Multivariate Statistik mit R

Reinhold Kosfeld

Inhalt

1. Einführung

1.1 Vorbemerkungen
1.2 Rechnen mit Zahlen und Funktionen
1.3 Variablen, Vektoren und Matrizen
1.4 Einlesen und Überprüfen von Datendateien

2. Uni- und bivariate Datenanalyse

2.1 Univariate Datenanalyse
2.2 Bivariate Datenanalyse
2.3 Statistische Auswertung im R-Commander

3. Faktorenanalyse

3.1 Bestimmung und Beurteilung der Korrelationsmatrix
3.2 Eigenwerte und Eigenvektoren der Korrelationsmatrix
3.3 Hauptkomponentenmethode und Faktormatrix
3.4 Faktorenanalyse im engeren Sinne (Hauptachsen-Faktorenanalyse)
3.5 Rotation der Faktormatrix
3.6 Bestimmung der Faktorwerte

4. Varianzanalyse

4.1 Daten, Variablen und Faktoren
4.2 Einfaktorielle Varianzanalyse
4.3 Zweifaktorielle Varianzanalyse ohne Interaktion
4.4 Zweifaktorielle Varianzanalyse mit Interaktion
4.5 Nonorthogonale Varianzanalyse

5. Clusteranalyse

5.1 Ähnlichkeitsmessung bei quantitativen Merkmalen

5.2 Ähnlichkeitsmessung bei qualitativen Merkmalen

5.3 Hierarchische Clusteranalyse und Dendrogramm

5.4 Partitionsverfahren

6. Diskriminanzanalyse

6.1 Streuungsmatrizen der Merkmalsvariablen

6.2 Optimale Gruppentrennung und Diskriminanzfunktion

6.3 Beurteilung der Merkmalsvariablen

6.4 Güte der Diskrimination

6.5 Klassifikation

1. Einführung
1.1 Vorbemerkungen

Die Statistiksoftware und Programmierumgebung R wird von einer internationalen Entwicklergemeinschaft als Open Source Software entwickelt und kann von jedermann entsprechend der GNU General Public License Version 2 (Juni 1991) frei genutzt und auch weiter verbreitet werden. Die Software wird auf der Homepage des R - Projektes (http://www.r-project.org) als Download bereitgestellt. Quellcode und Binärdateien für diverse Betriebssysteme werden durch CRAN bereitgestellt (http://CRAN.R-project.org).

R ist eine hochflexible, interpretierte Programmiersprache und –umgebung zur statistischen und grafischen Datenanalyse. Es handelt sich hierbei um keine vollständige, grafische Benutzeroberfläche (GUI), jedoch sind Werkzeuge zu ihrer Entwicklung vorhanden. In dem R-Commander (library(Rcmdr)) lassen sich aktuell bereits einige Methoden der Datenanalyse menügesteuert ausführen. Bei der Ausführung von R-Funktionen erfolgt in der Regel keine Ausgabe aller errechneten Werte. Vielmehr werden einige Ergebnisse erst einmal in Objekten zwischengespeichert. Sie können später jederzeit abgerufen werden. Andererseits speichert R keine Ergebnisse ab, wenn kein Objekt hierzu angegeben wird.

R kann sowohl interaktiv oder im Batch-Modus genutzt werden. Der Batch-Modus sollte bei größeren Befehlsfolgen verwendet werden, in dem eine Skript-Datei ausgeführt wird. Das Programm ist bereit, wenn das Zeichen > erscheint. Befehle können durch ein Semikolon (;) oder durch den Beginn einer neuen Zeile voneinander getrennt werden. Ist ein Befehl nicht vollständig, erscheint das Pluszeichen (+). Bereits verwendete Befehle können mit den Pfeiltasten (↑ und ↓) abgerufen werden. Mit dem Zeichen # wird ein Kommentar eingeleitet. Als Dezimalzeichen wird der Punkt (.), nicht das Komma (,) verwendet. Die ESC-Taste unterbricht die gerade ausgeführte Rechenprozedur unterbrochen.

Alle in einer R-Sitzung vorhandenen Objekte können mit dem Befehl ls() angezeigt. Ein Objekt obj lässt sich mit dem remove-Befehl rm(obj) löschen. Texte wie z.B. „Multivariate Statistik macht Spaß“ können unter Verwendung von Anführungszeichen mit einem Gleichheitszeichen (=) oder Pfeil (<-) einem Objekt obj zugewiesen werden:

```
> obj = "Multivariate Statistik macht Spaß"   oder   >obj <- "Multivariate Statistik macht Spaß"
> obj
[1] " Multivariate Statistik macht Spaß ".
```

Hier wird bei Zuweisungen allein von dem Gleichheitszeichen (=) Gebrauch gemacht. In diesem Fall ist das Objekt obj vom Typ character (String-Objekt (Textzeichenfolge)).

Die verfügbaren Objekte in der aktuellen R-Umgebung werden mit dem Befehl ls() in der R-Konsole angezeigt:

```
> ls()
 [1] "obj"
```

Das aktuelle Arbeitsverzeichnis (working directory) kann mit dem getwd()-Befehl überprüft werden:

```
> getwd()
[1] "C:/Users/Kosfeld/Documents/Multivariate Statistik/Multivariate Statistik mit R/Daten"
```

Es kann über das R-Menü File – Change Dir… oder mit dem setwd()-Befehl geändert werden:

```
> setwd("C:/Users/Kosfeld/Documents/Multivariate Statistik/Multivariate Statistik mit R/Daten").
```

Verlassen werden kann das Programm R durch den quit-Befehl q() oder durch Anklicken des Symbols x des Programmfensters RGui (64-bit)..

1.2 Rechnen mit Zahlen und Funktionen

Das Programm R kann als Taschenrechner benutzt werden, indem Rechenoperationen wie Addition (+), Subtraktionen (-), Multiplikationen (*), Divisionen (/) und Potenzierung (^) direkt ausgeführt werden:

```
> 3+4
[1] 7

> 7-12
[1] -5

> 5*6
[1] 30

> 9/4
[1] 2.25
```

Die Exponentiation mit einer beliebigen Basis erfolgt mit dem Zeichen ^,

```
> 2^4
[1] 16,
```

während sie zur Basis e (Eulersche Zahl: e = 2.718282) mit der exp-Funktion ausgeführt wird:

```
>exp(3)
[1] 20.08554
```

Die Quadratwurzel wird mit der sqrt-Funktion gezogen:

```
>sqrt(64)
[1] 8
```

Der Logarithmus zur Basis e wird mit der log-Funktion gebildet,

```
> log(8)
[1] 2.079442
```

und der Logarithmus zur Basis 10 mit der log10-Funktion:

```
> log10(1000)
[1] 3
```

Neben den arithmetischen Funktionen exp, sqrt und log führen wir an dieser Stelle die trigonometrischen Funktionen sin, cos und tan sowie ihre Umkehrfunktionen asin, acos und atan ein. Als Argument der Sinusfunktion muss der Winkel in Bogenmaß z.B. 2 (= 2*180/pi = 114.5916°) angegeben werden:

```
> sin(2)
[1] 0.9092974
```

Die Kreiskonstante π ist in R unter pigespeichert:

```
>pi
[1] 3.141593
```

Mit der Konstanten pi lässt sich der mit der Arcussinus-Funktion asin in Bogenmaß ermittelte Winkel,

```
>asin(0.9)
[1] 1.119770,
```

in einen Winkel in Grad transformieren:

```
>asin(0.9)*180/pi
[1] 64.15807
```

Mit der Funktion round werden Zahlen auf ganze Stellen,

```
>round(1.8786)
[1] 2
```

oder auf die durch das Argument digit=2 angegebenen Dezimalstellen gerundet:

```
>round(1.8786, digit=2)
[1] 1.88
```

Übungsaufgaben

1.2.1 Berechnen Sie den Ausdruck $4 \cdot (6\frac{1}{2} - \frac{1}{2}) / 2$!

1.2.2 Berechnen Sie den Ausdruck $(8\frac{1}{3} + 2\frac{1}{2})^2 / \sqrt{2}$ und runden Sie ihn auf drei Dezimalstellen!

1.2.3 Berechnen Sie den Ausdruck $\log_{10} 1000 + \sqrt[3]{27}$!

1.2.4 Berechnen Sie den Ausdruck $x^2 = \log_2 16$!

1.2.5 Berechnen Sie den Ausdruck $\ln x = \sqrt{8}$!

1.3 Variablen, Vektoren und Matrizen

In den vorherigen Beispielen geht nach Auswertung das Ergebnis nach seiner Ausgabe am Bildschirm verloren. Um die Ergebnisse der Rechenoperationen in einer Sitzung verfügbar zu haben, müssen sie unter einem Variablennamen gespeichert werden.

Das Ergebnis wird nicht mit der Zuweisung zu einer Variablen ausgegeben, sondern erst mit einem weiteren Befehl durch Nennung des Variablennamens:

```
> a = log10(1000)
> a
[1] 3
```

Das Objekt a ist hier der Spezialfall eines numerischen Vektors mit nur einem Element. Ein Vektor ist in R die einfachste Datenstruktur oder der einfachste Objekttyp.

Allgemein lässt sich ein Vektor durch Zuweisung von Zahlen mit Hilfe der Funktion c() definieren:

```
> x = c(3, 6, 7, 1, 9, 4)
> x
[1] 3 6 7 1 9 4
```

Ein Vektor mit einer Folge aufeinanderfolgender natürlicher Zahlen wird durch

```
> y = 1:6
> y
[1] 1 2 3 4 5 6
```

kreiert. Eine allgemeinere Sequenz von Zahlen kann mit dem seq()-Befehl durch Angabe des Inkrements z.B. by = 0.5

```
> a = seq(1, 4, by = 0.5)
```

```
>a
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0
```

oder der Länge z.B. length = 7

```
> b = seq(1, 4, length=7)
> b
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0
```

einem Vektor zugewiesen werden. Das erste Argument gibt dabei jeweils den Startwert (from = 1), das zweite den Endwert (to = 4) an.

Unter Verwendung der einfachen Operatoren +, -, *, / und ^ werden Operationen mit Vektoren elementweise ausgeführt. In den folgenden Beispielen werden die Vektoren x,

```
> x
[1] 3 6 7 1 9 4,
```

und y,

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

verwendet:

```
> d = x + 3
> d
[1]  6  9 10  4 12  7

> e = 2*x
> e
[1]  6 12 14  2 18  8

> f = x + y
> f
[1]  4  8 10  5 14 10

> g = x * y
> g
[1]  3 12 21  4 45 24
```

Um das Skalarprodukt der beiden Vektoren x und y zu bilden, muss der Operator %*% verwendet werden:

```
> t(x)%*%y
    [,1]
[1,]  109
```

Hierbei wird der Zeilenvektor t(x) mit dem Spaltenvektor y multipliziert. Die Transponation eines Spaltenvektors in einen Zeilenvektor oder eines Zeilenvektors in einen Spaltenvektor erfolgt mit der Funktion t().

Das äußere Produkt der beiden Vektoren x und y ergibt sich durch Multiplikation des Spaltenvektors x mit dem Zeilenvektor t(y) unter Verwendung des Multiplikationsoperators%*%:

```
> x%*%t(y)
 [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    3    6    9   12   15   18
[2,]    6   12   18   24   30   36
[3,]    7   14   21   28   35   42
[4,]    1    2    3    4    5    6
[5,]    9   18   27   36   45   54
[6,]    4    8   12   16   20   24
```

Das Ergebnis ist in diesem Fall eine 6x6-Matrix.

Aus der Sicht der Statistik lassen sich in einem Vektor die Werte eines Merkmals (Variablen) darstellen, die an n Untersuchungseinheiten erhoben worden sind. Allgemein werden bei einer Erhebung jedoch die Daten von m Merkmalen erfasst. Diese Daten lassen sich zweckmäßig in einer nxm-Matrix anordnen.

Eine Matrix lässt sich z.B. aus den Werten eines Vektors durch Angabe der Zeilen- und Spaltenzahl (nrow und ncol) mit der Funktion matrix erzeugen:

```
> A = matrix(c(3,5,6,2,8,4), nrow=3, ncol=2)
> A
 [,1] [,2]
[1,]    3    2
[2,]    5    8
[3,]    6    4
```

Das erste Argument data ist hierbei mit der Funktion c() spezifiziert. Standardmäßig wird die Matrix spaltenweise ausgefüllt. Sollen die Werte zeilenweise eingetragen werden, muss das Argument byrow gleich TRUE gesetzt werden:

```
> B = matrix(1:6, nrow=3, ncol=2, byrow=TRUE)
> B
[,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
```

Mit colnames() können den Spalten einer Matrix Namen zugeordnet werden:

```
>colnames(B) = c("b1", "b2")
```

```
> B
b1 b2
[1,]  1  2
[2,]  3  4
[3,]  5  6
```

Entsprechend lassen sich mit rownames() den Zeilen einer Matrix Namen zuordnen. Beides gleichzeitig wird mit dem Befehl dimnames() bewerkstelligen:

```
>dimnames(B) = list(c("B1", "B2", "B3"),c("b1", "b2"))
> B
 b1 b2
B1  1  2
B2  3  4
B3  5  6
```

Die Dimensionen von Matrizen können mit der dim-Funktion abgefragt werden:

```
>dim(B)
[1] 3 2
```

Der erste Wert gibt die Zeilenzahl, der zweite die Spaltenzahl an.

Für manche Berechnungen ist es zweckmäßig, die Zeilen- und Spalten einer Matrix als Vektoren zu definieren. Enthalten die beiden Spalten der Matrix B die Werte zweier Variablen, können diese extrahiert und z.B. den Variablenvektoren b1 und b2zugeordnet werden:

```
> b1 = B[,1]
> b1
B1 B2 B3
 1  3  5
> b2 = B[,2]
> b2
B1 B2 B3
 2  4  6.
```

Mit den Funktionen rbind und cbind lassen sich umgekehrt Vektoren durch zeilen- bzw. spaltenweises Zusammenfügen zu einer Matrix verbinden. So erhält man durch eine zeilenweise Verknüpfung der Vektoren b1 und b2 die Matrix B1,

```
> B1 = rbind(b1,b2)
> B1
   B1 B2 B3
b1  1  3  5
b2  2  4  6,
```

und durch spaltenweise Verknüpfung die Matrix B2:

```
> B2 = cbind(b1,b2)
> B2
b1 b2
B1  1  2
B2  3  4
B3  5  6.
```

Übungsaufgaben

1.3.1Weisen Sie der Variablen x den Wert $4^3 - 6 \cdot 3$ und der Variablen y den Ausdruck $3 \cdot x + 5$ zu. Berechnen Sie den Ausdruck $z = \left(\frac{1}{3}x^2 - x \cdot \sqrt{y}\right)$!

1.3.2 Berechnen Sie $s = \frac{\sqrt[3]{(y^3 - 3 \cdot x + 4)}}{y - 2}$ unter Verwendung der in Aufgabe 1.3.1 angegebenen Zuweisungen für x und y!

1.3.3 Definieren Sie einen Vektor x aus den Werten 3, 5, 8, 2, 4, 1!

1.3.4 Erzeugen Sie einen Vektor y mit den Komponenten 4, 5, 6, 7, 8, 9 auf drei unterschiedliche Arten!

1.3.5 Addieren und multiplizieren Sie die in den Aufgaben 1.3.3 und 1.3.4 gebildeten Vektoren x und y elementweise und speichern Sie die Ergebnisse unter den Objektnamen s bzw. m!

1.3.6 Bilden Sie das Skalarprodukt der beiden Vektoren x und y (s. Aufg. 1.3.3 und 1.3.4)!

1.3.7 Bilden Sie aus den beiden Vektoren x und y (s. Aufg. 1.3.3 und 1.3.4) 2x3-Matrizen A und B!

1.3.8 Multiplizieren Sie die Matrix A mit der transponierten Matrix B (s. Aufg. 1.3.7) und speichern Sie das Matrixprodukt unter dem Objektnamen AB!

1.3.9 Füllen Sie eine 3x3-Matrix C zeilenweise mit den Werten 1 bis 9!

1.3.10 Weisen Sie den Spalten der in Aufg. 1.3.9 gebildeten Matrix C die Namen c1, c2 und c3 zu!

1.3.11 Verbinden Sie die in den Aufgaben 1.3.3 und 1.3.4 gebildeten Vektoren x und y zu einer 6x2-Matrix D!

1.4 Einlesen und Überprüfen von Datendateien

Multivariate Datensätze sind in der Regel in Datendateien gespeichert. Sie müssen für eine multivariate Analyse in R eingelesen werden. Für ASCII-Dateien steht hierzu der Befehl read.table zur Verfügung. Sind die Daten in einer .xls- oder .xlsx-Datei gespeichert, können sie mit einem Tabellenkalkulationsprogramm wie z.B. Excel in eine CSV-Textdatei (Comma Separated Values) umgewandelt werden. Bei Verwendung des Dezimalpunkts (Dezimalkommas) lassen sie sich dann mit dem read.csv-Befehl (read.csv2-Befehl) in R einlesen.

Zur Illustration der multivariaten Verfahren wird häufig auf einen regionalen Datensatz zurückgegriffen, der aus der in Excel erstellten CSV-Datei Regionen12 eingelesen werden soll:

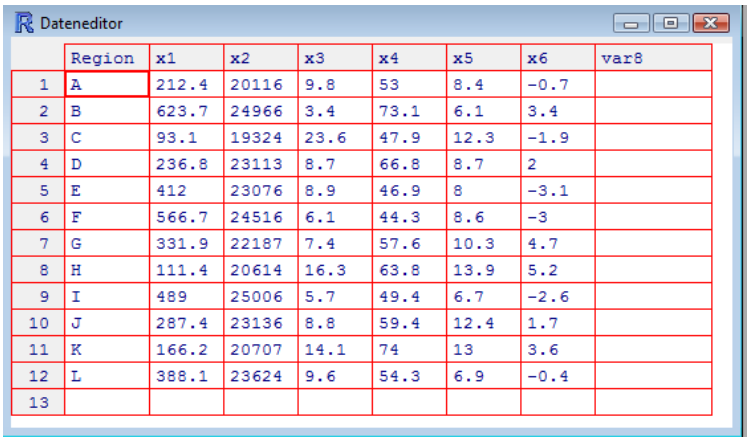
```
>Regionen12 = read.csv(file="Regionen12.csv", dec=".", sep=";", header=TRUE)
> Regionen12
  Region   x1    x2   x3   x4   x5   x6
1      A 212.4 20116  9.8 53.0  8.4 -0.7
2      B 623.7 24966  3.4 73.1  6.1  3.4
3      C  93.1 19324 23.6 47.9 12.3 -1.9
4      D 236.8 23113  8.7 66.8  8.7  2.0
5      E 412.0 23076  8.9 46.9  8.0 -3.1
6      F 566.7 24516  6.1 44.3  8.6 -3.0
7      G 331.9 22187  7.4 57.6 10.3  4.7
8      H 111.4 20614 16.3 63.8 13.9  5.2
9      I 489.0 25006  5.7 49.4  6.7 -2.6
10     J 287.4 23136  8.8 59.4 12.4  1.7
11     K 166.2 20707 14.1 74.0 13.0  3.6
12     L 388.1 23624  9.6 54.3  6.9 -0.4
```

Neben dem Dateinamen (file) sind in dem read.csv-Befehl einige optionale Parameter angegeben. Das Argument dec gibt an, dass den Punkt in der Datendatei Regionen12.csv als Dezimalzeichen verwendet worden ist. Die Werte selbst sind – nicht sichtbar - in dieser Datei durch Semikola voneinander getrennt (sep=";"). Der logische Wert TRUE des Arguments headerzeigt an, dass die erste Zeile der Datei Regionen12.csv die Variablennamen enthält.

Der Inhalt der Datei Regionen12.csv wird dem R-Objekt Regionen12 zugewiesen. Die erste Spalte enthält die Namen der Regionen, für die Beobachtungswerte von sechs quantitativen Variablen vorliegen, unter dem Variablennamen Region. Die Beobachtungswerte der metrisch skalierten Variablen befinden sich in den Spalten nachfolgenden Spalten mit den Bezeichnungen x_1, x_2, …, x_6, die die Variablennamen wiedergeben.

Mit den R-Befehlen fix und edit wird ein Dateneditor mit dem angegebenen Datensatz geöffnet:

```
> fix(Regionen12).
```



Dateneditor

	Region	x1	x2	x3	x4	x5	x6	var8
1	A	212.4	20116	9.8	53	8.4	-0.7	
2	B	623.7	24966	3.4	73.1	6.1	3.4	
3	C	93.1	19324	23.6	47.9	12.3	-1.9	
4	D	236.8	23113	8.7	66.8	8.7	2	
5	E	412	23076	8.9	46.9	8	-3.1	
6	F	566.7	24516	6.1	44.3	8.6	-3	
7	G	331.9	22187	7.4	57.6	10.3	4.7	
8	H	111.4	20614	16.3	63.8	13.9	5.2	
9	I	489	25006	5.7	49.4	6.7	-2.6	
10	J	287.4	23136	8.8	59.4	12.4	1.7	
11	K	166.2	20707	14.1	74	13	3.6	
12	L	388.1	23624	9.6	54.3	6.9	-0.4	
13								

Bei Verwendung des edit-Befehls wird der Datensatz nach Schließen des Dateneditors zusätzlich in der R-Konsole aufgelistet.

Mit dem Befehl

```
>class(Regionen12)
[1] "data.frame"
```

wird der Typ des R-Objekts Regionen12 ausgegeben. Regionen12 ist ein Objekt vom Typ data.frame, d.h. ein Datensatz, der wie ein Matrixobjekt aus Zeilen und Spalten besteht. In den Zeilen stehen die Beobachtungswerte der statistischen Einheiten (Regionen), in den Spalten die Variablen. Im Unterschied zu einem matrix-Objekt kann ein data.frame-Objekt jedoch auch nicht-nummerische Werte z.B. für qualitative Variablen enthalten.

Der str-Befehl

```
> str(Regionen12)
'data.frame':   12 obs. of  7 variables:
 $ Region: Factor w/ 12 levels "A","B","C","D",..: 1 2 3 4 5 6 7 8 9 10 ...
$ x1    : num  212.4 623.7 93.1 236.8 412 ...
 $ x2    : int  20116 24966 19324 23113 23076 ...
 $ x3    : num  9.8 3.4 23.6 8.7 8.9 6.1 7.4 16.3 5.7 8.8 ...
 $ x4    : num  53 73.1 47.9 66.8 46.9 44.3 57.6 63.8 49.4 59.4 ...
 $ x5    : num  8.4 6.1 12.3 8.7 8 8.6 10.3 13.9 6.7 12.4 ...
 $ x6    : num  -0.7 3.4 -1.9 2 -3.1 -3 4.7 5.2 -2.6 1.7 ...
```

gibt Aufschluss über die Struktur des data.frame-Objekts Regionen12.Die quantitativen Variablen x_1, x_2, …, x_6 sind durch die Kennzeichnung numbzw. intals numerische Vektoren ausgewiesen. Die qualitative Variable Region ist als factor-Objekt definiert, das vergleichbar mit einem character-Objekt ist.[1]

Der Datensatz Regionen12 enthält folgende sozioökonomische Merkmale:

x_1: Einwohnerdichte (Einwohner pro qkm) (ED)
x_2: Bruttoinlandsprodukt (BIP) pro Kopf (Euro) (BIP)
x_3: Anteil der Erwerbstätigen in der Landwirtschaft (Prozent) (EL)
x_4: Wachstumsrate des BIP (in den letzten zehn Jahren, in Prozent) (WBIP)
x_5: Geburtenrate (Lebendgeborene je 1000 Einwohner) (GEB)
x_6: Wanderungssaldo (Zu- minus Fortwanderungen je 1000 Einwohner) (WS)

Auf sie kann durch Verknüpfung des data.frame-NamensRegionen12mit dem jeweiligen Variablennamen unter Verwendung des Dollarzeichens ($) separat zugegriffen werden wie z.B.

[1] Die Unterschiede zwischen einem factor- und character-Objekt werden bei der Varianzanalyse (Kap. 3) aufgezeigt.

```
> Regionen12$Region
 [1] A B C D E F G H I J K L
Levels: A B C D E F G H I J K L
```

oder

```
> Regionen12$x1
 [1] 212.4 623.7  93.1 236.8 412.0 566.7 331.9 111.4 489.0 287.4 166.2 388.1.
```

Mit dem Befehl

```
>attach(Regionen12)
```

lassen sich die Variablen des Datensatzes Regionen12 direkt zugänglich machen z.B.

```
> Region
 [1] A B C D E F G H I J K L
Levels: A B C D E F G H I J K L
```

oder

```
> x1
 [1] 212.4 623.7  93.1 236.8 412.0 566.7 331.9 111.4 489.0 287.4 166.2 388.1.
```

Die Umwandlung der Variablen Region in einen alphanumerischen Vektor lässt sich mit dem Befehl

```
> Region = as.character(Region)
```

vornehmen:

```
>class(Region)
[1] "character".
```

Aus dem str-Befehl

```
>str(Region)
chr [1:12] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L"
```

geht hervor, dass die character-Variable im Unterschied zur factor-Variablen Zeichenketten („characterstrings“) enthält, die in Anführungszeichen (" ") stehen.

Der Suchpfad zum Datensatz Regionen12 wird mit dem Befehl

```
>detach(Regionen12)
```

wieder aufgehoben.

Um mit den Variablen x1 (Einwohnerdichte) und x2 (BIP pro Kopf) ohne Rückgriff auf den Datensatz Region12 Berechnungen durchführen zu können, werden sie als Vektoren ED und BIP definiert:

```
> ED = Regionen12$x1
> ED
 [1] 212.4 623.7  93.1 236.8 412.0 566.7 331.9 111.4 489.0 287.4 166.2 388.1
>class(ED)
[1] "numeric"
>str(ED)
num [1:12] 212.4 623.7 93.1 236.8 412 ...

> BIP = Regionen12$x2
> BIP
 [1] 20116 24966 19324 23113 23076 24516 22187 20614 25006 23136 20707 23624
>class(BIP)
[1] "integer"
>str(BIP)
int [1:12] 20116 24966 19324 23113 23076 24516 22187 20614 25006 23136 ...
```

Übungsaufgaben

1.4.1 Erstellen Sie eine CSV-Datei Produktivitaet24.csv mit den Variablen Arbeiter, PM (= Produktionsmethode), AG (= Arbeitergruppe) und PROD (= Produktivität) aus der Excel-Datei Produktivitaet24.xlsxundlesen Sie sie als data.frame-Objekt Produktivitaet24 in R ein!

1.4.2 Rufen Sie den Dateneditor mit den Befehlen fix und edit auf und erhöhen Sie die Produktivität des ersten Arbeiters von 69 auf 79! Mit welchem der beiden Befehle bleibt diese Änderung erhalten? Machen Sie mit dem entsprechenden Befehl die Änderung wieder rückgängig!

1.4.3 Definieren Sie die Variable Prod des data.frame-ObjektsProduktivitaet24 als Vektor Prod!

1.4.4 Definieren Sie die Variablen PM und AG des data.frame-Objekts Produktivitaet24 als

- Faktoren PMf und AGf,
- alphanumerische Vektoren PMa und AGa!

1.4.5 Lesen Sie die CSV-Datei Documenta mit den Variablen v103, v104, …, v112,

- Ich wollte ein kulturelles Ereignis erleben (v103),
- Ich wollte mir einen Überblick über das zeitgenössische Kunstgeschehen verschaffen(v104),
- Ich wollte mir bestimmte Künstler/ -innen anschauen (v105),
- Ich habe die DOCUMENTA IX aus beruflichen Interessen besucht (v106),
- Wir wollten etwas Gemeinsames unternehmen, da bot es sich die DOCUMENTA IX einfach an (v107),
- Ich interessiere mich speziell für die aktuellen Entwicklungstrends in der bildenden Kunst (v108),
- Mir kommt es sehr auf das documenta-Flair an (v109),
- Ich wollte etwas ästhetisch Schönes erleben (v110),

- Ich wollte mich auf dem Gebiet der Kunst weiterbilden (v111),
- Ich/ wir wollte(n) etwas Unterhaltsames unternehmen (v112)

die auf einer fünfstufigen Ratingskala (1: trifft völlig zu; 5: trifft überhaupt nicht zu) gemessen sind, als data.frame-Objekt Documenta in R ein!

1.4.6 Ermitteln Sie die Anzahl der Beobachtungen mit zwei verschiedenen R-Funktionen! Überprüfen Sie das Ergebnis im Dateneditor!

1.4.7 Definieren Sie die Variablen v103 und v104 als numerische Vektoren gleichen Namens!

2. Uni- und bivariate Datenanalyse

2.1 Univariate Datenanalyse

Unter univariater Analyse versteht man die separate Auswertung der Daten im Hinblick auf die einzelnen Merkmale. Neben grafischen Darstellungen gehört hierzu die Berechnung von Kenngrößen wie Mittelwerte, Quantile und Streuungsmaße, auf die hier eingegangen wird.

Durch Anwendung der Funktion summary auf das data.frame-Objekt Regionen12gibt R einige Lagemaße für alle hierin enthaltenen Variablen aus:

```
>summary(Regionen12)
 Region    x1          x2          x3          x4          x5          x6
 A    :1   Min.  : 93.1   Min.  :19324   Min.  : 3.400   Min.  :44.30   Min.  : 6.100   Min.  :-3.1000
B    :1   1st Qu.:200.8   1st Qu.:20684   1st Qu.: 7.075   1st Qu.:49.02   1st Qu.: 7.725   1st Qu.:-2.0750
 C    :1   Median :309.6   Median :23095   Median : 8.850   Median :55.95   Median : 8.650   Median : 0.6500
 D    :1   Mean   :326.6   Mean   :22532   Mean   :10.200   Mean   :57.54   Mean   : 9.608   Mean   : 0.7417
E    :1   3rd Qu.:431.2   3rd Qu.:23847   3rd Qu.:10.875   3rd Qu.:64.55   3rd Qu.:12.325   3rd Qu.: 3.4500
F    :1   Max.   :623.7   Max.   :25006   Max.   :23.600   Max.   :74.00   Max.   :13.900   Max.   : 5.2000
```

Neben den beiden Mittelwerten arithmetisches Mittel (Mean) und Median werden für jede Variable das 1. und 3. Quartil sowie der kleinste und größte Wert ausgegeben.

Während diese Kenngrößen bei quantitativen Variablen interpretierbar sind, ist ihre Angabe bei alphanumerischen Variablen wie z.B. Region nicht sinnvoll. Eine Unterdrückung der Ausgabe für diese Variable kann durch die Kennzeichnung [-1] hinter dem Namen des Datensatzes Regionen12 erreicht werden:

```
>summary(Regionen12[-1])
x1          x2          x3          x4          x5          x6
Min.  : 93.1   Min.  :19324   Min.  : 3.400   Min.  :44.30   Min.  : 6.100   Min.  :-3.1000
1st Qu.:200.8   1st Qu.:20684   1st Qu.: 7.075   1st Qu.:49.02   1st Qu.: 7.725   1st Qu.:-2.0750
Median :309.6   Median :23095   Median : 8.850   Median :55.95   Median : 8.650   Median : 0.6500
 Mean   :326.6   Mean   :22532   Mean   :10.200   Mean   :57.54   Mean   : 9.608   Mean   : 0.7417
 3rd Qu.:431.2   3rd Qu.:23847   3rd Qu.:10.875   3rd Qu.:64.55   3rd Qu.:12.325   3rd Qu.: 3.4500
 Max.   :623.7   Max.   :25006   Max.   :23.600   Max.   :74.00   Max.   :13.900   Max.   : 5.2000
```

Allgemein wird durch die Angabe [-j] hinter dem Namen eines data.frame-Objekts bewirkt, dass die j-te Spalte bei der Ausführung der Funktion unberücksichtigt bleibt.

Da in der Multivariaten Statistik Berechnungen häufig vorteilhaft mit Hilfe der Matrizenrechnung durchgeführt werden können, werden die Beobachtungswerte der quantitativen Variablen des Datensatzes Regionen12 einem Matrixobjekt zugewiesen:

```
> REG12 = as.matrix(Regionen12[-1]).
```

Weist man den Zeilen zusätzlich die Regionsnamen zu,

```
> row.names(REG12) = c("A","B","C","D","E","F","G","H","I","J","K","L")
```

oder

```
> row.names(REG12) = Regionen12$Region,
```

erhält man das Matrixobjekt mit den vollständigen Zeilen- und Spaltenlabels:

```
> REG12

     x1     x2    x3   x4    x5   x6
A 212.4 20116  9.8 53.0  8.4 -0.7
B   623.7   24966  3.4 73.1  6.1  3.4
C    93.1   19324 23.6 47.9 12.3 -1.9
D   236.8   23113  8.7 66.8  8.7  2.0
E   412.0   23076  8.9 46.9  8.0 -3.1
F   566.7   24516  6.1 44.3  8.6 -3.0
G   331.9   22187  7.4 57.6 10.3  4.7
H   111.4   20614 16.3 63.8 13.9  5.2
I   489.0   25006  5.7 49.4  6.7 -2.6
J   287.4   23136  8.8 59.4 12.4  1.7
K  166.2   20707 14.1 74.0 13.0  3.6
L   388.1   23624  9.6 54.3  6.9 -0.4
>class(REG12)
[1] "matrix"
>str(REG12)
num [1:12, 1:6] 212.4 623.7 93.1 236.8 412 ...
 - attr(*, "dimnames")=List of 2
  ..$ :chr [1:12] "A" "B" "C" "D" ...
  ..$ :chr [1:6] "x1" "x2" "x3" "x4" ....
```

Die Anwendung der Funktion summaryauf die Matrix REG12 und den reduzierten Datensatz Regionen12 führt zu einem identischen Ergebnis.

