Hauptkomponentenanalyse mit R

(prinicipal components analysis: PCA; analyse en composantes principales: ACP). Verfasst von Paul Ruppen, Version 24. Juni 2022

Das Ziel der Hauptkomponentenanalyse (PCA) besteht darin, p metrisch skalierte, korrelierte Variablen mit n Daten (= Zeilen der Datenmatrix) durch ein kleinere Anzahl unkorrelierter Variablen (= Hauptkomponenten) zu ersetzen, welche möglichst viel Information der ursprünglichen Variablenmenge erfassen. Dies kann die Interpretation der Daten erleichtern, da es eventuell einfacher ist, zwei oder drei Variablen zu interpretieren als z.B. 20 oder 30 Variablen mit komplizierten Interaktionsmustern, s. (Bartholomew, Steele, Galbraith & Moustaki, 2008, S. 117 ff.). Information wird dabei als Varianz definiert. Es geht darum, die erste Hauptkomponente so festzulegen, dass sie maximal viel Varianz der totalen Varianz erfasst. Die zweite Hauptkomponente wird dann orthogonal zur ersten so festgelegt, dass sie maximal viel Varianz der verbleibenden Varianz erfasst, usw. Es handelt sich um ein rein datenmanipulatives Verfahren ohne stochastischen Hintergrund. Wird das Verfahren zu Ende geführt, resultieren $m \leq p$ Variablen, auf welche die gesamte ursprüngliche Varianz aufgeteilt ist. Dabei ist m die Dimension des durch die p Variablen aufgespannten Raums. Da Hauptkomponenten orthogonal zueinander liegen, sind diese unkorreliert. Oft weisen viele der zuletzt berechneten Hauptkomponenten so wenig Varianz auf, dass man sie ohne bedeutenden Informationsverlust ausser Acht lassen kann. Die verbleibenden Hauptkomponenten mit viel Varianz gilt es dann sinnvoll inhaltlich zu interpretieren − ein intuitives und nicht rechnerisches Verfahren.

Es ist üblich, die ursprünglichen Daten für die PCA zu standardisieren (Mittelwert der standardisierten Daten ist 0, die Varianz 1). Dies erfolgt, um den Einfluss der Skalierung auf die Varianzen bei den Variablen vergleichbar zu halten. Eine Variable in kg oder dieselbe Variable in t gemessen führte zu sehr unterschiedlichen Ergebnissen bei der PCA! Bei standardisierten Variablen beträgt die Gesamtvarianz $m \leq p$. Die PCA setzt metrisch skalierte Daten voraus. Sie wird aber in Psychologie und Erziehungswissenschaften fast immer auf Daten angewendet, die dieser Anforderung nicht genügen. Die PCA wird in diesen Wissenschaften oft „Faktorenanalyse“ genannt, wobei dieser Name sehr unterschiedliche Verfahren bezeichnen kann.

1 Berechnung − Theorie

Seien p standardisierte Daten-Variablen $Z_i \in \mathbb{R}^{n\times 1}$ in einem p-dimensionalen Raum gegeben, den sie aufspannen. Wegen der Standardisierung gilt $\bar{Z}_i = 0$ und − $\mathrm{Var}(Z_i) = 1$. Zu finden sind die neuen Variablen $H_1, ..., H_p$ mit $H_i \in \mathbb{R}^{n\times 1}$ und $\mathbf{H} = (H_1, ..., H_p) \in \mathbb{R}^{n\times p}$,

die sogenannten Hauptkomponenten, so dass mit $\mathbf{Z} = (Z_1, ..., Z_p) \in \mathbb{R}^{n \times p}$ und $\mathbf{a}_j \in \mathbb{R}^{p \times 1}$:

$$H_j = a_{1j}Z_1 + ... + a_{pj}Z_p = \mathbf{Z}\mathbf{a}_j$$
$$H_j^t H_i = 0 \text{ für } j \neq i,$$

und die Varianz

$$\text{Var}(H_j) = \text{Var}(\mathbf{Z}\mathbf{a}_j) = \mathbf{a}_j^t \text{Var}(\mathbf{Z})\,\mathbf{a}_j$$

jeweils maximal ist. Sei $\Sigma :=\text{Var}(\mathbf{Z}) \in \mathbb{R}^{p \times p}$.

Gesucht ist in einem ersten Schritt die Hauptkomponente H_1 mit maximaler Varianz

$$\text{Var}(H_1) = \text{Var}(\mathbf{Z}\mathbf{a}_1) = \mathbf{a}_1^t \text{Var}(\mathbf{Z})\,\mathbf{a}_1 = \mathbf{a}_1^t \Sigma \mathbf{a}_1$$

Setzt man die Nebenbedingung $\mathbf{a}_i^t \mathbf{a}_i = 1$, so ergeben sich orthogonale H_i, wie sich in Kürze zeigen wird. Es ist also eine Maximierungsaufgabe unter Nebenbedingung zu lösen. Dazu verwendet man die Lagrange-Methode:

Bemerkung 1. *Lagrange-Methode: Man maximiert $f(\mathbf{y})$ unter der Nebenbedingung $g(\mathbf{y}) = c$, indem man $L(\mathbf{y}, \lambda) = f(\mathbf{y}) - \lambda[g(\mathbf{y}) - c]$ maximiert − mit dem Lagrangemultiplikatoren λ. Man leitet $L(\mathbf{y}, \lambda)$ ab und bestimmt die Nullstellen.*

Im vorliegenden Fall erhält man:

$$L(\mathbf{a}_1) = \mathbf{a}_1^t \Sigma \mathbf{a}_1 - \lambda_1 \left(\mathbf{a}_1^t \mathbf{a}_1 - 1\right)$$

$$\frac{\partial L(\mathbf{a}_1)}{\partial \mathbf{a}_1} = 2\Sigma\mathbf{a}_1 - 2\lambda_1 \mathbf{a}_1$$

und

$$2\Sigma\mathbf{a}_1 - 2\lambda_1\mathbf{a}_1 = 0$$
$$(\Sigma - \lambda_1 \mathbf{I})\,\mathbf{a}_1 = 0.$$

Es entsteht das Eigenwertproblem, denn damit gilt $\Sigma\mathbf{a}_1 = \lambda_1\mathbf{a}_1$.

Wenn p der Rang der Matrix Σ ist, wird Σ p Eigenwerte haben. Man nehme an, dass alle verschieden sind und nach Grösse absteigend $\lambda_1 > \lambda_2 > ... > \lambda_p$ geordnet sind. Welcher Eigenwert muss nun zur Bestimmung der ersten Hauptkomponente gewählt werden? Wegen $\mathbf{a}_1^t \mathbf{a}_1 = 1$, gilt

$$\text{Var}(H_1) = \text{Var}(\mathbf{Z}\mathbf{a}_1) = \mathbf{a}_1^t \Sigma \mathbf{a}_1 = \mathbf{a}_1^t \lambda_1 \mathbf{a}_1 = \lambda_1 \mathbf{a}_1^t \mathbf{a}_1 = \lambda_1$$

Die Varianz auf der ersten Hauptkomponente wird also am grössten, wenn man für λ_1 den grössten Eigenwert wählt. Diesem entspricht dann der Eigenvektor $\mathbf{a}_1$, der zur Berechnung der ersten Hauptkomponente $H_1 = \mathbf{Z}\mathbf{a}_1$ verwendet wird.

Als nächstes wird die zweite Hauptkomponenten H_2 mit der zweitgrössten Varianz berechnet. Es soll wieder gelten $\mathbf{a}_2^t \mathbf{a}_2 = 1$. Zudem sollen H_1 und H_2 unkorreliert sein, d.h. der folgende Ausdruck soll 0 ergeben:

$$\text{Cov}(H_1, H_2) = \text{Cov}(\mathbf{Z}\mathbf{a}_2, \mathbf{Z}\mathbf{a}_1) = \mathbf{a}_2^t \text{Cov}(\mathbf{Z}, \mathbf{Z})\,\mathbf{a}_1 = \mathbf{a}_2^t \Sigma \mathbf{a}_1 \quad (1)$$

Da $\Sigma \mathbf{a}_1 = \lambda_1 \mathbf{a}_1$, erhält man

$$\mathbf{a}_2^t \Sigma \mathbf{a}_1 = \lambda_1 \mathbf{a}_2^t \mathbf{a}_1$$

was 0 ergibt, wenn $\mathbf{a}_2^t \mathbf{a}_1 = 0$ (zweite Nebenbedingung). Man erhält die zu maximierende Funktion

$$L(\mathbf{a}_2) = \left(\mathbf{a}_2^t \Sigma \mathbf{a}_2\right) - \lambda_2 \left(\mathbf{a}_2^t \mathbf{a}_2 - 1\right) - \delta \left(\mathbf{a}_2^t \mathbf{a}_1 - 0\right)$$

und deren Ableitung

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{a}_2} &= 2\Sigma \mathbf{a}_2 - 2\lambda_2 \mathbf{a}_2 - \delta \mathbf{a}_1 \\ &= 2\left(\Sigma - \lambda_2 \mathbf{I}\right) \mathbf{a}_2 - \delta \mathbf{a}_1 \end{aligned}$$

