge „<-“ weist dem Variablennamen einen Wert zu.

Die Variablen können Zahlen, Vektoren oder Matrizen sowie Text enthalten.

Beispiel für die Verwendung einer numerischen Variablen. Bei dem Zeichen „>“ handelt es sich um die Eingabeaufforderung von R:

> A <- 2.5	A wird der Wert 2,5 zugewiesen
> A	Anzeige der Variablen A
[1] 2.5	R meldet das Ergebnis
> a*2	Multiplikation von a mit 2
Error: Object "a" not found	Achtung! „a“ wurde klein geschrieben
> A*2	Multiplikation von A mit 2
[1] 5	R meldet das Ergebnis
>	

Der Befehl zur Eingabe von Vektoren lautet „c()“.

Beispiel für die Verwendung eines Vektors:

> a <- c(1,2,3)	definiert den Vektor (1, 2, 3)
> b <- c(4,5,6)	definiert den Vektor (4, 5, 6)
> a	Ausgabe des Vektors a
[1] 1 2 3	

In R können auch Textvektoren definiert werden, dies ist sehr nützlich, wenn man später eine graphische Ausgabe der Daten durchführen möchte. Die Texte werden in Anführungszeichen gesetzt und durch Kommata voneinander getrennt.

Beispiel für Textvektoren:

```
> zeilennamen <- c("rot", "gelb", "gruen")  
> spaltennamen <- c("Test A", "Test B", "Test C")
```

Die Eingabe von Matrizen ist leider etwas komplizierter. Der Befehl lautet „matrix()“. Die Elemente der Matrix werden als Vektor eingegeben, man muß R anschließend mitteilen, in wieviele Zeilen bzw. Spalten dieser Vektor „zerlegt“ werden soll. Dies geschieht durch das Argument „nrow=x“. Wird R nichts weiter mitgeteilt, dann geht das Programm davon aus, daß die Elemente der Matrix Spalte für Spalte im Vektor enthalten sind. Das Argument „byrow=T“ bewirkt, daß die Elemente Zeile für Zeile aus dem Vektor ausgelesen werden.

Beispiel für die Eingabe einer Matrix:

```
> A <- matrix(c(1,2,3,4,5,6,7,8,9), nrow=3)
```

Ausgabe:

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

```
> A <- matrix(c(1,2,3,4,5,6,7,8,9), byrow=T, nrow=3)
```

Ausgabe:

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

Der Matrix können während der Eingabe Zeilen- und Spaltenbezeichnungen zugeordnet werden. Die Bezeichnungen für die Zeilen und Spalten sind selbst Textvektoren.

Beispiel:

```
> A <- matrix(c(1,2,3,4,5,6,7,8,9), byrow=T, nrow=3,
dimnames = list(zeilennamen, spaltennamen))
> A
      Test A Test B Test C
rot      1      2      3
gelb     4      5      6
gruen    7      8      9
```

2. Rechnen mit Vektoren und Matrizen

Die Addition von Vektoren und Matrizen erfolgt durch das Zeichen „+“. Beispiel:

```
> x1 <- c(6,4,2)
> x2 <- c(3,4,5)

> x1+x2
[1] 9 8 7
> X <- matrix(c(2,4,6,8,10,12), byrow=T, nrow=2)
> Y <- matrix(c(3,5,7,9,11,13), byrow=T, nrow=2)
> X+Y
      [,1] [,2] [,3]
[1,]    5    9   13
[2,]   17   21   25
```

Die Multiplikation von Vektoren und Matrizen nach den Regeln der linearen Algebra erfolgt nicht durch das Zeichen „*“, sondern durch die Zeichenfolge „%*%“. Das Zeichen „*“ bewirkt die elementweise Multiplikation.

Das Transponieren eines Vektors bzw. einer Matrix erfolgt durch den Befehl „t()“. Beispiel:

```
> x1*x2
[1] 18 16 10
> X*Y
      [,1] [,2] [,3]
[1,]    6   20   42
[2,]   72  110  156
> x1%*%t(x2)
      [,1] [,2] [,3]
[1,]   18   24   30
[2,]   12   16   20
[3,]    6    8   10
> t(x1)%*%x2
      [,1]
[1,]   44
> X%*%t(Y)
      [,1] [,2]
[1,]   68  140
[2,]  158  338
> t(X)%*%Y
      [,1] [,2] [,3]
[1,]   78   98  118
[2,]  102  130  158
[3,]  126  162  198
```

Einzelne Elemente einer Matrix lassen sich ansprechen, indem man den entsprechenden Zeilen- bzw. Spaltenbereich in eckigen Klammern hinter der Matrix angibt. Beispiel:

```
> A <- t(X) %*% Y
> A
      [,1] [,2] [,3]
[1,]   78   98  118
[2,]  102  130  158
[3,]  126  162  198
```

```
> A[2:3,2:3]
      [,1] [,2]
[1,]  130  158
[2,]  162  198
> A[1:3,1]
[1]  78 102 126
```