Deskriptive Kenngrößen wie z.B. das arithmetische Mittel (mean), die Varianz (var) und die Standardabweichung (sd) lassen sich für einzelne Variablen wie z.B.

```
> ED = REG12[,1]
```

```
> ED
  A     B     C     D     E     F     G     H     I     J     K     L
212.4 623.7  93.1 236.8 412.0 566.7 331.9 111.4 489.0 287.4 166.2 388.1

>class(ED)
[1] "numeric"
>str(ED)
 Named num [1:12] 212.4 623.7 93.1 236.8 412 ...
 - attr(*, "names")= chr [1:12] "A" "B" "C" "D" ...

> BIP = REG12[,2]
> BIP
 A     B     C     D     E     F     G     H  I     J     K     L
20116 24966 19324 23113 23076 24516 22187 20614 25006 23136 20707 23624

>class(BIP)
[1] "numeric"
>str(BIP)
Named num [1:12] 20116 24966 19324 23113 23076 ...
 - attr(*, "names")= chr [1:12] "A" "B" "C" "D" ...
```

berechnen:

```
>ED.m = mean(ED)
>ED.m
[1] 326.5583
>ED.v = var(ED)
>ED.v
[1] 30236.13
> ED.s = sd(ED)
> ED.s
[1] 173.8854

>BIP.m = mean(BIP)
>BIP.m
[1] 22532.08
>BIP.v = var(BIP)
>BIP.v
[1] 3766251
> BIP.s = sd(BIP)
> BIP.s
[1] 1940.683.
```

Hierbei ist zu beachten, dass die Funktionen var und sd von dem Faktor 1/(n-1) Gebrauch machen, wodurch im Falle von Stichproben eine erwartungstreue Schätzung der Kenngrößen der Grundgesamtheit gesichert ist.

Die Mittelwerte (Summen) aller quantitativen Variablen des Datensatzes Regionen12 können mit der Funktion colMeans (colSums) berechnet werden:

```
> REG12.m = colMeans(REG12)

> REG12.m
x1        x2        x3        x4        x5        x6
3.265583e+02 2.253208e+04 1.020000e+01 5.754167e+01 9.608333e+00 7.416667e-01.
```

Eine übersichtlichere Ausgabe der Variablenmittelwerte als Dezimalzahlen ohne Zehnerpotenzen erhält man unter Verwendung der round-Funktion:

```
> REG12.m = round(colMeans(REG12), 3)
> REG12.m
   x1       x2       x3       x4       x5       x6
 326.558 22532.083   10.200   57.542    9.608    0.742.
```

Dasselbe Ergebnis lässt sich mit der apply-Funktion erzielen:

```
> REG12.m = round(apply(REG12, 2, mean), 3)
> REG12.m
x1       x2       x3       x4       x5       x6
 326.558 22532.083   10.200   57.542    9.608    0.742.
```

Das Argument margin=2(margin=1) weist R an, die Mittelwerte der Spalten (Zeilen) zu berechnen.

Mit der Funktion apply lassen sich durch eine entsprechende Spezifikation des Arguments FUN andere Kenngrößen wie z.B. die Varianz und Standardabweichung bestimmen:

```
> REG12.var = round(apply(REG12, 2, var), 3)
> REG12.var
x1              x2              x3          x4          x5          x6
 30236.130     3766250.811     29.958    101.841      7.221      9.415

> REG12.sd = round(apply(REG12, 2, sd), 3)
> REG12.sd
x1       x2       x3       x4       x5       x6
 173.885 1940.683    5.473   10.092    2.687    3.068.
```

Übungsaufgaben

2.1.1 Transformieren Sie das data.frame-Objekt documenta in ein gleichnamiges Matrix-Objekt und beschreiben Sie die Variablen v103, v104, v105 und v106 mit der summary-Funktion! Unter welchen Annahmen sind die verschiedenen Lagemaße interpretierbar?

2.1.2 Berechnen Sie separat das arithmetische Mittel, die Varianz und Standardabweichung der in Aufg. 1.4.7 definierten Variablen v103 und v104 und interpretieren Sie sie! Welche Annahme über die Skaleneigenschaft der Daten wird hierbei im Allgemeinen unterstellt?

2.1.3 Definieren Sie ein Matrix-Objekt documenta4 der Variablen v103, v104, v105 und v106 und berechnen Sie einen Mittelwertvektor dieser Variablen unter Verwendung zweier verschiedener R-Funktionen!

2.1.4 Berechnen Sie die Varianzen und Standardabweichungen der Variablen v103, v104, v105 und v106 unter Verwendung des Matrix-Objekts documenta4!

2.2 Bivariate Datenanalyse

Zum Zwecke einer Beschreibung von Zusammenhängen zwischen zwei Merkmalen werden in der bivariaten Analyse Kenngrößen wie die Kovarianz,dieKorrelation, Koeffizienten einer linearen Einfachregression und das Bestimmtheitsmaß berechnet. Da zur Beurteilung der Eignung der Faktorenanalyse eines multivariaten Datensatzes jedoch die Signifikanz der Korrelationskoeffizienten von Bedeutung ist, wird die Deskription hier durch einen Test dieser Kenngröße ergänzt.

Die Kovarianz ist eine Maßzahl der Verbundstreuung zwischen zweie Variablen. Berechnet werden sollhierdie Kovarianz zwischen der Einwohnerdichte (ED) und dem BIP pro Kopf (BIP). Hierzu wird die in der Funktion covvoreingestellteBerechnungsmethode (method="pearson") verwendet:

```
>cov.ED_BIP = cov(ED,BIP)
>cov.ED_BIP
[1] 306152.1.
```

Der Wert von 306152,1 gibt die durchschnittliche Summe der Kreuzprodukte $(x_i - \bar{x})(y_i - \bar{y})$ an, die analog zur Varianz mit dem Faktor 1/(n-1) gebildet wird. Anhand der Kovarianz kann die Richtung, jedoch nicht die Stärke des Zusammenhangs zwischen zwei Variablen erkannt werden. Hier weist die Kovarianz einen positiven Zusammenhang zwischen der Einwohnerdichte und dem Bruttoinlandsprodukt pro Kopf aus.

Um die Maßeinheiten auszuschalten, in denen die Variablen gemessen werden (Einwohner je qkm bei ED und 1000 Euro bei BIP), dividiert man die Kovarianz durch das Produkt ihrer Standardabweichungen. Das sich ergebene normierte Maß liegt zwischen -1 und 1 und heißt Pearsonscher Korrelationskoeffizient:

```
>cor.ED_BIP = cov.ED_BIP/(ED.s*BIP.s)
>cor.ED_BIP
[1] 0.9072343.
```

Man erhält dasselbe Ergebnis mit cor-Funktion (Voreinstellung: method="pearson"):

```
>cor.ED_BIP = cor(ED,BIP)
>cor.ED_BIP
[1] 0.9072343.
```

Der Korrelationskoeffizient von 0,907 gibt einen starken linearen Zusammenhang zwischen der Einwohnerdichte und dem Bruttoinlandsprodukt pro Kopf wieder. Auf den Rangkorrelationskoeffizienten nach Spearman (method="spearman") und Kendalls Tau (method="kendall"), die nur eine Ordinalskala voraussetzen, soll hier nicht eingegangen werden.

Ob eine Korrelation substanziell ist oder nicht lässt sich mit einem Signifikanztest beantworten. Die aus einer Stichprobe berechneten Korrelationskoeffizienten können aufgrund des Stichprobenfehlers ungleich 0 sein auch wenn die Variablen in der Grundgesamtheit nicht korreliert sind. Der t-Test über den Korrelationskoeffizienten ρder Grundgesamtheit unterstellt als Nullhypothese H_0: $\rho = 0$. Im zweiseitigen Fall wird sie abgelehnt, wenn die Wahrscheinlichkeit der beobachteten Abweichung des empirischen Korrelationskoeffizienten r von 0 kleiner ist als das vorgegebene Signifikanzniveau α. Diese Wahrscheinlichkeit heißt p-Wert.

Das Programm R gibt bei dem t-Test über den Korrelationskoeffizienten der VariablenED und BIP einen p-Wert von 0,00004622 aus, der jedes übliche Signifikanzniveau α (α=0,10, α=0,05, α=0,01) unterschreitet,

```
>cortest.ED_BIP = cor.test(ED,BIP)
>cortest.ED_BIP

Pearson's product-moment correlation

data:  ED and BIP
t = 6.8206, df = 10, p-value = 4.622e-05
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.6954040 0.9740065
sample estimates:
cor
0.9072343,
```

so dass der berechnete Korrelationskoeffizient von 0,907 hochsignifikant ist.

Alternativ kann der Test wie manuell üblich durch einen Vergleich des t-Werts mit dem kritischen Wert durchgeführt werden. Hierzu muss bei einem zweiseitigen Test der α-Wert des Signifikanzniveaus,

```
>alpha = 0.01
>alpha
[1] 0.01,
```

als Argument der Quantilsfunktion der t-Verteilung qt halbiert werden:

```
> t10.01 = qt(alpha/2,df=10, lower.tail=FALSE)
> t10.01
[1] 3.169273.
```

Das zweite Argument der qt-Funktion gibt die Zahl der Freiheitsgrade df (12 Beobachtungen minus 2 geschätzte Parameter) an und das dritte Argument weist R an, das obere Quantil zu berechnen. Da der empirische t-Wert von 6,8206 größer als der kritische Wert von 3,169273 ist, führt auch dieser Vergleich auf dem 1%-Signifikanzniveau zu einer Ablehnung der Nullhypothese einer fehlenden Korrelation zwischen ED und BIP.

Mit der Funktion cor.testwird außerdem das zu dem Punktschätzer von 0,907 das 95%-Konfidenzintervall für den Korrelationskoeffizienten der Grundgesamtheit berechnet. Die berechneten Größen sind zusammen mit Informationen zu dem Test als Attribute htest-Objekts cortest.ED_BIP,

```
>class(cortest.ED_BIP)
[1] "htest"
```

verfügbar wie z.B.

```
>cortest.ED_BIP$statistic
t
6.820619
>cortest.ED_BIP$p.value
[1] 4.621999e-05
> cortest.ED_BIP$conf.int
[1] 0.6954040 0.9740065
attr(,"conf.level")
[1] 0.95.
```

Alle Attribute dieses Objekts lassen sich mit der str-Funktion anzeigen.

Wie lässt sich der starke Zusammenhang zwischen dem BIP pro Kopf und der Einwohnerdichte nutzen, um aus bekannten Werten einer der beiden Variablen eine Aussage über die erwartete Größenordnung der anderen Variablen zu machen? Verwendet man z.B. die Einwohnerdichte als Indikator der Agglomerationskraft, könnte aufgrund von Skalenerträgen (economies of scale) oder externen Effekten ein positiver Einfluss dieser Variablen auf das BIP pro Kopf begründet werden. In diesem Fall könnte der Zusammenhang durch eine Regression des BIP pro Kopf (BIP) auf die Einwohnerdichte (ED) modelliert werden.

Zur Schätzung eines linearen Regressionsmodells mit der Kleinst-Quadrate-Methode ist in R die Funktion lm verfügbar. Mit dem Argument formular erfolgt eine symbolische Wiedergabe der Modellgleichung, in dem die abhängige Variable BIP durch eine Tilde (~) von der unabhängigen Variablen ED getrennt wird:

```
> RegModel.BIP_ED = lm(BIP~ED)
```

Das Ergebnis der Kleinst-Quadrate-Schätzung wird dem lm-Objekt RegModel.BIP_ED zugeordnet.

```
>class(RegModel.BIP_ED)
[1] "lm".
```

Ein kurzer Regressionsoutput wird durch Eingabe des ObjektnamesRegModel.BIP_ED und anschließender Betätigung der Return-Tast abgerufen:

```
>RegModel.BIP_ED

Call:
lm(formula = BIP ~ ED)

Coefficients:
(Intercept)        ED
  19225.56     10.13.
```

Hier werden nebender symbolischen Modellgleichung allein die Kleinst-Quadrate-Schätzer der Regressionskoeffizienten ausgegeben. Das absolute Glied von 19225,559, das per Voreinstellung in der lm-Funktion mitberücksichtigt wird, ist hier nur eine technische Konstante, die die Lage der Regressionsgerade festlegt.[2] Der Regressionskoeffizient der Variablen ED (= Steigungsmaß) gibt an, um wie viel das BIP pro Kopf im Mittel zunimmt, wenn die Einwohnerdichte um eine Einheit steigt.

Bei Verwendung der Funktion summary wir ein langer Regressionsoutput mit Signifikanztests und diversen Kenngrößen ausgegeben von denen an dieser Stelle allein das Bestimmheitsmaß (Determinationskoeffizient) R^2 von Interesse ist. Der Determinationskoeffizient ist durch das Verhältnis der durch die Regression erklärte Varianz des BIP pro Kopf (= Varianz der Regressionswerte), BIPf.v, und der bereits berechneten Varianz der abhängigen Variablen BIP definiert. Mit den Regressionswerten BIPf, die unter dem Attributnamen fitted.valuesimlm-Objekts RegModel.BIP_ED verfügbar sind,

```
>BIPf = RegModel.BIP_ED$fitted.values
>BIPf
     1      2      3      4      5      6      7      8      9     10     11     12
21376.19 25540.75 20168.23 21623.25 23397.21 24963.61 22586.17 20353.53 24176.87 22135.59 20908.40
23155.22,
```

ist die erklärte Varianz gleich

```
>BIPf.v = var(BIPf)
> BIPf.v
[1] 3099904.
```

Da die gesamte Varianz der abhängigen Variablen BIP,

[2] Eine Interpretation der Regressionsgleichung ist ohne weitere Annahmen nur im Stützbereich, d.h. innerhalb des beobachteten Wertebereichs der unabhängigen Variablen zulässig.

```
> BIP.v
[1] 3766251,
```

beträgt, nimmt das Bestimmtheitsmaß einen Wert von

```
> R2 = BIPf.v/BIP.v
> R2
[1] 0.8230741
```

an. Der Determinationskoeffizient ist nicht als Attribut des lm-Objekts RegModel.BIP_ED verfügbar, jedoch aus dem erweiterten Regressionsoutputs abrufbar:

```
> R2 = summary(RegModel.BIP_ED)$r.squared
> R2
[1] 0.8230741.
```

82,3% der Varianz des BIP pro Kopf lässt sich somit auf die Veränderlichkeit der Einwohnerdichte zurückführen.

Durch die Regression des BIP pro Kopf auf die Einwohnerdichte ist eine Erklärung des starken Zusammenhangs zwischen diesen beiden Variablen gegeben. Wenn z.B. aufgrund einer hohen Produktivität und dem damit verbundenen höheren Lohnniveau Arbeitskräfte in die dadurch gekennzeichneten Regionen migrieren, könnte die Umkehrregression von Relevanz sein. In der Faktorenanalyse wird von einem weiteren Erklärungsmuster der beobachteten Zusammenhänge zwischen einer Reihe von Variablen ausgegangen.

Übungsaufgaben

2.2.1 Berechnen Sie die Kovarianzen zwischen den Variablen v103, v104 und v105 des Datensatzes Documenta und interpretieren Sie sie!

2.2.2 Bestimmen Sie den Pearsonschen Korrelationskoeffizienten bei den Variablen v103, v104 und v105 unter Verwendung

- der Kovarianzen und Standardabweichungen,
- der Funktion cor!

Interpretieren Sie die berechneten Werte!

2.2.3 Welche Werte nehmen der Rangkorrelationskoeffizient nach Spearman und Kendalls Tau bei den Variablen v103, v104 und v105 an?

2.2.4 Testen Sie den Pearsonschen Korrelationskoeffizienten zwischen den Variablen v103 und v104 zweiseitig auf Signifikanz (α=0,05) unter Verwendung

- des p-Werts,
- des kritischen Werts!

2.2.5 Welche Informationen hinsichtlich der Signifikanz des Korrelationskoeffizienten der Variablen v103 und v104 gibt das 95%-Konfidenzintervall wieder?

2.2.6 Regressieren Sie die Variable v104 auf die Variable v103 und interpretieren Sie den kurzen Regressionsoutput!

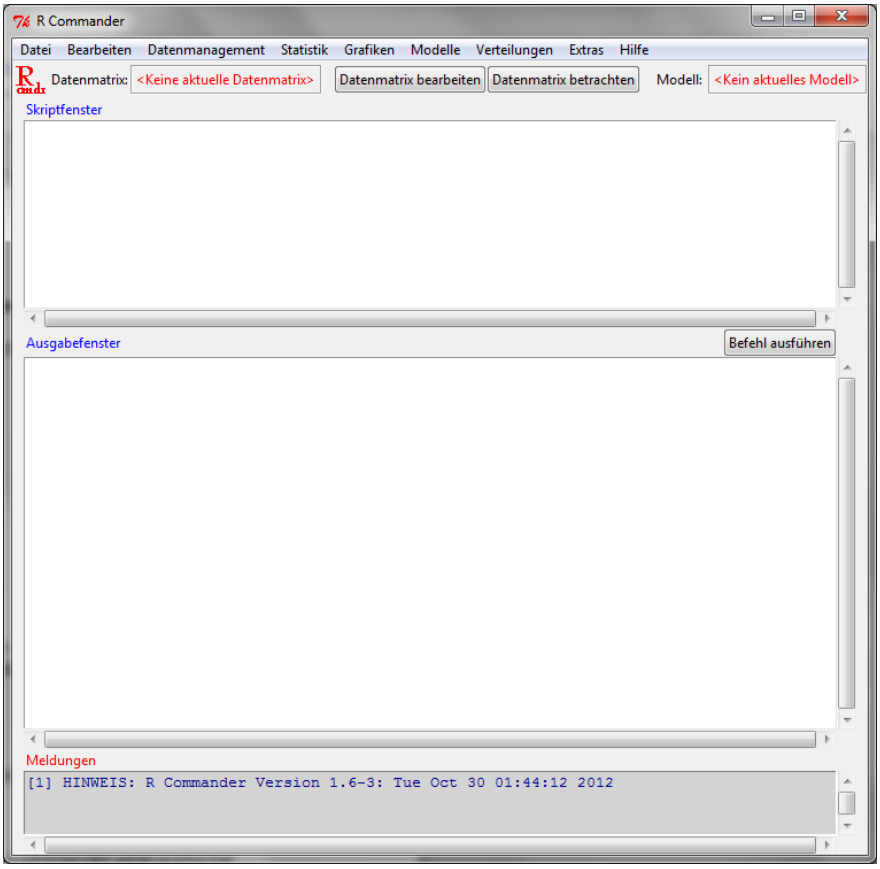
2.2.7 Bestimmen Sie den Determinationskoeffizienten unter Verwendung der in Aufg. 2.2.2 ermittelten Ergebnisse undinterpretieren Sie ihn!

2.3 Statistische Auswertung im R-Commander

Der R-Commander bietet eine einfache grafische Benutzeroberfläche, die mit dem Befehl

```
> library(Rcmdr)
```

aufgerufen werden kann:



Nach Schließen des R-Commanders in einer Sitzung kann er mit

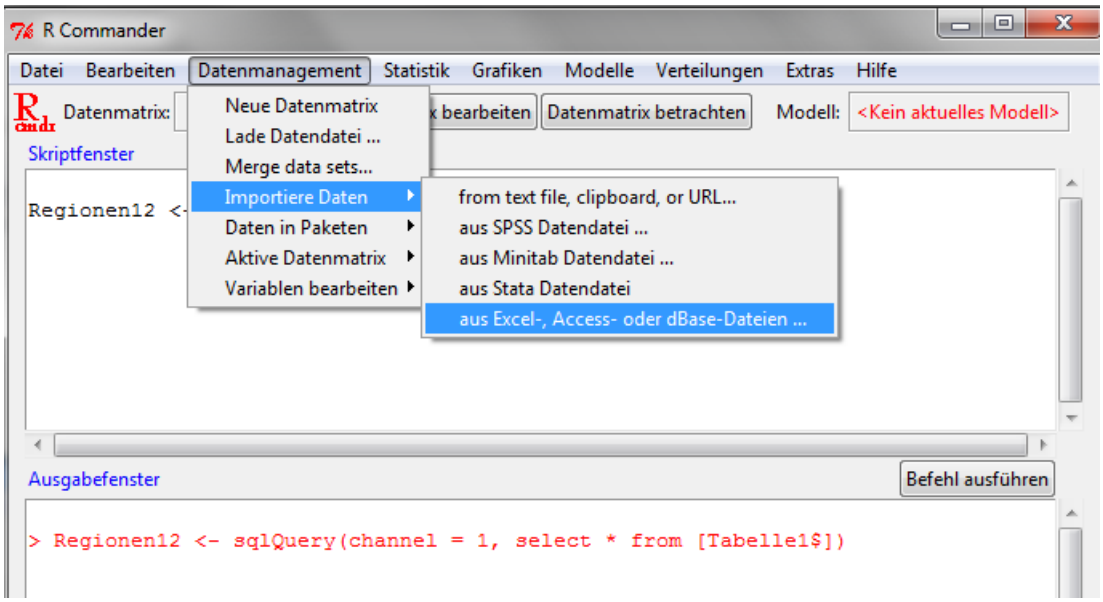
```
> Commander()
```

wieder geöffnet werden.

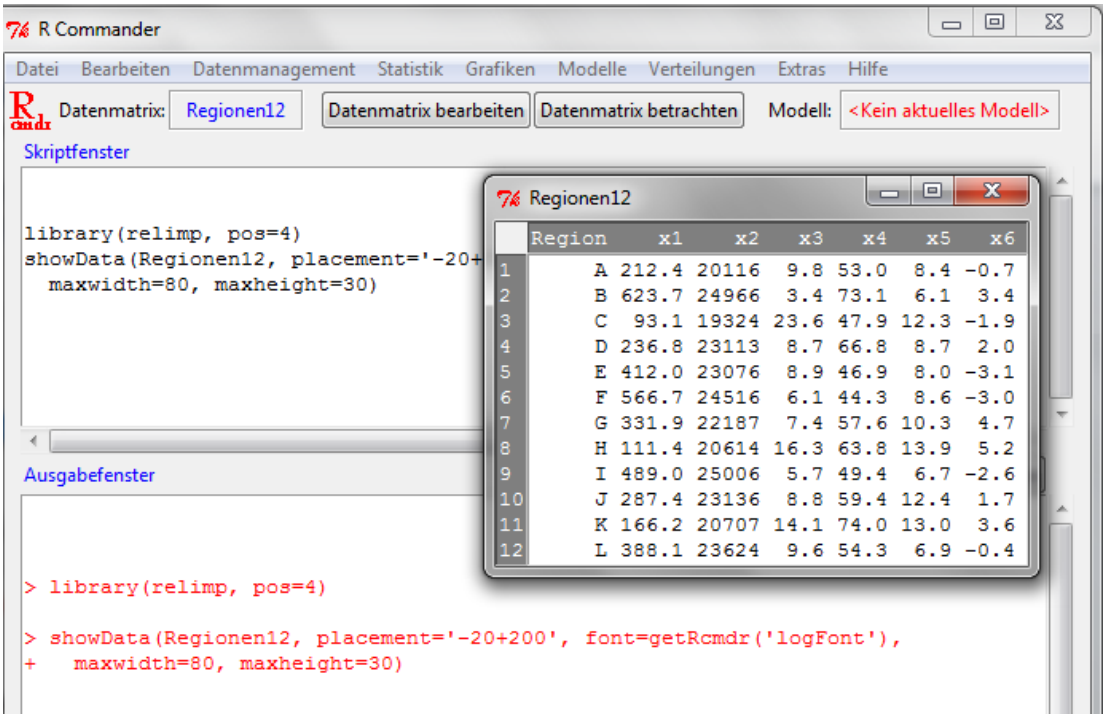
Der R-Commander erleichtert das Datenmanagement und Erstellen von Skripten, bietetjedoch nur eine eingeschränkte Funktionalität bei der Erzeugung von Grafiken und Durchführung von Datenanalysen über ein Menü.

Der Import der Excel-Datei Regionen12.xls erfolgt über die Menüpunkte

- Datenmanagement
  - Importiere Daten
    - aus Excel-, Access- oder dBase-Dateien
      - Name der Datenmatrix angeben: Regionen12.xlsx:

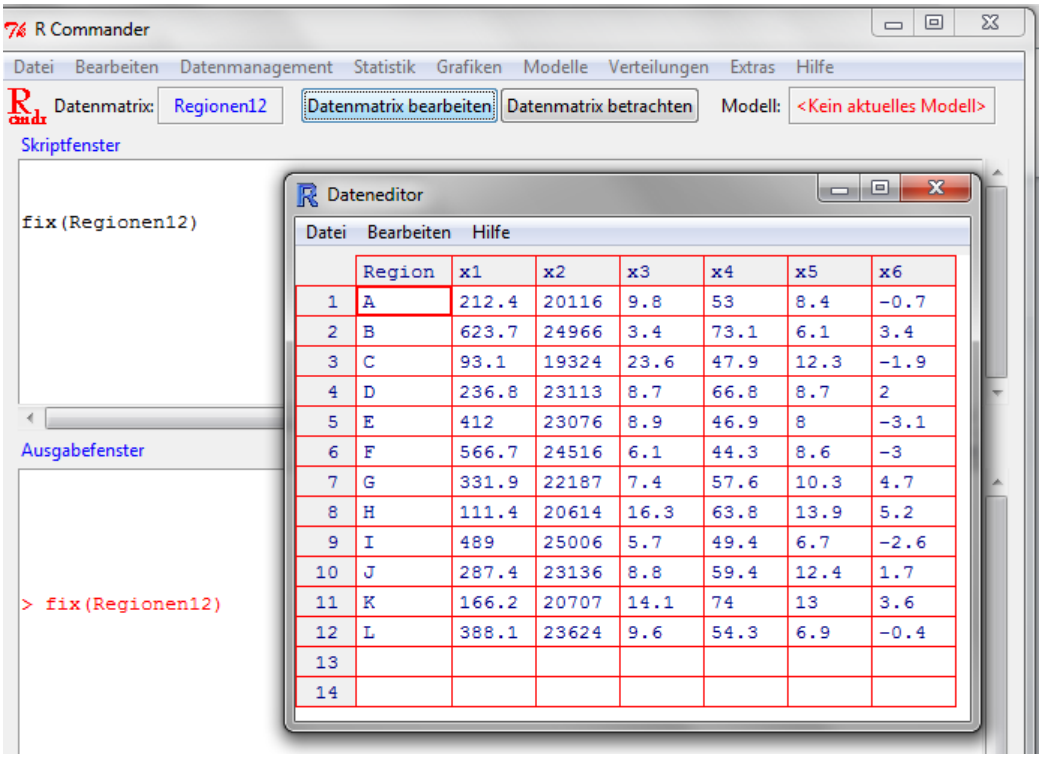


In dem Feld **Datenmatrix** wird **Regionen12** als aktueller Datensatz angezeigt.[3] Er wird nach Anklicken der Schaltfläche **Datenmatrix betrachten** in einem Fenster wiedergegeben:



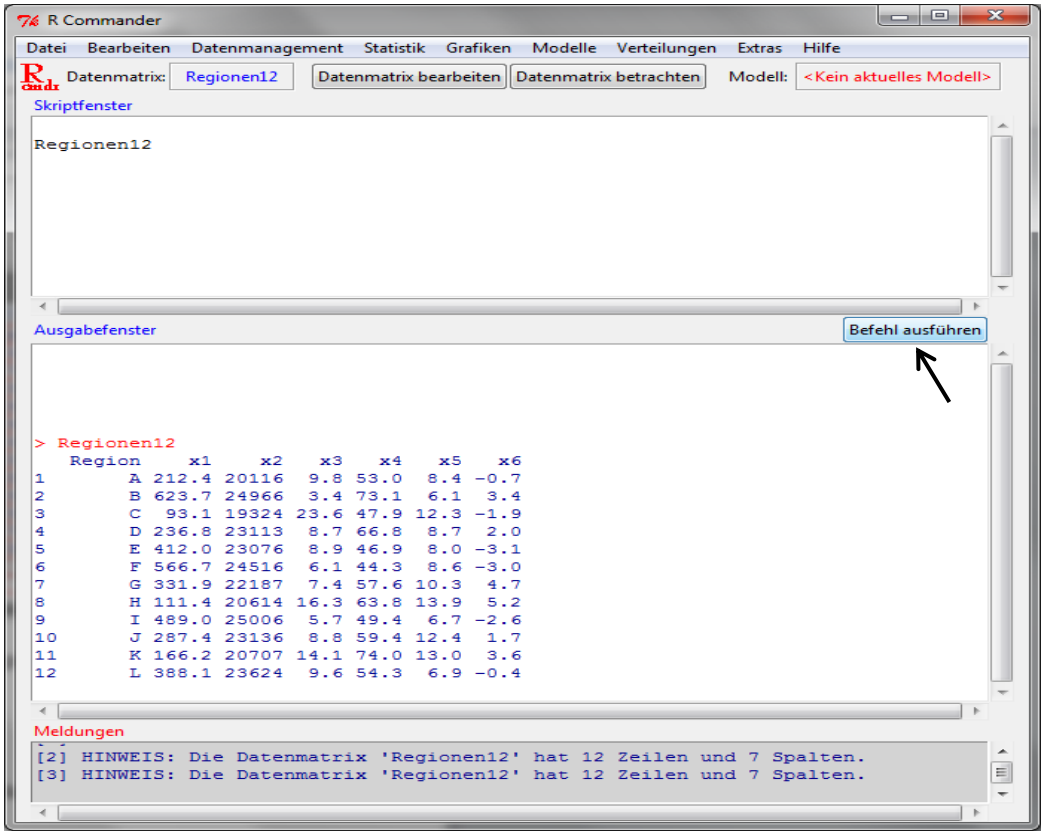
[3] **Regionen12** ist entgegen der Bezeichnung jedoch kein **matrix**-, sondern ein **data.frame**-Objekt.

Nach Anklicken der Schaltfläche Datenmatrix bearbeiten wird der Datensatz Regionen im Dateneditor angezeigt, wo er bei Bedarf verändert werden könnte:



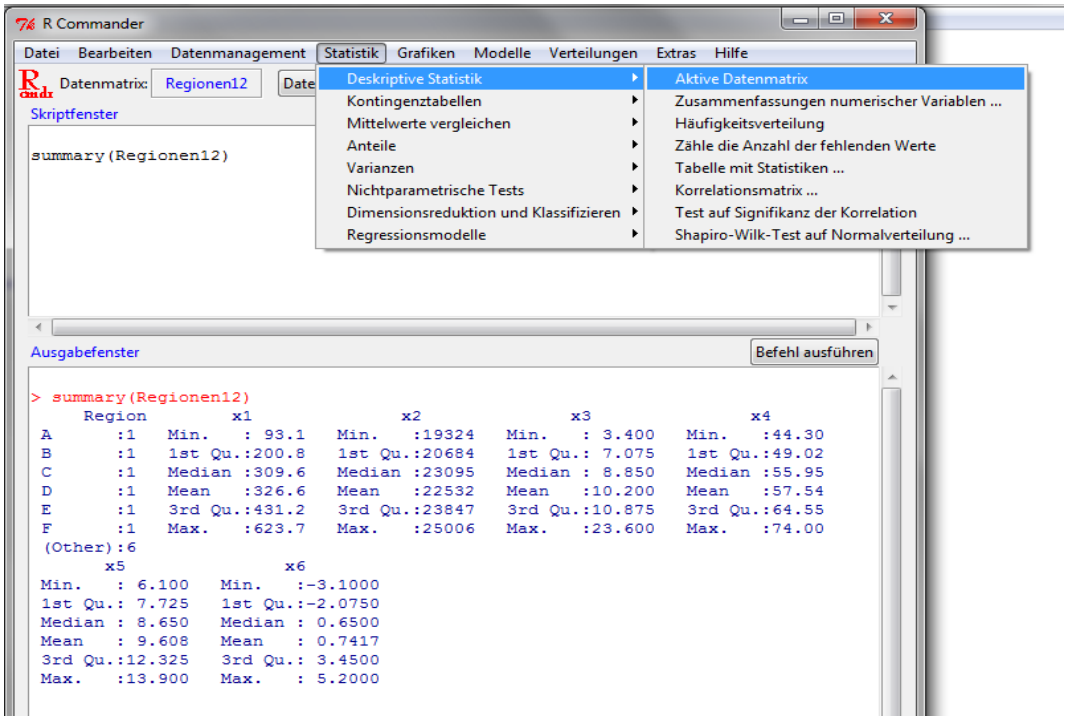
	Region	x1	x2	x3	x4	x5	x6
1	A	212.4	20116	9.8	53	8.4	-0.7
2	B	623.7	24966	3.4	73.1	6.1	3.4
3	C	93.1	19324	23.6	47.9	12.3	-1.9
4	D	236.8	23113	8.7	66.8	8.7	2
5	E	412	23076	8.9	46.9	8	-3.1
6	F	566.7	24516	6.1	44.3	8.6	-3
7	G	331.9	22187	7.4	57.6	10.3	4.7
8	H	111.4	20614	16.3	63.8	13.9	5.2
9	I	489	25006	5.7	49.4	6.7	-2.6
10	J	287.4	23136	8.8	59.4	12.4	1.7
11	K	166.2	20707	14.1	74	13	3.6
12	L	388.1	23624	9.6	54.3	6.9	-0.4
13							
14							

Der Datensatz lässt sich durch Eingabe des Namens Regionen12 und Anklicken der Schaltfläche Befehl ausführen wie in der R-Konsole im R-Commander auflisten:



Eine zusammenfassende Beschreibung der Variablen des aktiven Datensatzes**Regionen12** durch verschiedene Lagemaße erhält man durch die Wahl der Menüpunkte

- Statistik
  - Deskriptive Statistik
    - Aktive Datenmatrix:

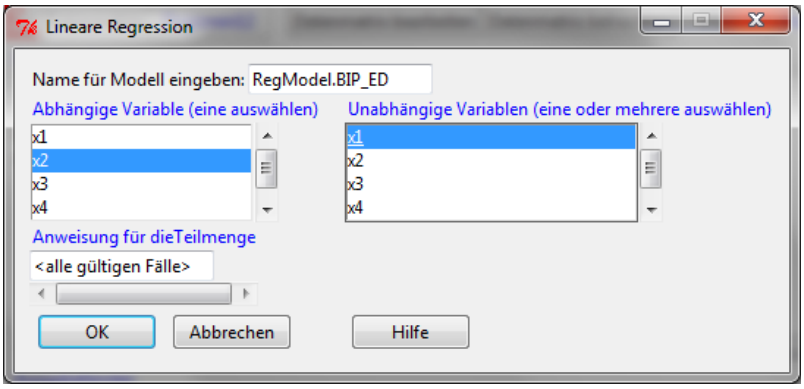


Sie erfolgt per Voreinstellung auch für die qualitative Variable **Region**, wodurch jedoch keine verwertbaren Informationen bereitgestellt werden. Bei Wahl des Untermenü-punkts **Zusammenfassung numerischer Variablen**kann die Ausgabe auf eine oder mehrere quantitative Variablen beschränkt werden.