Man setzt mit 0 gleich:

$$2\left(\Sigma - \lambda_2 \mathbf{I}\right) \mathbf{a}_2 - \delta \mathbf{a}_1 = 0$$

Man multipliziert von links her mit $\mathbf{a}_1^t$ und erhält:

$$\begin{aligned} \mathbf{a}_1^t \left(2\left(\Sigma - \lambda_2 \mathbf{I}\right) \mathbf{a}_2 - \delta \mathbf{a}_1\right) &= 2\mathbf{a}_1^t \Sigma \mathbf{a}_2 - 2\lambda_2 \mathbf{a}_1^t \mathbf{I} \mathbf{a}_2 - \delta \mathbf{a}_1^t \mathbf{a}_1 \\ &= 2\mathbf{a}_1^t \Sigma \mathbf{a}_2 - \delta \\ &= 0 \end{aligned}$$

Da (s. (1)) $\mathbf{a}_1^t \Sigma \mathbf{a}_2 = Cov(Z_1, Z_2) = Cov(Z_2, Z_1) = \mathbf{a}_2^t \Sigma \mathbf{a}_1 = 0$, gilt $\delta = 0$. Man erhält als zu lösende Gleichung:

$$\left(\Sigma - \lambda_2 \mathbf{I}\right) \mathbf{a}_2 = 0$$

Es ist der zweitgrösste Eigenwert mit seinem normierten Eigenvektor zu wählen, um $H_2 = \mathbf{Z}\mathbf{a}_2$ zu berechnen. Mit analogen Überlegungen erhält man die übrigen Hauptkomponenten.

Fasst man die Eigenvektoren als Spalten zu einer Matrix $\mathbf{A}$ und die Hauptkomponenten zu einer Matrix $\mathbf{H}$ zusammen, gilt

$$\mathbf{H} = \mathbf{Z}\mathbf{A}$$

Bemerkung 2. *Gemäss eben erfolgter Herleitung gilt*

$$\Sigma \mathbf{a}_i = \lambda_i \mathbf{a}_i \tag{2}$$

Durch Zusammenfassen der Eigenvektoren $\mathbf{a}_i$ erhält man die Matrix $\mathbf{A}$ und durch Aufreihung der Eigenwerte $\lambda_1, ..., \lambda_p$ auf der Diagonalen einer Diagonalmatrix erhält man eine Eigenwert-Matrix $\mathbf{\Lambda}$. Damit gilt dann gemäss (2).

$$\Sigma \mathbf{A} = \mathbf{A}\mathbf{\Lambda}$$

Durch Multiplikation auf beiden Seiten von rechts her durch $\mathbf{A}^{-1}$ erhält man (für orthogonale Matrizen gilt $\mathbf{A}^{-1} = \mathbf{A}^t$):

$$\begin{aligned} \Sigma &= \mathbf{A}\mathbf{\Lambda}\mathbf{A}^{-1} \\ &= \mathbf{A}\mathbf{\Lambda}\mathbf{A}^t \end{aligned}$$

Durch Multiplikation auf beiden Seiten von links her mit $\mathbf{A}^{-1}$ *gilt*

$$\begin{aligned}\mathbf{A}^{-1}\Sigma\mathbf{A} &= \mathbf{A}^t\mathbf{\Sigma}\mathbf{A}\\ &= \mathbf{\Lambda}\end{aligned}$$

Schliesslich gilt für

$$\mathbf{H} = \mathbf{Z}\mathbf{A}$$

$$Var(\mathbf{H}) = \mathbf{\Lambda}$$

denn

$$Var(\mathbf{H}) = Var(\mathbf{Z}\mathbf{A}) = \mathbf{A}^t\, Var(\mathbf{Z})\, \mathbf{A} = \mathbf{A}^t\Sigma\mathbf{A} = \mathbf{\Lambda},$$

Damit stellt die Matrix der Eigenvektoren einen Zusammenhang zwischen den beiden Kovarianzmatrizen Σ *und* $\mathbf{\Lambda}$ *her.*

Bemerkung 3. *Werden die Hauptkomponenten verwendet, um die alten Variablen zu ersetzen und weitere Berechnungen anzustellen, werden, diese oft standardisiert. Sie werden also durch die Standardabweichung (die Wurzel aus den Eigenwerten) dividiert.*

$$\mathbf{H}^s = \mathbf{H}\mathbf{\Lambda}^{-\frac{1}{2}}$$

Möchte man die berechneten Hauptkomponenten als Index verwenden, um bisher nicht untersuchte Objekt i mit Ausprägungen $\mathbf{z}_i$ *zu messen, so rechnet man wegen*

$$\mathbf{H}^s = \mathbf{H}\mathbf{\Lambda}^{-\frac{1}{2}} = \mathbf{Z}\mathbf{A}\mathbf{\Lambda}^{-\frac{1}{2}}$$

mit

$$\mathbf{z}_i^t\mathbf{A}\mathbf{\Lambda}^{-\frac{1}{2}}.$$

Bemerkung 4. *Bei quadratischen Matrizen gilt:* $Spur(BC) = Spur(CB)$ *– die Spur ist die Summe der Diagonaleinträge einer quadratischen Matrix. Es gilt:*

$$Spur(\Lambda) = Spur\left(\mathbf{A}^t\Sigma\mathbf{A}\right) = Spur\left(\Sigma\mathbf{A}\mathbf{A}^t\right) = Spur(\Sigma) = \sum_{i=1}^{m} Var(Z_i) = p$$

Damit ist die Summe der Varianzen der ursprünglichen standardisierten Variablen mit der Summe der Varianzen der Hauptkomponenten identisch. Damit kann man sagen, dass die i-te Hauptkomponente den Anteil

$$\frac{\lambda_i}{\sum_{j=1}^{p}\lambda_j}$$

an der Gesamtvarianz der ursprünglichen standardisierten Variablen auf sich trägt. Oder man sagt, dass die ersten m Hauptkomponenten

$$\frac{\sum_{j=1}^{m}\lambda_j}{\sum_{j=1}^{p}\lambda_j}100\%$$

der Gesamtvarianz erklären.

Es stellt sich die Frage, welche Hauptkomponenten man berücksichtigen möchte. Es gibt dafür verschiedene, nicht äquivalente Kriterien:

- Man stellt die Eigenwerte λ_j auf der Ordinate und den Index j auf der Abszisse dar. Weist der Graph einen deutlichen Knick auf, so werden die Eigenwerte links vom Knick (inklusive Knick oder ohne Knick) als wichtig betrachtet. Dieses Verfahren heisst Scree-Test.

- Man wählt die Hauptkomponenten, deren Eigenwerte grösser als 1 sind. Die Varianzen dieser Hauptkomponenten sind dabei grösser als die Varianzen der standardisierten Ursprungsvariablen, die ja mit 1 identisch sind.

- Man setzt eine prozentuale Grösse als Schwelle. Man verwendet die Hauptkomponenten, so dass der prozentuale Anteil der Summe derer Eigenwerte diese Schwelle knapp überschreitet. Man wählt also das $\min(m_1)$, so dass z.B.

$$\frac{1}{m}\sum_{i}^{m_1}\lambda_i > 0.9$$

- unter der Voraussetzung, dass die Daten eine multivariate Normalverteilung besitzen, kann ein Test von Bartlett (s. z.B. (Fahrmeir, Hamerle & Tutz, 1996, S. 670)) durchgeführt werden. Es handelt sich um einen Test auf Gleichheit von Varianzen. Es wird überprüft, ob sich die Eigenwerte, die man nicht mehr berücksichtigen will, noch signifikant unterscheiden.

  $$H_0 : \lambda_{m_1+1} = ... = \lambda_m$$

  Man testet $\lambda_{m-1} = \lambda_m$, $\lambda_{m-2} = ... = \lambda_m$ etc. bis ein Testergebnis signifikant, H_0 also verworfen wird. Dies sei erstmals der Fall bei m_{m_1+1}. Dann werden die ersten m_{m_1+1} Hauptkomponenten verwendet. Die Teststatistik ist gegeben durch

$$B = (n-1)\left[-\ln\left(\prod_{j=1}^{m}\lambda_j\right) + \ln\left(\prod_{j=1}^{m_1}\lambda_j\right) + (m-m_1)\ln\left(\frac{m-\sum_{i}^{m_1}\lambda_i}{m-m_1}\right)\right]$$