Die Determinante einer quadratischen Matrix erhält man durch den Befehl „det()“. Beispiel:

```
> A
      [,1] [,2] [,3]
[1,]   78   98  118
[2,]  102  130  158
[3,]  126  162  198
> det(A)
[1] 0
```

Die Inverse einer quadratischen Matrix erhält man durch den Befehl „solve()“. Beispiel:

```
> B <- matrix(c(1,2,3,4), byrow=T, nrow=2)
> solve(B)
      [,1] [,2]
[1,] -2.0  1.0
[2,]  1.5 -0.5
> B %*% solve(B)
      [,1] [,2]
[1,]    1 1.110223e-16
[2,]    0 1.000000e+00
```

Eigenwerte und Eigenvektoren einer (symmetrischen) Matrix liefert der Befehl „eigen()“. Beispiel:

```
> C <- B %*% t(B)
> eigen(C)

$values
[1] 29.8660687  0.1339313

$vectors
      [,1] [,2]
[1,] 0.4045536  0.9145143
[2,] 0.9145143 -0.4045536
```

Eigenwerte und Eigenvektoren lassen sich einzeln ansprechen. Hierfür verwendet man den Befehl „eigen()\$values“ bzw. „eigen()\$vectors“. Beispiel:

```
> L <- eigen(C)$values
> E <- eigen(C)$vectors
> L
[1] 29.8660687 0.1339313

> E
      [,1]      [,2]
[1,] 0.4045536 0.9145143
[2,] 0.9145143 -0.4045536
```

Weitere Befehle:

Die Summe der Elemente eines Vektors lassen sich durch „sum()“ bestimmen. Beispiel:

```
> a <- c(1,2,3)
> sum(a)
[1] 6
```

Das Format einer Matrix läßt sich durch die Befehle „nrow()“ bzw. „ncol()“ bestimmen. Beispiel:

```
> Y <- matrix(c(3,5,7,9,11,13), byrow=T, nrow=2)
> Y
      [,1] [,2] [,3]
[1,] 3    5    7
[2,] 9    11   13
> nrow(Y)
[1] 2
> ncol(Y)
[1] 3
```

3. Erstellung von Programmen und graphische Ausgabe

Eine Folge von Befehlen läßt sich zu einem Programm zusammenstellen. Es handelt sich hierbei um eine Textdatei mit der Endung „.r“.

Programmstart

Der Aufruf eines Programms erfolgt durch den Befehl „source(„Programm.r“)“. Beispiel:

```
> source("optscal.r")
```

Kommentare

An beliebiger Stelle können innerhalb des Programms Kommentare eingegeben werden. Dies erfolgt, indem man dem Kommentar das Zeichen „#“ voranstellt. Beispiel:

```
I <- nrow(X)           # Ermittlung der Zeilenzahl von X
```

Löschen aller Variablen

Alle im Speicher vorhandenen Variablen werden durch den Befehl „rm(list=ls())“ gelöscht. Mit dem Befehl „ls()“ kann man sich anzeigen lassen, welche Variablen verwendet werden.

Erstellen von speziellen Matrizen

Manchmal ist es hilfreich, Matrizen zu erstellen, die z.B. nur aus Nullen oder nur aus Einsen bestehen. Hierfür gibt es den Befehl „rep()“. Beispiel für eine Matrix vom format (3,3), die nur aus Einsen besteht:

```
EinsII <- matrix(rep(1, times=9), ncol = 3)
```

Graphische Ausgabe

Die Graphische Ausgabe von zweidimensionalen Koordinaten erfolgt über den Befehl „plot()“. Das erste Argument ist der Vektor, der die Koordinaten für die erste Achse enthält, das zweite Element ist der Vektor mit den Elementen für die zweite Achse. Das Argument „pch=“ gibt an, welches Zeichen für die Darstellung verwendet werden soll (Standard ist „o“). Mit den beiden folgenden Argumenten „xlab=“ und „ylab=“ kann man Texte für die Achsenbeschriftung bestimmen. Beispiel:

```
plot(K12[1:I,1],K12[1:I,2], pch=".",xlab="1. Achse", ylab="2. Achse")
```

Um den Punkten Texte zuzuordnen, kann man den Befehl „text()“ verwenden. Das erste Argument ist die erste Koordinate für den entsprechenden Punkt, das zweite die zweite Koordinaten, als drittes Argument wird der Text eingegeben. Beispiel:

```
text(K12[i,1],K12[i,2], "Text")
```

Verwendung von Schleifen

Da man üblicherweise mehr als nur einen Punkt bezeichnen möchte, empfiehlt sich die Verwendung von Schleifen. Hat man z.B. I Punkte, so lautet der Befehl für I Durchläufe mit dem Laufindex i „for (i in 1:I)“. Die entsprechende Befehlsfolge, die mit jedem Durchlauf ausgeführt werden soll, wird in geschweifte Klammern eingekleidet. Beispiel:

```
for (i in 1:I) {text(K12[i,1],K12[i,2],rnamen[i])}
```