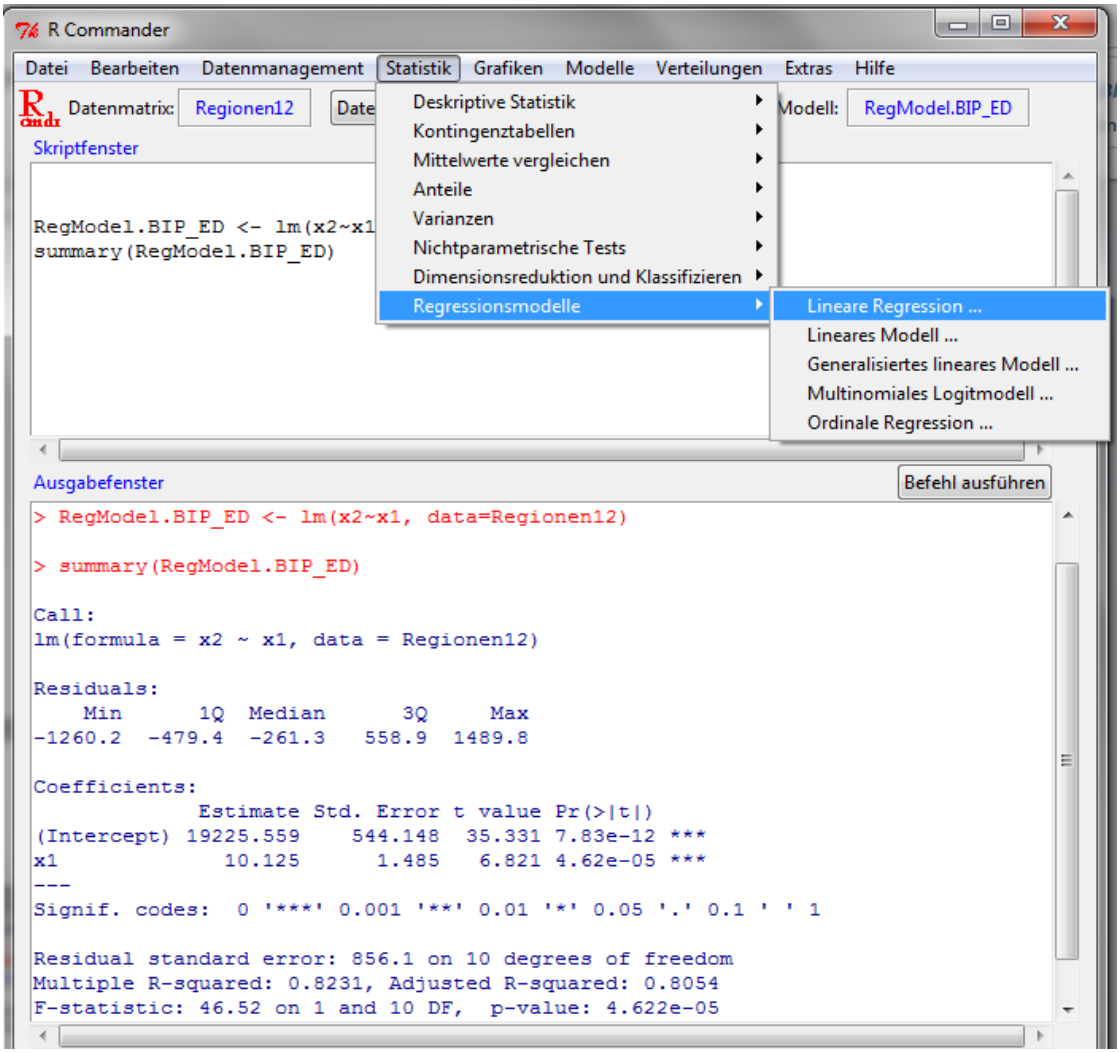
Die bei der bivariaten Analyse durchgeführte Regression des BIP pro Kopf (**x2**) auf die Einwohnerdichte (**x1**) lässt sich ebenfalls über das Menü im R-Commander bewerkstelligen:

- Statistik
  - Regressionsmodelle
    - Lineare Regression.

Hierzu sind in einem sich öffnenden Fenster der Name für das Modell anzugeben und die abhängige und eine oder mehrere unabhängigen Variablen zu benennen:



Ausgegeben wird im R-Commander das komplette Ergebnis der Kleinst-Quadrate-Schätzung, das in der R-Konsole durch die Funktion `summary` bereitgestellt wird.



Wie bereits gesehen lassen sich Befehle auch direkt ohne Verwendung des Menüs im Scriptfenstereingeben und durch Anklicken der entsprechenden Schaltfläche ausführen. Außerdem ist es ohne weiteres möglich, die aus dem Menü im Skriptfenster erzeugten Befehle nachträglich zu ändern.

In dem Menü Statistiksind auch Prozeduren zur multivariaten Analyse wie die Faktorenanalyse, Varianzanalyse und Clusteranalyse verfügbar. Jedoch ist die Faktorenanalyse in einer Variante implementiert, die in der Regel nicht in der Anwendung eingesetzt wird. Außerdem lässt sich durch zusätzliche Anwendung der Matrizenrechnung in R die Methodik der Verfahren anschaulich nachzuvollziehen. Nicht im Menü enthalten sind Prozeduren zur Diskriminanzanalyse.

Übungsaufgaben

2.3.1Öffnen Sie den R-Commander und importieren Sie den Datensatz Documenta! Überprüfen Sie, ob im Feld Datenmatrix der Name des importierten Datensatzes angezeigt wird!

2.3.2 Geben Sie drei unterschiedliche Darstellungen des Datensatzes Documenta im R-Commander wieder!

2.3.3 Bestimmen Sie das arithmetische Mittel, die drei Quartile sowie den kleinsten und größten Wert der Variablen v103, v104, v105 und v106 unter Verwendung des R-Menüs!

2.3.4 Schätzen Sie das lineare Regressionsmodell mit der abhängigen Variablen v104 und der unabhängigen Variablen v103 unter Verwendung des R-Menüs! Interpretieren Sie die Regressionskoeffizienten und das Bestimmtheitsmaß!

3. Faktorenanalyse
3.1 Bestimmung und Beurteilung der Korrelationsmatrix

Eine Faktorenanalyse setzt voraus, dass die Variablen eines Gegenstandsbereichs substanziell korreliert sind. Nur dann kann es sinnvoll sein, gemeinsame Faktoren, die hinter den Variablen stehen, aus der Korrelationsmatrix zu extrahieren. Die gemessenen Korrelationen zwischen den Variablen müssen in diesen Fall signifikant von null abweichen. Dies kann von Signifikanztests über einzelne Korrelationskoeffizienten beurteilt werden. Ein globaler Test (Omnibus-Test) über die Eignung einer Faktorenanalyse bei einem gegebenen Datensatz ist der Bartlett-Sphärentest.

Die Korrelationsmatrix **R** kann nach Standardisierung der nxm-Beobachtungsmatrix **X** direkt unter Verwendung des Matrizenkalküls bestimmt werden. Mit der standardisierten Beobachtungsmatrix **Z** erhält man **R** aus

(3.1) $\mathbf{R} = \frac{1}{n-1}\mathbf{Z}'\mathbf{Z}$,

wobei das Apostroph (‘) die Transponierte kennzeichnet. In R lässt sich die Standardisierung

(3.2) $z_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j}$

kompakt mit dem Befehl `scale` bewerkstelligen. Für die Datenmatrix `REG12` des 12-Regionen-Beispiels erhält man die standardisierte Datenmatrix `ZREG12`:

```
> ZREG12 = scale(REG12)
> ZREG12
x1          x2          x3          x4          x5          x6
A -0.65651480 -1.2449654 -0.07308063 -0.450043283 -0.4496692 -0.4698356
B  1.70883627  1.2541546 -1.24237066  1.541707907 -1.3055911  0.8663443
C -1.34259888 -1.6530692  2.44820101 -0.955412988  1.0016768 -0.8609127
D -0.51619249  0.2993362 -0.27405235  0.917427683 -0.3380272  0.4100878
E  0.49136771  0.2802707 -0.23751204 -1.054505087 -0.5985251 -1.2519898
F  1.38103415  1.0222775 -0.74907643 -1.312144544 -0.3752412 -1.2194000
G  0.03071947 -0.1778154 -0.51156439  0.005780372  0.2573968  1.2900112
H -1.23735714 -0.9883547  1.11447956  0.620151386  1.5971008  1.4529599
I  0.93418811  1.2747659 -0.82215705 -0.806774839 -1.0823071 -1.0890410
J -0.22519622  0.3111877 -0.25578219  0.184146151  1.0388908  0.3123185
K -0.92220704 -0.9404335  0.71253611  1.630890796  1.2621748  0.9315238
L  0.35392085  0.5626455 -0.10962094 -0.321223554 -1.0078791 -0.3720664

attr(,"scaled:center")
x1        x2        x3        x4        x5        x6
3.265583e+02 2.253208e+04 1.020000e+01 5.754167e+01 9.608333e+00 7.416667e-01
```

```
attr(,"scaled:scale")
      x1        x2        x3        x4        x5        x6
173.885393 1940.683078   5.473407  10.091622   2.687161   3.068449.
```

Bei der Ausgabe der standardisierten Datenmatrix in der R-Konsole werden die Mittelwerte und Standardabweichung der Variablen als Attribute des matrix-Objekts ZREG12 mit angegeben.

Unter Verwendung der Anzahl der Regionen (Beobachtungen),

```
> n = dim(ZREG12)[1] oder   > n = nrow(ZREG12)
> n
[1] 12,
```

ergibt sich aus (3.1) die Korrelationsmatrix **R**:

```
> R = (1/(n-1))*t(ZREG12)%*%ZREG12
> R
  x1        x2        x3        x4        x5        x6
x1  1.0000000  0.90723433 -0.83389081 -0.16101263 -0.7865920 -0.30934486
x2  0.9072343  1.00000000 -0.84531953 -0.05351755 -0.7109732 -0.22018038
x3 -0.8338908 -0.84531953  1.00000000 -0.06708459  0.7186602  0.03891881
x4 -0.1610126 -0.05351755 -0.06708459  1.00000000  0.2258691  0.83177028
x5 -0.7865920 -0.71097315  0.71866021  0.22586911  1.0000000  0.45387020
x6 -0.3093449 -0.22018038  0.03891881  0.83177028  0.4538702  1.00000000.
```

Die Hauptdiagonale von **R** enthält die Eigenkorrelationen der Variablen, die stets gleich 1 sind. Aufgrund der Symmetrie spiegeln sich die Korrelationen zwischen den Variablen an der Hauptdiagonale. So findet man die bereits berechnete Korrelation von 0,907 zwischen der Einwohnerdichte (x1) und dem BIP pro Kopf (x2) im Schnittpunkt der 1. Zeile und 2. Spalte als auch der 2. Zeile und 1. Spalte.

Die Korrelationsmatrix **R** lässt sich ohne explizite Matrizenmultiplikation direkt mit der cor-Funktion bestimmen, indem das matrix-Objekt REG12 als Argument angegeben wird:

```
> R = cor(REG12).
```

Wie bereits bekannt, gibt das Quadrat des Korrelationskoeffizienten das Bestimmtheitsmaß (Determinationskoeffizient) einer linearen Regression einer der beiden Variablen auf die andere Variable an. Die Matrix der quadrierten Korrelationskoeffizienten(Determinationskoeffizienten) spiegelt damit wider, wie viel Prozent ihrer Varianzen die Variablen paarweise gemeinsam haben. Im 12-Regionen-Beispiel,

```
> D = R^2
```

```
> D
 x1  x2        x3    x4        x5        x6
x1 1.00000000 0.823074131 0.695373884 0.025925067 0.61872705 0.095694240
x2 0.82307413 1.000000000 0.714565108 0.002864128 0.50548282 0.048479398
x3 0.69537388 0.714565108 1.000000000 0.004500343 0.51647250 0.001514674
x4 0.02592507 0.002864128 0.004500343 1.000000000 0.05101685 0.691841804
x5 0.61872705 0.505482824 0.516472499 0.051016853 1.00000000 0.205998157
x6 0.09569424 0.048479398 0.001514674 0.691841804 0.20599816 1.000000000,
```

haben die Variablen ED (x1) und BIP (x2) z.B. einen gemeinsamen Anteil von 82,3%, was dem in der bivariaten Analyse berechneten R^2 entspricht.

Ein globaler Test (Omnibus-Test) über die Eignung einer Faktorenanalyse für einen gegebenen Datensatz ist der Bartlett-Sphärentest, der die Nullhypothese

H_0: Es gibt keine gemeinsamen Faktoren

prüft. Die Teststatistik

(3.3) $v = -[n-1-(2m+5)/6]\cdot \ln|\mathbf{R}|$

ist bei Normalverteilung der m Variablen χ^2-verteilt mit m(m-1)/2 Freiheitsgraden. Sie nimmt für den Datensatz Regionen12 den Wert

```
> m = dim(ZREG12)[2] oder   > m = ncol(ZREG12)
> m
[1] 6
> v = -(n-1-(2*m+5)/6)*log(det(R))
+ )
> v
[1] 48.88061
```

an. Da sie das 95%-Quantil $\chi^2_{15;0,95}$,

```
> vq95.df15 = qchisq(0.95,m*(m-1)/2)
> vq95.df15
[1] 24.99579,
```

übersteigt, wird die Nullhypothese fehlender gemeinsamer Faktoren bei einem Signifikanzniveau von 5% zurückgewiesen. Der p-Wert von

```
>p.v = 1 - pchisq(v,m*(m-1)/2)
>p.v
[1] 1.831945e-05
```

zeigt an, dass die Überschreitungswahrscheinlichkeit des ermittelten Wertes der Prüfgröße v praktisch gleich 0 ist. Der Bartlett-Sphärentest lässt daher eine Extraktion gemeinsamer Faktoren aus der Korrelationsmatrix R als erfolgreich erscheinen.

Mit dem R-Basispaket ist es nicht möglich, Signifikanztests über alle einzelnen Korrelationen abzurufen. Hierzu muss das R-Paket Hmisc geladen werden,

```
>library(Hmisc),
```

wonach auf die Funktion rcorr zurückgegriffen werden kann.[4] Der Aufruf der Funktionrcorrmit Angabe der Datenmatrix REG12bewirkt, dass zusätzlich zur Korrelationsmatrix eine Matrix der Signifikanzwerte (p-Werte) ausgegeben wird:

```
>rcorr(REG12)
 x1   x2   x3   x4   x5   x6
x1  1.00  0.91 -0.83 -0.16 -0.79 -0.31
x2  0.91  1.00 -0.85 -0.05 -0.71 -0.22
x3 -0.83 -0.85  1.00 -0.07  0.72  0.04
x4 -0.16 -0.05 -0.07  1.00  0.23  0.83
x5 -0.79 -0.71  0.72  0.23  1.00  0.45
x6 -0.31 -0.22  0.04  0.83  0.45  1.00

n= 12

P
     x1   x2   x3   x4   x5   x6
x1       0.0000 0.0007 0.6171 0.0024 0.3279
x2 0.0000       0.0005 0.8688 0.0095 0.4917
x3 0.0007 0.0005       0.8359 0.0085 0.9044
x4 0.6171 0.8688 0.8359       0.4803 0.0008
x5 0.0024 0.0095 0.0085 0.4803       0.1383
x6 0.3279 0.4917 0.9044 0.0008 0.1383
```

Die Matrix der Signifikanzwerte ist wie die Korrelationsmatrix symmetrisch. Man erkennt hieraus z.B. eine Signifikanz der Korrelation zwischen den Variablen ED (x1) und BIP (x2) auf allen gängigen Signifikanzniveaus αanhand des p-Wertes von 0.0000. Von den m(m-1)/2=15 Merkmalskorrelationen des Datensatzes Regionen12 sind 7 auf dem 5%- und 1%-Niveau signifikant. Damit bestätigen auch die individuellen Signifikanztests der Korrelationen, dass zwischen den Variablen des regionalen Datensatzes substanziell Gemeinsamkeiten existieren, die mit Hilfe einer Faktorenanalyse aufgedeckt werden könnten.

[4]Mit der im Basispaket enthaltene Funktion cor.test lässt sich nur ein Signifkanztest über einzelne Korrelationskoeffizienten unter Angabe der beiden Variablenvektoren durchführen. Eine Alternative zu der im R-Paket Hmisc enthaltenen Funktion rcorr ist die Funktionrcorr.adjust des Pakets Rcmdr, die zusätzlich adjustierte p-Werte angibt und auch nach Schließen des Commanders in der R-Konsole verfügbar bleibt.

Übungsaufgaben

3.1.1 Berechnen Sie die Korrelationsmatrix für die Variablen des Datensatzes Documenta unter Anwendung der Matrizenrechnung! Überprüfen Sie das Ergebnis mit anhand der cor-Funktion!

3.1.2 Bestimmen Sie die Matrix der Determinationskoeffizienten für das Documenta-Beispiel und interpretieren Sie die berechneten Werte für die Variablenpaare v103 und v104 sowie v106 und v111!

3.1.3 Beurteilen Sie anhand des Bartlett-Sphärentests die Eignung einer Faktorenanalyse für den Datensatz Documenta!

3.1.4 Ermitteln Sie die Signifikanzwerte (p-Werte) der Korrelationen zwischen den Variablen der zur Documenta erhobenen Daten!

a) Interpretieren Sie die Testergebnisse für die Variablenpaare v103 und v104 sowie v106 und v111!

b) Zu welcher Beurteilung der Eignung einer Faktorenanalyse gelangen Sie anhand der Matrix der p-Werte der Merkmalskorrelationen?

3.2 Eigenwerte und Eigenvektoren der Korrelationsmatrix

Die Analyse der Korrelationsmatrix hat ergeben, dass sich der 12-Regionen-Datensatz grundsätzlich für eine Faktorenanalyse eignet. Bei der Extraktion der gemeinsamen Faktoren aus den Merkmalskorrelationen stößt man auf ein Eigenwertproblem, das mit den Methoden der linearen Algebra gelöst werden kann. In der Hauptkomponentenanalyse werden dabei die Eigenwerte λ_k der Korrelationsmatrix R bestimmt, die die potenziellen Varianzen der gemeinsamen Faktoren spiegeln. Sie geben an, wie viel Prozent der gesamten Varianz der Merkmale auf den jeweils extrahierten Faktor zurückgeführt werden kann.

Die Eigenwerte λ_k der mxm-Korrelationsmatrix **R** ergeben sich als nicht-triviale Lösung der charakteristischen Gleichung

(3.4) $\left|\mathbf{R} - \lambda \cdot \mathbf{I}\right| = 0$,

d.h. als Wurzeln (Nullstellen) eines Polynoms m-ten Grades. I ist eine mxm-Einheitsmatrix. Bei sechs Variablen (m=6) im 12-Regionen-Beispiel ist insgesamt eine standardisierte Merkmalsvarianz von 6 (=m) zu erklären. Trägt man die Determinante $\left|\mathbf{R} - \lambda \cdot \mathbf{I}\right|$ in einem kartesischen Koordinatensystem gegen λ ab, ist die Abzisse in einem Bereich von 0 bis 6 zu betrachten:

```
>lambda = seq(from=0, to=6, by=0.1).
```

Um die Determinate $\left|\mathbf{R}-\lambda\cdot\mathbf{I}\right|$ in Abhängigkeit von λ im Intervall 0 bis 6 (=m) zu berechnen, ist vorab die Einheitsmatrix zu definieren, wozu der bereits ermittelte Wert von m (= 6) verwendet wird:

```
> I= diag(1,nrow=m,ncol=m).
```

Jetzt muss noch ein Vektor definiert werden, der die Werte der Determinante in Abhängigkeit der Abzissenwerte lambda aufnimmt:

```
>detR=rep(0, length(lambda)).
```

Initialisiert man den Index s der Werte des Vektors detR,

```
> s = 0,
```

dann erhält man die Determinantenwerte für jede Komponente l des Vektors lambda unter Verwendung einer for-Schleife:

```
>for (l in lambda) {
s=s+1;
detl=det(R-l*I);
detR[s] = detl;
}.
```

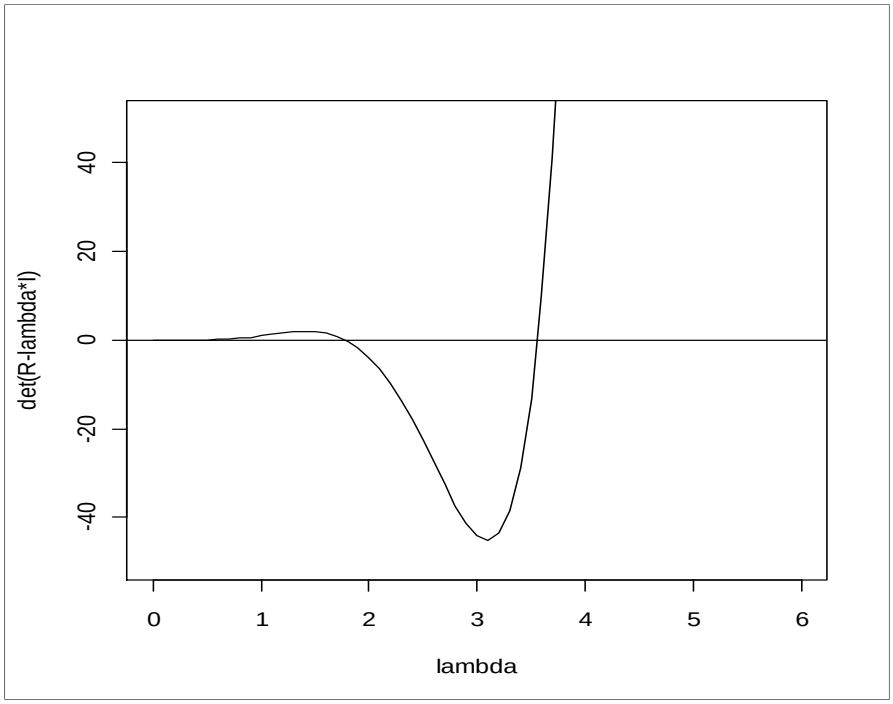
Die for-Schleife enthält mehrere Befehle, die durch ein Semikolon (;) voneinander getrennt sind. Obwohl sie hier der Übersichtlichkeit halber in getrennte Zeilen aufgelistet sind, können sie in der R-Konsole in einer Zeile stehen.

Die Determinantenfunktion lässt sich mit dem plot-Befehl grafisch darstellen:

```
>plot(lamba, detR, ylim=c(-50,50), type="l", ylab="det(R-lambda*I)")
>abline(h=0)
>box(which="figure").
```

Aus Abbildung 3.1 geht hervor, dass die Kurve der Determinante die Nulllinie für $\lambda>1$ zweimal schneidet. Die größte Wurzel (=Eigenwert) liegt zwischen 3 und 4 und die zweitgrößte Wurzel zwischen 1 und 2. Die Determinante liegt für viele λ-Werte, die kleiner als 1 sind, nahe bei Null, so dass die kleineren Wurzeln bei der gegebenen Skalierung der Grafik nicht exakt lokalisiert werden können. Da die Determinantenfunktion jedoch ein Polynom sechsten Grades ist, gibt es vier weitere Wurzeln (Nullstellen), die deutlich kleiner als 1 sind.

Abbildung 3.1: Grafische Darstellung der Determinantenfunktion $\left|\mathbf{R}-\lambda\cdot\mathbf{I}\right|$



Eine Berechnung der Eigenwerte und der zugehörgien Eigenvektoren der Korrelationsmatrix **R** erfolgt mit dem Befehl eigen:

```
> Eigen = eigen(R).
```

Das Ergebnis ist in dem Listenobjekt Eigen gespeichert[5],

```
>class(Eigen)
[1] "list",
```

das die Attribute values und vectors enthält:

```
> str(Eigen)
List of 2
$ values :num [1:6] 3.562 1.782 0.301 0.182 0.102 ...,
 $ vectors: num [1:6, 1:6] -0.504 -0.484 0.456 0.15 0.473 ....
```

Aus dem Vektor der Eigenwerte ew gehen die exakten Wurzeln der Determinante $\left|\mathbf{R}-\lambda\cdot\mathbf{I}\right|$ hervor:

```
>ew = Eigen$values
>ew
[1] 3.56196327 1.78222288 0.30123955 0.18161127 0.10186866 0.07109437
```

[5] Ein Listenobjekt besteht in R aus unterschiedlichen Typen von Objekten. Hier enthält das list-Objekt den Vektor values der Eigenwerte und die Matrix vectors der Eigenvektoren.

```
>class(ew)
[1] "numeric"
```

Sie summieren sich zu dem Wert 6 (= m):

```
>ew_sum = sum(ew)
>ew_sum
[1] 6.
```

Die Matrix EVder zugehörigen Eigenvektoren ist durch

```
> EV = Eigen$vectors
> EV
[,1]       [,2]       [,3]       [,4]       [,5]       [,6]
[1,] -0.5035195 -0.09882825  0.2204702 -0.2172156  0.5909047  0.5401325
[2,] -0.4838541 -0.17921213  0.3890376 -0.4318150 -0.2011615 -0.5962280
[3,]  0.4559119  0.29603911 -0.1231536 -0.6199821  0.4895797 -0.2554845
[4,]  0.1499901 -0.68023128 -0.3696214 -0.4738515 -0.2763334  0.2779802
[5,]  0.4728113 -0.03220923  0.7903421 -0.1186835 -0.2155621  0.3003641
[6,]  0.2417647 -0.63774980  0.1536723  0.3786217  0.4978457 -0.3464176
>class(EV)
[1] "matrix"
```

gegeben. Die Eigenvektoren sind auf die Länge 1 normiert, d.h. ihr Skalarprodukt ist gleich 1, z.B.

```
> t(EV[,1])%*%EV[,1]
 [,1]
[1,]    1
```

und sie stehen orthogonal (senkrecht) zueinander, d.h. das Skalarprodukt zweier unterschiedlicher Eigenvektoren ist gleich 0, z.B.

```
>round(t(EV[,1])%*%EV[,2],6)
    [,1]
[1,]    0.
```

Die Normierung auf 1 und die Orthogonalität der Eigenvektoren zeigt sich insgesamt daran, dass das Matrixprodukt t(EV)%*%EV eine Einheitsmatrix ist:

```
>round(t(EV)%*%EV,6)
 [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    0    0    0    0    0
[2,]    0    1    0    0    0    0
[3,]    0    0    1    0    0    0
[4,]    0    0    0    1    0    0
[5,]    0    0    0    0    1    0
[6,]    0    0    0    0    0    1.
```

Übungsaufgaben

3.2.1 Berechnen Sie die Determinante $|\mathbf{R} - \lambda \cdot \mathbf{I}|$ für λ-Werte im relevanten Bereich unter Verwendung eines Inkrements von 0,1 für den Datensatz`Documenta`!!

3.2.2 Stellen Sie die in Aufg. 3.2.1 berechnete Determinantenfunktion grafisch dar und interpretieren Sie sie! Geben Sie die approximativ die Wurzeln (Nullstellen) der Determinate $|\mathbf{R} - \lambda \cdot \mathbf{I}|$ an, die größer als 1 sind!

3.2.3 Berechnen Sie die Eigenwerte der Korrelationsmatrix `R`des Documenta-Beispiel und interpretieren Sie sie!

3.2.4 Ermitteln Sie die Eigenvektoren der Korrelationsmatrix `R` des Documenta-Beispiels und zeigen Sie beispielhaft anhand von zwei ausgewählten Vektoren ihre Orthogonalität und Normierung auf 1!

3.3 Hauptkomponentenmethode und Faktormatrix

Die Eigenvektoren der Korrelationsmatrix **R** sind nicht identisch mit den Faktorvektoren $\mathbf{a}_k$ wie sie mit der Hauptkomponentenmethode bestimmt werden. Sie lassen sich jedoch unter Verwendung der Eigenwerte λ_k aus ihnen berechnen. Hierbei ist die Frage nach der Anzahl der gemeinsamen Faktoren zu stellen.

Nach dem Kaiser-Kriterium werden diejenigen Faktoren als substanziell interpretiert, deren Eigenwerte größer als 1 sind. Da in dem 12-Regionen-Beispiel diese Bedingung für die beiden ersten Eigenwerte λ_1(= 3,562) und λ_2(= 1.782) erfüllt ist,müssten hiernach zwei gemeinsame Faktoren extrahiert werden.

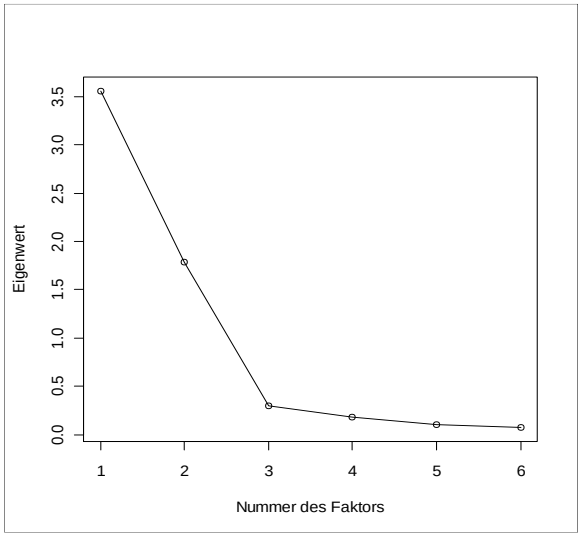
Der Scree-Test stellt ein alternatives Kriterium für die Wahl der Faktorenzahl bereit. Er lässt sich mit dem Befehl

```
>plot(ew, type="o", xlab="Nummer des Faktors", ylab="Eigenwert")
> box(which="figure")
```

erzeugen. Aus dem in Abbildung 3.2 wiedergegebenen Screeplot geht hervor, dass der scharfe Knick („Ellbogen“) bei dem dritten Faktor liegt.

Während die Eigenwerte der Faktoren 1 und 2 auf einer steil ansteigenden Geradenliegen, verlaufen die Eigenwerte der Faktoren 3 bis 6 näherungsweise parallel zur Abzisse. Nach dem Scree-Test werden die Faktoren als substanziell interpretiert, die vor dem „Knick“ liegen. Die nachfolgenden Faktoren werden als Zufallsfaktoren angesehen. Damit legt der Scree-Test in Einklang mit dem Kaiser-Kriterium die Extraktion von zwei gemeinsamen Faktoren nahe.

Abbildung 3.2: Screeplot



Bei einem kumulierten Varianzkriterium von z.B. 90% wären dagegen drei gemeinsame Faktoren zu extrahieren, da die ersten beiden Faktoren zusammen 89,1% der gesamten Varianz der manifesten Variablen erklären und erst beim dritten Faktor mit 94,1% die gesetzte Marke überschritten wird:

```
> ew.pro = ew/m
> ew.pro
[1] 0.59366055 0.29703715 0.05020659 0.03026855 0.01697811 0.01184906
>ew.cumpro = cumsum(ew.pro)
>ew.cumpro
[1] 0.5936605 0.8906977 0.9409043 0.9711728 0.9881509 1.0000000
```

Bei der Kumulierung der anteilsmäßigen Eigenwerteist von der Funktion `cumsum` Gebrauch gemacht worden, die die Werte eines Vektors bis zu dem jeweiligen Element summiert.

Das kumulierte Varianzkriterium von 90% wird bei dem 12-Regionen-Beispiel bei einer Extraktion von zwei Faktoren nur knapp unterschritten. Dagegen weisen das Kaiser-Kriterium und der Scree-Test einheitlich eine Zwei-Faktoren-Lösung aus, die daher hier gewählt wird.

Die unrotierte Faktormatrix `A` der Zwei-Faktoren-Lösung ergibt sich, indem man die auf die ersten beiden Spalten reduzierte der Matrix der Eigenvektoren,

```
> EV.F1F2 = cbind(EV[,1],EV[,2])
> EV.F1F2
  [,1]       [,2]
[1,] -0.5035195 -0.09882825
[2,] -0.4838541 -0.17921213
[3,]  0.4559119  0.29603911
[4,]  0.1499901 -0.68023128
[5,]  0.4728113 -0.03220923
[6,]  0.2417647 -0.63774980,
```

mit der Diagonalmatrix der Quadratwurzeln der beiden größten Eigenwerte,

```
> ewD.F1F2 = diag(sqrt(c(ew[1],ew[2])))
> ewD.F1F2
  [,1]    [,2]
[1,] 1.887316 0.000000
[2,] 0.000000 1.334999,
```

multipliziert:

```
> A = EV.F1F2%*%ewD.F1F2
> A
[,1]       [,2]
[1,] -0.9503005 -0.1319356
[2,] -0.9131858 -0.2392480
[3,]  0.8604499  0.3952120
[4,]  0.2830788 -0.9081082
[5,]  0.8923446 -0.0429993
[6,]  0.4562864 -0.8513955.
```

Die Faktorvektoren (= Spalten der Faktormatrix) sind bis auf eine Spiegelung der Ladungen an der Abzisse, d.h. bis auf das Vorzeichen, determiniert. Da die größten Faktorladungen der beiden Faktorvektoren negativ sind, werden sie der besseren Interpretation halber noch mit dem Faktor -1 multipliziert:

```
> A = A*(-1)
> A
[,1]       [,2]
[1,]  0.9503005  0.1319356
[2,]  0.9131858  0.2392480
[3,] -0.8604499 -0.3952120
[4,] -0.2830788  0.9081082
[5,] -0.8923446  0.0429993
[6,] -0.4562864  0.8513955.
```

Obwohl die endgültige Interpretation der Faktorenextraktion erst nach Rotation der Faktoren erfolgen kann, wird das Faktorenmuster hier bereits anhand der unrotierten Faktormatrix deutlich. Die Variablen x1 (Einwohnerdichte), x2 (BIP pro Kopf), x3 (Anteil der Erwerbstätigen in der Landwirtschaft) und x5 (Geburtenrate) laden absolut hoch auf dem ersten Faktor, wobei die Ladungen der beiden letzteren Variablen negativ sind. Faktor 1 kann daher als „Verstädterung“ („Urbanisierung“) interpretiert werden. Der zweite Faktor ist dagegen mit den Variablen x4 (Wachstumsrate des BIP) und x6 (Wanderungssaldo) hoch korreliert, was seine Interpretation als „Attraktivität“ begründet.