Die Teststatistik ist chi-quadrat-verteilt mit $\frac{(m-m_1+2)(m-m_1-1)}{2}$ Freiheitsgraden. n = Anzahl Datensätze. Jede der Hauptkomponenten muss mit einem Q-Q-Plot auf Normalverteilung überprüft werden. Der Test verhält sich gemäss (Fahrmeir et al., 1996, S. 670) sehr grob und ist entsprechend nicht sehr nützlich. Sind die Voraussetzungen nicht gegeben, kann man den Levene-Test verwenden. Es handelt sich um einen Kruskal-Wallis-Test auf die Beträge der Daten der verbleibenden Achsen. Mit dem folgenden R-Befehl kann man die beiden Tests berechnen (Eingabe: Hauptkomponenten des prcomp-Befehls für x, Voreinstellung ist der Bartlett-Test, bei levene=T wird der Levenetest durchgeführt)

```
bartlett.pca=function(x, levene=F){m = length(x[,1]);n = length(x[1,])
dat=cbind(x[,1],rep(1,m))
for (i in 2:n) dat= rbind(dat,cbind(x[,i],rep(i,m)))
dat=dat[order(dat[,2],decreasing=T),]
ausgabe=matrix(rep(NA,(n-1)*6),nrow=(n-1),ncol=6)
colnames(ausgabe)=c("PCx bis", "PCy", "Testwert","df","p-Wert",Änzahl_Daten")
for (j in 2:n) {
if (levene == F) test=bartlett.test(dat[1:(m*j),1],g=dat[1:(m*j),2]) else
test=kruskal.test(abs(dat[1:(m*j),1]),g=dat[1:(m*j),2] )
ausgabe[j-1,1]=n; ausgabe[j-1,2]=n-(j-1);ausgabe[j-1,3]=test$statistic
ausgabe[j-1,4]=test$parameter; ausgabe[j-1,5]=test$p.value
ausgabe[j-1,6]=j*m}
ausgabe}
```

Wird der obige Befehl auf die obigen dat-Daten ausgeführt, erhält man mit

```
res=prcomp(dat)
bartlett.pca(res$x)
```

die folgenden Ergebnisse:

	PCx bis	PCy	Testwert	df	p-Wert	Anzahl_Daten
1	5	4	154.220	1	0.000	100
2	5	3	262.135	2	0.000	150
3	5	2	328.253	3	0.000	200
4	5	1	427.661	4	0.000	250

Mit

```
res=prcomp(dat)
bartlett.pca(res$x, levene=T)
```

erhält man

	PCx bis	PCy	Testwert	df	p-Wert	Anzahl_Daten
1	5	4	54.513	1	0.000	100
2	5	3	96.085	2	0.000	150
3	5	2	112.877	3	0.000	200
4	5	1	137.425	4	0.000	250

Die Tests scheiden nicht eine Gruppe von Hauptkomponenten aus, deren Varianzen sich nicht unterscheiden.

- Schliesslich können inhaltliche Überlegungen eine Rolle spielen. Man berücksichtigt Achsen, so dass die entsprechenden Hauptkomponenten sinnvoll interpretierbar sind.

2 Einige geometrische Überlegungen

Wird auf eine Datenmatrix $\mathbf{Z}$ die Hauptachsentransformation angewendet, so werden die Punkte, die zeilenweise durch die Koordinaten von $\mathbf{Z}$ gegeben sind, im durch die Eigenvektoren aufgespannten Vektorraum beschrieben. Die Werte der Hauptkomponenten stellen die Koordinaten der Punkte bezüglich der neuen Eigenvektor-Achsen $\mathbf{x}_j := \{x | x = t\mathbf{a}_j,$ mit $t \in \mathbb{R}\}$ dar. Die Varianz der Hauptkomponente H_i wird maximal, wenn die Summe der quadrierten, orthogonalen Distanzen $\|d\|^2$ der Punkte auf die neue Achse minimal wird. Wegen des Satzes von Pythagoras gilt ja

$$\|h_i\|^2 + \|d\|^2 = \|P\|^2$$

Offensichtlich ist $\|h_i\|^2$ maximal, wenn $\|d\|^2$ minimal ist. Es geht bei der PCA dabei natürlich nicht darum, das $\|h_i\|^2$ eines einzelnen Punktes zu maximieren − dieses wäre maximal, wenn $\|d\|^2 = 0$ (die Achse würde also durch den Punkt verlaufen). Es ist die Summe aller quadrierten h_i zu maximieren (s. Abbildung 1).

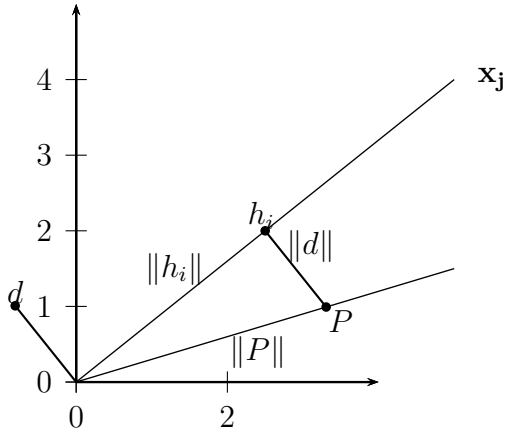

Abbildung 1: Abbildung zur Bemerkung, dass Maximierung der Streuung zur Minimierung der orthogonalen Abstände führt. $\|a\|^2 + \|d\|^2 = \|P\|^2$

Im Gegensatz zur Regression werden bei der Hauptkomponentenmethode also nicht die vertikalen Summenquadrate von einer Gerade minimiert, sondern die kürzesten Abstände der Punkt zu den Hauptachsen gesucht. Dazu noch ein

Beispiel 5. *Die folgenden Daten im zweidimensionalen Raum werden orthogonal auf die erste Hauptachse projiziert (PC2 gibt dann die Koordinaten der Punkte bezüglich der zweiten Achse an, deren Betrag die Distanzen der Punkte von der ersten Achse sind):*

id	*Z1*	*Z2*	*PC1=H_1*	*PC2=H_2*
1	*-0.14798*	*-0.01135*	*-0.12484617*	*-0.05011611*
2	*-1.19731*	*-0.23361*	*1.16296997*	*-0.04507080*
3	*0.99292*	*0.15288*	*-1.29985595*	*0.16958621*
4	*0.53165*	*0.10032*	*-0.86504457*	*0.04051044*
5	*1.81554*	*0.20739*	*-1.96120794*	*0.51385965*
6	*-0.37939*	*-0.25100*	*0.71360788*	*0.50544696*
7	*-1.30782*	*-0.26820*	*1.33112043*	*-0.01201512*
8	*-0.07482*	*0.02778*	*-0.28337141*	*-0.11920569*
9	*0.55637*	*0.13864*	*-0.99160582*	*-0.05583144*
10	*1.43925*	*0.37713*	*-2.22488623*	*-0.20982057*
11	*-0.16901*	*-0.02625*	*-0.06865612*	*-0.01963454*
12	*-1.41266*	*-0.42594*	*1.85398006*	*0.38268111*
13	*-1.95917*	*-0.41652*	*2.16062762*	*0.02123847*
14	*0.99973*	*0.26339*	*-1.62543058*	*-0.14766342*
15	*-0.96308*	*-0.08666*	*0.59240498*	*-0.32929745*

Man erhält die Abbildung 2:

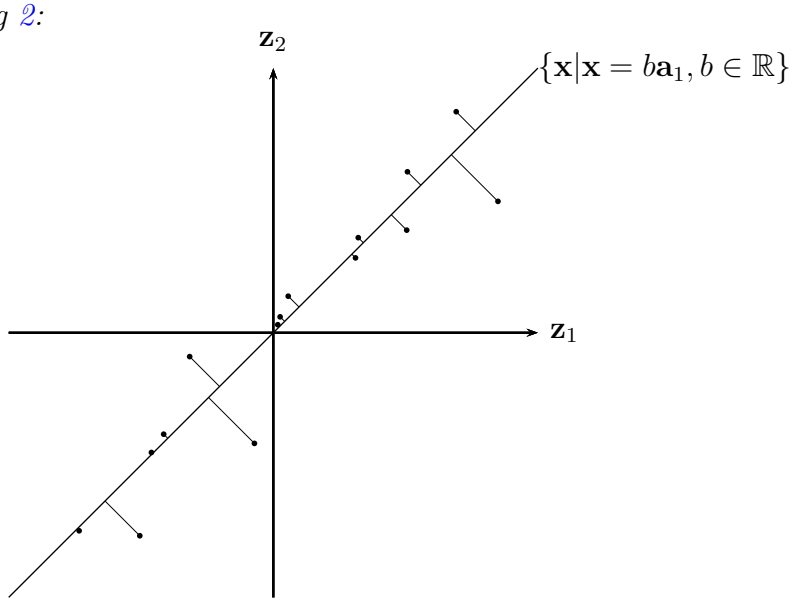

Abbildung 2: Orthogonale Projektion der Daten Z auf die Achse $\{x|x = b\mathbf{a}_1, b \in \mathbb{R}\}$

Die Hauptkomponentenmethode besteht also darin, in einem ersten Schritt das orthogonale Koordinatensystem so zu drehen, dass nachher auf der ersten Achse (durch 1. Eigenvektor aufgespannter Vektorraum) die Koordinaten der Punkte bezüglich dieser Achse maximale Varianz aufweisen.

Beispiel 6. *(Dreidimensionaler Raum). Es hat drei Datenvariablen, z.B. Im ersten Schritt*

	V1	V2	V3
1	0.24	0.91	0.08
2	-0.55	-0.61	-0.41
3	0.88	0.78	0.89
4	-1.74	-1.30	-1.58
5	1.54	1.41	1.55
6	0.43	0.12	0.37
7	-0.06	-0.50	-0.25
8	0.13	-0.50	0.24
9	-0.68	-0.56	-0.82
10	0.47	0.65	0.45
11	-1.62	-1.69	-1.80
12	0.90	0.87	0.86
13	0.46	0.86	0.68
14	0.99	0.97	1.00
15	-1.39	-1.40	-1.26

wird die Achse so gedreht, dass sie sie mitten in und längs der Punktwolke liegt – s. folgende Abbildung 3.

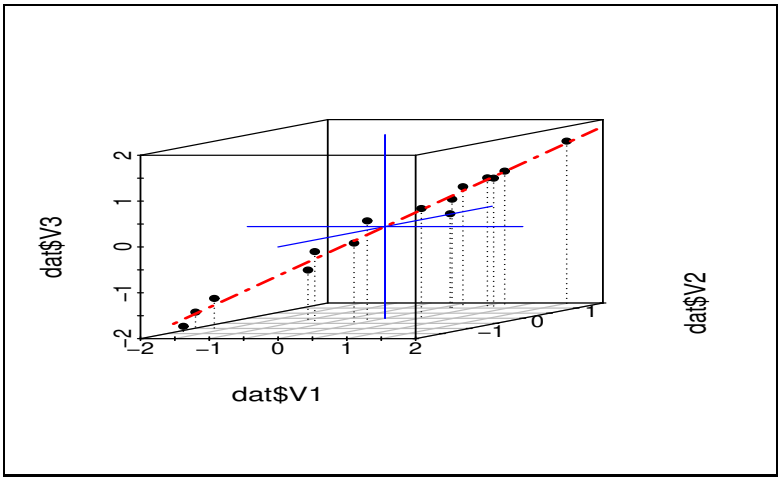

Abbildung 3: Die horizontale Achse wird im dreidimensionalen Raum in die gestrichelte Achse gedreht, die durch den ersten normierten Eigenvektor gegeben ist

Im zweiten Schritt wird die zweite zur ersten Achse orthogonale Achse um die erste Achse so gedreht, dass die Koordinaten der Punkte bezüglich der zweiten Achse maximale Varianz aufweist, etc.

3 Ladungen

Betrachtet man

$$H_j = a_{1j}Z_1 + ... + a_{pj}Z_p = \mathbf{Z}\mathbf{a}_j$$

so sieht man, dass die Komponenten $(a_{1j}, ..., a_{pj})$ der Eigenvektoren $\mathbf{a}_j$ die ursprünglichen Variablen $Z_1, ..., Z_p$ bei der Berechnung der Hauptkomponente H_j gewichten. Man spricht deshalb auch von *Ladungen („loadings")*. Gewöhnlich wird der Ladungsbegriff jedoch anders verwendet. Um den Zusammenhang der ursprünglichen Variablen mit den Hauptkomponenten leichter interpretierbar macht, verwendet man den Ladungsbegriff für die Korrelationen $\text{Kor}(H_j, Z_i)$ zwischen den Hauptkomponenten und den ursprünglichen Variablen. Diese erhält man nämlich alternativ aus den Koeffizienten $\mathbf{a}_j$ mittels

$$\text{Kor}\left(H_j, Z_i\right) = a^*_{ij} := a_{ij}\sqrt{\lambda_j}$$

Die folgenden Überlegungen können dies einsichtig machen.

Wegen

$$\begin{aligned}\mathbf{H} &= \mathbf{Z}\mathbf{A}\\ \mathbf{Z} &= \mathbf{H}\mathbf{A}^{-1} = \mathbf{H}\mathbf{A}^t\\ Z_i &= \sum_{k=1}^{p} a_{ik}H_k\end{aligned}$$

gilt

$$\text{Cov}\left(H_j, Z_i\right) = \text{Cov}\left(H_j, \sum_{k=1}^{p} a_{ik}H_k\right) = \sum_{k=1}^{p} a_{ik}\text{Cov}\left(H_j, H_i\right) = a_{ij}\text{Var}\left(H_j\right) = a_{ij}\lambda_j$$

Damit ist dann, da die Z_i standardisiert sind und entsprechend die Standardabweichung 1 aufweisen,

$$\text{Kor}\left(H_j, Z_i\right) = \frac{\text{Cov}\left(H_j, Z_i\right)}{\sqrt{\text{Var}\left(H_j\right)}\sqrt{\text{Var}\left(Z_i\right)}} = \frac{\lambda_j a_{ij}}{\sqrt{\lambda_j}} = a_{ij}\sqrt{\lambda_j} = a^*_{ij}$$

$a_{ij}\sqrt{\lambda_j} = a^*_{ij}$ liefert also die Korrelation zwischen H_j und Z_i. Die Matrix der Korrelationskoeffizienten zwischen $\mathbf{H}$ und $\mathbf{Z}$ ist dann gegeben durch

$$\text{Kor}\left(\mathbf{H}, \mathbf{Z}\right) = \mathbf{A}\Lambda^{\frac{1}{2}}$$

Damit kann man auf dem Hintergrund einer interpretierbaren Grösse (Pearson-Korrelationskoeffizienten) den Zusammenhang zwischen den standardisierten Originalvariablen und den Hauptkomponenten beurteilen, was für die Interpretation nützlich sein kann. Gewöhnlich werden diese Korrelationen „Ladungen" genannt.

Bemerkung 7. *Die Varianz der j-ten Hauptkomponente ist gleich der Summe der quadrierten Korrelationskoeffizienten zwischen dieser Hauptkomponente und allen p Originalvariablen (die Eigenvektoren sind normiert, d.h.* $\sum_{i=1}^{p} a_{ij}^2 = 1$*):*

$$\sum_{i=1}^{p} Kor(H_j, Z_i)^2 = \sum_{i=1}^{p} \left(a_{ij}^*\right)^2 = \lambda_j \sum_{i=1}^{p} a_{ij}^2 = \lambda_j.$$

Bemerkung 8. *Es gilt*

$$\sum_{j=1}^{p} Kor(H_j, Z_i)^2 = 1$$

d.h. die quadrierte Summe der Korrelationen zwischen den Hauptkomponenten H_j *und der Variable* Z_i *beträgt 1. Denn mit der Korrelationsmatrix* $\Sigma = Kor(\mathbf{Z})$ *der standardisierten Ausgangsariablen* $\mathbf{Z}$ *gilt:*

$$\Sigma = \mathbf{A}\Lambda\mathbf{A}^t = \sum_{j=1}^{p} \lambda_j \mathbf{a}_j \mathbf{a}_j^t$$

Damit ist ($\sigma_{jj} = 1$ *ist die j-te Komponente auf der Diagonalen von* Σ *und damit der Korrelation von* Z_j *mit sich selber):*

$$1 = \sigma_{jj} = \sum_{j=1}^{p} \lambda_j \left(a_{jj}\right)^2 = \sum_{j=1}^{p} \left(\sqrt{\lambda_j} a_{jj}\right)^2 = \sum_{j=1}^{p} \left(a_{jj}^*\right)^2.$$

Berücksichtigt man $k < p$ *Hauptkomponenten gemäss einem der obigen Kriterien und summiert* $\sum_{j=1}^{k} Kor(H_j, Z_i)^2$*, erhält man Zahlen zwischen 0 und 1, die ausdrücken, wie gut die Hauptkomponenten die* Z_i *reproduzieren. Diese Zahlen werden „Kommunalitäten" genannt.*