Bildet man die Skalarprodukte der beiden Faktorvektoren A[,1] und A[,2], erhält man die Eigenwerte der beiden gemeinsamen Faktoren:

```
> la1 = t(A[,1])%*% A[,1]
> la1
     [,1]
[1,] 3.561963
> la2 = t(A[,2])%*% A[,2]
> la2
[,1]
[1,] 1.782223.
```

Anhand der kompletten Faktormatrix A6 erkennt man, dass die Extraktion zusätzlicher Faktoren zu keinen substanziell interpretierbaren Faktorladungen (Faustregel: $a_{jk} \geq 0{,}5$) führen würde, da die Ladungen der potenziell extrahierbaren Faktoren 3 bis 6 durchweg kleiner als 0,5 sind:

```
> A6 = EV%*% diag(sqrt(ew))
> A6
          [,1]       [,2]        [,3]        [,4]        [,5]        [,6]
[1,] -0.9503005 -0.1319356  0.12100571 -0.09256834  0.18859827  0.14401837
[2,] -0.9131858 -0.2392480  0.21352445 -0.18402174 -0.06420444 -0.15897541
[3,]  0.8604499  0.3952120 -0.06759324 -0.26421078  0.15625852 -0.06812119
[4,]  0.2830788 -0.9081082 -0.20286780 -0.20193596 -0.08819698  0.07411931
[5,]  0.8923446 -0.0429993  0.43378159 -0.05057799 -0.06880067  0.08008766
[6,]  0.4562864 -0.8513955  0.08434350  0.16135295  0.15889675 -0.09236716.
```

Die Faktorextraktion nach der Hauptkomponentenmethode kann in R kompakt mit der Funkion principal durchgeführt werden, die in dem R-Paket psych,

```
> library(psych),
```

verfügbar ist.[6] Nach Spezifkation der Datenmatrix REG12 und der Anzahl der gemeinsamen Faktoren erhält man mit Setzen des Arguments rotate="none" die unrotierte Zwei-Faktoren-Lösung der Hauptkomponentenmethode:

```
> REG12.PCA = principal(REG12, nfactors=2, rotate="none")
> REG12.PCA
Principal Components Analysis
Call: principal(r = REG12, nfactors = 2, rotate = "none")
Standardized loadings (pattern matrix) based upon correlation matrix
PC1   PC2   h2  u2
x1 -0.95  0.13 0.92 0.080
x2 -0.91  0.24 0.89 0.109
x3  0.86 -0.40 0.90 0.103
x4  0.28  0.91 0.90 0.095
x5  0.89  0.04 0.80 0.202
x6  0.46  0.85 0.93 0.067
```

[6] Die im Basispaket enthaltene R-Funktion prcomp verwendet nicht die für eine Faktorenanalyse erforderliche Normierung, so dass auf sie hier nicht zurückgegriffen wird.

```
  PC1  PC2
SS loadings    3.56 1.78
Proportion Var 0.59 0.30
Cumulative Var 0.59 0.89

Test of the hypothesis that 2 components are sufficient.

The degrees of freedom for the null model are  15  and the objective function was  5.99
The degrees of freedom for the model are 4  and the objective function was  0.73
The number of observations was  12  with Chi Square =  4.96  with prob <  0.29

Fit based upon off diagonal values = 0.99>.
```

Die unter den Bezeichnungen PC1 und PC2aufgelisteten Faktorladungen lassen sich in ein matrix-Objekt transformieren:

```
> A = round(matrix(REG12.PCA$loadings, nrow=6, ncol=2), digits=3)
> A
       [,1]     [,2]
[1,] -0.950  0.132
[2,] -0.913  0.239
[3,]  0.860 -0.395
[4,]  0.283  0.908
[5,]  0.892  0.043
[6,]  0.456  0.851.
```

Spiegelt man die Komponenten des Vektors A[,1] der besseren Interpretierbarkeit halber an der Abzisse,

```
> A[,1] = A[,1]*(-1),
```

erhält man die mit der Matrizenrechnung erhaltene unrotierte Faktormatrix

```
> A
  [,1]   [,2]
[1,]  0.950  0.132
[2,]  0.913  0.239
[3,] -0.860 -0.395
[4,] -0.283  0.908
[5,] -0.892  0.043
[6,] -0.456  0.851.
```

Daneben sind unter h2 und u2 die Kommunalitäten bzw. Einzelrestvarianzen ausgewiesen.

Es folgen die Größen SS loadings, Proportion Var und CumulativeVar, die sich sich auf die beiden größten Eigenwerte der Korrelationsmatrix **R** beziehen. Sie geben die Varian-

zen der Faktoren sowie ihre einfachen und kumulierten Anteile an der gesamten Streuung der beobachteten Variablen an.

Im Anschluss daran wird das Ergebnis eines χ^2-Test über die Gültigkeit der Nullhypothese „Die Anzahl von p=2 gemeinsamen Faktoren ist hinreichend“ wiedergegeben. Da der p-Wert von 0,29 größer als die üblicherweise verwendeten Signifikanzniveaus α (0,01; 0.05; 0,10) ist, lässt sich die Nullhypothese nicht verwerfen.

Der zum Schluss ausgewiesene fit.off-Wert,

```
> fit.PCA = REG12.PCA$fit.off
> fit.PCA
[1] 0.9944024,
```

gibt den Anteil der gesamten quadrierten Merkmalskorrelationen an, der durch das Zwei-Faktoren-Modell erklärt werden kann.

Übungsaufgaben

3.3.1 Erstellen Sie ein Screeplot für die Korrelationsmatrix des Datensatzes Documenta! Wie viel gemeinsame Faktoren legt der Scree-Test nahe?

3.3.2 Bestimmen Sie die prozentualen Anteile kumulierten Varianzen der Faktoren des Datensatzes documenta! Geben Sie an,welchen kumulierten Varianzkriterien

- dem Kaiser-Kriterium,
- dem Scree-Test

entsprechen würde!

3.3.3 Geben Sie für den Datensatz documentadieunrotiertenFaktormatrizen an, die sich auf der Grundlage

- des Kaiser-Kriteriums,
- des Scree-Tests

ergeben!

3.3.4 Zeigen Sie, dass die Summen der Ladungsquadrate der extrahierten Faktoren mit ihren Eigenwerten übereinstimmen!

3.3.5 Geben Sie die komplette Faktormatrix nach der Hauptkomponentenmethode für die beiden in Aufg. 3.3.3 berechneten Faktorlösungen an! Beurteilen Sie die beiden Faktorlösungen anhand der darin enthaltenen Informationen!

3.3.6 Interpretieren Sie die mit der Funktion principal ermittelte unrotierte Zwei-Faktoren-Lösung für die Documenta-Daten!

3.4 Faktorenanalyse im engeren Sinne (Hauptachsen-Faktorenanalyse)

Bei der Faktorenanalyse im engeren Sinne werden die Einsen auf der Hauptdiagonale der Korrelationsmatrix**R** durch Schätzer für dieKommunalitäten ersetzt. Die gemeinsamen Faktoren werden dann aus der auf diese Weise gebildeten reduzierten Korrelationsmatrix $\mathbf{R}_h$ extrahiert. Hier werden als Anfangsschätzer der Kommunalitäten die multiplen Determinationskoeffizienten verwendet, die sich aus den Hauptdiagonalelementen r^{jj} der inversen Korrelationsmatrix $\mathbf{R}^{-1}$ nach der Formel $1-1/r^{jj}$ berechnen lassen:

```
>Ri = solve(R)    # Invertierung der Korrelationsmatrix
>Ri
         x1          x2          x3          x4          x5       x6
x1  8.02905079 -4.8171174  1.4693935  0.8217622  1.6869194 -0.08323604
x2 -4.81711736  7.0103547  2.3992012 -1.0882342 -0.8514125  1.25160962
x3  1.46939346  2.3992012  5.5453862 -0.6520599 -1.9781598  2.20717832
x4  0.82176220 -1.0882342 -0.6520599  3.7923220  1.1312872 -3.62782076
x5  1.68691941 -0.8514125 -1.9781598  1.1312872  3.9396145 -2.31768149
x6 -0.08323604  1.2516096  2.2071783 -3.6278208 -2.3176815  5.23337054
> h2.md = diag(1-(1/Ri))
> h2.md
   x1       x2       x3       x4       x5       x6
0.8754523 0.8573539 0.8196699 0.7363093 0.7461681 0.8089186
> H2.md = diag(h2.md)
> H2.md
   [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
[1,] 0.8754523 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
[2,] 0.0000000 0.8573539 0.0000000 0.0000000 0.0000000 0.0000000
[3,] 0.0000000 0.0000000 0.8196699 0.0000000 0.0000000 0.0000000
[4,] 0.0000000 0.0000000 0.0000000 0.7363093 0.0000000 0.0000000
[5,] 0.0000000 0.0000000 0.0000000 0.0000000 0.7461681 0.0000000
[6,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.8089186
> Rh = R - I + H2.md    # Bildung der reduzierten Korrelationsmatrix Rh
> Rh
         x1          x2          x3          x4          x5          x6
x1  0.8754523  0.90723433 -0.83389081 -0.16101263 -0.7865920 -0.30934486
x2  0.9072343  0.85735387 -0.84531953 -0.05351755 -0.7109732 -0.22018038
x3 -0.8338908 -0.84531953  0.81966991 -0.06708459  0.7186602  0.03891881
x4 -0.1610126 -0.05351755 -0.06708459  0.73630931  0.2258691  0.83177028
x5 -0.7865920 -0.71097315  0.71866021  0.22586911  0.7461681  0.45387020
x6 -0.3093449 -0.22018038  0.03891881  0.83177028  0.4538702  0.80891856.
```

In der Hauptachsen-Faktorenanalyse werden die Eigenwerte und Eigenvektoren der reduzierten Korrelationsmatrix $\mathbf{R}_h$ bestimmt:

```
>Eigen.Rh = eigen(Rh)
>ew.Rh = Eigen.Rh$values
>ew.Rh
[1]  3.386477366  1.561301695  0.076223705 -0.009122723 -0.045182143 -0.125825930
```

```
>EV.Rh = Eigen.Rh$vectors
>EV.Rh
  [,1]   [,2]     [,3]     [,4]     [,5]     [,6]
[1,]  0.5115873 -0.09465814  0.25193286 -0.2316419  0.77012760 -0.1381797
[2,]  0.4896021 -0.17824868  0.49559453 -0.3258241 -0.56369511  0.2428780
[3,] -0.4565467  0.29472913  0.07435157 -0.7390839  0.15712984  0.3581018
[4,] -0.1433465 -0.66516719 -0.29644013 -0.4601743 -0.11300407 -0.4739170
[5,] -0.4616315 -0.04200401  0.76449010  0.1439330  0.01865455 -0.4238186
[6,] -0.2376582 -0.65436248  0.11430607  0.2479085  0.22659129  0.6240675.
```

Vergleicht man die Summen der Eigenwerte der reduzierten Korrelationsmatrix $\mathbf{R}_h$und der Kommunalitäten h_j^2 miteinander, zeigt sich, dass sie identisch sind:

```
>ewRh.sum = sum(ew.Rh)
> ewRh.sum
[1] 4.843872
> h2md.sum = sum(h2.md)
> h2md.sum
[1] 4.843872.
```

Unter Verwendung der beiden größten Eigenwerte und zugehörigen Eigenvektoren lässt sich die sich unmittelbar aus den Anfangsschätzern der Kommunalitäten resultierende unrotierte Faktormatrix Anr bestimmen:

```
> a1nr = EV.Rh[,1]*sqrt(ew.Rh[1])
> a1nr
[1]  0.9414426  0.9009846 -0.8401547 -0.2637918 -0.8495120 -0.4373478
> a2nr = EV.Rh[,2]*sqrt(ew.Rh[2])*(-1)
> a2nr
[1]  0.11827729  0.22272540 -0.36827011  0.83114010  0.05248488  0.81763939
>Anr = cbind(a1nr, a2nr)
>Anr
a1nr        a2nr
[1,]  0.9414426  0.11827729
[2,]  0.9009846  0.22272540
[3,] -0.8401547 -0.36827011
[4,] -0.2637918  0.83114010
[5,] -0.8495120  0.05248488
[6,] -0.4373478  0.81763939.
```

Um die iterierte Faktorenlösung der Hauptachsen-Faktorenanalyse zu ermitteln, kann auf die R-Funktion fa zurückgegriffen werden, die in dem R-Paket psych verfügbar ist.[7] Es empfiehlt sich hier, es zusammen mit dem Paket GPArotation für die Faktorrotation zu laden:

[7]Mit der allgemein verfügbaren R-Funktion factanal erfolgt eine Schätzung der Faktormatrix unter Verwendung der Maximum-Likelihood-Methode. Eine Hauptachsen-Faktorenanalyse kann mit dieser Funktion nicht durchgeführt werden.

```
>library(psych)
>library(GPArotation).
```

Danach können praktisch alle Extraktions- und Rotationsverfahren zur Durchführung einer Faktorenanalyse genutzt werden.

Bei dem Aufruf der fa-Funktion muss eine Daten- oder Korrelationsmatrix mit der Anzahl der zu extrahierenden Faktoren (nfactors) angegeben werden. Zur Durchführung einer Hauptachsen-Faktorenanalyse ist die Faktorisierungsmethode fm="pa" zu wählen. Durch die Spezifikation Arguments rotate="none" wird die unrotierte Faktormatrix ermittelt. Voreingestellt ist eine maximale Zahl von 50 Iterationen (max.iter=50) für die Konvergenz der Faktorlösung. Unter Berücksichtigung der gesetzten Maximalzahl (max.iter) wird die Lösung iteriert, solange die Veränderungen der Kommunalitäten größer als der gesetzte Wert min.err (Voreinstellung: 0.001) ist.

Um zu verhindern, dass die Kommunalität der Variablen Wanderungssaldo (x_6) den Wert 1 überschreitet, wird die maximale Iterationszahl für den Datensatz Regionen12 auf 10 begrenzt. Die Meldung maximum iteration exceeded zeigt an, dass aufgrund der Begrenzung der Iterationszahl das Konvergenzkriterium min.err=0.000001 bei der ermittelten Faktorlösung nicht erfüllt ist:[8]

```
> REG12.FA = fa(REG12, nfactors=2,fm="pa", rotate="none", min.err=0.000001,max.iter=10)
maximum iteration exceeded
> REG12.FA
Factor Analysis using method =  pa
Call: fa(r = REG12, nfactors = 2, rotate = "none", min.err = 1e-06,
max.iter = 10, fm = "pa")

Standardized loadings (pattern matrix) based upon correlation matrix
  PA1   PA2   h2     u2
x1 -0.95  0.13 0.92 0.07957
x2 -0.90  0.23 0.86 0.14208
x3  0.84 -0.38 0.86 0.14059
x4  0.27  0.80 0.71 0.29209
x5  0.84  0.04 0.70 0.29910
x6  0.47  0.88 1.00 0.00038

PA1  PA2
SS loadings  3.41 1.64
Proportion Var 0.57 0.27
Cumulative Var 0.57 0.84

Test of the hypothesis that 2 factors are sufficient.
The degrees of freedom for the model are 4  and the objective function was  0.25
The number of observations was  12  with Chi Square =  1.73  with prob<  0.79
```

[8]Auf die Ausgabe einiger ergänzender Kenngrößen, die an dieser Stelle nicht von Relevanz sind, wird hier verzichtet.

```
Fit based upon off diagonal values = 1.
```

Als erstes werden wird die unrotierte Faktormatrix (Faktorvektoren PA1 und PA2) zusammen mit den Kommunalitäten (h2) und Einzelrestvarianzen (u2) ausgewiesen. Hier ergibt sich gegenüber der Hauptkomponentenmethode kein substanzieller Unterschied bei der Interpretation der Faktorenmatrix. Die Kommunalität h_6^2 von 1 zeigt an, dass die Variable Wanderungssaldo (x6) vollständig durch die beiden gemeinsamen Faktoren erklärt wird. Dementsprechend nimmt die Einzelrestvarianz dieser Variablen den Wert 0 an.

Mit dem print-Befehl können nach Angabe der Faktormatrix (loadings) durch eine Spezifikation des Arguments digits die auszugebenden Dezimalstellen der Faktorladungen festgelegt werden. Hierbei werden jedoch standardmäßig Ladungen, die absolut kleiner als 0,1 sind, unterdrückt. Um die komplette Faktormatrix zu erhalten, wird der Schwellenwert (cutoff) für die zu unterdrückenden Faktorladungen gleich 0 gesetzt:

```
>print(REG12.FA$loadings, digits=3, cutoff=0)

Loadings:
PA1   PA2
x1 -0.950  0.134
x2 -0.897  0.231
x3  0.844 -0.383
x4  0.268  0.798
x5  0.836  0.039
x6  0.466  0.885

 PA1   PA2
SS loadings    3.408 1.638
Proportion Var 0.568 0.273
Cumulative Var 0.568 0.841
```

Die unter der Faktormatrix angegebenen Größen SS loadings,Proportion Var und CumulativeVarbeziehen sich auf die beiden größten Eigenwerte der reduzierten Korrelationsmatrix $\mathbf{R}_h$, die den Varianzen der beiden Faktoren entsprechen.

Anschließend wird mit der Funktion fa das Ergebnis eines χ^2-Test über die Gültigkeit der Nullhypothese „Die Anzahl von p=2 gemeinsamen Faktoren ist hinreichend“ geprüft. Der p-Wert von 0,79 überschreitet alle gängigen Signifikanzniveaus, so dass die Nullhypothese einer Zwei-Faktoren-Lösung nicht verworfen werden kann.

Schließlich gibt der fit.off-Wert,

```
> fit.FA = REG12.FA$fit.off
> fit.FA
[1] 0.9989562,
```

den Anteil der gesamten quadrierten Merkmalskorrelationen an, der durch das faktorenanalytische Modell erklärt wird.

Definiert man unter Verwendung der Faktorenzahl

```
> p = 2
```

die unrotierte Faktormatrix Anr,

```
>Anr = matrix(loadings(REG12.FA), nrow=m, ncol=p)
>round(Anr, 3)
[,1]  [,2]
[1,] -0.950  0.134
[2,] -0.897  0.231
[3,]  0.844 -0.383
[4,]  0.268  0.798
[5,]  0.836  0.039
[6,]  0.466  0.885,
```

dann erhält man die reproduzierte Korrelationsmatrix Rrep (=$\tilde{\mathbf{R}}_{\mathrm{h}}$) als Produkt der Faktormatrix Anr und ihrer Transponierten:

```
>Rrep = Anr%*%t(Anr)
>Rrep
>round(Rrep, 3)
[,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,]  0.920  0.883 -0.853 -0.147 -0.789 -0.324
[2,]  0.883  0.858 -0.846 -0.056 -0.741 -0.213
[3,] -0.853 -0.846  0.859 -0.080  0.691  0.054
[4,] -0.147 -0.056 -0.080  0.708  0.255  0.830
[5,] -0.789 -0.741  0.691  0.255  0.701  0.424
[6,] -0.324 -0.213  0.054  0.830  0.424  1.000.
```

Die Differenz zwischen der beobachteten und reproduzierten Korrelationsmatrix, $\mathbf{R} - \tilde{\mathbf{R}}_{\mathrm{h}}$, gibt die Restkorrelationsmatrix Rrest wieder:

```
>Rrest = R - Rrep
>round(Rrest, 3)
  x1    x2    x3    x4    x5    x6
x1  0.080  0.024  0.020 -0.014  0.003  0.014
x2  0.024  0.142  0.000  0.002  0.030 -0.007
x3  0.020  0.000  0.141  0.012  0.028 -0.016
x4 -0.014  0.002  0.012  0.292 -0.029  0.001
x5  0.003  0.030  0.028 -0.029  0.299  0.030
x6  0.014 -0.007 -0.016  0.001  0.030  0.000.
```

Während auf der Hauptdiagonale von Rrest die Einzelrestvarianzen stehen, spiegeln die Nichthauptdiagonalelemente die Teile der Korrelationen zwischen den Variablen,

die nach der Extraktion der beiden gemeinsamen Faktoren unerklärt bleiben (=Residualkorrelationen). Da sie nahe bei 0 liegen – alle Residualkorrelationen sind absolut kleiner oder gleich 0,03 – liegt es nahe, sie als zufällig betrachtet werden, was sich auch mit der Funktion corr.p zeigt:

```
>corr.p(Rrest,12,adjust="none")
Call:corr.p(r = Rrest, n = 12, adjust = "none")
Correlation matrix
x1x2   x3   x4   x5   x6
x1  0.08  0.02  0.02 -0.01  0.00  0.01
x2  0.02  0.14  0.00  0.00  0.03 -0.01
x3  0.02  0.00  0.14  0.01  0.03 -0.02
x4 -0.01  0.00  0.01  0.29 -0.03  0.00
x5  0.00  0.03  0.03 -0.03  0.30  0.03
x6  0.01 -0.01 -0.02  0.00  0.03  0.00
Sample Size
[1] 12
Probability values (Enteries above the diagonal are adjusted for multiple tests when appropriate.)
x1   x2   x3   x4   x5   x6
x1 0.81 0.94 0.95 0.97 0.99 0.96
x2 0.94 0.66 1.00 0.99 0.93 0.98
x3 0.95 1.00 0.66 0.97 0.93 0.96
x4 0.97 0.99 0.97 0.36 0.93 1.00
x5 0.99 0.93 0.93 0.93 0.34 0.93
x6 0.96 0.98 0.96 1.00 0.93 1.00.
```

Während der χ^2-Test global bestätigt hat, dass zwei gemeinsame Faktoren die Merkmalskorrelationen hinreichend erklären können, weisen die t-Tests Auskunft aus, dass keine der 15 Restkorrelationen des Zwei-Faktoren-Modells mehr signifikant ist.

Übungsaufgaben

3.4.1 Berechnen Sie für den Datensatz documenta die Anfangsschätzer für die Kommunalitäten nach der Methode der multiplen Determinationskoeffizienten und bestimmen Sie hiermit die reduzierte Korrelationsmatrix!

3.4.2 Geben Sie die Eigenwerte der reduzierten Korrelationsmatrix an! Können Sie bereits unter Kenntnis der reduzierten Korrelationsmatrix den Wert angeben, zu dem sich ihre Eigenwerte summieren müssen?

3.4.3 Welche Faktormatrizen ergeben sich im ersten Schritt mit den in Aufg. 3.4.1 berechneten Kommunalitätenschätzern nach der Hauptachsen-Faktorenanalyse bei einer

- Zwei-Faktoren-Lösung,
- Drei-Faktorenlösung?

3.4.4 Wie verändert sich die in Aufg. 3.4.3 ermittelte Faktormatrix in der iterierten Lösung (cutoff=0.000001) bei der Extraktion von zwei und drei gemeinsamen Faktoren?

3.4.5 Geben Sie die endgültigen Kommunaliäten der in Aufg. 3.4.4 ermittelten iterierten Lösungen an!

3.4.6 Bestimmen Sie die Restkorrelationsmatrizen der Zwei- und Drei-Faktorenlösung die und interpretieren Sie sie!

3.4.7 Testen Sie die Annahmen, dass

- zwei gemeinsame Faktoren,
- drei gemeinsame Faktoren

für den Datensatz documenta hinreichend sind (α=0,05) und interpretieren Sie das Testergebnis!

3.5 Rotation der Faktormatrix

Das Verfahren der Faktorenextraktion ist so ausgestaltet, dass der erste Faktor einen möglichst großen Teil der Varianz der beobachteten Variablen erklärt. Der zweite und evtl. weitere Faktoren leisten ihre Erklärungsbeiträge jeweils für die noch nicht erklärten Streuungen der Variablen. Daher werden die Faktorladungen des Faktors F_2 im Mittel absolut kleiner sein als die des Faktors F_1. Auf diese Weise können sich viele Ladungen ergeben, die absolut zwischen 0,3 und 0,7 liegen. Sie erschweren die Interpretation der Faktorenlösung, da ihr Erklärungsgehalt nicht unbedingt vernachlässigbar, aber auch nicht hoch ist.

Aus diesem Grund wird eine Rotation der Faktoren, d.h. eine Drehung des Koordinatensystems, in dem die Variablenvektoren liegen, mit dem Ziel vorgenommen, eine Einfachstruktur zu erreichen. Sie ist durch Faktorladungen gekennzeichnet, die tendenziell nahe bei 0 und 1 statt in einem schwer interpretierbaren mittleren Bereich liegen. Eine Rotation ist zulässig, da sie zu einer beobachtungsäquivalenten Faktormatrix führt, bei der die erklärten Varianzanteile der Variablen (= Kommunalitäten) unverändert bleiben.

Man unterscheidet orthogonale (rechtwinklige) und oblique (schiefwinklige) Rotationsverfahren. Bei orthogonaler Rotation bleiben die gemeinsamen Faktoren unkorreliert, wohingegen diese Eigenschaft bei obliquer Rotation verloren geht. In dem Paket stats ist als orthogonales Rotationsverfahren die Varimax-Methode und als obliques Rotationsverfahren die Promax-Methode implementiert.

Bei der Varimax-Methode werden die Variablenvektoren standardmäßig bei der Rotation auf die Länge 1 normiert (normalize = TRUE), so dass alle Variablen gleichgewichtig berücksichtigt werden (Kaiser-Normalisierung). Mit dem Stoppkriterium eps von 0,000001 wird die unrotierte Faktormatrix Anr des Datensatzes Regionen12 in die rotierte Faktorlösung

```
> REG12.vari = varimax(Anr, eps=0.000001)
> REG12.vari
$loadings
```

```
Loadings:
       [,1]   [,2]
[1,]  0.946 -0.162
[2,]  0.925
[3,] -0.921 -0.107
[4,]         0.841
[5,] -0.785  0.292
[6,] -0.174  0.985

                 [,1]  [,2]
SS loadings     3.243 1.803
Proportion Var  0.540 0.301
Cumulative Var  0.540 0.841

$rotmat
         [,1]       [,2]
[1,] 0.9522748 -0.3052420
[2,] 0.3052420  0.9522748
```

transformiert. Bei der Ausgabe der rotierten Faktormatrix sind Ladungen, die kleiner oder gleich 0,1 sind, unterdrückt worden. In kompletter Form lautet sie

```
> Arr = matrix(loadings(REG12.vari), nrow=m, ncol=p)
> round(Arr, 3)
       [,1]   [,2]
[1,]  0.946 -0.162
[2,]  0.925 -0.054
[3,] -0.921 -0.107
[4,] -0.011  0.841
[5,] -0.785  0.292
[6,] -0.174  0.985.
```

Durch die Rotation hat sich die Varianz des ersten Faktors zugunsten des zweiten Faktors ohne eine Veränderung ihrer Summe verringert.

Der Rotationswinkel alpha.vari ergibt sich aus den Elementen der Transformations-matrix rotmat:

```
> alpha.vari = asin(REG12.vari$rotmat[2,1])*180/pi
> alpha.vari
[1] 17.77272
```

oder

```
> alpha.vari = acos(REG12.vari$rotmat[1,1])*180/pi
> alpha.vari
[1] 17.77272.
```

Dass die Kommunalitäten nicht von der Rotation tangiert worden sind, erkennt man anhand der Ergebnisse ihrer Berechnung für die rotierte und unrotierte Faktorlösung:

```
> h2.vari = apply(Arr^2, 1, sum)  # Kommunalitäten der Varimax-rotierten HFA
> h2.vari
[1] 0.9204332 0.8579163 0.8594086 0.7079052 0.7009017 0.9996202

> h2.nr = apply(Anr^2, 1, sum)   # Kommunalitäten der unrotierten HFA
> h2.nr
[1] 0.9204332 0.8579163 0.8594086 0.7079052 0.7009017 0.9996202.
```

Ein visueller Vergleich der unrotierten und rotierten Faktorenlösung lässt sich durch eine Darstellung der Variablenvektoren im Faktorenraum vornehmen. Hierzu werden vorab den Zeilen und Spalten der Faktormatrizen Anr und Arr Label zugeordnet:

```
> rowlab = c("ED","BIP","EL","WBIP","GEB","WS")
> class(rowlab)
[1] "character"
> collab = c("F1","F2")
> class(collab)
[1] "character"

> rownames(Anr) = rowlab
> colnames(Anr) = collab
> Anr
  F1         F2
ED    0.9499522  0.13425391
BIP   0.8969300  0.23115557
EL   -0.8442273 -0.38299991
WBIP -0.2676951  0.79764940
GEB  -0.8362939  0.03891264
WS   -0.4658831  0.88463164

> rownames(Arr) = rowlab
> colnames(Arr) = collab
> Arr
   F1         F2
ED   0.94559545 -0.1621187
BIP   0.92468225 -0.0536571
EL   -0.92084404 -0.1070275
WBIP -0.01144317  0.8412932
GEB  -0.78450385  0.2923276
WS   -0.17362197  0.9846195.
```

Um beide Koordinatensysteme in einer Grafik einander gegenüberzustellen, wird der Plotbereich in eine Zeile und zwei Spalten unterteilt:

```
> par(mfrow=c(1, 2)).
```

Die Variablenvektoren der unrotierten und rotierten Lösung können jeweils mit dem plot-Befehl in beiden Faktorenräumen dargestellt werden:[9]

```
>plot(Anr, main="Unrotierte Faktoren", cex.main=1, font.main=1, xlim=c(-1.4,1), ylim=c(-1,1), xlab=
"Faktor 1", ylab="Faktor 2")
> text(Anr, labels=rownames(Anr), cex=0.8, adj=1.2)
> abline(h=0, v=0)
> box(which="figure").

>plot(Arr, main="Rotierte Faktoren", cex.main=1, font.main=1, xlim=c(-1.4,1), ylim=c(-1,1), xlab=
"Faktor 1", ylab="Faktor 2")
> text(Arr, labels=rownames(Arr), cex=0.8, adj=1.2)
> abline(h=0, v=0)
> box(which="figure").
```

Abbildung 3.3: Variablenvektoren im Faktorenraum

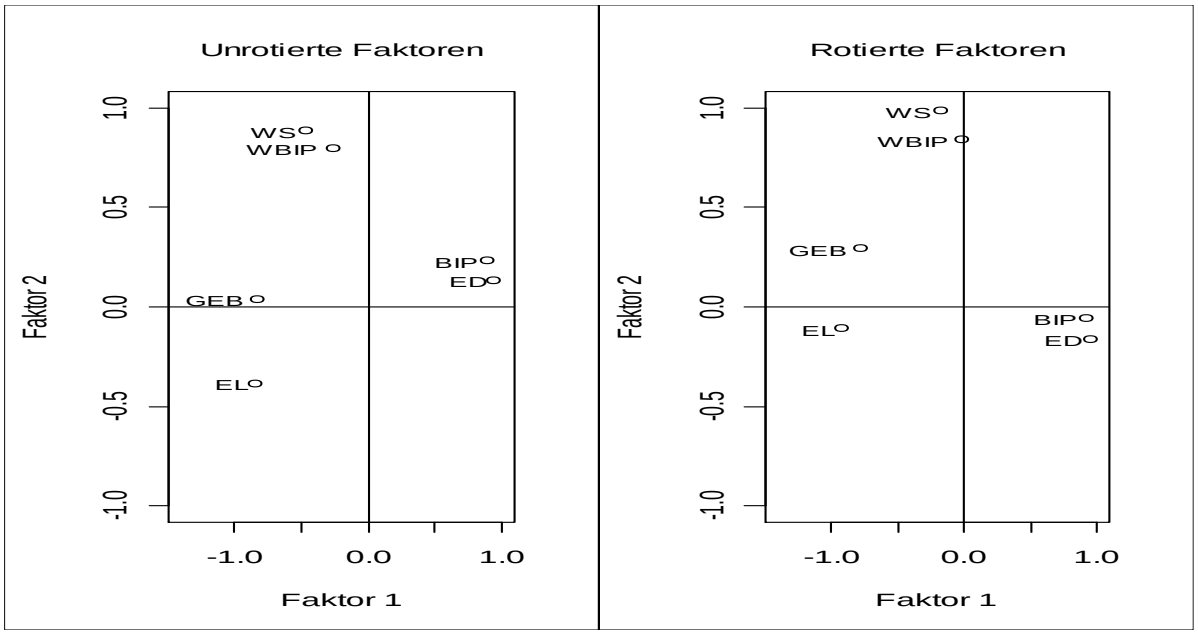


Abbildung 3.3 zeigt die Variablenvektoren ohne und mit Rotation im Faktorenraum. Die Konfigurationen unterscheiden sich nicht, d.h. ihre relative Lage zueinander ist bei beiden Lösungen identisch. Doch liegen die Variablenvektoren BIP, ED und EL im rotierten Faktorraum näher an der F_1-Achse als im unrotierten Faktorraum. Gleiches gilt für die Variablenvektoren WS und WBIP in Bezug auf die F_2-Achse. Eine geringfügig ungünstigere Lage im Hinblick auf das Kriterium einer Einfachstruktur ist allein für den Variablenvektor GEB zu konstatieren.

Eine schiefwinklige Faktorenlösung, bei der die beiden Faktoren nicht senkrecht aufeinander stehen, erhält man mit dem promax-Befehl:

[9] Diese Darstellung lässt sich im Prinzip auch erzeugen, indem man den biplot-Befehl mit dem R-Objekt REG12.FA der Klasse „psych“ „fa“ ausführt. Allerdings wird hierbei eine evtl. Spiegelung der Ladungen eines Faktors an einer der Achsen des Koordinatensystems außer Acht gelassen.

```
> REG12.pro = promax(Anr)
> REG12.pro
$loadings

Loadings:
        F1      F2
ED      0.938
BIP     0.932
EL     -0.950  -0.192
WBIP   0.104   0.858
GEB   -0.757   0.228
WS             0.990

                    F1    F2
SS loadings      3.238 1.812
Proportion Var   0.540 0.302
Cumulative Var 0.540 0.842

$rotmat
         [,1]          [,2]
[1,] 0.9256160 -0.2258895
[2,] 0.4408423  1.0000404.
```

Weitere Rotationsmethoden sind in der Funktion fa des Pakets psych implementiert, wobei z.T. auf das Paket GPArotation zurückgegriffen wird.

Übungsaufgaben

3.5.1 Bestimmen Sie für den Datensatz documenta die Varimax-rotierte

- Zwei-Faktoren-Lösung,
- Drei-Faktoren-Lösung!

Interpretieren Sie hierbei die

- Größen SS loadings, Proportion Var, Cumulative Var,
- Matrix rotmat!

3.5.2 Geben Sie die Varimax-rotierte Faktormatrix der

- Zwei-Faktoren-Lösung,
- Drei-Faktorenlösung

des Datensatzes documenta als matrix-Objekt an und interpretieren Sie sie!

3.5.3 Was versteht man unter einer beobachtungsäquivalenten Faktorstruktur? Zeigen Sie die Beobachtungsäquivalenz anhand der Zwei-Faktoren-Lösung des Datensatzes documenta auf!

3.5.4 Stellen Sie die Variablenvektoren der unrotierten und Varimax-rotierten Zwei-Faktoren-Lösung des Datensatzes documenta grafisch dar! Welche Unterschiede zwischen den beiden Diagrammen lassen sich hierbei besonders herausstellen?

3.5.5 Skizzieren Sie die Unterschiede zwischen der Varimax- und Promax-rotierten Faktormatrix im Zwei-Faktoren-Fall für den Datensatz documenta!

3.6 Bestimmung der Faktorwerte

Durch Matrizenmanipulation lässt sich aus der Strukturgleichung des faktorenanalytischen Modells die Beziehung

(3.5) $\mathbf{F}'_{KQ} = (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'\mathbf{Z}'$

herleiten, durch die zugleich ein Kleinst-Quadrate-Schätzer (KQ-Schätzer) gegeben ist. Er wird hier mit der rotierten Faktormatrix der Hauptkomponentenmethode,

```
> Arr.PCA = matrix(loadings(varimax(A, eps=0.000001)), nrow=m, ncol=p)
> Arr.PCA
           [,1]       [,2]
[1,]  0.945560150 -0.1607483
[2,]  0.942564041 -0.0475713
[3,] -0.939095828 -0.1171496
[4,]  0.004112915  0.9510710
[5,] -0.837468055  0.3100972
[6,] -0.178022649  0.9489178,
```

bestimmt:

```
> Ft.PCA = solve(t(Arr.PCA)%*%Arr.PCA)%*%t(Arr.PCA)%*%t(ZREG12)
> F.PCA = t(Ft.PCA)
> F.PCA
          [,1]       [,2]
1  -0.45616537 -0.5524149
2   1.64162926  1.3044181
3  -1.89615768 -1.2235309
4   0.18278482  0.6954310
5   0.31536218 -1.1733190
6   0.85412299 -1.1669651
7   0.07910656  0.7274160
8  -1.24518448  0.9996086
9   1.03019843 -0.8872220
10 -0.11702410  0.4041017
11 -0.87984495  1.2794162
12  0.49117234 -0.4069399.
```

Mit einer Kennzeichnung der Zeilen (=Regionen) und Spalten (= Faktoren),

```
> colnames(F.PCA) = c("F1","F2")
> rownames(F.PCA) = LETTERS[1:12],
```

hat man

```
> F.PCA
          F1          F2
A -0.45616537 -0.5524149
B  1.64162926  1.3044181
C -1.89615768 -1.2235309
D  0.18278482  0.6954310
E  0.31536218 -1.1733190
F  0.85412299 -1.1669651
G 0.07910656  0.7274160
H -1.24518448  0.9996086
I  1.03019843 -0.8872220
J -0.11702410  0.4041017
K -0.87984495  1.2794162
L  0.49117234 -0.4069399.
```

Man erhält die geschätzte Faktorwertematrix auch direkt mit der Funktion principal des Pakets psych,

```
> library(psych),
```

indem man dem Argument scores den Wahrheitswert TRUE zuordnet:

```
> REG12.PCA = principal(REG12, nfactors=2, rotate="varimax", scores=TRUE)
> REG12.PCA$scores
        PC1         PC2
A  0.45741293 -0.5510943
B -1.64445231  1.2995904
C  1.89874767 -1.2182003
D -0.18457796  0.6947322
E -0.31214031 -1.1737803
F -0.85070760 -1.1688064
G -0.08103339  0.7270702
H  1.24208543  1.0026188
I -1.02747052 -0.8896084
J  0.11589703  0.4043440
K  0.87614314  1.2812184
L -0.48990412 -0.4080842.
```

Aufgrund der Voreinstellung varimax ist es hier nicht erforderlich, das Argument rotate zu spezifizieren. Die Abweichungen der mit Hilfe der Matrizenrechnung und der R-Funktion principal ermittelten Faktorwerte liegen im Tausendstel-Bereich.

Da die Faktorwerte standardisiert sind, d.h. einen Mittelwert von 0 und eine Standardabweichung von 1 haben,

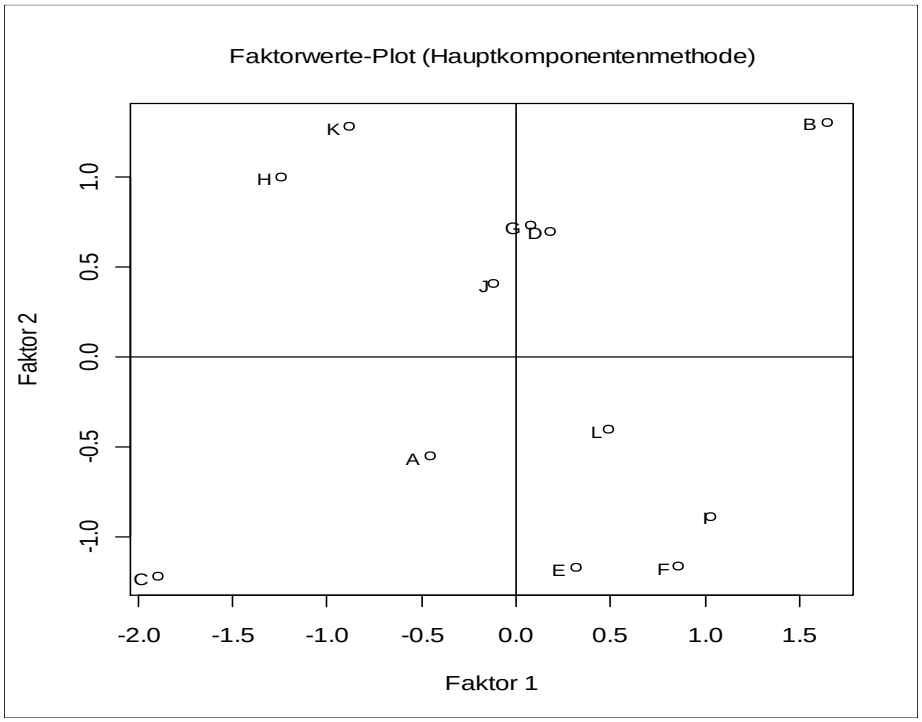
```
> round(apply(F.PCA,2,mean),3)
F1 F2
 0  0
> round(apply(F.PCA,2,sd),3)
F1 F2
 1  1,
```

lassen sie sich gut interpretieren. Hierzu werden sie in einem Faktorwerte-Plot dargestellt:

```
> plot(F.PCA, main="Faktorwerte-Plot (Hauptkomponentenmethode)", cex.main=1, font.main=1, xlab=
"Faktor 1", ylab="Faktor 2")
> text(F.PCA, labels=rownames(F.PCA), cex=0.8, adj=1.6)
> abline(h=0, v=0)
> box(which="figure").
```

Abbildung 3.4 gibt den Faktorwerte-Plot für die Hauptkomponentenlösung wieder. Die verstädterten Regionen B, D und G und die ländlichen Räume H, J und K stellen sich als attraktiv heraus. Positiv ragen hinsichtlich der Attraktivität insbesondere die Regionen B und K heraus. Eher unattraktiv sind die verstädterten Gebiete E, F, I und L sowie die ländlichen Regionen A und C. Als am wenigsten attraktiv erweisen sich dabei die Regionen C, E und F.

Abbildung 3.4: Faktorwerte-Plot für die Hauptkomponentenlösung



Bei der Hauptachsen-Faktorenanalyse wird in der Regel der Regressionsschätzer

(3.6) $\mathbf{F}'_{\mathrm{REG}} = \mathbf{A}'\mathbf{R}^{-1}\mathbf{Z}'$

für die Berechnung der Faktorwerte verwendet:

```
> Ft.FA = t(Arr)%*%Ri%*%t(Z)
> F.FA = t(Ft.FA)
> rownames(F.FA) = LETTERS[1:12]
> F.FA
          F1          F2
A -0.50622040 -0.6306372
B  1.62435663  1.1466765
C -1.85151350 -1.1325208
D -0.01070417  0.5709693
E  0.32886529 -1.2515213
F  1.00898571 -1.2145024
G  0.12546943  1.3262958
H -1.19049765  1.3046489
I  0.95567396 -0.8653254
J -0.02870932  0.2086810
K -0.80882225  0.6673076
L  0.35311628 -0.1300721.
```

Ohne Matrizenrechnung erhält man die oben wiedergegebene Faktorwertematrix bei der Hauptachsen-Faktorenanalyse durch Spezifikation des Arguments rotate:

```
> REG12.FA = fa(REG12, nfactors=2, fm="pa", rotate="varimax", scores="regression", min.err=
0.000001, max.iter=10)
maximum iteration exceeded
> REG12.FA$scores
         PA1         PA2
A  0.50510897 -0.6315277
B -1.62233464  1.1495354
C  1.84951610 -1.1357798
D  0.01170971  0.5709496
E -0.33106890 -1.2509402
F -1.01112306 -1.2127235
G -0.12313343  1.3265148
H  1.19279348  1.3025502
I -0.95719644 -0.8636409
J  0.02907680  0.2086301
K  0.80999622  0.6658821
L -0.35334481 -0.1294501.
```

Das Argument scores ist bereits per Default auf regression gesetzt. Erneut liegen die Abweichungen zu den mit der Matrizenrechnung berechneten Faktorwerten im Tausendstel-Bereich.

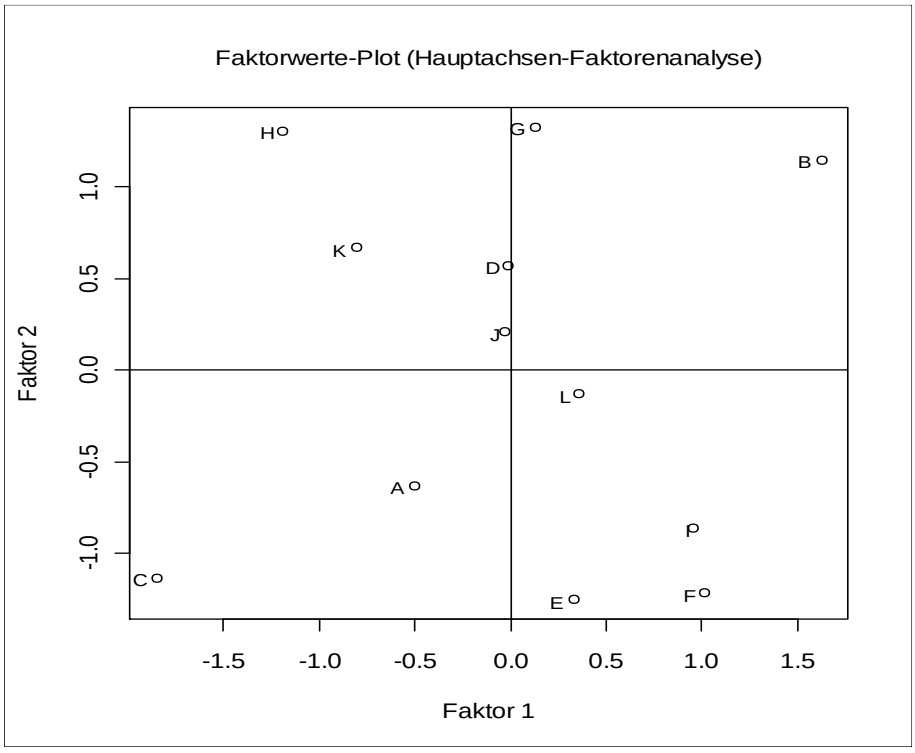
Da die Mittelwerte der beiden Faktoren wiederum gleich 0 und die Standardabweichungen approximativ gleich 1 sind,

```
> round(apply(F.FA,2,mean),3)
F1 F2
 0  0
> round(apply(F.FA,2,sd),3)
F1    F2
0.981 1.004,
```

ist auch hier die Interpretation der Faktorwerte eindeutig. Hierzu erstellen wir wieder ein Faktorwerte-Plot:

```
> plot(F.FA, main="Faktorwerte-Plot (Hauptachsen-Faktorenanalyse)", cex.main=1, font.main=1, xlab=
"Faktor 1", ylab="Faktor 2")
> text(F.FA, labels=rownames(F.FA), cex=0.8, adj=1.6)
> abline(h=0, v=0)
> box(which="figure").
```

Abbildung 3.5: Faktorwerte-Plot für die Hauptachsen-Faktorenanalyse



Aus Abbildung 3.5 gehen die Gebiete B und H erneut als Regionen mit vergleichbar hohen Attraktivitätswerten hervor. Doch stellt sich jetzt der verstädterte Raum G als attraktivste Region heraus, während der ländliche Raum K im Hinblick auf die Attraktivität zurückfällt. Keine nennenswerten Veränderungen ergeben sich dagegen bei den weniger attraktiven Regionen.

Unter Verwendung der berechneten Faktorwertematrizen lassen sich die standardisierten Beobachtungen der Variablen bis auf die Werte der Einzelrestfaktoren reproduzieren. Beispielhaft sei hier die reproduzierte standardisierte Beobachtungsmatrix $\hat{\mathbf{Z}}$ für die Hauptkomponentenlösung des Datensatzes `Regionen12` betrachtet:

```
> Zt.PCA = Arr.PCA%*%Ft.PCA
> Z.PCA = t(Zt.PCA)
> Z.PCA
        [,1]        [,2]        [,3]        [,4]        [,5]        [,6]
A -0.34253206 -0.40368598  0.49309817 -0.5272620  0.21072163 -0.4429885
B  1.34257627  1.48528784 -1.69445924  1.2473462 -0.97031569  0.9455384
C -1.59625069 -1.72904509  1.92400990 -1.1714635  1.20855802 -0.8234712
D  0.06104472  0.13920384 -0.25312191  0.6621560  0.06257474  0.6273670
E  0.48680289  0.35306536 -0.15870147 -1.1146126 -0.62794864 -1.1695248
F  0.99521227  0.86057967 -0.66539386 -1.1063537 -1.07717330 -1.2594072
G -0.04213084  0.03995888 -0.15950513  0.6921496  0.15932042  0.6761752
H -1.33808217 -1.22121880  1.05224382  0.9455774  1.35277803  1.1702174
I  1.11673397  1.01323430 -0.86351736 -0.8395740 -1.13788331 -1.0252994
J -0.17561197 -0.12952635  0.06255649  0.3838481  0.22331475  0.4042923
K -1.03761026 -0.89017371  0.67637564  1.2131970  1.13358540  1.3706931
L  0.52984787  0.48232005 -0.41358506 -0.3850086 -0.53753205 -0.4735923.
```

Die Diagonalelemente der reproduzierten Korrelationsmatrix

```
> Rrep.PCA = cor(Z, Z.PCA)
>Rrep.PCA
        [,1]        [,2]        [,3]        [,4]        [,5]        [,6]
x1  0.9594195  0.95273664 -0.91863427 -0.1568544 -0.9428561 -0.33259154
x2  0.9374315  0.94401027 -0.92969694 -0.0433716 -0.9006249 -0.22048652
x3 -0.9066561 -0.93248339  0.94686365 -0.1212220  0.8404358  0.05810457
x4 -0.1553761 -0.04385276 -0.12181062  0.9512234  0.3264428  0.93414310
x5 -0.8779269 -0.85235871  0.79294610  0.3066038  0.8933741  0.45939241
x6 -0.3347414 -0.22577970  0.05926686  0.9485999  0.4967158  0.96594548,

> r.PCA = diag(Rrep.PCA)
> round(r.PCA, 3)
[1] 0.959 0.944 0.947 0.951 0.893 0.966,
```

geben die Korrelationskoeffizienten gleicher Spalten von $\mathbf{Z}$ und $\hat{\mathbf{Z}}$ an. Ihre Quadrate sind Determinationskoeffizienten, die die Kommunalitäten der beobachteten Variablen spiegeln:

```
> h2.PCA = r.PCA^2
> round(h2.PCA, 3)
[1] 0.920 0.891 0.897 0.905 0.798 0.933.
```

Übungsaufgaben

3.6.1 Berechnen Sie den KQ-Schätzer der Faktorwertematrix für die Varimax-rotierte Zwei-Faktorenlösung der Hauptkomponentenmethode für den Datensatz documenta! Kennzeichnen Sie die Zeilen und Spalten der Faktorwertematrix mit Faktoren- und Variablenkürzeln!

3.6.2 Erstellen Sie ein Faktorwerte-Plot unter Verwendung der in Aufg. 3.6.1 bestimmten Faktorwertematrix!

3.6.3 Berechnen Sie den Regressionsschätzer mit der Varimax-rotierte Faktormatrix der

- Zwei-Faktoren-Lösung,
- Drei-Faktorenlösung

der Hauptachsen-Faktorenanalyse für den Datensatz `documenta`! Welche Veränderungen treten hierbei auf?

3.6.4 Bestimmung Sie die in den Aufg. 3.6.1 und 3.6.3 für eine Zwei-Faktorenlösung berechneten Faktorwertematrizen mit den R-Funktionen `principal` und `fa`!?

3.6.5 Berechnen Sie unter Verwendung der Zwei-Faktoren-Lösung der Hauptkomponentenmethode die reduzierte standardisierte Beobachtungsmatrix $\hat{\mathbf{Z}}$ der Documenta-Daten! Zeigen Sie, wie sich hiermit die Kommunalitäten der beobachteten Variablen erhalten lassen!

4. Varianzanalyse
4.1 Daten, Variablen und Faktoren

Das Modell der Varianzanalyse (ANOVA) geht von einer Beziehung zwischen einer quantitativen abhängigen Variablen und einer oder mehreren qualitativen unabhängigen Variablen (Faktoren) aus. Es wird geprüft, ob die Faktoren einzeln oder in einer Kombination einen substanziellen Einfluss auf die abhängige Variable haben. In diesem Fall unterscheiden sich die Werte, die die abhängige Variable annimmt, in den durch die Faktoren gebildeten Gruppen (Faktorstufen bzw. Zellen) signifikant voneinander.

Die Varianzanalyse wird hier an einem Datensatz Produktivitaet24 aufgezeigt, in dem die von Arbeitern produzierten Teile (Produktivität) nach der Produktionsmethode und der Arbeitergruppe untergliedert ist. Nach Einlesen der in Excel erstellten Datei Produktivitaet24.csv sind die Daten in dem data.frame-Objekt Prod24 verfügbar:

```
>Prod24 = read.csv2(file="Produktivitaet24.csv")
>class(Prod24)
[1] "data.frame"
>str(Prod24)
'data.frame':   24 obs. of  4 variables:
$ Arbeiter: num  1 2 3 4 5 6 7 8 9 10 ...
 $ PM     : num  1 1 1 1 1 1 2 2 2 2 ...
 $ AG     : num  1 1 2 2 3 3 1 1 2 2 ...
$ PROD    : num  69 77 61 67 65 69 79 83 62 64 ...
> head(Prod24)
Arbeiter PM AG PROD
1        1  1 1   69
2        2  1 1   77
3        3  1  2   61
4        4  1  2   67
5        5  1  3   65
6        6  1  3   69.
```

Der Datensatz Prod24, der aus 24 Zeilen (=Beobachtungen) und 4 Spalten (=Variablen) besteht,

```
>dim(Prod24)
[1] 24  4,
```

kann im R-Editor betrachtet und überprüft werden:

>fix(Prod24)

Dateneditor

	Arbeiter	PM	AG	PROD	var5	var6	var7	var8
1	1	1	1	69				
2	2	1	1	77				
3	3	1	2	61				
4	4	1	2	67				
5	5	1	3	65				
6	6	1	3	69				
7	7	2	1	79				
8	8	2	1	83				
9	9	2	2	62				
10	10	2	2	64				
11	11	2	3	72				
12	12	2	3	78				
13	13	3	1	80				
14	14	3	1	78				
15	15	3	2	74				
16	16	3	2	76				
17	17	3	3	80				
18	18	3	3	74				
19	19	4	1	75				
20	20	4	1	67				
21	21	4	2	70				
22	22	4	2	62				
23	23	4	3	70				
24	24	4	3	76				

Für die Varianzanalyse müssen die Variablen PM (Produktionsmethode) und AG (Arbeitsgruppe) als Faktoren definiert werden:

```
> Prod24$PM = as.factor(Prod24$PM)
> class(Prod24$PM)
[1] "factor"
> Prod24$PM
 [1] 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3 4 4 4 4 4 4
Levels: 1 2 3 4
> Prod24$AG = as.factor(Prod24$AG)
> class(Prod24$AG)
[1] "factor"
> Prod24$AG
 [1] 1 1 2 2 3 3 1 1 2 2 3 3 1 1 2 2 3 3 1 1 2 2 3 3
Levels: 1 2 3.
```

Außerdem soll die Variable Arbeiter, da sie die Arbeiter nur durchnummeriert, als qualitative Variable (character variable) definiert werden:

```
> Prod24$Arbeiter = as.character(Prod24$Arbeiter)
> class(Prod24$Arbeiter)
[1] "character"
```

```
> Prod24$Arbeiter
 [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14" "15" "16" "17" "18" "19" "20"
"21" "22" "23" "24".
```

Mit dem Befehl

```
>str(Prod24)
'data.frame':   24 obs. of  4 variables:
 $ Arbeiter: chr  "1" "2" "3" "4" ...
 $ PM     : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 2 2 2 2 ...
 $ AG     : Factor w/ 3 levels "1","2","3": 1 1 2 2 3 3 1 1 2 2 ...
 $ PROD   : num  69 77 61 67 65 69 79 83 62 64 ...
```

kann die Struktur des Datenobjekts Prod24 angezeigt werden.

Man erhält einen Zugriff auf die Variablen des Datensatzes Prod24 unter Verwendung der Kürzel mit Hilfe des attach-Befehls. Alternativ können die Variablen als separate R-Objekte definiert werden:

```
> Arbeiter = Prod24$Arbeiter
> PM = Prod24$PM
> AG = Prod24$AG
> PROD = Prod24$PROD.
```

Während man mit der Funktion summary einen Überblick über die Verteilung der Produktivität (PROD) erhält,

```
>summary(PROD)
Min. 1st Qu.Median    Mean 3rd Qu.    Max.
  61.00   67.00   73.00   72.00   77.25   83.00,
```

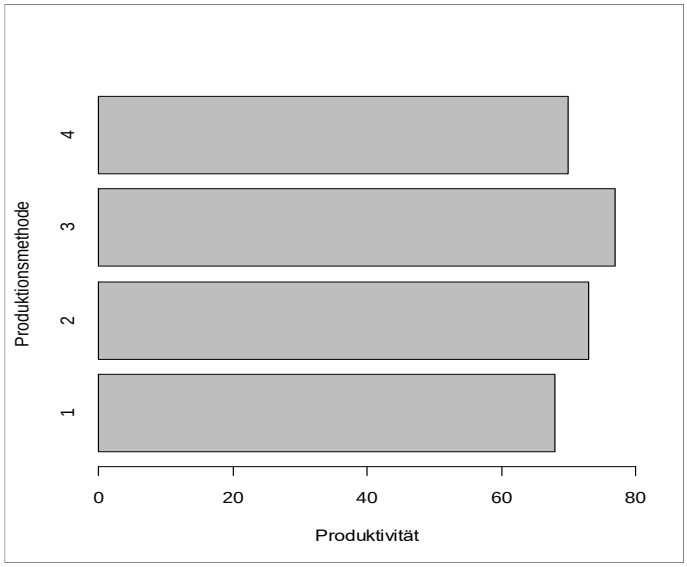
werden Mittelwerte dieser Variablen in den Stufen des Faktors Produktionsmethode (PM) mit der Funktion tapply bestimmt,

```
>PROD.mPM = tapply(PROD, PM, mean)
>PROD.mPM
 1   2   3   4
68 73 77 70,
```

die durch ein horizontales Balkendiagramm grafisch wiedergegeben werden können (Abbildung 4.1):

```
> barplot(PROD.mPM, xlim = c(0,80), xlab="Produktivität", ylab="Produktionsmethode", horiz = TRUE)
> box(which="figure").
```

Abbildung 4.1: Barplot der Produktivitätsmittelwerte in den Stufen des Faktors PM



Mit der Funktion boxplot können die Verteilungen der Produktivität (PROD) innerhalb der Stufen des Faktors Produktionsmethode (PM) transparent gemacht werden:

```
> boxplot(PROD~PM)
> box(which="figure").
```

Abbildung 4.2: Boxplots der Produktivität in den Stufen des Faktros PM

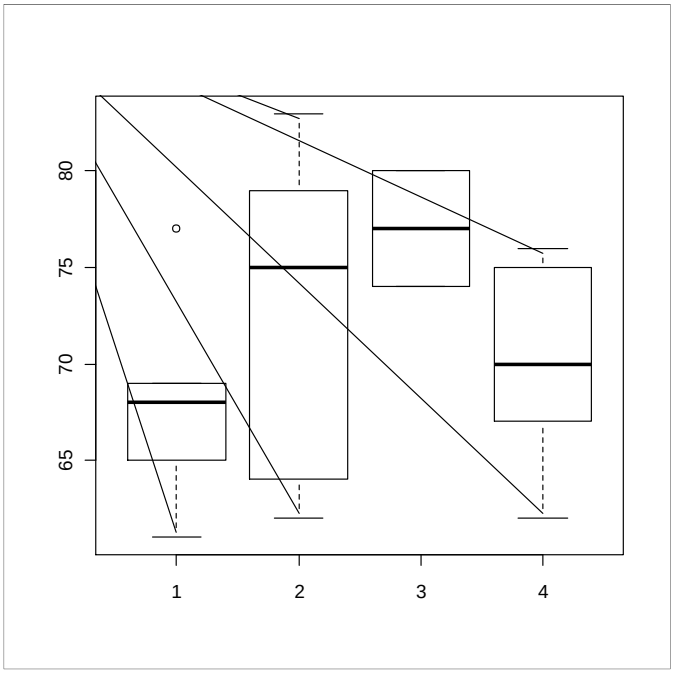


Abbildung 4.2 spiegelt die unterschiedliche zentrale Tendenz der Produktivität in den Stufen des Faktors Produktionsmethode anhand der durch fette Linien eingezeichneten Werte des Median wider. Klar erkennbar ist die große Streuung der Produktivitätswerte innerhalb der zweiten Produktionsmethode. Ein Ausreißer wird bei der ersten Produktionsmethode identifiziert.

Übungsaufgaben

4.1.1 Lesen Sie die csv-Datei Wanderungssaldo als data.frame-Objekt Saldo mit den Variablen Region, WS (= Wanderungssaldo), Attraktivität, Urbanität und Standort in R ein! Geben Sie den Datensatz in der R-Konsole aus und stellen Sie den Datentyp der Variablen fest!

4.1.2 Definieren Sie die Variablen des Datensatzes Saldo als eigenständige R-Objekte: Region als String-Variablen, WS als quantitative Variable, Attract (= Attraktivität), Urban (= Urbanität) und Stand (= Standort) als Faktoren! Erläutern Sie den Datentyp factor!

4.1.3 Berechnen Sie die Mittelwerte des Wanderungssaldos (WS) separat für die Stufen der Faktoren Attract und Urban und stellen Sie sie grafisch dar!

4.1.4 Erstellen Sie Boxplots des Wanderungssaldos (WS) für die Stufen des Faktors Attract und interpretieren Sie sie!

4.2 Einfaktorielle Varianzanalyse

Bei der einfachen Varianzanalyse untersuchen wir den Einfluss der Produktionsmethode (PM) auf die Produktivität der Arbeiter (PROD). Hierbei wird zunächst die Berechnung der Prüfgröße, des kritischen Wertes und des p-Wertes ausführlich nachvollzogen, ehe von der R-Funktion aov Gebrauch gemacht wird.

Mit dem Gesamtmittelwert der Produktivität,

```
> PROD.m = mean(PROD)
> PROD.m
[1] 72,
```

erhält man die totale Abweichungsquadratsumme

```
> QT = sum((PROD-PROD.m)^2)
> QT
[1] 954.
```

Zur Bestimmung der durch den Faktor Produktionsmethode (PM) erklärten Teil der totalen Abweichungsquadratsummme, QA, benötigt man die Mittelwerte der vier Stufen dieses Faktors, PROD.mPM, die bereits für den barplot gebildet worden sind.

Unter Verwendung der Größen

```
> N = length(PROD)          # Gesamtzahl der Beobachtungen
> N
[1] 24
> p = length(PROD.mPM)      # Anzahl der Stufen des Faktors A
> p
[1] 4
>nA = N/p                   # Anzahl der Beobachtungen je Stufe des Faktors A
>nA
[1] 6
```

ergibt sich für die erklärte Abweichungsquadratsumme QA der Wert

```
> QA = sum(nA*(PROD.mPM-PROD.m)^2)
> QA
[1] 276.
```

Die nicht-erklärte Abweichungsquadratsumme (Restabweichungsquadratsumme), QR, ergibt sich aus der Differenz der beiden berechneten Abweichungsquadratsummen QT und QA:

```
> QR = QT - QA
> QR
[1] 678.
```

Man erhält die mittleren Abweichungsquadratsummen MQA und MQR, indem man die Abweichungsquadratsummen QA und QR durch die jeweiligen Freiheitsgrade p-1 und N-p dividiert:

```
> MQA = QA/(p-1)
> MQA
[1] 92
> MQR = QR/(N-p)
> MQR
[1] 33.9.
```

Zur Überprüfung der Nullhypothese, dass die Produktionsmethode keinen Einfluss auf die Produktivität hat, wird als Teststatistik der F-Wert berechnet,

```
> F = MQA/MQR
```

```
> F
[1] 2.713864,
```

der bei ihrer Gültigkeit F-verteilt ist mit p-1 und N-p Freiheitsgraden.

Bei Wahl eines Signifikanzniveaus α von 5% ergibt sich für 3 und 20 Freiheitsgrade (FG) der kritische Wert

```
> qf95 = qf(0.95, p-1, N-p)    # 95%-Quantil der F-Verteilung mit p-1 und N-p FG
> qf95
[1] 3.098391.
```

Da F < qf95 ist, kann die Nullhypothese eines fehlenden Einflusses der Produktionsmethode (PM) auf die Produktivität (PROD) auf der Basis des Modells der einfaktoriellen Varianzanalyse nicht statistisch gesichert nachgewiesen werden.

Alternativ kann der F-Test unter Verwendung des p-Wertes durchgeführt werden:

```
>pval = 1 - pf(F, p-1, N-p)    # Wahrscheinlichkeit > F für eine F-Verteilung mit p-1 und N-p FG
>pval
[1] 0.07212854.
```

Wegen α < pval kann auch hiermit die Nullhypothese nicht verworfen werden.

Mit der R-Funktion aov werden die ausführlich ermittelten Ergebnisse der einfaktoriellen Varianzanalyse der Produktivität (PROD) in Abhängigkeit von der Produktionsmethode (PM) bestätigt:

```
> PROD.aovPM = aov(PROD~PM)
> summary(PROD.aovPM)
            Df    Sum    SqMean   SqFvalue   Pr(>F)
PM           3    276     92.0     2.7139    0.07213 .
Residuals   20    678     33.9
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.
```

Übungsaufgaben

4.2.1 Berechnen Sie für den Datensatz Saldo die für eine einfaktorielle Varianzanalyse des Wanderungssaldos (WS) in Abhängigkeit vom Faktor Attraktivität (Attract) benötigten Abweichungsquadratsummen Q_T, Q_A und Q_R!

4.2.2 Prüfen Sie unter Verwendung der in Aufg. 4.2.2 berechneten Abweichungsquadratsummen mit Hilfe

- der Teststatistik,
- des p-Werts,

ob der Faktor Attraktivität (Attract) einen signifikanten Einfluss auf den Wanderungssaldo (WS) hat (α=0,05)!

4.2.3 Testen Sie den Einfluss der Attraktivität (Attract) auf die Wanderungssaldo (WS) mit der R-Funktion aov (α=0,05)! Interpretieren Sie den mit der summary-Funktion ausgegebenen Output des Tests!

4.3 Zweifaktorielle Varianzanalyse ohne Interaktion

Im Modell der zweifaktoriellen Varianzanalyse wird die Produktivität (PROD) in Abhängigkeit von der Produktionsmethode (PM) und der Arbeitsgruppe (AG) untersucht. Hierbei lassen wir zunächst eine mögliche Wechselwirkung zwischen der Produktionsmethode und der Arbeitsgruppe außer Acht.

Als erstes betrachten wir die gemeinsame Häufigkeitsverteilung der Faktoren PM (Faktor A) und AG (Faktor B). Sie wird mit der Funktion tapply erstellt, in der die Zellen durch eine Liste der Faktoren definiert werden:

```
>hjk = tapply(PROD, list(PM, AG), length)
>class(hjk)
[1] "matrix"
>rownames(hjk) = c("A1", "A2", "A3", "A4")
>colnames(hjk) = c("B1", "B2", "B3")
>hjk
   B1 B2 B3
A1  2  2  2
A2  2  2  2
A3  2  2  2
A4  2  2  2.
```

Sie zeigt, dass in den Kombinationen der Faktorstufen (=Zellen) Aj*Bk jeweils n=2 Beobachtungen vorliegen, so dass es sich um ein balanciertes Design handelt.

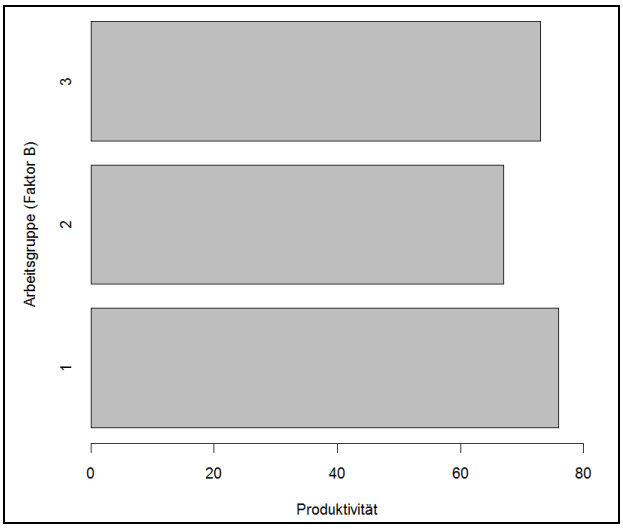
Den Gesamtmittelwert PROD.m (= 72) und die Mittelwerte der Stufen des Faktors A, PROD.mPM (68, 73, 77, 70), sind bereits in der einfaktoriellen Varianzanalyse berechnet worden. Jetzt werden zusätzlich noch die Mittelwerte der Stufen des Faktors B, PROD.mAG, bestimmt,

```
>PROD.mAG = tapply(PROD, AG, mean)
>PROD.mAG
 1  2  3
76 67 73,
```

und in Form eines Balkendiagramms dargestellt (Abbildung 4.3):

```
>barplot(PROD.mAG, xlim = c(0,80), xlab="Produktivität", ylab="Arbeitsgruppe", horiz = TRUE).
```

Abbildung 4.3: Barplot der Produktivitätsmittelwerte in den Stufen des Faktors AG



In den Stufen des Faktors Arbeitsgruppe (Faktor B) zeigen sich wie bei der Untergliederung nach der Produktionsmethode unterschiedliche durchschnittliche Produktivitätswerte. Zu testen ist, ob diese Produktivitätsunterschiede auch signifikant sind. Außerdem ist zu prüfen, ob die Produktionsunterschiede im Hinblick auf die Produktionsmethode (Faktor A) auch bei Einbeziehung der Arbeitsgruppe (Faktor B) nicht-signifikant bleiben.