Bemerkung 9. *Sei* z_{ij} *das i-te Datum der Standardisierung* Z_j *der Variable* X_j*. Damit gilt einerseits, dass*

$$\sum_{i=1}^{n} z_{ij} z_{ik} = \sum_{i=1}^{n} \frac{x_{ij} - \bar{x}_j}{s_j} \frac{x_{jk} - \bar{x}_k}{s_k}$$

bis auf den Faktor $\frac{1}{n}$ *die Korrelation der standardisierten Variablen* Z_j *und* Z_k *ist. Andererseits ist dieser Ausdruck ein Skalarprodukt, womit wegen*

$$\cos(\phi) = \frac{\mathbf{a}^t\mathbf{b}}{\|\mathbf{a}\| \, \|\mathbf{b}\|}$$
$$\mathbf{a}^t\mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \cos(\phi)$$

(ϕ *ist der Winkel zwischen* $\mathbf{a}$ *und* $\mathbf{b}$*) gilt*

$$r_{jk} := \frac{1}{n}\mathbf{z}_j^t\mathbf{z}_k = \frac{1}{n} \|\mathbf{z}_j\| \, \|\mathbf{z}_k\| \cos(\phi)$$

Zuletzt gilt

$$\|\mathbf{z}_j\|^2 = \sum_{i=1}^{n} \left(\frac{z_{ij} - \bar{z}_j}{s_j} \right)^2 = \frac{1}{s_j^2} \frac{n}{n} \sum_{i=1}^{n} (z_{ij} - \bar{z}_j)^2 = \frac{n}{s_j^2} s_j^2 = n$$

und damit

$$\begin{aligned} r_{jk} &= \frac{1}{n} \|\mathbf{z}_j\| \, \|\mathbf{z}_k\| \cos(\phi) \\ &= \frac{1}{n} \sqrt{n} \sqrt{n} \cos(\phi) \\ &= \cos(\phi) \end{aligned}$$

Die Korrelation zwischen den standardisierten Variablen Z_j und Z_k ist also mit dem Winkel zwischen diesen Variablen identisch.

4 Berechnung mit R

In einem ersten Schritt werden die Variablen standardisiert. Wir betrachten die Beispieldaten dat.txt. dat=read.table("pfad\\dat.txt"). Das R-Dataframe dat kann man standardisieren mit

```
Z=scale(dat)
```

Damit ist dann z.B. mean(Z[,1])=0 (bis auf Rundungsfehler) und sd(Z[,1])=1. Als nächstes wird die Korrelationsmatrix berechnet, die im Falle standardisierter Variablen mit der Kovarianzmatrix identisch ist:

```
Sigma=cor(Z)
```

Im Beispiel erhält man:

	V1	V2	V3	V4	V5
V1	1.000	0.924	0.402	0.502	0.617
V2	0.924	1.000	0.413	0.533	0.604
V3	0.402	0.413	1.000	0.105	0.354
V4	0.502	0.533	0.105	1.000	0.367
V5	0.617	0.604	0.354	0.367	1.000

Die Eigenwerte und Eigenvektoren erhält man mit

```
eig=eigen(Sigma)
```

und daraus die Eigenwerte mit

```
eig$values
```

3.02605993 0.90222311 0.54246759 0.45457181 0.07467756

sowie die Eigenvektoren mit

```
eig$vectors
```

	1	2	3	4	5
1	-0.532	-0.043	0.101	0.466	0.698
2	-0.536	-0.056	0.034	0.445	-0.714
3	-0.311	0.783	-0.497	-0.206	0.020
4	-0.369	-0.613	-0.561	-0.416	0.040
5	-0.443	0.079	0.654	-0.607	-0.022

Die Hauptkomponenten erhält man mittels $\mathbf{H} = \mathbf{ZA}$:

```
(H=Z%*%eig$vectors)
```

und die standardisierten Hauptkomponenten erhält mittels $\mathbf{H}^* = \mathbf{ZA}\Lambda^{-\frac{1}{2}}$:

```
(Hstern=H%*%solve(diag(sqrt(eig$values))))
```

Verwendet man Hstern als Messskala für neu erhobene Daten, so kann man fürs Individuum mit den neu erhobenen Daten $\mathbf{z}_n := c(1.5, 1.3, 0.25, 0.63, 0.4)$ berechnen : $\mathbf{z}_n^t\mathbf{A}\Lambda^{-\frac{1}{2}}$:

```
z1%*%eig$vectors%*%solve(diag(sqrt(eig$values)))
```

Man erhält:

[,1]	[,2]	[,3]	[,4]	[,5]
-1.139709	-0.311319	-0.02804978	1.071014	0.51024

Das Untersuchungsobjekt mit den Messwerten $\mathbf{z}_n$ hätte auf der ersten Hauptkomponente einen geschätzten Messwert von -1.14. Man erhält folgende Ladungen:

```
cor(Z,H)
```

	1	2	3	4	5
V1	-0.93	-0.04	0.07	0.31	0.19
V2	-0.93	-0.05	0.02	0.30	-0.20
V3	-0.54	0.74	-0.37	-0.14	0.01
V4	-0.64	-0.58	-0.41	-0.28	0.01
V5	-0.77	0.08	0.48	-0.41	-0.01

Man sieht, dass alle ursprünglichen Variablen mit der ersten Hauptkomponente negativ korreliert sind. Da die Variablen nicht interpretiert sind, kann man diesem Umstand keinen inhaltlichen Sinn geben, ausser dass alle Variablen mit der ersten Hauptkomponente mit gleichem Vorzeichen korreliert sind. Die zweite Hauptkomponente weist positive und negative Korrelationen auf. Drei Korrelationen sind schwach, zwei relativ stark. Die zweite Hauptkomponente unterscheidet einerseits zwischen der Gruppe der positiv korrelierten und der negativ korrelierten, andererseits zwischen den schwach und den stark mit

der zweiten Hauptkomponente korrelierten Variablen. Es werden inhaltliche Beispiele mit entsprechenden Interpretationen folgen.

Wie viele Hauptkomponenten sollte man berücksichtigen? Ein Scree-Plot ergibt mit

```
plot(eig$values,type="l",main= "Scree-Plot",ylab="Eigenwerte",xlab="Hauptkomponenten")
points(eig$values)
```

die Abbildung 4

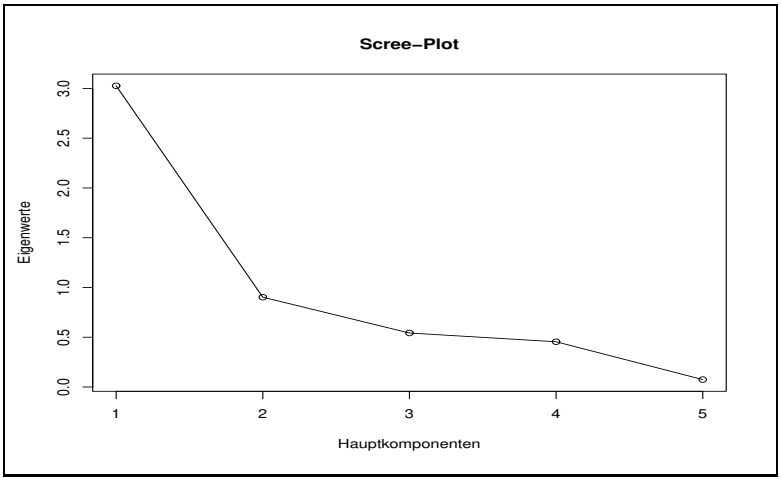

Abbildung 4: Screeplot der dat-Daten-Beispiels

Die Abbildung legt nahe, dass man eine oder zwei Hauptkomponenten berücksichtigen sollte, wobei die zweite bereits einen Eigenwert unter 1 aufweist. Man könnte noch die Kommunalitäten für die beiden Fälle analysieren:

```
komm=cor(Z,H)^2
komm[,1]
rowSums(komm[,1:2])
```

	Kommunalität HK1	Kommunalität HK1 und HK2
V1	0.8576380	0.8592843
V2	0.8681997	0.8710733
V3	0.2933104	0.8468269
V4	0.4124163	0.7509013
V5	0.5944955	0.6001972

Man sieht, dass durch die Berücksichtigung der zweiten Hauptkomponente sich die Kommunalitäten bei drei Variablen praktisch nicht verändern, bei zweien allerdings stark. Schwach ist der Anstieg bei den Variablen, die mit der ersten Hauptkomponente stark korrelieren, stark ist der Anstieg bei den Variablen, die mit der zweiten Hauptkomponente stark korrelieren. Da die Variablen nicht interpretiert sind, können Inhalte keine Hilfe für die Anzahl

der zu berücksichtigenden Hauptkomponenten spielen. Die obigen Ergebnisse (Eigenwert der zweiten Hauptkomponente kleiner als 1, schwache Korrelation der meisten Variablen mit der 2. Hauptkomponenten, schwacher Anstieg der meisten Kommunalitäten durch Berücksichtigung der 2. Hauptkomponenten) führen eher dazu, nur eine Hauptkomponente zu berücksichtigen.