Hierzu werden die Abweichungsquadratsummen QT, QA, QB und QR benötigt, wobei die beiden ersteren Größen aus der einfaktoriellen Varianzanalyse entnommen werden können. Die durch den Faktor B (Faktor AG) erklärte Abweichungsquadratsumme ergibt sich wie folgt:

```
> q = length(PROD.mAG)      # Anzahl der Stufen des Faktors B
> q
[1] 3
>nB = N/q                   # Anzahl der Beobachtungen je Stufe des Faktors B
>nB
[1] 8
```

```
> QB = sum(nB*(PROD.mAG-PROD.m)^2)
> QB
[1] 336.
```

Die unerklärte Abweichungsquadratsumme (Restabweichungsquadratsumme, QR, wird wieder residual bestimmt:

```
> QR = QT - QA - QB
> QR
[1] 342.
```

Da die mittlere Abweichungsquadratsumme MQA (= 92) bereits aus der einfaktoriellen Varianzanalyse bekannt ist, müssen nur noch die Größen MQB und MQR berechnet werden:

```
> MQB = QB/(q-1)
> MQB
[1] 168
> MQR = QR/(N-p-q+1)
> MQR
[1] 19.
```

Die Teststatistiken FA und FB nehmen die Werte

```
> FA = MQA/MQR
> FA
[1] 4.842105
> FB = MQB/MQR
> FB
[1] 8.842105.
```

an. Sie sind beide wegen

```
> qfA95 = qf(0.95, p-1, N-p-q+1)    # 95%-Quantil der F-Verteilung mit p-1 und N-p-q+1 FG
> qfA95
[1] 3.159908
> qfB95 = qf(0.95, q-1, N-p-q+1)    # 95%-Quantil der F-Verteilung mit q-1 und N-p-q+1 FG
> qfB95
[1] 3.554557
```

auf einem Niveau von 5% signifikant. Im Modell der zweifachen Varianzanalyse tragen damit beide Faktoren A und B, die Produktionsmethode und die Arbeitsgruppe, statistisch gesichert zur Erklärung der Produktivitätsunterschiede bei.

Das Testergebnis bestätigt sich bei Verwendung der jeweiligen p-Werte,

```
>pvalA = 1 - pf(FA, p-1, N-p-q+1)  # Wahrscheinlichkeit > F für eine F-Verteilung mit p-1 und N-p-q+1 FG
>pvalA
[1] 0.01213908
>pvalB = 1 - pf(FB, q-1, N-p-q+1)  # Wahrscheinlichkeit > F für eine F-Verteilung mit q-1 und N-p-q+1 FG
>pvalB
[1] 0.002114305,
```

die kleiner als das vorgegebene Signifikanzniveau α von 0,05 sind.

Kompakt lassen sich die ausführlich ermittelten Ergebnisse der zweifaktoriellen Varianzanalyse ohne Interaktion wiederum mit der R-Funktion aov erzielen:

```
>PROD.aovPMAG = aov(PROD~PM+AG)
>summary(PROD.aovPMAG)
            Df  Sum Sq  Mean Sq  F value   Pr(>F)
PM           3    276       92    4.8421  0.012139 *
AG           2    336      168    8.8421  0.002114 **
Residuals   18    342       19
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.
```

Übungsaufgaben

4.3.1 Geben Sie die gemeinsamen Häufigkeitsverteilung der Faktoren Attraktivität (Attract) und Urbanität (Urban) des Datensatzes Saldo an und interpretieren Sie sie!

4.3.2 Berechnen Sie für den Datensatz Saldo die für eine zweifaktorielle Varianzanalyse des Wanderungssaldos (WS) in Abhängigkeit von den Faktoren Attraktivität (Attract) und Urbanität (Urban) benötigten Abweichungsquadratsummen Q_T, Q_A, Q_B und Q_R!

4.3.3 Prüfen Sie unter Verwendung der in Aufg. 4.3.2 berechneten Abweichungsquadratsummen mit Hilfe

- der Teststatistik,
- des p-Werts,

ob der Faktor Attraktivität (Attract) einen signifikanten Einfluss auf die Wanderungssaldos (WS) hat (α=0,05)!

4.3.4 Prüfen Sie unter Verwendung der in Aufg. 4.3.2 berechneten Abweichungsquadratsummen, ob

- der Teststatistik,
- des p-Werts,

ob der Faktor Urbanität (Urban) einen signifikanten Einfluss auf die Wanderungssaldos (WS) hat (α=0,05)!

4.3.4 Testen Sie die separaten Einflüsse der Attraktivität (Attract) und Urbanität (Urban) auf den Wanderungssaldo (WS) mit der R-Funktion aov (α=0,05)! Interpretieren Sie den mit der summary-Funktion ausgegebenen Output des Tests!

4.4 Zweifaktorielle Varianzanalyse mit Interaktion

Da die totale Abweichungsquadratsumme QT und die durch die Faktoren A und B erklärten Abweichungsquadratsummen QA und QB durch die Einbeziehung der Interaktion AxB unverändert bleiben, muss bei den systematischen Einflüssen im saturierten Modell nur noch die auf die Wechselwirkung (Interaktion) zurückführbare Abweichungsquadratsumme QAB berechnet werden:

Sie wird hier residual bestimmt, da die Restabweichungsquadratsumme QR einfacher aus den Residuen, d.h. den Differenzen zwischen den Beobachtungswerten der Produktivität und den Zellenmittelwerten PROD.mPMAG,

```
> PROD.mPMAG = tapply(PROD, list(PM, AG), mean)
> PROD.mPMAG
   1  2  3
1 73 64 67
2 81 63 75
3 79 75 77
4 71 66 73,
```

berechnet werden kann. Bei bereits ermittelter Anzahl der Faktorstufen p (= 4) und q (= 3) benötigt man hierzu noch die Anzahl der Beobachtungen je Zelle:

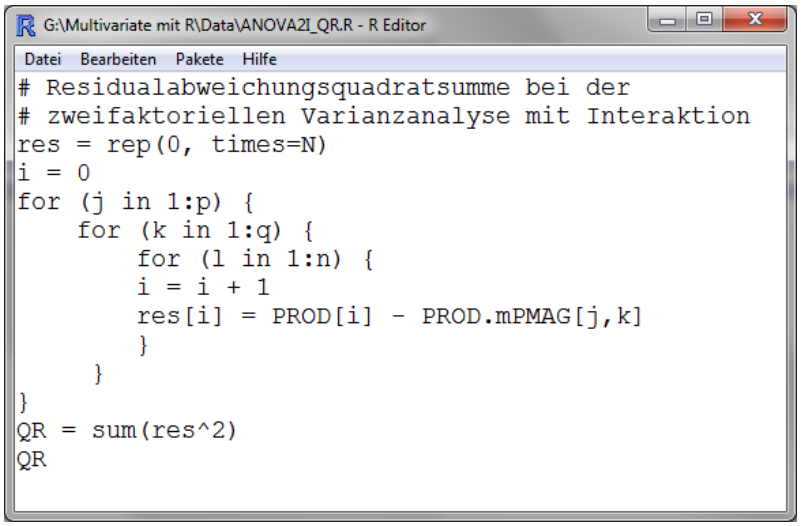
```
> n = hjk[1,1]        oder        n = N/(p*q)
> n
[1] 2.
```

Wir berechnen die Quadratsumme der Residuen unter Verwendung der for-Schleife:

```
# Residualabweichungsquadratsumme bei der
# zweifaktoriellen Varianzanalyse mit Interaktion
res = rep(0, times=N)
i = 0
for (j in 1:p) {
  for (k in 1:q) {
    for (l in 1:n) {
    i = i + 1
    res[i] = PROD[i] - PROD.mPMAG[j,k]
    }
  }
}
QR = sum(res^2)
QR.
```

Die Befehle können in ein Skriptfenster eingegeben werden, das in dem R-Menü mit – Datei – Neues Skript geöffnet werden kann. Nach Speicherung des Skripts als R-Datei ANOVA2I_QR.R ist es nach Wahl der Menüpunkte – Datei – Öffne Skript … jederzeit in R zur Bearbeitung oder Ausführung verfügbar.

Um das Skript



auszuführen, wählen wir in dem Fenster die Menüpunkte – Bearbeiten – Alles auswählen und anschließend – Bearbeiten – Alles ausführen. Der Inhalt des Skriptfensters wird daraufhin – in roter Schrift - im R-Commander zusammen mit dem berechneten QR-Wert von 190 ausgegeben.

Damit ergibt sich die Abweichungsquadratsumme der Interaktion, QAB, der Wert

```
> QAB = QT - QA - QB - QR
```

```
> QAB
[1] 152.
```

Auch die mittleren Abweichungsquadratsummen können für die Haupteffekte A und B aus der zweifaktoriellen Varianzanalyse ohne Interaktion übernommen werden. Für die Interaktion AxB und die Residuen erhält man:

```
> MQAB = QAB/((p-1)*(q-1))
> MQAB
[1] 25.33333

> MQR = QR/(N-p*q)
> MQR
[1] 15.83333.
```

Mit der Restabweichungsquadratsumme MQR als Bezugsgröße ergeben sich die Prüfgrößen

```
> FA = MQA/MQR
> FA
[1] 5.810526

> FB = MQB/MQR
> FB
[1] 10.61053

> FAB = MQAB/MQR
> FAB
[1] 1.6
```

der F-Tests. Während die beiden Haupteffekte A Produktionsmethode) und B (Arbeitsgruppe) auf dem 5%-Niveau signifikant sind,

```
> qfA95 = qf(0.95, p-1, N-p*q)    # 95%-Quantil der F-Verteilung mit p-1 und N-p·q FG
> qfA95
[1] 3.490295

> qfB95 = qf(0.95, q-1, N-p*q)    # 95%-Quantil der F-Verteilung mit q-1 und N-p·q FG
> qfB95
[1] 3.885294,
```

gilt dies für die Interaktion AxB nicht:

```
> qfAB95 = qf(0.95, (p-1)*(q-1), N-p*q)    # 95%-Quantil der F-Verteilung mit (p-1)(q-1) und N-p·q FG
> qfAB95
[1] 2.996120.
```

Für eine Wechselwirkung zwischen der Produktionsmethode und der Arbeitsgruppe gibt es somit keine statistisch gesicherte Evidenz.

Zu denselben Testentscheidungen gelangt man auch mit Hilfe der p-Werte:

```
>pvalA = 1 - pf(FA, p-1, N-p*q)
>pvalA
[1] 0.01085874
>pvalB = 1 - pf(FB, q-1, N-p*q)
>pvalB
[1] 0.002221300
>pvalAB = 1 - pf(FAB, (p-1)*(q-1), N-p*q)
>pvalAB
[1] 0.2297649.
```

Auch hier können die Ergebnisse wieder mit der Funktion aov erhalten werden. Hierzu sind die beiden Faktoren PM und AG bei ihrer Spezifikation multiplikativ zu verknüpfen:

```
> PROD.aovPMIAG = aov(PROD~PM*AG)
> summary(PROD.aovPMIAG)
            Df  Sum Sq  Mean Sq  F value    Pr(>F)
PM           3    276    92.000   5.8105  0.010859 *
AG           2    336   168.000  10.6105  0.002221 **
PM:AG        6    152    25.333   1.6000  0.229765
Residuals   12    190    15.833
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.
```

Übungsaufgaben

4.4.1 Bestimmen Sie die Mittelwerte des Wanderungssaldos (WS) für die Zellen, die aus den Faktoren Attraktivität (Attract) und Urbanität (Urban) des Datensatzes Saldo gebildet werden!

4.4.2 Berechnen Sie für den Datensatz Saldo die

- Abweichungsquadratsumme der Residuen,
- Abweichungsquadratsumme der Interaktion

des saturierten Modells der zweifaktoriellen Varianzanalyse unter Verwendung der Faktoren Attraktivität (Attract) und Urbanität (Urban)!

4.4.3 Testen Sie die Faktoren Attraktivität (Attract) und Urbanität (Urban) sowie ihre Interaktion unter Verwendung der in den Aufg. 4.3.2 und 4.4.2 berechneten Abweichungsquadratsummen auf der Basis

- der Prüfgröße,
- des p-Werts,

auf Signifikanz (α=0,05)!

4.4.4 Führen Sie die in Aufg. 4.4.3 durchgeführten Signifikanztests unter Verwendung der R-Funktion aov durch (α=0,05)! Interpretieren Sie den mit der summary-Funktion ausgegebenen Output des Tests!

4.5 Nonorthogonale Varianzanalyse

Eine nonorthogonale Varianzanalyse wird mit einem geringfügig modifizierten Datensatz von Produktivitätsdaten durchgeführt. Hierzu lesen wir die um zwei Arbeiter erweiterten Daten aus der in Excel erstellten Datei Produktivitaet26.csv in R ein:

```
> Prod26 = read.csv2(file="Produktivitaet26.csv").
```

Die ersten Zeilen des Datensatzes Prod26 lauten:

```
> head(Prod26)
  Arbeiter PM AG PROD
1        1  1  1   69
2        2  1  1   77
3        3  1  1   70
4        4  1  1   74
5        5  1  2   61
6        6  1  2   67.
```

In dem modifizierten Datensatz werden die Variablen PM (Produktionsmethode) und AG (Arbeitsgruppe) für die Varianzanalyse erneut als Faktoren definiert:

```
> PM26 = as.factor(Prod26$PM)
> AG26 = as.factor(Prod26$AG).
```

Außerdem soll die Variable Arbeiter wiederum als qualitative Variable (character variable) definiert werden:

```
> Arbeiter26 = as.character(Prod26$Arbeiter).
```

Die Variable PROD wird als quantitative Größe übernommen:

```
.> PROD26 = Prod26$PROD.
```

Anhand der gemeinsamen Häufigkeitsverteilung der Faktoren PM26 (Faktor A) und AG26 (Faktor B) erkennt man das unbalancierte Design des Datensatzes Prod26:

```
> hjk = tapply(PROD26, list(PM26, AG26), length)
> hjk
   1  2  3
1  4  2  2
2  2  2  2
3  2  2  2
4  2  2  2.
```

Nicht alle Kombinationen der Faktorstufen (=Zellen) Aj*Bk sind jetzt gleich stark besetzt. Die Zelle A1xB1 ist mit 4 Beobachtungen doppelt so stark besetzt wie die übrigen Zellen.

Die R-Funktion aov ist für balancierte Designs konzipiert. Im Falle eines unbalancierten Designs führt sie die nonorthogonale Varianzanalyse automatisch unter Verwendung des hierarchischen Ansatzes (Typ I) aus. Bei dieser Anpassungsmethode hängt das Ergebnis davon ab, welcher der beiden Faktoren den anderen Faktor untergliedert.

Wird der Faktor A (Produktionsmethode) nach den Stufen des Faktors B (Arbeitsgruppe) untergliedert, führt die hierarchische Varianzanalyse zu dem folgenden Ergebnis:

```
> PROD26.aovNON1a = aov(PROD26~PM26*AG26)
> summary(PROD26.aovNON1a)
            Df Sum Sq Mean Sq  F value     Pr(>F)
PM26         3    252   84.000   5.9095  0.0080109 **
AG26         2    360  180.000  12.6633  0.0007246 ***
PM26:AG26    6    151   25.167   1.7705  0.1773270
Residuals   14    199   14.214
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.
```

Im umgekehrten Fall einer Untergliederung des Faktors B (Arbeitsgruppe) nach den Stufen des Faktors A (Produktionsmethode) ergibt sich ein davon abweichendes Ergebnis:

```
> PROD26.aovNON1b = aov(PROD26~AG26*PM26)
> summary(PROD26.aovNON1b)
            Df  Sum Sq  Mean Sq  F value   Pr(>F)
AG26         2   310.4  155.200  10.9186  0.001389 **
PM26         3   301.6  100.533   7.0727  0.003978 **
AG26:PM26    6   151.0   25.167   1.7705  0.177327
Residuals   14   199.0   14.214
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.
```

In beiden Fällen wird die totale Abweichungsquadratsumme Q_T von 962 exakt auf die Haupteffekte A und B, die Interaktion AxB und die Zufallsfaktoren aufgeteilt. Dabei bleiben die Abweichungsquadratsummen der Residuen und Interaktion (Q_R und Q_{AxB}) unverändert. Dagegen steigen (fallen) die den Faktoren A und B zuzurechnenden Haupteffekte, wenn einer der beiden Faktoren dem anderen Faktor untergeordnet (übergeordnet) wird.

Für die R-Funktion Anova benötigt man das Zusatzpaket car:

```
> library(car).
```

Der Typ II-Test (experimenteller Ansatz) verkörpert das Prinzip der Marginalität. Jeder Faktor wird nach allen anderen getestet, wobei die Interaktionen zunächst ignoriert werden:

```
> PROD26.AnovaNON2 = Anova(lm(PROD26~PM26*AG26), type="II")
> PROD26. AnovaNON2
Anova Table (Type II tests)

Response: PROD26
            Sum Sq  Df  F value    Pr(>F)
PM26        301.6    3   7.0727  0.0039783 **
AG26        360.0    2  12.6633  0.0007246 ***
PM26:AG26   151.0    6   1.7705  0.1773270
Residuals   199.0   14
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.
```

Wie bei der orthogonalen Varianzanalyse erweisen sich die beiden Haupteffekte A und B als signifikant. Jedoch zeigen die niedrigeren p-Werte eine höhere Signifikanz an. Der Interaktionseffekt bleibt dagegen nach wie vor nicht signifikant.

Der Typ III-Test verletzt das Prinzip der Marginalität. Er muss sehr sorgfältig in einer Anwendung eingesetzt werden, da der Hypothesentest auf dieser Basis aufgrund der Verletzung der Marginalitätsbedingung häufig nicht sinnvoll ist, was auch hier der Fall ist. Aus diesem Grund wird hierauf seine Anwendung verzichtet.

Übungsaufgaben

4.5.1 Bilden Sie ein `data.frame`-Objekt `Saldo22` durch Ausschluss der beiden Regionen W und X aus dem Datensatz `Saldo`! Definieren Sie die Variablen des neuen Datensatzes `WS22`, `Attract22` und `Urban22` analog zum Datensatz `Saldo`!

4.5.2 Geben Sie die gemeinsame Häufigkeitsverteilung der Faktoren `Attract22` und `Urban22` an und bestimmen Sie bestimmen Sie die Mittelwerte des Wanderungssaldos (`WS`) der Faktorkombinationen!

4.5.3 Testen Sie die Effekte der Faktoren `Attract22` und `Urban22` auf den Wanderungssaldos (`WS`) im saturierten varianzanalytischen Modell unter Verwendung des hierarchischen Ansatzes auf Signifikanz (α=0,05) und interpretieren Sie die Testergebnisse!

4.5.4 Testen Sie die Effekte der Faktoren `Attract22` und `Urban22` auf den Wanderungssaldos (`WS`) im saturierten varianzanalytischen Modell unter Verwendung des experimentellen Ansatzes auf Signifikanz (α=0,05) und interpretieren Sie die Testergebnisse!

5. Clusteranalyse

Die Clusteranalyse besteht im Kern aus zwei Arbeitsschritten. Im ersten Arbeitsschritt wird die Ähnlichkeit der Objektpaare gemessen. Die Wahl eines Ähnlichkeitsmaßes hängt von dem Skalenniveau der verwendeten Variablen ab. Während bei quantitativen Merkmalen Distanzmaße bevorzugt werden, wird bei qualitativen Variablen die Ähnlichkeit in der Regel anhand übereinstimmender Merkmalskategorien bestimmt. Im zweiten Schritt erfolgt eine Gruppierung der Objekte (statistische Einheiten) aufgrund ihrer Ähnlichkeit. Hierzu werden hierarchische Verfahren eingesetzt, wenn die Gruppenzahl (Clusterzahl) nicht bekannt ist. Bei bekannter Gruppenzahl erfolgt die Gruppierung (Clusterung) mit Hilfe von Partitionsverfahren. Sie können auch eingesetzt werden, um eine mit hierarchischen Verfahren ermittelte Clusterlösung zu optimieren.

5.1 Ähnlichkeitsmessung bei quantitativen Merkmalen

Anhand des bereits verwendeten Datensatzes Regionen12 soll die Durchführung einer Clusteranalyse in R für quantitative Variablen illustriert werden. Mit Hilfe der Clusteranalyse werden die12 Regionen aufgrund der sechs Merkmale in homogene Gruppen (Cluster) eingeteilt, die sich untereinander möglichst deutlich unterscheiden, d.h. heterogen sein, sollen. Dies kann zum Beispiel im Rahmen der Strukturpolitik von Bedeutung sein, wenn förderungsrelevante Regionen zu identifizieren sind.

Bei metrisch skalierten Variablen erfolgt die Messung der paarweisen Ähnlichkeit der Objekte in der Regel mit Hilfe von Distanzmaßen. Je größer die Distanz zweier Beobachtungen in einem m-dimensionalen reellen Zahlenraum ist, umso unähnlicher sind die durch sie repräsentierten Objekte.

Die Distanz kann in R mit der Funktion dist berechnet werden, wozu die Datenmatrix REG12 herangezogen wird, die unter Ausschluss der Spalte Regionen aus dem Datensatz Regionen12 erzeugt worden ist:

```
> class(REG12)
[1] "matrix"
> str(REG12)
 num [1:12, 1:6] 212.4 623.7 93.1 236.8 412 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:12] "A" "B" "C" "D" ...
  ..$ : chr [1:6] "x1" "x2" "x3" "x4" ....
```

Als Zeilenbezeichnungen sind hierbei die Namen der Regionen verwendet worden.

Da die Distanzmaße durch die Streuung der Variablen stark beeinflusst werden, wird zu ihrer Berechnung die standardisierte Datenmatrix Z herangezogen:

```
> Z = scale(REG12)
> Z
          x1          x2          x3           x4          x5          x6
A -0.65651480 -1.2449654 -0.07308063 -0.450043283 -0.4496692 -0.4698356
B  1.70883627  1.2541546 -1.24237066  1.541707907 -1.3055911  0.8663443
C -1.34259888 -1.6530692  2.44820101 -0.955412988  1.0016768 -0.8609127
D -0.51619249  0.2993362 -0.27405235  0.917427683 -0.3380272  0.4100878
E  0.49136771  0.2802707 -0.23751204 -1.054505087 -0.5985251 -1.2519898
F  1.38103415  1.0222775 -0.74907643 -1.312144544 -0.3752412 -1.2194000
G  0.03071947 -0.1778154 -0.51156439  0.005780372  0.2573968  1.2900112
H -1.23735714 -0.9883547  1.11447956  0.620151386  1.5971008  1.4529599
I  0.93418811  1.2747659 -0.82215705 -0.806774839 -1.0823071 -1.0890410
J -0.22519622  0.3111877 -0.25578219  0.184146151  1.0388908  0.3123185
K -0.92220704 -0.9404335  0.71253611  1.630890796  1.2621748  0.9315238
L  0.35392085  0.5626455 -0.10962094 -0.321223554 -1.0078791 -0.3720664
attr(,"scaled:center")
        x1           x2           x3           x4           x5           x6
3.265583e+02 2.253208e+04 1.020000e+01 5.754167e+01 9.608333e+00 7.416667e-01
attr(,"scaled:scale")
       x1          x2          x3         x4         x5         x6
 173.885393 1940.683078    5.473407   10.091622   2.687161   3.068449.
```

Unter den standardisierten Daten sind die Mittelwerte und Standardabweichungen der Variablen ausgewiesen.

Bei Anwendung der dist-Funktion kann das gewünschte Distanzmaß als Argument gewählt werden. Voreinstellung ist die euklidische Distanz. Die unter Verwendung aller sechs Merkmale bestimmten euklidischen Distanzen der zwölf betrachteten Regionen werden in folgender Form ausgewiesen:

```
> ED = dist(Z)
> ED
        A        B        C        D        E        F        G        H        I        J        K
B 4.437655
C 3.083645 6.776875
D 2.258684 2.886892 4.338580
E 2.161091 3.894615 4.127675 2.781323
F 3.325593 3.710110 5.191331 3.460545 1.311897
G 2.368153 3.222141 4.350507 1.594708 2.969128 3.428452
H 3.293213 5.374258 3.228417 3.004484 4.622530 5.397857 2.666050
I 3.217359 3.188392 5.372927 3.021539 1.359738 1.020342 3.337313 5.409713
J 2.422973 3.639992 3.885212 1.590068 2.679030 3.148616 1.403022 2.528569 3.168383
K 3.163073 4.708066 3.696859 2.452918 4.453335 5.221353 2.601375 1.291836 5.121532 2.348693
L 2.151168 2.947865 4.359989 1.856930 1.262743 1.943002 2.298838 4.176005 1.451511 2.309333 3.902170.
```

Da die Eigendistanzen der Objekte stets gleich 0 sind, bleiben die Hauptdiagonalelemente der Distanzmatrix ED in der Voreinstellung leer. Aufgrund der Symmetrieeigenschaften ist bei einem Aufruf der Funktion dist außerdem die Ausgabe einer unteren Dreiecksmatrix voreingestellt. Mit einem Wert der euklidischen Distanz von 1,020 erweisen sich die Regionen F und I auf der Basis der ausgewählten Merkmale als am ähnlichsten. Die beiden unähnlichsten Regionen sind dagegen B und C mit einer euklidischen Distanz von 6,777.

Vor allem für varianzorientierte Klasssifikationsverfahren ist die quadrierte euklidische Distanz von Bedeutung. Man erhält die aus diesem Distanzmaß bestehende Distanzmatrix QED durch elementweise Quadrierung von ED:

```
> QED = ED^2
> QED
        A         B         C         D         E         F         G         H         I         J         K
B 19.692778
C  9.508866 45.926041
D  5.101653  8.334147 18.823279
E  4.670315 15.168024 17.037698  7.735756
F 11.059571 13.764915 26.949913 11.975369  1.721074
G  5.608146 10.382190 18.926907  2.543094  8.815721 11.754286
H 10.845253 28.882652 10.422673  9.026923 21.367782 29.136864  7.107822
I 10.351401 10.165843 28.868344  9.129700  1.848888  1.041097 11.137658 29.264999
J  5.870800 13.249539 15.094870  2.528317  7.177202  9.913781  1.968472  6.393659 10.038649
K 10.005030 22.165890 13.666767 6.016805 19.832191 27.262527 6.767153 1.668840 26.230086  5.516361
L 4.627525 8.689910 19.009506 3.448190 1.594521 3.775256 5.284657 17.439018 2.106885 5.333019 15.226934.
```

Die Rangfolge der ähnlichen Objektpaare bleibt hierbei unverändert. Allerdings ändern sich hierdurch die Abstandsverhältnisse, was durchaus einen Einfluss auf eine exploratorisch zu ermittelnde Gruppenzahl haben kann. Auch wenn die Gruppenzahl unverändert bleibt kann sich eine andere Gruppierung der Objekte ergeben.

Ein weiteres Distanzmaß stellt die City-Block-Distanz dar, die mit dem Argument method = „manhattan“ angefordert werden kann:

```
> CBD = dist(Z, method="manhattan")
> CBD
        A         B         C         D         E         F         G         H         I         J         K
B 10.217614
C  5.963262 16.180876
D  4.244632  6.196266  9.984610
E  4.373022  8.617824  8.543391  4.957674
F  6.666881  6.922920 10.688394  6.991466  2.656751
G  5.115604  7.363475  9.664735  3.648570  5.650995  7.247894
H  7.064773 11.956417  6.588539  6.672664 10.924574 13.069577  5.821684
I  6.468086  5.742625 10.935731  6.941526  2.916422  2.115210  7.197954 13.019638
```

J 5.075077 8.119657 8.135647 2.529087 5.206126 7.252756 2.938253 5.816822 7.202816
K 6.549978 9.502667 7.507931 5.467474 10.313937 12.458940 5.928021 2.632116 12.409000 5.206184
L 3.639386 6.578228 9.602648 3.988511 2.570272 4.597093 4.719963 9.737760 3.281878 4.213261 9.127122.

Die Rangfolge der Ähnlichkeit zwischen den Objektpaaren verändert sich bei der Verwendung der City-Block-Metrik gegenüber der euklidischen Distanz. Zwar erweisen sich auf der Grundlage der City-Block-Distanz nach wie vor die Regionen F und I am ähnlichsten, aber den Rang zwei nehmen jetzt z. B. die Regionen D und J ein, während hier vorher die Regionen E und L platziert waren. In der CBD-Distanzmatrix gehen die Merkmalswertdifferenzen proportional zu ihrem Ausmaß in die Objektdistanz ein. Dagegen besitzen größere Merkmalswertdifferenzen bei der euklidischen Distanz einen überproportionalen Einfluss.

Übungsaufgaben

5.1.1 Lesen Sie die csv-Datei Kaeufer30 als data.frame-Objekt Kaeufer30 in R ein! Geben Sie den Datensatz in der R-Konsole aus und stellen Sie den Datentyp der Variablen fest!

5.1.2 Selektieren Sie die Daten der elf Kunden 2, 11, 13, 15 – 17, 21, 22, 27 – 29, um das Matrixobjekt Kaeufer11 unter Verwendung der beiden quantitativen Variablen Einkommen und Alter des Datensatzes Kaeufer30 zu definieren! Standardisieren Sie die beiden Variablen Einkommen und Alter für den kompletten Datensatz Käufer30 und erstellen Sie hiermit die standardisierte Beobachtungsmatrix Z11 der elf ausgewählten Kunden!

5.1.3 Stellen Sie die standardisierten Werte der elf in Aufg. 5.1.2 ausgewählten Kunden der Variablen Einkommen und Alter unter Verwendung ihrer Labels in einem Koordinatensystem dar! Welche Kundengruppen lassen sich hierin identifizieren?

5.1.4 Bestimmen Sie für die in Aufg. 5.1.2 ausgewählten elf Kunden die Distanzmatrix mit der

- euklidischen Distanz (ED),
- quadrierten euklidischen Distanz (QED),
- City-Block-Distanz (CBD)!

Treten bei den jeweils drei ähnlichsten und unähnlichsten Kundenpaaren Unterschiede zwischen den verwendeten Distanzmaßen auf?

5.1.5 Welche drei Kundenpaare erweisen sich anhand der Tschebyscheff-Distanz als

- am ähnlichsten,
- am unähnlichsten?

5.2 Ähnlichkeitsmessung bei qualitativen Merkmalen

Im Fall von nominalskalierten Merkmalen greift man auf Ähnlichkeitsmerkmale zurück. Einen Spezialfall stellen hierbei dichotome (zweiwertige) Variablen dar. Grundsätzlich kann man qualitative Variablen mit mehr als zwei Ausprägungen (= polytome Variablen) in Dummy-Variablen (= dichotome Variablen) überführen. Der Einfachheit halber werden wir uns bei der Bestimmung von Ähnlichkeiten nur auf letzteren Fall konzentrieren.

Hierzu werden die beispielhaft bei acht Konsumenten erhobenen Daten für vier dichotome Merkmale in R eingelesen:

```
> Konsumenten8 = read.csv2(file="Konsumenten8.csv")
> Konsumenten8
  Konsument X1 X2 X3 X4
1         1  1  1  1  0
2         2  1  0  0  0
3         3  0  1  0  1
4         4  1  0  1  0
5         5  1  0  0  0
6         6  1  1  0  1
7         7  1  1  1  1
8         8  0  0  0  0
>class(Konsumenten8)
[1] "data.frame"
>str(Konsumenten8)
'data.frame':   8 obs. of  5 variables:
$ Konsument: int  1 2 3 4 5 6 7 8
 $ X1      : int  1 1 0 1 1 1 1 0
 $ X2      : int  1 0 1 0 0 1 1 0
 $ X3      : int  1 0 0 1 0 0 1 0
 $ X4      : int  0 0 1 0 0 1 1 0.
```

Zusätzlich ist die Nummer des Konsumenten unter dem Variablennamen `Konsument` erfasst. Die dichotomen Variablen sind wie folgt definiert:

- `X1`: Produktkauf (ja 1/nein 0)
- `X2` Erwerbstätigkeit (ja 1/nein 0)
- `X3`: Soziale Stellung (hoch 1/niedrig 0)
- `X4`: Bildung (hoch 1/niedrig 0).

Vierfeldertafeln für die Paarvergleiche erhält man mit der Funktion `xtabs`, mit der allgemein Kontingenztabellen für diskrete Merkmale erzeugt werden können. Hierzu sind in dem Argument `formula` die Beobachtungsvektoren der beiden zu vergleichenden Objekte unter Voranstellung der Tilde (~) zu benennen und durch ein Plus-

zeichen (+) voneinander zu trennen. Für die Kunden 1 und 2 ergibt sich z.B. die Vierfeldertafel

```
> xtabs(formula=~t(Konsumenten8[1,][-1])+t(Konsumenten8[2,][-1]))
                          t(Konsumenten8[2, ][-1])
t(Konsumenten8[1, ][-1])        0        1
                        0       1        0
                        1       2        1.
```

Sie weist jeweils eine Übereinstimmung in den mit 1 und 0 kodierten Eigenschaften aus(a=1, d=1). Bei zwei Merkmalen besitzt Konsument 1, aber nicht Konsument 2 die mit 1 kodierte Eigenschaft 1 (b=2). Der umgekehrte Fall tritt dagegen nicht auf (c=0).

Mit der R-Funktion daisy, die durch Laden des Pakets cluster verfügbar ist,

```
>library(cluster)
```

lassen sich der Simple-Matching-Koeffizient und der Jaccard-Koeffizient als Unähnlichkeitsmaße (dissimilarity coefficients: d(i,j)) bestimmen. Man erhält sie in der üblichen Form als Ähnlichkeitsmaße (similarity coefficients: s(i,j)) durch die anschließende Transformation

(5.1) $s(i,j) = 1 - d(i,j)$.