Alternativ und kürzer kann man mit R wie folgt rechnen:

```
(res=prcomp(dat,scale=T))
```

und erhält (s. Abbildung 5):

```
Standard deviations (1, .., p=5):
[1] 1.7395574 0.9498543 0.7365240 0.6742194 0.2732720

Rotation (n x k) = (5 x 5):
          PC1         PC2         PC3         PC4         PC5
V1 -0.5323696  0.04271658 -0.10092570  0.4662205 -0.69800454
V2 -0.5356376  0.05643576 -0.03361196  0.4454597  0.71438313
V3 -0.3113329 -0.78326425  0.49678591 -0.2058642 -0.01981476
V4 -0.3691723  0.61250937  0.56053654 -0.4156590 -0.03962851
V5 -0.4432365 -0.07949641 -0.65397766 -0.6074962  0.02198525
```

Abbildung 5: Ausgabe des prcomp-Befehls

Man erhält unmittelbar statt der Eigenwerte die Wurzel aus diesen und die Eigenvektoren. Die Daten müssen nicht vorgängig standardisiert und in eine Korrelationsmatrix umgewandelt werden, wenn man "scale=T" angibt. Die Vorzeichen sind bei manchen Eigenvektoren umgekehrt. Eigenvektoren sind nur bis auf die Multiplikation mit einer von 0 verschiedenen Konstante bestimmt. Man kann einen Eigenvektor also mit -1 multiplizieren. Dadurch verändern auch die entsprechenden Hauptkomponenten und die Korrelationen $\mathbf{a}_i^*$ das Vorzeichen. Weist z.B. die erste Hauptkomponenten mit allen Ursprungsvariablen Z_i eine negative Korrelation auf, kann man die Hauptkomponente mit -1 multiplizieren, um überall positive Korrelationen zu erhalten.

Die einzelnen Grössen erhält man mittels

res$sdev	Standardabweichungen (Wurzeln aus Eigenwerten)
res$rotation	Matrix der Eigenvektoren
res$center	Mittelwerte der nicht-standardisierten Ausgangsvariablen
res$scale	Standardabweichungen der nicht-standardisierten Ausgangsvariablen
res$x	nicht standardisierte Hauptkomponenten

Die standardisierten Hauptkomponenten $\mathbf{H}^* = \mathbf{ZA}\Lambda^{-\frac{1}{2}}$ erhält man mittels

```
Z%*%res$rotation%*%solve(diag(res$sdev))
```

oder schneller mittels direkter Standardisierung der nicht standardisierten Hauptkomponenten: „scale(res$x)“.

Berechnung der Werte des obigen $\mathbf{z}_n$ auf den Hauptkomponenten mittels $\mathbf{z}_n^t \mathbf{A} \Lambda^{-\frac{1}{2}}$:

```
z1%*%res$rotation%*%solve(diag(res$sdev))
```

Es ergeben sich bis auf manche Vorzeichen dieselben Ergebnisse wie oben.

Die Ladungsmatrix, d.h. die Korrelationen zwischen $\mathbf{H}_i$ und $\mathbf{Z}_i$, erhält man mittels:

```
cor(res$x,scale(dat))
```

Auch hier ergeben sich wiederum Vorzeichenwechsel bezüglich der obigen Resultaten, die sonst bis auf Transposition identisch sind.

Die Kommunalitäten berechnen wir bei Berücksichtigung nur einer Hauptkomponente mit

```
(cor(res$x,scale(dat))^2)[1,]
```

und bei Berücksichtigung zweier Hauptkomponenten

```
rowSums((cor(res$x,scale(dat))^2)[1:2,]).
```

Alternativ kann man − gemäss Theorie auf der Seite 10 − die Ladungsmatrix mit

```
eig$vectors%*%diag(sqrt(eig$values))
```

berechnen und die Kommunalitäten für die erste Hauptkomponente mit

```
((eig$vectors%*%diag(sqrt(eig$values)))^2)[,1]
```

und für die ersten zwei Hauptkomponenten mittels

```
rowSums(((eig$vectors%*%diag(sqrt(eig$values)))^2)[,1:2])
```

Mit dem von „prcomp(x,scale=T)“ erzeugten R-Objekt x (bei korr=F, Voreinstellung) oder mit der Korrelationsmatrix x (bei korr=T) kann die Ladungsmatrix und die Kommunalitäten auch z.B. mit der folgenden Funktion rechnen:

```
Komm=function(x,korr=F){
if (korr==F) {lad=x$rotation%*%diag(x$sdev);lad
 colnames(lad)=colnames(x$rotation)
 rownames(lad)=rownames(x$rotation) } else
{eig=eigen(x); lad=eig$vectors%*%diag(sqrt(eig$values))
 lad}
m = length(lad[,1]); comm=matrix(rep(0,m^2),ncol=m)
comm[,1]=lad[,1]^2
for (i in 2:m) comm[,i]=rowSums(lad[,1:i]^2)
colnames(comm)=paste("V",1:m,sep=)
rownames(comm)=paste("PC",1:m,sep=)
list('Ladungsmatrix'=lad,'Kommunalitaeten'=comm)}
```

Man erhält mit dem R-Objekt res (Resultat des Befehls prcomp(dat,scale=T))

Komm(res)

die Ladungen

	PC1	PC2	PC3	PC4	PC5
V1	-0.926	0.041	-0.074	0.314	-0.191
V2	-0.932	0.054	-0.025	0.300	0.195
V3	-0.542	-0.744	0.366	-0.139	-0.005
V4	-0.642	0.582	0.413	-0.280	-0.011
V5	-0.771	-0.076	-0.482	-0.410	0.006

und die Kommunalitäten

	PC1	PC2	PC3	PC4	PC5
V1	0.858	0.859	0.865	0.964	1.000
V2	0.868	0.871	0.872	0.962	1.000
V3	0.293	0.847	0.981	1.000	1.000
V4	0.412	0.751	0.921	1.000	1.000
V5	0.594	0.600	0.832	1.000	1.000

. Alternativ erhält man mit

```
Sigma = cor(dat)
Komm(Sigma,korr=T)
```

dasselbe Resultat.

Bemerkung 10. *SPSS: Mit „Dimensionsreduktion“, „Faktoranalyse“ erhält man die PCA. Unter dem Titel „Kommunalitäten“ werden die Kommunalitäten bezüglich der berücksichtigten Hauptkomponenten geliefert. In der Voreinstellung werden dabei die Hauptkomponenten berücksichtigt, deren Eigenwert grösser als 1 ist (kann geändert werden). Unter „Erklärte Gesamtvarianz“ erscheinen die Eigenwerte (Gesamt), % der Varianz, Kumulierte %. Schliesslich wird unter „Komponentenmatrix“ die Matrix der Korrelationen (Ladungen) bezüglich der berücksichtigten Hauptkomponenten geliefert (mit umgekehrten Vorzeichen). Die berücksichtigten standardisierten Hauptkomponenten werden in die Datendatei geschrieben, wenn man „scores“, „als Variablen speichern“, „Regression“ anklickt. Klickt man „Koeffizientenmatrix der Faktorscores anzeigen“, werden die $\mathbf{a}_i^*$ für die berücksichtigten Hauptkomponenten angegeben.*

5 Graphische Darstellungen

Wenn es nicht zu viele Daten hat, kann man die Punkte, die durch die beiden ersten Hauptachsen gegeben sind, in einem Streudiagramm darstellen. Dies ist sinnvoll, wenn ein

bedeutender Teil der Gesamtvarianz durch die beiden ersten Hauptachsen erfasst wird. Im obigen Beispiel ergibt sich mit

```
plot(res$x[,1:2],main="Plot der Datenpunkte bezüglich der ersten zwei Hauptachsen")
abline(0,0)
lines(c(0,0),c(-3,3))
```

die Abbildung 6

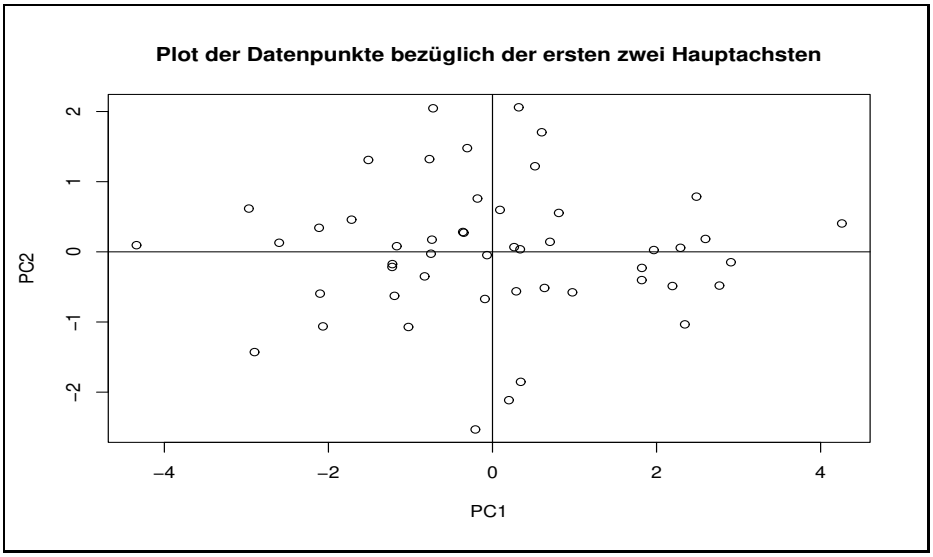

Abbildung 6: Darstellung der Daten in den Koordinaten der ersten zwei Hauptkomponenten