Für die acht Konsumenten wird die untere Dreiecksmatrix des Simple-Matching-Koeffizienten als Unähnlichkeitsmaß (dissimilarity coefficient) durch die Spezifikation symm (symmetrische binäre Variablen) des Arguments type in der R-Funktion daisy erzeugt. Das bedeutet, dass die nichtvorhandene Eigenschaft (0) einer Variablen bei der Messung der Übereinstimmungen genauso behandelt wird wie die vorhandene Eigenschaft. Die Spezifikation gower des Arguments metric erfolgt automatisch im Falle nicht-nummerischer Spalten des verwendeten Datensatzes:

```
>SMCd = daisy(Konsumenten8[-1], metric = "gower", type = list(symm=(1:4)))
>SMCd
Dissimilarities :
1   2   3   4   5   6   7
2 0.50
3 0.75 0.75
4 0.25 0.25 1.00
5 0.50 0.00 0.75 0.25
6 0.50 0.50 0.25 0.75 0.50
7 0.25 0.75 0.50 0.50 0.75 0.25
8 0.75 0.25 0.50 0.50 0.25 0.75 1.00

Metric :  mixed ;  Types = S, S, S, S
Number of objects : 8
```

```
>class(SMCd)
[1] "dissimilarity" "dist"
>str(SMCd)
Classes 'dissimilarity', 'dist'  atomic [1:28] 0.5 0.75 0.25 0.5 0.5 0.25 0.75 0.75 0.25 0 ...
  ..- attr(*, "Size")= int 8
  ..- attr(*, "Metric")= chr "mixed"
  ..- attr(*, "Types")= chr [1:4] "S" "S" "S" "S".
```

Aus der SMCd-Matrix wird nun die Matrix des Simple-Matching-Koeffizienten(SMC) als Ähnlichkeitsmaß (similarity coefficient) unter Anwendung der Transformation (5.1) gewonnen:

```
> SMC = 1 - SMCd
> SMC
Dissimilarities :
    1    2    3    4    5    6    7
2 0.50
3 0.25 0.25
4 0.75 0.75 0.00
5 0.50 1.00 0.25 0.75
6 0.50 0.50 0.75 0.25 0.50
7 0.75 0.25 0.50 0.50 0.25 0.75
8 0.25 0.75 0.50 0.50 0.75 0.25 0.00

Metric :  mixed ;  Types = S, S, S, S
Numberofobjects : 8.
```

Dagegen wird die untere Dreiecksmatrix des Jaccard-Koeffizienten als Unähnlichkeitsmaß (dissimilarity coefficient) durch die Spezifikation asymm (asymmetrische binäre Variablen) des Arguments type in der R-Funktion daisy erzeugt. Hier wird die nicht-vorhandene Eigenschaft (0) einer Variablen bei der Messung der Übereinstimmungen zwischen zwei Objekten außer Acht gelassen:

```
>Jd = daisy(Konsumenten8[-1], metric = "gower", type = list(asymm=(1:4)))
>Jd
Dissimilarities :
          1         2         3         4         5         6         7
2 0.6666667
3 0.7500000 1.0000000
4 0.3333333 0.5000000 1.0000000
5 0.6666667 0.0000000 1.0000000 0.5000000
6 0.5000000 0.6666667 0.3333333 0.7500000 0.6666667
7 0.2500000 0.7500000 0.5000000 0.5000000 0.7500000 0.2500000
8 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000

Metric :  mixed ;  Types = A, A, A, A
Number of objects : 8
```

```
>class(Jd)
[1] "dissimilarity" "dist"
>str(Jd)
Classes 'dissimilarity', 'dist'  atomic [1:28] 0.667 0.75 0.333 0.667 0.5 ...
  ..- attr(*, "Size")= int 8
  ..- attr(*, "Metric")= chr "mixed"
  ..- attr(*, "Types")= chr [1:4] "A" "A" "A" "A".
```

Die Matrix des Jaccard-Koeffizient (J) als Ähnlichkeitsmaß (similarity coefficient) erhält man aus der Jd-Matrix wieder mittels der Transformation (5.1):

```
> J = 1 - Jd
> J
Dissimilarities :
1       2       3       4       5       6       7
2 0.3333333
3 0.2500000 0.0000000
4 0.6666667 0.5000000 0.0000000
5 0.3333333 1.0000000 0.0000000 0.5000000
6 0.5000000 0.3333333 0.6666667 0.2500000 0.3333333
7 0.7500000 0.2500000 0.5000000 0.5000000 0.2500000 0.7500000
8 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000

Metric :  mixed ;  Types = A, A, A, A
Numberofobjects : 8.
```

Man beachte, dass trotz der Transformation (5.1) die Bezeichnung Dissimilarities bei den Ähnlichkeitsmaßen SMC und J beibehalten worden ist. Die unterschiedlichen Werte in beiden Ähnlichkeitsmatrizen erklären sich aus dem Vorhandensein von Übereinstimmungen in den nicht im Vordergrund stehenden Eigenschaften. Die Wahl des Ähnlichkeitskoeffizienten muss in dem konkreten Anwendungsbereich getroffen werden. Hierbei kommt es entscheidend darauf an, ob die Ähnlichkeit anhand bestimmter oder beliebiger Eigenschaften festgestellt werden soll.

Übungsaufgaben

5.2.1 Bei sieben Studierenden der Lehrveranstaltung „Multivariate Statistik“ sind die Merkmale Geschlecht (weiblich 1, männlich 0), Wohnort (Kassel 1, nicht Kassel 0), Semester (niedrig 1, hoch 0), Studienfach (VWL 1, BWL 0) ermittelt worden:

Studierender	Geschlecht	Wohnort	Studienfach	Semester
1	1	1	1	1
2	1	0	0	0
3	0	1	1	1
4	1	0	1	0
5	0	1	0	0
6	1	1	0	1
7	0	1	0	0

Definieren Sie ein data.frame-Objekt Studenten7, das die kodierten Werte der fünf in der Tabelle wiedergegebenen qualitativen Variablen in R enthält! Überprüfen Sie den Variablentyp mit dem str-Befehl!

5.2.2 Erstellen Sie eine Vierfeldertafel für einen Paarvergleich der Studierenden 4 und 5 und interpretieren Sie sie!

5.2.3 Bestimmen Sie für die sieben Studierenden die Ähnlichkeitsmatrix unter Verwendung

- des Simple-Matching-Koeffizienten (SMC),
- des Jaccard-Koeffizienten (J)

und erläutern Sie die Unterschiede!

5.3 Hierarchische Clusteranalyse und Dendrogramm

Unter Verwendung der berechneten Distanz- oder Ähnlichkeitmaße werden die Objekte zu Gruppen zusammengefasst. Beispielhaft greifen wir hierzu auf den Datensatz Regionen12 zurück. Da von vornherein nicht bekannt ist, wie viel Gruppen vorhanden sind, wird für das Regionenbeispiel eine hierarchische Clusteranalyse durchgeführt.

In R kann eine hierarchische Clusteranalyse mit der im Paket stats enthaltenen Funktion hclust ausgeführt werden. Als Input dieser Funktion ist ein Objekt des Typs dist erforderlich, das die Distanzen zwischen den Untersuchungseinheiten enthält. Durch Spezifikation des Arguments method wird ein hierarchisches Verfahren wie z.B. das Single-Linkage-, Complete-Linkage-, Average-Linkage- oder Ward-Verfahren gewählt.

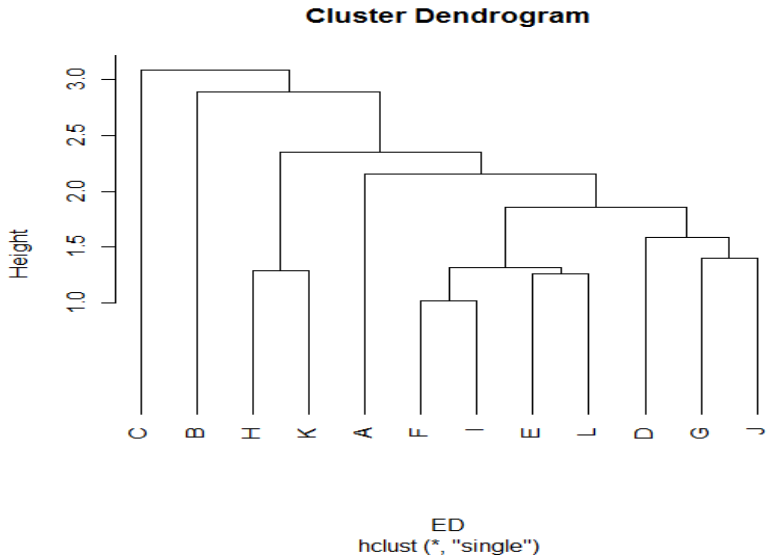
Bei den ersteren dreiVerfahren wird zur Messung der Unähnlichkeit der Regionen, die euklidische Distanz verwendet, die den direkten Abstand zwischen zwei Objektvektoren im m-dimensionalen Merkmalsraum wiedergibt. Bei dem Ward-Verfahren wird auf die quadrierte euklidische Distanz zurückgegriffen. Beide Distanzmatrizen, ED und QED, sind bereits für die 12 Regionen bei der Messung der Unähnlichkeit der Objektpaare ermittelt worden.

Um Aufschluss über potenzielle Ausreißer zu erhalten, führen wir als erstes eine hierarchische Clusteranalyse mit den Single-Linkage-Verfahren durch:

```
> Reg12hcl.single = hclust(ED, method="single")
> plot(Reg12hcl.single, hang=-1).
```

Der negative Wert des Arguments hang in der plot-Funktion gibt an, dass die Labels der Regionen im Dendrogramm unter der Nullmarke angebracht werden sollen.

Abbildung 5.1: Dendrogramm des Single-Linkage-Verfahrens (Zwölf-Regionen-Beispiel)



Wie aus dem in Abbildung 5.1 wiedergegebenen Dendrogramm ersichtlich ist, ergibt sich bei einer Anwendung des Single-Linkage-Verfahrens keine klare Gruppenstruktur. Die Heterogenität steigt von Stufe zu Stufe relativ moderat an. Allein die beiden Regionen B und C scheinen eine größere Distanz zu dem sich auf der drittletzten Stufe des hierarchischen Klassifikationsprozesses ergebenden Cluster der übrigen Regionen zu haben. Es könnte sich hierbei um Ausreißer handeln, die ggf. von vornherein separiert werden könnten. Um ihren Einfluss auf das Klassifikationsverhalten verschiedener Verfahren transparent zu machen, müsste die hierarchische Klassifikation mit und ohne die mit dem Single-Linkage-Verfahren identifizierten beiden potenziellen Ausreißerregionen B und C durchgeführt werden, worauf hier jedoch verzichtet werden soll.

Das Complete-Linkage-Verfahren ist in R voreingestellt, so dass man diese Methode nicht notwendig explizit angeben muss:

```
> Reg12hcl.complete =hclust(ED, method = "complete")    oder    > Reg12hcl.complete = hclust(ED)
> plot(Reg12hcl.complete, hang=-1).
```

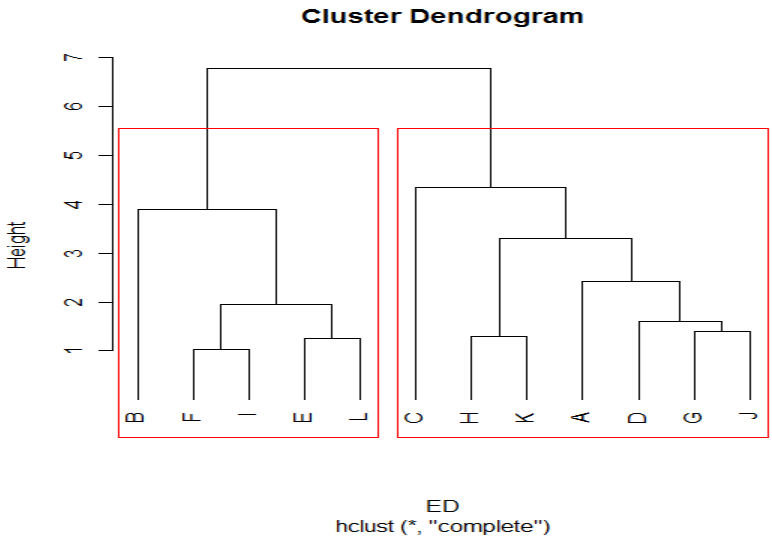
Die Funktion rect.hclust zeichnet aufgrund der Spezifikation k=2 und border="red" rote Rechtecke um die beiden Cluster, die sich mit dem Complete-Linkage-Verfahren identifizieren lassen:

```
> rect.hclust(Reg12hcl.complete, k=2, border="red").
```

In dem ersten Cluster befinden sich die Regionen B, F, I, E und L (s. Abbildung 5.2), die mehrheitlich durch eine hohe Bevölkerungsdichte und ein hohes Bruttoinlandsprodukt bei unterdurchschnittlichen Ausprägungen in den übrigen Merkmalen

gekennzeichnet sind. Allein die Region B, die erst relativ spät in die Gruppe integriert wird,

Abbildung 5.2: Dendrogramm des Complete-Linkage-Verfahrens (Zwölf-Regionen-Beispiel)



weicht insofern merklich ab als sie im Unterschied zu den übrigen Gruppenmitgliedern ein positives BIP-Wachstum bei einem positiven Wanderungssaldo aufweist.

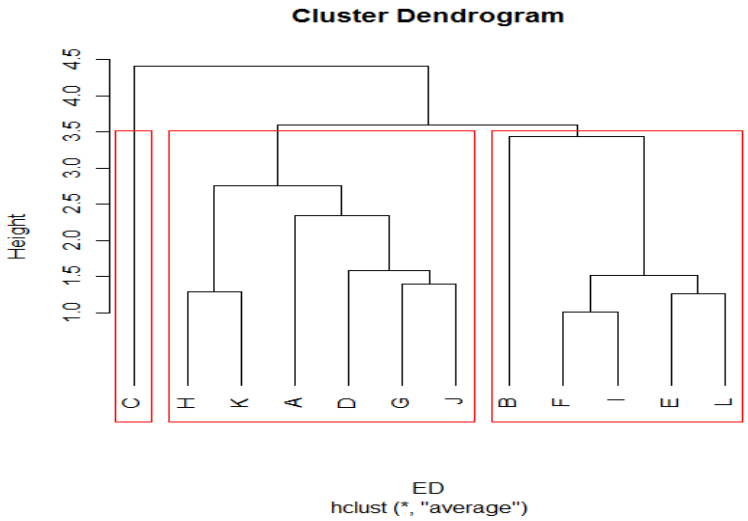
Nicht ganz so eindeutig sind die Verhältnisse innerhalb des zweiten Clusters zu dem die Regionen A, C, D, G, H, J und K gehören (s. Abbildung 5.2). Die potenzielle Ausreißerregion C fällt hier z.B. insbesondere durch ihren weit über dem Durchschnitt liegenden Anteil der in der Landwirtschaft beschäftigten Erwerbspersonen auf. Wie die Region B wird die Region C erst am Ende des hierarchischen Klassifikationsprozesses einer bestehenden Gruppe zugeordnet.

Das Average-Linkage-Verfahren wird durch die Spezifikation „average“ des Arguments method in der hclust-Funktion aufgerufen:

```
> Reg12hcl.average = hclust(ED, method="average")
> plot(Reg12hcl.average, hang=-1)
> rect.hclust(Reg12hcl.average, k= 3, border="red").
```

Abbildung 5.3 zeigt, dass die Ausreißerregion C bei Anwendung des Average-Linkage-Verfahrens keiner der beiden Gruppen zugeordnet werden kann. Die potenzielle Ausreißerregion B erhöht zwar die Heterogenität des aus den Regionen E, F, I und L bestehenden Clusters 1 beträchtlich. Doch erfolgt ihre Zuordnung noch vor der Fusion dieses Clusters mit dem aus dem Regionen A, D, G, H, J und K bestehenden Cluster 2. Abgesehen von der fehlenden Zuordnung der Region C stimmen die beiden „entdeckten“ Cluster mit den vom Complete-Linkage-Verfahren ermittelten Gruppen überein.

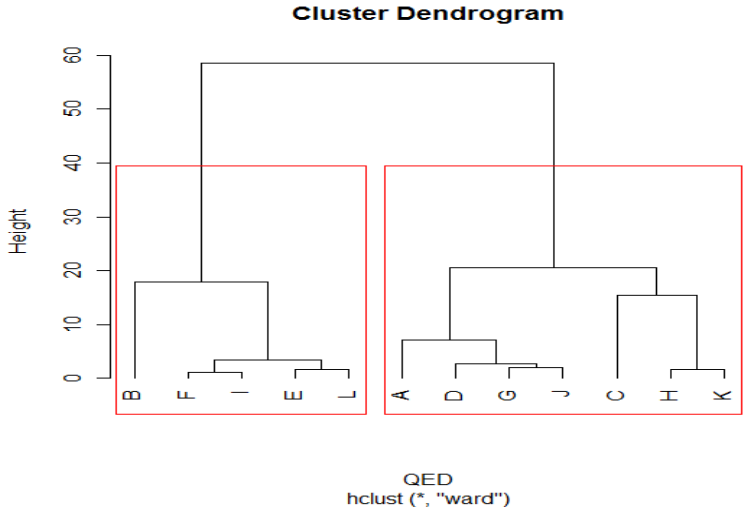
Abbildung 5.3: Dendrogramm des Average-Linkage-Verfahrens (Zwölf-Regionen-Beispiel)



Schließlich betrachten wir noch die hierachische Klassifikation mit dem Ward-Verfahren, bei der die Zwei-Cluster-Lösung am deutlichsten hervortritt (s. Abbildung 5.4):

```
> Reg12hcl.ward = hclust(QED, method="ward")      # Verwendung der quadrierten euklidischen Distanz
> plot(Reg12hcl.ward, hang=-1)
> rect.hclust(Reg12hcl.ward, k=2, border="red").
```

Abbildung 5.4: Dendrogramm des Ward-Verfahrens (Zwölf-Regionen-Beispiel)



Das Cluster 1 besteht aus den Regionen B, E, F, I und L, das Cluster 2 enthält die Regionen A, C, D, G, H, J und K. Zwar wird durch die Eingruppierung der potentiellen Ausreißer B und C die Heterogenität der Partition erkennbar erhöht, doch bleibt der Anstieg deutlich hinter demjenigen zurück, der sich bei der Fusion der beiden oben genannten Cluster ergibt. Aus diesem Grund muss sie nicht notwendig als unakzeptabel angesehen werden. Hieran wird deutlich, dass die Ausreißeridentifikation mit Hilfe des Single-Linkage-Verfahrens differenziert zu betrachten ist.

Ansonsten unterscheidet sich die ermittelte Clusterstruktur jedoch nicht von den mit dem Complete-Linkage- und Average-Linkage-Verfahren gefundenen Gruppierungen.

Mit der Funktion hclust ermittelt R auch weitere Informationen, die das Programm in Objekten dieses Typs speichert. Informationen darüber, welche Cluster in den einzelnen Stufen fusioniert worden sind, findet man unter dem Attribut merge. Bei Einsatz des Ward-Verfahrens ergibt sich z.B. folgende Zuordnungsfolge:

```
> Reg12hcl.ward$merge
      [,1] [,2]
 [1,]   -6   -9
 [2,]   -5  -12
 [3,]   -8  -11
 [4,]   -7  -10
 [5,]   -4    4
 [6,]    1    2
 [7,]   -1    5
 [8,]   -3    3
 [9,]   -2    6
[10,]    7    8
[11,]    9   10.
```

Im Dendrogramm erkennt man zum Beispiel nicht eindeutig, ob im ersten Schritt die Objekte F und I oder E und L oder aber H und K vereinigt wurden. Die Fusionsfolge unter merge zeigt an, dass zuerst die Region 6 (= Region F) und 9 (= Region I) vereinigt wurden. Eine negative Zahl bedeutet hierin, dass das Objekt mit der entsprechenden Nummer mit einem anderen Objekt oder Cluster fusioniert wird. Die positiven Zahlen beziehen sich immer auf den Arbeitsschritt, in dem das Cluster zuletzt gebildet oder erweitert worden ist. So geben die Zahlen „1“ und „2“ im sechsten Schritt z.B. an, dass das in Arbeitsschritt 1 gebildete Cluster mit dem in Arbeitsschritt 2 gebildeten Cluster fusioniert wurde.

Unter dem Attribut height werden die Werte des Heterogenitätsmaßes gespeichert, bei dem die Vereinigung der Cluster in den n-1 Stufen erfolgte. Für das Ward-Verfahren werden in den 11 Arbeitsschritten folgende Werte des verwendeten Varianzkriteriums ausgewiesen:

```
> Reg12hcl.ward$height
 [1] 1.041097 1.594521 1.668840 1.968472 2.724783 3.408243 7.116986 15.503347 17.906704
20.563770 58.503238.
```

Übungsaufgaben

5.3.1 Setzen Sie das Single-Linkage-Verfahren unter Verwendung der in Aufg. 5.1.4 berechneten euklidischen Distanzen zur Identifikation von Ausreißern in dem Datensatz Kaeufer30 ein!

5.3.2 Ermitteln Sie eine hierarchische Klassifikation der 30 Käufer (Datensatz Kaeufer30) unter Verwendung der euklidischen Distanz mit Hilfe

- des Complete-Linkage-Verfahrens,
- des Average-Linkage-Verfahrens!

Wieviel Käufergruppen lassen sich anhand der Dendrogramme identifizieren? Begründen Sie die Wahl der Clusterzahl! Unterscheiden sich die von den beiden Verfahren aufgedeckten Clusterstrukuren voneinander?

5.3.3 Geben Sie die in Aufg. 5.3.2 in dem hclust-Objekt des Complete-Linkage-Verfahrens gespeicherte Zuordnungsfolge wieder! Erläutern Sie sie anhand zweier Stufen mit negativen und positiven Werten!

5.3.4 Explorieren Sie die Clusterstruktur der 30 Käufer (Datensatz Kaeufer30) unter Anwendung des Ward-Verfahrens! Welche Unterschiede zu den hierarchischen Klassifikationen mit dem Complete-Linkage- und Average-Linkage-Verfahren zeigen sich hier?

5.4 Partitionsverfahren

Im Gegensatz zu den hierarchischen Klassifikationsverfahren gehen die partitionierenden Verfahren von einer gegebenen Zerlegung der Objektmenge in G Cluster aus, die jedoch nicht als optimal angesehen wird. Diese Ausgangspartition soll durch den Austausch von Objekten zwischen Clustern verbessert werden. Die Anzahl der Cluster bleibt hierbei konstant. Wir beschränken uns hier darauf, die Implementierung des iterativen Minimal-Distanz-Verfahrens in Form der k-Means-Methode in R vorzustellen, das metrisch skalierte Variablen voraussetzt.

Die Arbeitsweise der k-Means Methode sei anhand des Regionenbeispiels exemplifiziert. Hierzu gehen wir von einer Drei-Gruppen-Lösung aus, für die die Ausgangspartition durch

- Cluster 1: Regionen A, C, H
- Cluster 2: Regionen B, E, F
- Cluster 3: Regionen D, G

gegeben ist. Um eine Vergleichbarkeit mit dem Beispiel im Skript „Multivariate Statistik“ (Kosfeld, 2012) herzustellen, werden bei der Ermittlung der Minimal-Distanz-Partition allein die Merkmale Bevölkerungsdichte (X1) und das BIP (X2) berücksichtigt.

Hierzu werden aus der kompletten standardisierten Datenmatrix Z die Werte der beiden ersten Merkmale für die Regionen A bis H ausgeschnitten:

```
> Z8.EDBIP =Z[1:8,1:2]
> Z8.EDBIP
          x1          x2
A -0.65651480 -1.2449654
B  1.70883627  1.2541546
C -1.34259888 -1.6530692
D -0.51619249  0.2993362
E  0.49136771  0.2802707
F  1.38103415  1.0222775
G  0.03071947 -0.1778154
H -1.23735714 -0.9883547.
```

Anschließend werden die Daten der drei Cluster separat in den matrix-Objekten cl1, cl2 und cl3 gespeichert:

```
> cl1 = Z8.EDBIP[c(1,3,8),]  # Daten des Clusters 1
> cl1
        x1          x2
A -0.6565148 -1.2449654
C -1.3425989 -1.6530692
H -1.2373571 -0.9883547

> cl2 = Z8.EDBIP[c(2,5,6),] # Daten des Clusters 2
> cl2
        x1          x2
B 1.7088363 1.2541546
E 0.4913677 0.2802707
F 1.3810342 1.0222775

> cl3 = Z8.EDBIP[c(4,7),] # Daten des Clusters 3
> cl3
        x1          x2
D -0.51619249  0.2993362
G  0.03071947 -0.1778154.
```

Da bei der k-Means-Methode die Distanzen der Objektvektoren von den Clusterzentren zu ermitteln sind, werden noch die Gruppenmittelwerte der Variablen berechnet,

```
> cl1m = colMeans(cl1)
> cl1m
      x1          x2
-1.078824 -1.295463

> cl2m = colMeans(cl2)
> cl2m
       x1          x2
1.1937460 0.8522343

> cl3m = colMeans(cl3)
> cl3m
       x1              x2
-0.24273651  0.06076039,
```

und zu einem matrix-Objekt verbunden:

```
> CLUSM = rbind(cl1m,cl2m,cl3m)
> CLUSM
              x1            x2
cl1m -1.0788236 -1.29546311
cl2m  1.1937460  0.85223429
cl3m -0.2427365  0.06076039.
```

Um die k-Means-Methode in R unter Verwendung der vorgegebenen Cluster auszuführen, sind in der Funktion kmeans die Datenmatrix und die Clusterzentren der Ausgangspartition zu spezifizieren:

```
> Reg12.kmeans = kmeans(Z8.EDBIP, centers = CLUSM)
> Reg12.kmeans
K-means clustering with 3 clusters of sizes 3, 2, 3

Cluster means:
         x1        x2
1 -1.078823605 -1.2954631
2  1.544935212  1.1382161
3  0.001964896  0.1339305

Clustering vector:
A B C D E F G H
1 2 1 3 3 2 3 1
```

```
Within cluster sum of squares by cluster:
[1] 0.49780265 0.08061061 0.65478902
 (between_SS / total_SS =  92.7 %)

Available components:

[1] "cluster"    "centers"    "totss"      "withinss"   "tot.withinss" "betweenss"   "size"    .
```

Der Output zeigt als erstes, dass das Cluster 1 der Minimal-Distanz-Partition wie in der Ausgangspartition aus drei Regionen besteht. Die Größe der anderen beiden Cluster hat sich dagegen verändert. Das Cluster 2, das sich vorher aus drei Regionen zusammengesetzt hat, enthält jetzt nur noch zwei Regionen. Umgekehrt hat sich die Zahl der Regionen im Cluster 3 von zwei auf drei erhöht:

```
> Reg12.kmeans$size
[1] 3 2 3.
```

Unter Cluster means werden die neuen Clusterschwerpunkte ausgewiesen, die in dem Atrribut centers verfügbar sind:

```
> Reg12.kmeans$centers
         x1          x2
1 -1.078823605 -1.2954631
2  1.544935212  1.1382161
3  0.001964896  0.1339305.
```

Das Attribut cluster enthält die im Output der Funktion kmeans unter der Überschrift Clustering vector wiedergegebenen Information über die endgültige Zuordnung der Regionen zu den drei Clustern:

```
> Reg12.kmeans$cluster
A B C D E F G H
1 2 1 3 3 2 3 1.
```

Ein Vergleich mit der Ausgangspartition zeigt, dass die Region E vom Cluster 2 in das Cluster 3 umgruppiert worden ist. Alle anderen Regionen hatten bereits die minimale Distanz zu ihrem Cluster.

Als weitere Attribute sind die totale (totss) Abweichungsquadratsumme sowie die Abweichungsquadratsummen innerhalb (withinss und tot.withinss) und zwischen (betweenss) den Clustern verfügbar:

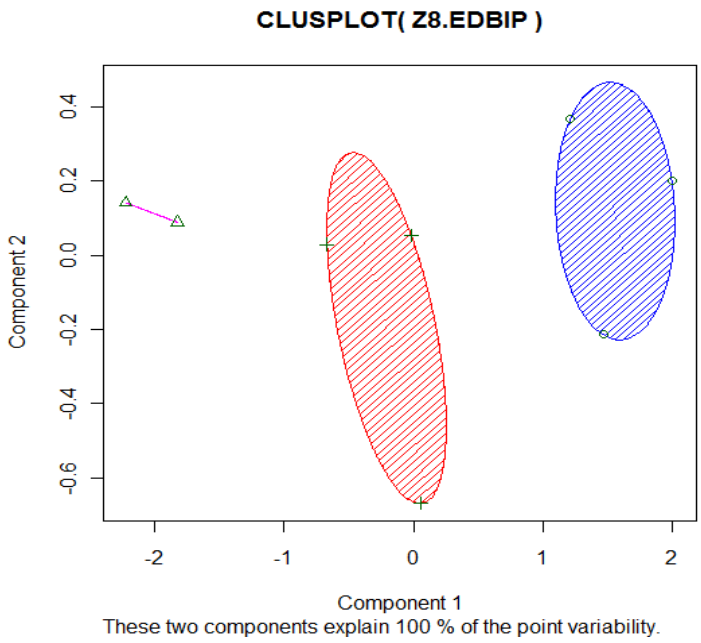
```
> Reg12.kmeans$tots
[1] 16.99307
> Reg12.kmeans$withins
[1] 0.49780265 0.08061061 0.65478902
> Reg12.kmeans$tot.withins
[1] 1.233202
> Reg12.kmeans$betweens
[1] 15.75987.
```

Zusätzlich wird im Output noch das prozentuale Verhältnis der Abweichungsquadratsumme zwischen den Clustern zur totalen Abweichungsquadratsumme angezeigt: (between_SS / total_SS = 92.7 %).

Die mit k-Means-Verfahren erhaltene Minimum-Distanz-Partition lässt sich mit der R-Funktion clusplot grafisch darstellen (s. Abbildung 5.5). Hierzu müssen als Argumente insbesondere die Datenmatrix und die Zuordnung der Regionen zu den Clustern spezifiziert werden:

```
> clusplot(Z8.EDBIP, clus=Reg12.kmeans$cluster, lines=0, shade=TRUE, color=TRUE).
```

Abbildung 5.5: Bivariater Clusterplot (Acht-Regionen-Beispiel)



Wenn man mit der Minimal-Distanz-Partition weiterarbeiten möchte, ist es sinnvoll, die Ausgangsdaten um die Clusterzuordnung zu ergänzen.

```
> Z8cl.EDBIP = data.frame(Z8.EDBIP, Reg12.kmeans$cluster)
```

```
> Z8cl.EDBIP
         x1          x2      Reg12.kmeans.cluster
A -0.65651480 -1.2449654            1
B  1.70883627  1.2541546            2
C -1.34259888 -1.6530692            1
D -0.51619249  0.2993362            3
E  0.49136771  0.2802707            3
F  1.38103415  1.0222775            2
G  0.03071947 -0.1778154            3
H -1.23735714 -0.9883547            1.
```

Im vorangegangenen Beispiel haben wir eine Startpartition vorgegeben. In R ist es jedoch auch möglich mit einer Zufallspartition zu beginnen. Hierfür muss man lediglich als zweites Argument die Anzahl der gewünschten Cluster angeben:

```
> kmeans(Z8.EDBIP. 3).
```

Für diesen kleinen Beispieldatensatz führen Zufallspartitionen meist zu demselben Ergebnis. Insbesondere können sich jedoch bei umfangreicheren Daten-sätzen unterschiedliche Minimal-Distanz-Partitionen ergeben. In diesen Fällen wird die Qualität der Lösung stärker durch die Startpartition beeinflusst. In der Regel wird es sich anbieten, eine mit einem hierarchischen Verfahren ermittelte Partition vorzu-geben.

Übungsaufgaben

5.4.1 Definieren Sie Teilmatrizen der standardisieren Werte des Datensatzes Kaeufer30 für vier Cluster, die aus dem Dendrogramm des Complete-Linkage-Verfahrens hervorgehen!

5.4.2 Berechnen Sie unter Verwendung der in Aufg. 5.4.1 definierten Teilmatrizen die Clusterschwerpunkte und speichern Sie sie in einem matrix-Objekt!

5.4.3 Bestimmen Sie für das Käufer-Beispiel (Datensatzes Kaeufer30) mit der k-Means-Methode eine Minimal-Distanz-Partition unter Verwendung der Vier-Cluster-Lösung des Complete-Linkage-Verfahrens und interpretieren Sie sie!

5.4.4 Überprüfen Sie die Stabilität der in Aufg. 5.4.3 ermittelten Minimal-Distanz-Partition unter Verwendung von drei Zufallspartitionen als Ausgangspartitionen!

5.4.5 Stellen die die in Aufg. 5.4.3 ermittelte Minimal-Distanz-Partition in Form eines bivariaten Clusterplots grafisch dar!

6. Diskriminanzanalyse

Im Unterschied zur Clusteranalyse geht die Diskriminanzanalyse von einer Kenntnis der Gruppenzugehörigkeit der Objekte aus. Die Diskriminanzanalyse zielt zum einen auf eine bestmögliche Diskrimination (=Trennung) der Gruppen durch eine Reihe zur Verfügung stehender Merkmale (=Merkmalsvariablen) ab. Zum anderen wird die Klassifikation neuer Objekte zu einer der bestehenden Gruppen thematisiert.

6.1 Streuungsmatrizen der Merkmalsvariablen

Ausgangspunkt einer Diskrimination der G Gruppen von Objekten die Diskriminanzfunktion

(6.1) $d_i = a_0 + a_1 \cdot x_{i1} + a_2 \cdot x_{i2} + \ldots + a_m \cdot x_{im}$

nach der sich die Diskriminanzwerte d_i der Objekte aus einer Linearkombination ihrer Variablenwerte x_{i1}, x_{i2}, …, x_{im} unter Addition einer Konstanten a_0 ergeben. Die Koeffizienten der Linearkombination a_1, a_2, …, a_m heißen Diskriminanzkoeffizienten. Zu ihrer Ermittlung werden die Streuungsmatrix innerhalb der Gruppen, **W**, und die Streuungsmatrix zwischen den Gruppen, **B**, der Merkmalvariablen benötigt, wozu hier ein Datensatz von 30 Käufern der Marken A und B eingelesen wird:

```
> Kaeufer30 = read.csv2("Kaeufer30.csv")
> class(Kaeufer30)
[1] "data.frame"
> str(Kaeufer30)
'data.frame':   30 obs. of  4 variables:
 $ Kunde    : num  1 2 3 4 5 6 7 8 9 10 ...
 $ Einkommen: num  28 23 25 37 33 32 23 30 30 33 ...
 $ Alter    : num  37 25 40 30 45 38 35 32 42 27 ...
 $ Gruppe   : num  1 1 1 1 1 1 1 1 1 1 ...
> Kaeufer30
   Kunde Einkommen Alter Gruppe
      1     1      28    37     1
      2     2      23    25     1
      3     3      25    40     1
      4     4      37    30     1
      5     5      33    45     1
      6     6      32    38     1
      7     7      23    35     1
      8     8      30    32     1
      9     9      30    42     1
     10    10      33    27     1
```

```
11  11    35  40   1
12  12    23  31   1
13  13    39  45   1
14  14    36  34   1
15  15    40  38   1
16  16    22  20   1
17  17    26  20   2
18  18    41  27   2
19  19    34  34   2
20  20    27  35   2
21  21    42  43   2
22  22    30  22   2
23  23    39  32   2
24  24    35  30   2
25  25    44  35   2
26  26    34  22   2
27  27    37  41   2
28  28    26  27   2
29  29    29  25   2
30  30    33  37   2
```

Als Merkmalsvariablen werden hierbei das Einkommen (X_1) und Alter (X_2) verwendet. Die ersten 16 Kunde bilden als Käufer der Marke A die Gruppe 1, während die letzten 14 Kunde als Käufer der Marke B der Gruppe 2 angehören. Anhand der Gruppenmittelwerte

```
> # Gruppe 1 (Marke A)
> n1 = 16
> X1.EA = subset(Kaeufer30, Gruppe==1)[,2:3]
> xm1.EA = colMeans(X1.EA)
> xm1.EA
Einkommen    Alter
  30.5625   34.9375
> # Gruppe 2 (Marke B)
> n2 = 14
> X2.EA = subset(Kaeufer30, Gruppe==2)[,2:3]
> xm2.EA = colMeans(X2.EA)
> xm2.EA
Einkommen    Alter
 34.07143  30.71429
```

ist erkennbar, dass die Käufer der Marke A (Gruppe 1) im Mittel ein niedrigeres Einkommen haben als die Käufer der Marke B (Gruppe 2), während es sich beim Alter genau umgekehrt verhält.

Aus der 30x2-Beobachtungsmatrix X.EA, die die Werte der Merkmalsvariablen Einkommen (X_1) und Alter (X_2) aller 30 Käufer enthält,

```
> X.EA = as.matrix(Kaeufer30[,2:3]).
```

lassen sich die Gesamtmittelwerte der Variablen Einkommen und Alter bestimmen:

```
# Gesamtmittelwerte
> n = 30
> m = 2
> xm.EA = colMeans(X.EA)
> xm.EA
Einkommen    Alter
 32.20000  32.96667.
```

Zur Berechnung der Streuungsmatrizen **W** und **B**,

(6.2) $\mathbf{W} = (\mathbf{X} - \overline{\mathbf{X}}^{grup})'(\mathbf{X} - \overline{\mathbf{X}}^{grup})$

und

(6.3) $\mathbf{B} = (\overline{\mathbf{X}}^{grup} - \overline{\mathbf{X}})'(\overline{\mathbf{X}}^{grup} - \overline{\mathbf{X}})$

werden die Matrix der Gesamtmittelwerte, $\overline{\mathbf{X}}$, der beiden Merkmalsvariablen,

```
> XM.EA = matrix(rep(xm.EA, n), nrow=n, ncol=m, byrow=TRUE)
> XM.EA
     [,1]    [,2]
 [1,] 32.2 32.96667
 [2,] 32.2 32.96667
 [3,] 32.2 32.96667
...
[30,] 32.2 32.96667,
```

und die Matrix Gruppenmittelwerte, $\overline{\mathbf{X}}^{grup}$,

```
> XM1.EA = matrix(rep(xm1.EA, n1), nrow=n1, ncol=m, byrow=TRUE)
> XM2.EA = matrix(rep(xm2.EA, n2), nrow=n2, ncol=m, byrow=TRUE)
> XMgrup.EA = rbind(XM1.EA, XM2.EA)
> XMgrup.EA
```

```
         [,1]     [,2]
 [1,] 30.56250 34.93750
...
[16,] 30.56250 34.93750
[17,] 34.07143 30.71429
...
[30,] 34.07143 30.71429
```

benötigt. Hiermit erhält man als Streuungsmatrix innerhalb der Gruppen, **W**,

```
> W = t(X.EA-XMgrup.EA)%*%(X.EA-XMgrup.EA)
```

oder

```
> W = crossprod(X.EA-XMgrup.EA)
> W
            Einkommen    Alter
Einkommen 1014.8661   599.8482
Alter      599.8482  1433.7946
```

und als Streuungsmatrix zwischen den Gruppen, **B**,

```
> B = t(XMgrup.EA-XM.EA)%*%(XMgrup.EA-XM.EA)
```

oder

```
> B = crossprod(XMgrup.EA-XM.EA)
> B
         [,1]      [,2]
[1,]  91.93393 -110.6482
[2,] -110.64821  133.1720.
```

Die totale Streuungsmatrix **T** ist durch die Summe der Intra-Gruppen-Streuungsmatrix **W** und der Inter-Gruppen-Streuungsmatrix **B**:

```
> T = W + B
> T
            Einkommen   Alter
Einkommen    1106.8    489.200
Alter         489.2   1566.967.
```

Übungsaufgaben

6.1.1 Fügen Sie dem `data.frame`-Objekt `Regionen12` die Gruppenvariable `Gruppe` hinzu, die für die Regionen C, H und K den Wert 1 (ländliche Regionen) und für die Regionen A, B, D, E, F, G, I, J, L den Wert 2 (verstädterte Regionen) annimmt!

6.1.2 Konvertieren Sie den in Aufg. 6.1.1.erweiterten Datensatz Regionen12 in ein matrix-Objekt X.reg unter Eliminierung der Variablen Region und Geben Sie die Teilmatrizen X.reg1 und X.reg2 für beide Gruppen an!

6.1.3 Bestimmen Sie

- die Matrix der Gesamtmittelwerte (XM.reg)

und

- die Matrix der Gruppenmittelwerte (XMgrup.reg)!

6.1.4 Berechnen Sie unter Verwendung der Matrizen X.reg, XM.reg und XMgrup.reg die

- Streuungsmatrix innerhalb der Gruppen (W.reg),
- Streuungsmatrix zwischen den Guppen (B.reg),
- totale Streuungsmatrix (T.reg)!

6.2 Optimale Gruppentrennung und Diskriminanzfunktion

Unter Verwendung der Intra-Gruppen-Streuungsmatrix **W** und der Inter-Gruppen-Streuungsmatrix **B** lässt sich das Diskriminanzkriterium

(6.4) $\lambda(\mathbf{a}) = \frac{B_d}{W_d} = \frac{\mathbf{a'Ba}}{\mathbf{a'Wa}}$

in Bezug auf den Vektor der Diskriminanzkoeffizienten $\mathbf{a} = (a_1\ a_2 \ldots a_m)'$ maximieren. Hierin werden die Intra- und Inter-Gruppen-Streuungen auf der Diskriminanzachse, W_d und B_d, auf die entsprechenden gruppenbezogenen Streuungen der Merkmalsvariablen zurückgeführt. In der Diskriminanzfunktion (6.1) werden diejenigen Gewichte a_1, a_2, …,a_m als Diskriminanzkoeffizienten verwendet, die zu einer bestmöglichen Trennung der Gruppen auf der Diskriminanzachse führen. Sie ist durch größtmögliche Abstände der Gruppenmittelwerte auf der Diskriminanzachse bei geringstmöglicher Streuung der Diskriminanzwerte innerhalb der Gruppen für einen gegebenen Datensatz charakterisiert.

Die Maximierung des Diskriminanzkriteriums λ führt zu der charakteristische Gleichung

(6.5) $(\mathbf{W}^{-1}\mathbf{B} - \lambda \cdot \mathbf{I}) \cdot \mathbf{a} = \mathbf{0}$

die das zu lösende Eigenwertproblem kennzeichnet. Speziell sind hier die Eigenwerte und Eigenvektoren der Matrix $\mathbf{W}^{-1}\mathbf{B}$,

```
> Wi = solve(W)
> Wi
             Einkommen       Alter
Einkommen  0.0013090528 -0.0005476607
Alter     -0.0005476607  0.0009265715
```

```
> WiB = Wi%*%B
> WiB
                [,1]       [,2]
Einkommen  0.1809440 -0.2177774
Alter     -0.1528721  0.1839911
```

zu bestimmen. Die Anzahl der von null verschiedenen Eigenwerte der Matrix $\mathbf{W}^{-1}\mathbf{B}$ wird durch ihren Rang rg bestimmt, der durch rg($\mathbf{W}^{-1}\mathbf{B}$) = min {m, G-1} gegeben ist. Mit m=2 und G=2 existiert in unserem Beispiel nur ein von Null verschiedener Eigenwert:

```
> eigen.val = round(eigen(WiB)$values, 4)
> eigen.val
[1] 0.3649 0.0000.
```

Mit der Funktion qr findet man bestätigt, dass der Rang der Matrix $\mathbf{W}^{-1}\mathbf{B}$ eins beträgt:

```
> qr(WiB)$rank
[1] 1.
```

Im Zwei-Gruppen-Fall entspricht der Eigenwert außerdem der Spur der Matrix $\mathbf{W}^{-1}\mathbf{B}$ d.h. der Summe ihrer Hauptdiagonalelemente:

```
> round(sum(diag(WiB)),4)
[1] 0.3649.
```

Der zu dem Eigenwert von 0,3649 zugehörige Eigenvektor lautet

```
> eigen.vec = eigen(WiB)$vectors[,1]
> eigen.vec
[1]  0.7638740 -0.6453654,
```

der auf die Länge 1 normiert ist:

```
> sum(eigen.vec^2)
[1] 1
> eigen.vec*eigen.vec
[1] 0.5835036 0.4164964
> eigen.vec%*%eigen.vec
    [,1]
[1,]    1.
```

Die Komponenten der auf die Länge eins normierten Eigenvektoren der Matrix $\mathbf{W}^{-1}\mathbf{B}$ heißen rohe Diskriminanzkoeffizienten ($\mathbf{a}$). Eigenvektoren sind jedoch allgemein nur bis auf eine multiplikative Konstante bestimmt. Ein beliebiges Vielfaches des ermittelten Eigenvektors ist daher ebenfalls eine Lösung. Wird der Lösungsvektor so normiert, dass eine einheitliche Varianz der Diskriminanzvariablen gleich eins ist, spricht man von normierten Diskriminanzkoeffizienen ($\tilde{\mathbf{a}}$). Hier wird die Normierung im Hinblick auf die gepoolte Varianz der Diskriminanzvariablen, $(s_d^{pool})^2$, vorgenommen, die dem gewogenen arithmetischen Mittel der gruppenspezifischen Varianzen entspricht. Dazu wird der ermittelte Lösungsvektor mit der Proportionalitätskonstanten $\gamma = \sqrt{(n-G)/\mathbf{a}'\mathbf{W}\mathbf{a}}$ multipliziert:[10]

```
> # Normierte Diskriminanzkoeffizienten
> gamma1 = sqrt((n-G)/t(eigen.vec)%*%W%*%eigen.vec)
> gamma1
     [,1]
[1,] 0.2163993
> a.norm = eigen.vec*gamma1
> a.norm
[1]  0.1653018 -0.1396566.
```

Der erhaltene Vektor a.norm enthält die normierten Diskriminanzkoeffizienten, die die Gewichte der beiden Merkmalsvariablen Einkommen und Alter bei der Bildung der Diskriminanzwerte repräsentieren. Das Einkommen geht mit einem positiven Gewicht von 0,1653 und das Alter mit einem negativen Gewicht von -0,1397 in die Diskriminanzfunktion ein. Da die Käufer der Marke A (Gruppe 1) durch ein niedriges Einkommen und hohes Alter gekennzeichnet sind, werden sie in der Regel niedrige Diskriminanzwerte aufweisen. Umgekehrt weisen Käufer der Marke B (Gruppe 2), die durch ein hohes Einkommen und niedriges Alter gekennzeichnet sind, tendenziell hohe Diskriminanzwerte auf.

Die normierten Diskriminanzkoeffizienten lassen sich direkt mit der Funktion lda ermitteln, die in dem R-Paket MASS verfügbar ist:

```
> library(MASS)
> Kaeufer30.lda = lda(Gruppe~Einkommen+Alter, data=Kaeufer30)
> Kaeufer30.lda
Call:
lda(Gruppe ~ Einkommen + Alter, data = Kaeufer30)
```

[10] Da gamma in R schon als spezielle mathematische Funktion belegt ist, wird hier der Wert des Koeffizienten γ dem R-Objekt gamma1 zugewiesen.

```
Prior probabilities of groups:
      1         2
0.5333333 0.4666667

Group means:
  Einkommen    Alter
1  30.56250 34.93750
2  34.07143 30.71429

Coefficients of linear discriminants:
                LD1
Einkommen  0.1653018
Alter         -0.1396566.
```

Zusätzlich zu den normierten Diskriminanzkoeffizienten enthält der Output die Gruppenmittelwerte und die A-priori-Wahrscheinlichkeiten der Gruppenzugehörigkeit.

Für eine vollständige numerische Spezifikation der Diskriminanzfunktion (6.1) wird noch die Konstante a_0 benötigt, die so gewählt wird, dass der kritische Diskriminanzwert d_c null wird. In Matrixnotation lautet die Bedingung hierfür $\mathbf{a}_0 = -\overline{\mathbf{X}} \cdot \tilde{\mathbf{a}}$:

```
> a0 = -XM.EA%*%a.norm
> a0
        [,1]
 [1,] -0.7187053
 [2,] -0.7187053
...
[30,] -0.7187053.
```

Damit ist die numerisch spezifizierte Diskriminanzfunktion durch

$$D = -0{,}7187 + 0{,}1653 \cdot X_1 - 0{,}1397 \cdot X_2$$

gegeben. Bei einem tendenziell niedrigen Einkommen und hohem Alter weisen die Käufer der Marke A (Gruppe 1) überwiegend negative Diskriminanzwerte auf:

```
> Kaeufer30.disval = a0 + X.EA%*%a.norm
> # Diskriminanzwerte der Gruppe 1
> Kaeufer30.disval[1:16]
 [1] -1.2575492 -0.4081790 -2.1724243  1.2077632 -1.5482930 -0.7359986
 [7] -1.8047449 -0.2286626 -1.6252286  0.9655258 -0.5194064 -1.2461185
[13] -0.5564822  0.4838350  0.5864158  0.1248022.
```

Umgekehrt weisen Käufer der Marke B (Gruppe 2) bei einem tendenziell hohen Einkommen und niedrigen Alter mehrheitlich positive Diskriminanzwerte auf:

```
> # Diskriminanzwerte der Gruppe 2
> Kaeufer30.disval[17:30]
 [1]  0.7860094  2.2879402  0.1532314 -1.1435378  0.2187364  1.1679034
 [7]  1.2590536  0.8771596  1.6665928  1.8291106 -0.3284594 -0.1915868
[13]  0.5836318 -0.4310402.
```

Alternativ können die Diskriminanzwerte unter Verwendung der mit der Funktion lda bestimmten normierten Diskriminanzkoeffizienten mit Hilfe der generischen R-Funktion predict berechnet werden. Das absolute Glied wird hierbei automaisch berücksichtigt:

```
> Kaeufer30.predict = predict(Kaeufer30.lda)
> Kaeufer30.predict$x
1  -1.2575492
2  -0.4081790
3  -2.1724243
...
16  0.1248022
17  0.7860094
18  2.2879402
19  0.1532314
...
30 -0.4310402.
```

Übungsaufgaben

6.2.1 Bestimmen Sie den größten Eigenwert eigval.reg des Produkts aus der inversen Intra-Gruppen-Matrix W.reg und der Inter-Gruppen-Matrix B.reg und den zugehörigen Eigenvektor eigvec.reg! Welche diskriminanzanalytischen Konzepte verkörpern diese Größen?

6.2.2 Berechnen Sie den Vektor der normierten Diskriminanzkoeffizienten anorm.reg unter Verwendung der in den Aufg. 6.1.4 und 6.2.1 ermittelten Ergebnisse! Ermitteln Sie die normierten Diskriminanzkoeffizienten zum Vergleich mit der R-Funktion lda!

6.2.3 Geben Sie die vollständig numerisch spezifizierte Diskriminanzfunktion unter Verwendung einer geeignet gewählten Konstanten a0.reg an!

6.2.4 Berechnen Sie die Diskriminanzwerte der Regionen unter Verwendung der

- Matrizenrechnung (Regionen12.disval) und
- R-Funktion predict

und interpretieren Sie sie!

6.3 Beurteilung der Merkmalsvariablen

Die normierten Diskriminanzkoeffizienten geben die Effekte an, die aus einer Veränderung der Werte der Merkmalsvariablen um eine Einheit auf die Diskriminanzvariable resultieren. Ihre Verwendung ist zur Bestimmung der Diskriminanzwerte der Untersuchungseinheiten zweckmäßig. Da sie von der Maßeinheit und der Streuung der Ausgangsvariablen abhängig sind, eignen sie sich jedoch nicht zur Beurteilung der Bedeutung der einzelnen Variablen für die Diskriminierung. Hierzu muss der Streuungseinfluss ausgeschaltet werden, was durch Multiplikation der normierten Diskriminanzkoeffizienten mit den gepoolten Standardabweichungen der Ausgangsvariablen erfolgt. Speziell lässt sich der Vektor der standardisierten Diskriminanzkoeffizienten ($\mathbf{a}^*$) aus der Beziehung

$$(6.6) \quad \mathbf{a}^* = \frac{1}{\sqrt{n-G}} \cdot \mathrm{diag}(\mathbf{W})^{1/2} \cdot \tilde{\mathbf{a}}$$

berechnen.

In dem Zwei-Gruppen-Fall der Käufer der Marken A und B sind die standardisierten Diskriminanzkoeffizienten des Einkommens und Alters absolut nahezu gleich:

```
> # Standardisierte Diskriminanzkoeffizienten
> a.stand = (1/sqrt(n-G))*diag(diag(W^0.5))%*%a.norm
> a.stand
        [,1]
[1,]  0.9951829
[2,] -0.9993691.
```

Allgemein lässt sich die relative Bedeutung der Merkmalsvariablen aufgrund der standardisierten Diskriminanzkoeffizienten anhand ihrer absoluten prozentualen Anteile erkennen, die in unserem Beispiel jeweils bei etwa 50% liegen:

```
> # Relative Bedeutung der Merkmalsvariablen
> # aufgrund der standardisierten Diskriminanzkoeffizienten
> pt.stand = abs(a.stand)/sum(abs(a.stand))*100
> pt.stand
       [,1]
[1,] 49.89506
[2,] 50.10494.
```

Aufgrund von Korrelationen zwischen den Merkmalsvariablen können sich allerdings Verzerrungen ergeben, von denen die standardisierten Diskriminanzkoeffizienten nicht bereinigt sind. Aus diesem Grund wird häufig zusätzlich auf Strukturkoeffizienten ($\mathbf{r}_{dx}^{pool}$) zurückgegriffen, die die Korrelationen zwischen den einzelnen Merkmals-

variablen und der Diskriminanzvariablen widerspiegeln. Man erhält sie aus der Beziehung

(6.7) $$\mathbf{r}_{dx}^{pool} = \frac{1}{\sqrt{n-G}} \cdot \mathrm{diag}(\mathbf{W})^{-1/2} \cdot \mathbf{W} \cdot \tilde{\mathbf{a}}:$$

```
> rdx.pool = (1/sqrt(n-G))*diag(diag(W^-0.5))%*%W%*%a.norm
> rdx.pool
        [,1]
[1,]  0.4982252
[2,] -0.5044930.
```

In unserem Beispiel von Käufern der Marken A und B ändert sich die relative Bedeutung der beiden Variablen Einkommen und Alter hierdurch praktisch nicht:

```
> # Relative Bedeutung der Merkmalsvariablen
> # aufgrund der Strukturkoeffizienten
> pt.rdx = abs(rdx.pool)/sum(abs(rdx.pool))*100
> pt.rdx
       [,1]
[1,] 49.68746
[2,] 50.31254.
```

Übungsaufgaben

6.3.1 Berechnen Sie die standardisierten Diskriminanzkoeffizienten astand.reg und interpretieren Sie sie!

6.3.2 Berechnen Sie die Strukturkoeffizienten rdx.reg und interpretieren Sie sie!

6.3.3 Vergleichen Sie die mit den standardisierte Diskriminanzkoeffizienten und de Strukturkoeffizienten ermittelten Bedeutungsrangfolgen der Merkmalsvariablen miteinander!

6.4 Güte der Diskrimination

Nach Ermittlung einer optimalen Diskriminanzfunktion auf der Grundlage des Diskriminanzkriteriums (6.4) stellt sich die Frage nach der Güte der Diskrimination. Darüber gibt die Beurteilung der Merkmalsvariablen, aus der ihre Bedeutungsrangfolge hervorgeht, noch keinen Aufschluss. Hierfür wurden verschiedene Maßzahlen und Tests konzipiert.

Das Diskriminanzkriterium (= Eigenwert) λ selbst kommt nicht unmittelbar für eine Beurteilung der Trennfähigkeit der Gruppen durch die verwendeten Merkmalsvariablen in Betracht, da es als Verhältnis der Inter- und Intra-Gruppen-Streuung auf der Diskriminanzachse (B_d/W_d) nicht normiert ist. Jedoch wird der Eigenwert λ als

Baustein eines Gütemaßes verwendet, dem eine ähnliche Interpretation zukommt wie dem Determinationskoeffizienten (Bestimmtheitsmaß) in der Regressionsanalyse.

Der quadrierte kanonische Korrelationskoeffizient r_c^2, der als Verhältnis der Streuung zwischen den Gruppen (B_d) und der totalen Streuung (T_d) auf der Diskriminanzachse auf das Intervall [0, 1] normiert ist, lässt sich unter Verwendung des Eigenwerts λ durch $r_c^2 = \lambda/(1+\lambda)$ berechnen. Üblicherweise wird jedoch in der Diskriminazanalyse der kanonische Korrelationskoeffizient $r_c = \sqrt{\lambda/(1+\lambda)}$ als Gütemaß herangezogen. In unserem Beispiel der beiden Käufergruppen nimmt der kanonische Korrelationskoeffizient den Wert,

```
> rc = sqrt(eigen.val[1]/(1+eigen.val[1]))
> rc
[1] 0.5170547,
```

der einen mittelstarken Zusammenhang zwischen der Gruppenvariablen und der Linearkombination der beiden Merkmalsvariablen Einkommen und Alter widerspiegelt.

Die gebräuchlichste Kenngröße zur Charakterisierung der Trennfähigkeit ist Wilks Lambda Λ. Da diese Kenngröße einen Bezug zwischen der Streuung innerhalb der Gruppen (W_d) und der totalen Streuung (T_d) auf der Diskriminanzachse herstellt, handelt es sich um ein inverses Gütemaß. Unter Verwendung des Eigenwerts λ ist Wilks Lambda durch $\Lambda = 1/(1+\lambda)$ gegeben. Der relativ hohe Λ-Wert von

```
> WL = 1/(1+eigen.val[1])
> WL
[1] 0.7326544
```

weist aus, dass die Gruppentrennung allein durch die beiden Merkmale Einkommen und Alter allenfalls schwach gelungen ist. Der Anteil der nicht-erklärten Streuung beträgt 73,3%.

Wilks Lambda bildet die Grundlage verschiedener Signifikanztests in der Diskriminanzanalyse. Ein Standardsignifikanztest in der Diskriminanzanalyse bezieht sich auf die Nullhypothese gleicher Gruppenmittelwerte auf der Diskriminanzachse. Er ist in dem R-Paket rrcov verfügbar,

```
> library(rrcov),
```

und wird mit der Funktion Wilks.test aufgerufen:

```
> Kaeufer30.WLtest = Wilks.test(Gruppe~Einkommen+Alter, data=Kaeufer30)
> Kaeufer30.WLtest

    One-way MANOVA (Bartlett Chi2)

data:  x
Wilks' Lambda = 0.7326, Chi2-Value = 8.4, DF = 2.0, p-value = 0.015
sample estimates:
  Einkommen    Alter
1   30.56250   34.93750
2   34.07143   30.71429.
```

Zusätzlich zum Testergebnis werden hierbei die Gruppenmittelwerte ausgegeben. Die χ^2-verteilte Prüfgröße Chi2-Value, die sich unter Verwendung von Wilks Lambda berechnet, nimmt in dem Käufer-Beispiel den Wert 8,4 an. Bei einem p-Wert von 0,015 ist die Prüfgröße auf dem 5%-Niveau signifikant. Trotz der durch die deskriptiven Gütemaße indizierten eher schwachen Trennschärfe der Merkmale Einkommen und Alter lassen sich mit ihnen die beiden Käufergruppen doch signifikant diskriminieren.

Das Testergebnis wird durch Hotellings T^2-Test, der die multivariate Verallgemeinerung des univariaten Zwei-Stichproben-Tests über Differenzen von Erwartungswerten ist, bestätigt:

```
> Kauefer30.T2test = T2.test(X1.EA, X2.EA)
> Kauefer30.T2test

    Two-sample Hotelling test

data:  X1.EA and X2.EA
T2 = 10.2182, F = 4.9266, df1 = 2, df2 = 27, p-value = 0.015
alternative hypothesis: true difference in mean vectors is not equal to (0,0)
sample estimates:
                 Einkommen    Alter
mean x-vector    30.56250   34.93750
mean y-vector    34.07143   30.71429.
```

Hier nimmt die aus Hotellings T^2 gebildete Prüfgröße F den Wert 4,9266 an, der bei gleichem p-Wert ebenfalls auf dem 5%-Niveau signifikant ist.

Übungsaufgaben

6.4.1 Berechnen Sie den kanonischen Korrelationskoeffizienten rc.reg und Wilks Lambda WL.reg für das Regionen-Beispiel und interpretieren Sie die beiden Gütemaße!

6.4.2 Testen Sie Wilks Lambda WL.reg auf Signifikanz (α=0,05) und interpretieren Sie das Testergebnis!

6.4.3 Welches Ergebnis weist Hotellings T^2-Test im Hinblick auf die Relevanz der verwendeten Merkmalsvariablen für eine Trennung zwischen ländlichen und städtischen Regionen aus (α=0,05)?

6.5 Klassifikation

In der Diskriminanzanalyse zielt die Klassifikation letztlich auf die Zuordnung neuer Objekte zu bestehenden Gruppen ab (externe Analyse). Angewandt auf die Objekte bestehender Gruppen bietet sie aber zusätzlich die Möglichkeit, die prognostische Relevanz der Diskrimination zu bewerten (interne Analyse).

Zur Klassifikation neuer Objekte ist in R die Funktion predict verfügbar. Hierfür definieren wir ein neues Objekt, deren Merkmalswerte im Unterschied zur Gruppenzugehörigkeit bekannt sind:

```
> Einkommen = 28
> Alter = 32
> Objneu = as.data.frame(cbind(Einkommen, Alter))
> Objneu
  Einkommen Alter
1        28    32.
```

Dabei ist darauf zu achten, dass das neu definierte Objekt Objneu vom Type data.frame ist, dessen Variablenbezeichnungen identisch mit denen des Datensatzes Kaeufer30 ist, für den die Diskriminanzfunktion geschätzt wurde. Nach Laden des R-Pakets MASS,

```
> library(MSS),
```

lässt sich die Funktion predict auf das lda-Objekt Kaeufer30.lda anwenden. Wählt man für das Argument data die Daten des neuen Objekts Objneu, erhält man die gewünschten Zuordnungsinformationen:

```
> predict(Kaeufer30.lda, data=Objneu)
```

```
$class
[1] 1
Levels: 1 2

$posterior
        1         2
1 0.6970725 0.3029275

$x
      LD1
1 -0.5592662.
```

Aufgrund des negativen Diskriminanzwertes LD1 ist das neue Objekt der Gruppe 1 zuzuordnen. Diese Zuordnung wird auch durch die höhere A-posteriori-Wahrscheinlichkeit des neuen Objekts für die Gruppe 1 im Vergleich zur Gruppe 2 angezeigt. Bei einem Wert von 0,697 ist die Wahrscheinlichkeit, dass ein Objekt mit dem gegebenen Merkmalsvektor der Gruppe 1 angehört, mehr als doppelt so groß wie die Zugehörigkeitswahrscheinlichkeit zur Gruppe 2.

Zur Beurteilung der Klassifikationsgüte kann die Trefferquote herangezogen werden. Ist die Gruppenzugehörigkeit der klassifizierten Objekte bekannt, kann durch Vergleich der vorhergesagten mit der tatsächlichen Gruppenzugehörigkeit eine Beurteilung der Klassifikation erfolgen. Der Prozentsatz richtig klassifizierter Objekte ist demnach eine Maßzahl für die Klassifikationsgüte. Die Resubstitutionsmethode vermittelt jedoch im Allgemeinen ein zu optimistisches Bild über die Güte der Klassifikation, da die Trefferquote auf Basis derselben Daten ermittelt wird, die in die Berechnung der Klassifikationsvorschriften eingehen.

Das Problem der Überschätzung der Trefferquote wird durch eine Kreuzvalidierung vermieden, für die die vorhandene Stichprobe in eine Lern- und Kontrollstichprobe aufgeteilt wird. Anhand der Lernstichprobe wird die Klassifikationsvorschrift ermittelt, die dann auf die Kontrollstichprobe angewandt wird. Sofern keine großen Stichproben vorliegen, kann die Datenbasis zur Ermittlung der Trefferquote hierbei jedoch stark reduziert werden. Die Trefferquote kann ein hohes Maß an Instabilität aufweisen, wenn sie nur auf einer geringen Kontrollstichprobe basiert. Dies trifft auch auf eine Klassifikationsregel zu, die aus einer kleinen Lernstichprobe ermittelt worden ist.

Einen Ausweg aus dieser Situation bietet die Leave-one-out-Methode, die als Spezialfall der Kreuzvalidierung in R implementiert ist. Hiebei wird ein Klassifikationsverfahren bei gegebener Größe n des ursprünglichen Datensatzes jeweils auf n-1 Untersuchungseinheiten angewendet und für die n-te Untersuchungseinheit getestet. Die Lernstichprobe besteht folglich aus n-1 Objekten, wobei das nicht berücksichtigte Objekt klassifiziert wird. Die Klassifikation eines Objektes basiert also auf den übrigen n-1 Objekten und wird n-mal, d.h. für jedes Objekt, durchgeführt. Dabei wird

jedes Mal eine neue Klassifikationsregel bestimmt. Hierdurch wird die Trefferquote fast unverzerrt geschätzt.

In R lässt sich die Leave-one-out-Methode für einen gegebenen Datensatz einsetzen, indem man den Befehl lda um das Argument CV=TRUE erweitert:

```
> Kaeufer30.ldaCV = lda(Gruppe~Einkommen+Alter, data=Kaeufer30, CV=TRUE).
```

Hierbei ergeben sich die Zuordnungswahrscheinlichkeiten

```
> Kaeufer30.ldaCV$posterior
          1          2
1  0.83066137 0.16933863
2  0.61098418 0.38901582
3  0.93775479 0.06224521
4  0.16793233 0.83206767
5  0.86971242 0.13028758
6  0.73064153 0.26935847
7  0.90276846 0.09723154
8  0.60422739 0.39577261
9  0.88200586 0.11799414
10 0.23100304 0.76899696
11 0.67151062 0.32848938
12 0.82273857 0.17726143
13 0.64844884 0.35155116
14 0.38057491 0.61942509
15 0.31359512 0.68640488
16 0.38358871 0.61641129
17 0.37806198 0.62193802
18 0.07822597 0.92177403
19 0.50682674 0.49317326
20 0.87496881 0.12503119
21 0.56838760 0.43161240
22 0.25664480 0.74335520
23 0.22869891 0.77130109
24 0.30804419 0.69195581
25 0.16325048 0.83674952
26 0.13314004 0.86685996
27 0.68960718 0.31039282
28 0.64480978 0.35519022
29 0.39954567 0.60045433
30 0.68965788 0.31034212,
```

aus denen sich in Verbindung mit der Variablen Gruppe des data-frame-Objekts Kaeufer30 eine Kreuzklassifikation der tatsächlichen und der mit Hilfe der Klassifikationsregeln ermittelten Gruppenzugehörigkeit erstellen lässt:

```
> CV.table = table(Kaeufer30$Gruppe, Kaeufer30.ldaCV$class)
> CV.table

     1  2
 1  11  5
 2   6  8
```

Danach werden aufgrund der Klassifikationsregeln 11 der 16 Käufer der Marke A (Gruppe 1) und 8 der 14 Käufer der Marke B (Gruppe 2) korrekt klassifiziert. Im ersten Fall kommt es jedoch zu 5, im zweiten Fall zu 6 Fehlklassifikationen.

Trefferquoten lassen sich getrennt für beide Gruppen,

```
> diag(prop.table(CV.table, 1))
      1         2
0.6875000  0.5714286,
```

und für den Datensatz insgesamt,

```
> sum(diag(prop.table(CV.table)))
[1] 0.6333333,
```

ausweisen.

Demnach werden in dem Käufer-Beispiel 63,3% der Beobachtungen aufgrund ihrer Diskriminanzwerte korrekt klassifiziert. Hierin spiegelt sich die vergleichsweise geringe Trennfähigkeit der beiden Merkmalsvariablen Einkommen und Alter nieder, die mit dem kanonischen Korrelationskoeffizienten und Wilks Lambda indiziert worden ist.

Übungsaufgaben

6.5.1 Definieren Sie eine neue Region Objneu.reg, das die Merkmalswerte 200, 18020, 6.8, 53, 11, 2.5 in den für den Datensatz Regionen12 definierten Variablen X_1 bis X_6 aufweist, als ein Objekt des Typs data-frame!

6.5.2 Bestimmen Sie unter Verwendung der Funktion predict die A-posteriori-Wahrscheinlichkeiten und den Diskriminanzwert der neuen Region Objneu.reg! Welche Schlüsse im Hinblick auf die Gruppenzugehörigkeit des neuen Objekts lassen sich daraus ziehen?

6.5.3 Ermitteln Sie die mit der Leave-one-out-Methode ermittelten Häufigkeiten der korrekten Zuordnungen und Fehlklassifikationen für den Datensatz Regionen12 in einer Kreuztabelle dar! Geben Sie die Trefferquoten innerhalb der beiden Gruppen und die Gesamtrefferquote an und interpretieren Sie sie!