Bei interpretierten Variablen wird man die Punkte beschriften, um eventuell Informationen aus der Verteilung der Punkte zu ziehen. Diesem Zweck dient auch die Bildung von Clustern und entsprechenden graphischen Darstellungen, die oft bei der PCA durchgeführt werden. Am Beispiel erhält man z.B. mit

```
clustdat=hclust(dist(res$x), method = "ward.D2")
plot(clustdat)
```

die Abbildung 7

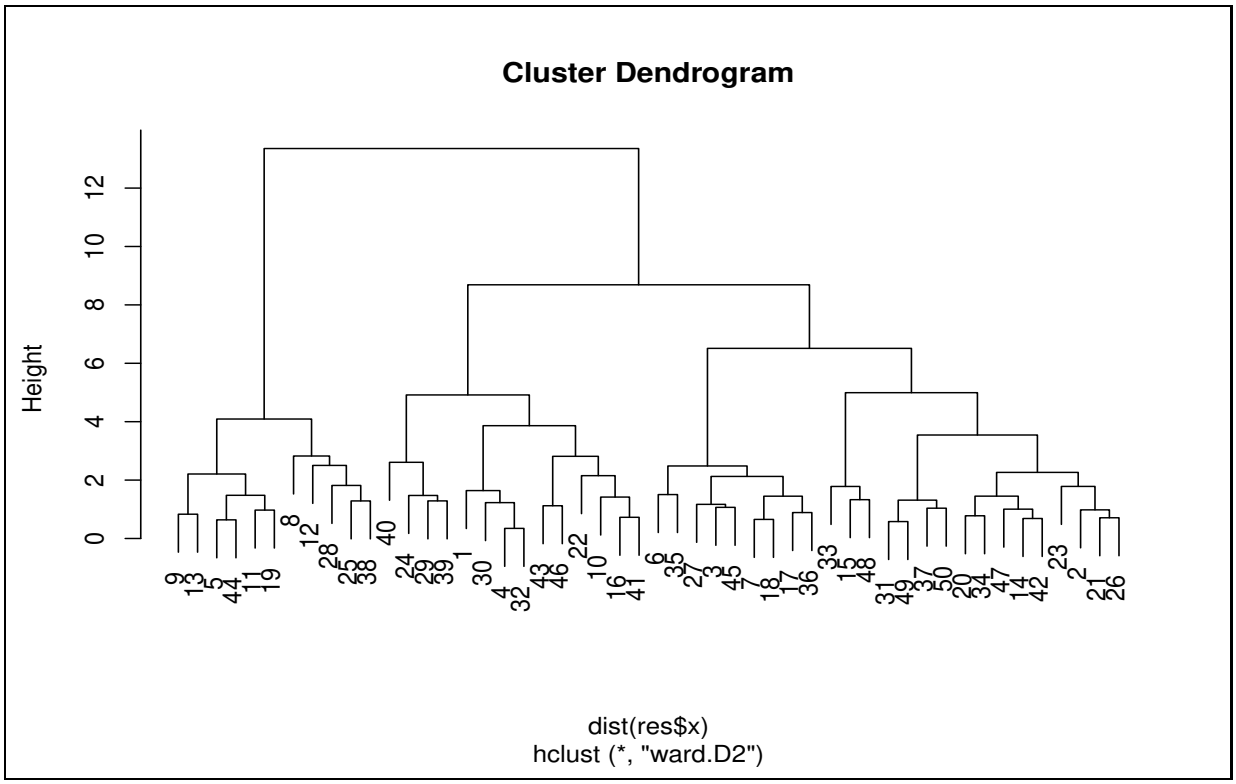

Abbildung 7: Clusterbildung der Daten, die durch die zwei Hauptkomponenten gegeben sind.

Es können drei bis vier Cluster entdeckt werden. Die Clusterbildung kann dann verwendet werden, um die obige Abbildung informativer zu gestalten. Man färbt die Daten gemäss Clustern ein − bei Schwarzweissgraphiken wird man unterschiedliche Symbole für die Datenpunkte wählen. Bei der Unterscheidung von 3 Clustern ergibt sich mit den Befehlen (es müssen zwei zusätzliche Pakete installiert und eingebunden werden):

```
library(ggplot2)
library(ggrepel)
```

```
dat3Clusters = cutree(clustdat, k = 3)
datDf = data.frame(res$x, "Cluster" = factor(dat3Clusters))
str(datDf)
```

ergibt:

```
'data.frame':   50 obs. of 6 variables:
$ PC1 : num -2.114 0.701 0.6 -1.221 1.967 ...
$ PC2 : num 0.3425 0.1428 1.7044 -0.1749 0.0256 ...
$ PC3 : num 0.10462 0.00367 0.09952 0.11032 1.69194 ...
$ PC4 : num 0.69 0.423 -0.401 1.027 0.186 ...
$ PC5 : num -0.0453 -0.391 -0.3451 -0.0471 -0.1226 ...
$ cluster: Factor w/ 3 levels "1","2","3": 1 2 2 1 3 2 2 3 3 1 ...
```

Mit

```
ggplot(datDf,aes(x=PC1, y=PC2,pch=Cluster)) +
geom_text_repel(aes(label = rownames(datDf))) +
theme_classic() +
geom_hline(yintercept = 0, color = "gray70") +
geom_vline(xintercept = 0, color = "gray70") +
geom_point(aes(color = Cluster), alpha = 0.55, size = 3) +
xlab("PC1") + ylab("PC2") + xlim(-5, 6) +
ggtitle("Plot von dat mit drei Clustern")
```

erhält man die Abbildung 8

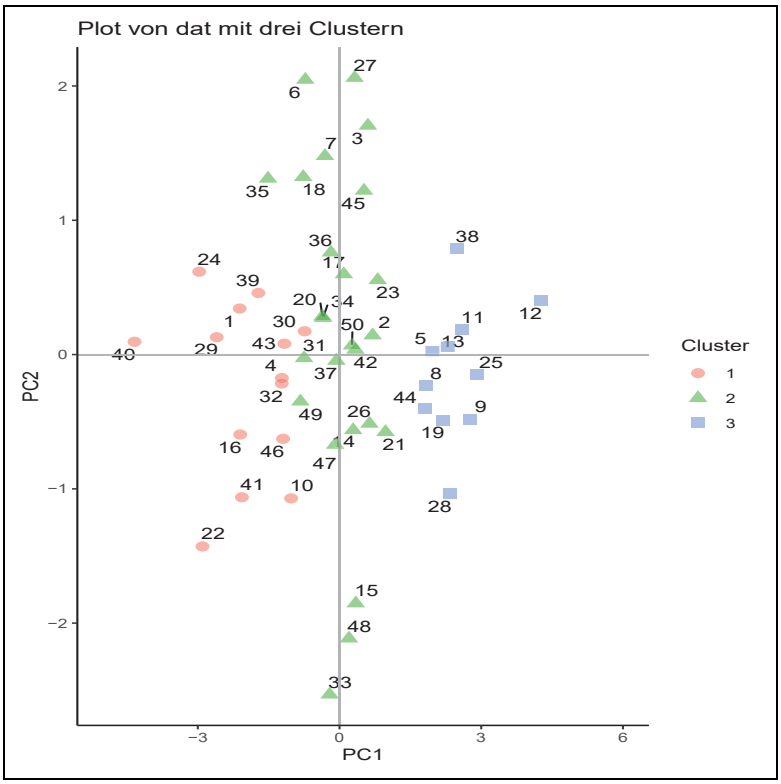

Abbildung 8: Darstellung der Daten in den Koordinaten der ersten zwei Hauptkomponenten mit Clusterbildung

Manchmal werden um die Cluster noch Kreise gebildet, um diese besser hervorzuheben. Fügt man im obigen ggplot-Befehl vor ggtitle „stat_ellipse()+“ hinzu, erhält man die Abbildung 9:

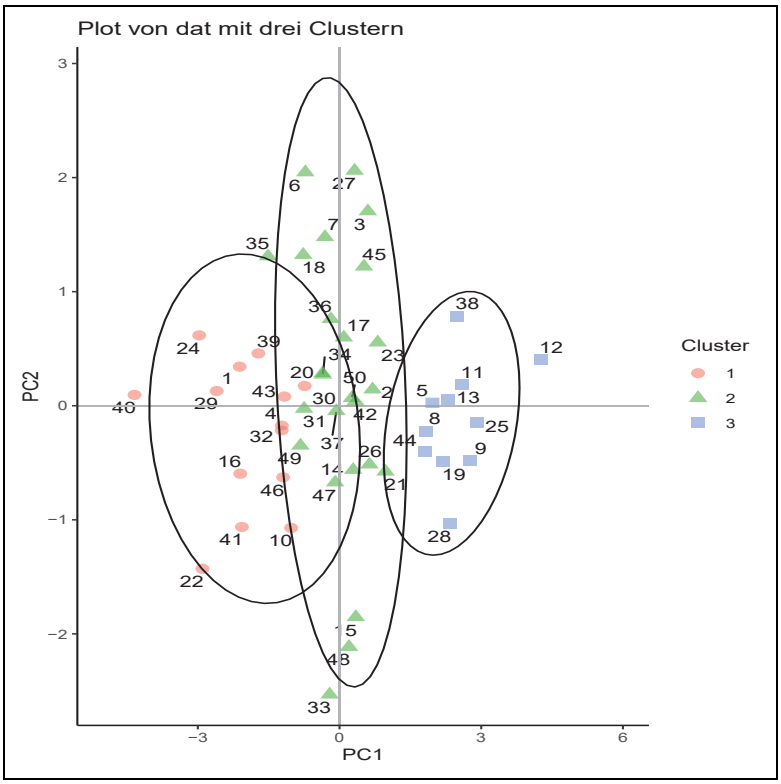

Abbildung 9: Darstellung der Daten in den Koordinaten der ersten zwei Hauptkomponenten mit Clusterbildung samt Ellipsen um die Cluster

Eine weitere graphische Darstellungsart besteht darin, dass man die Ladungen der ersten zwei Hauptkomponenten graphisch darstellt. Mit

```
datkom=Komm(res)
plot(datkom[[1]][,1:2],ylim=c(-0.7,0.7),
main="Plot der Ladungen der ersten zwei Hauptkomponenten")
abline(0,0)
text(x=datkom[[1]][3:5,1],y=datkom[[1]][3:5,2]+0.07,labels=rownames(datkom[[1]])[3:5])
text(x=datkom[[1]][1,1],y=datkom[[1]][1,2]+0.07,labels=rownames(datkom[[1]])[1])
text(x=datkom[[1]][2,1]+0.02,y=datkom[[1]][2,2],labels=rownames(datkom[[1]])[2])
```

erhält man die Abbildung 10

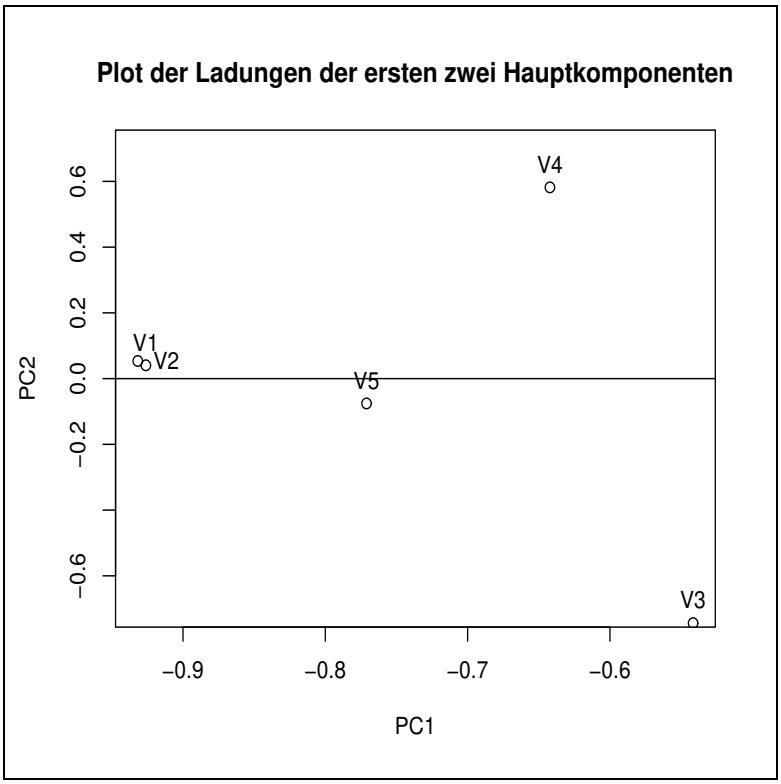

Abbildung 10: Punkteplot der Ladungen der erste zwei Hauptkomponenten

Man sieht sofort, dass alle Variablen mit der ersten Hauptkomponente negativ korreliert sind, allerdings unterschiedlich stark. Mit der zweiten Hauptkomponente sind manche Variablen negativ, manche positiv korreliert. Auch auf diese Punkte könnte man eine Cluster-Analyse durchführen. Manchmal werden die Punkte mit Pfeilen mit dem O-Punkt des Koordinatensystems verbunden: fügt man bei den obigen Befehlen den Befehl

```
arrows(0,0,datkom[[1]][,1],datkom[[1]][,2])
```

hinzu, erhält man die Abbildung 11

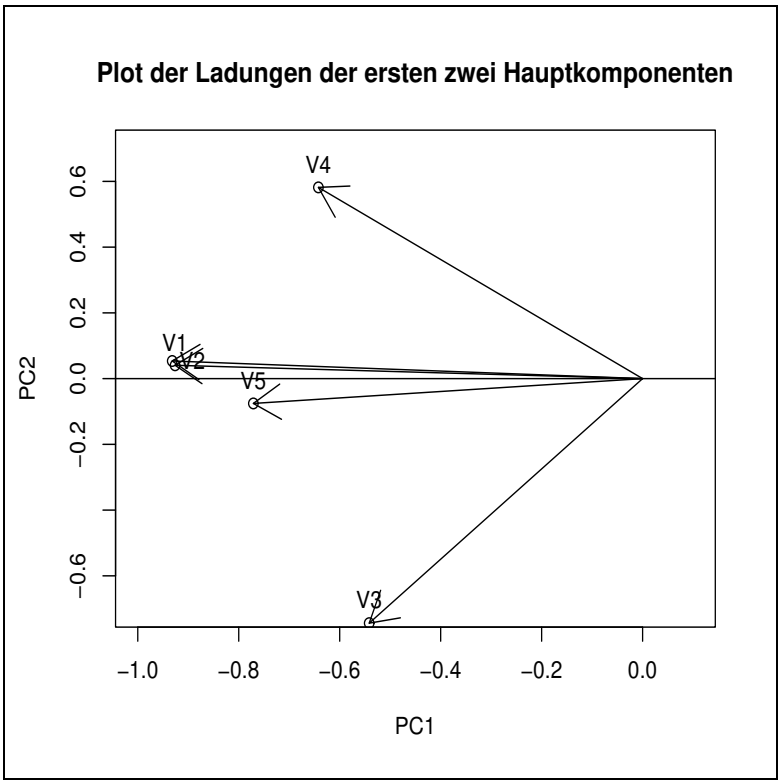

Abbildung 11: Punkteplot der Ladungen der erste zwei Hauptkomponenten mit Pfeilen

Man sieht, welche Variablen nahe beieinanderliegen. Je näher die Pfeile beieinanderliegen, desto mehr korrelieren die entsprechenden Variablen. Die Distanz der Pfeilspitze auf x-Achse drückt die Stärke der Korrelation mit der 1. Hauptkomponente aus, die Distanz der Pfeilspitze auf der y-Achse drückt die Stärke der Korrelation mit der 2. Hauptkomponente aus.

Schliesslich kann man beide Darstellungsarten in einem einzigen Plot vereinigen, den sogenannten Biplots. Dabei werden die beiden Darstellungsarten allerdings nicht einfach übereinandergelegt (für Details s. (Gabriel, 1971)). Die Pfeile entsprechen den Korrelationen (Ladungen) zwischen den Variablen und den Hauptkomponenten. Man sieht, welche Variable (gemeinsam) zu den Hauptkomponenten beitragen. Daten, welche auf beitragenden Variablen höhere (tiefere) Werte haben, liegen bezüglich entsprechender Hauptkomponenten weiter weg vom 0-Punkt. Ansprechende Darstellungen erhält man mit dem Paketen ggplot und ggbiplot.

```
install.packages("devtools")
library(devtools)
devtools::install_github("ggbiplot")
library(ggbiplot)
ggbiplot(res,labels=rownames(dat)
```

ergibt die Abbildung 12:

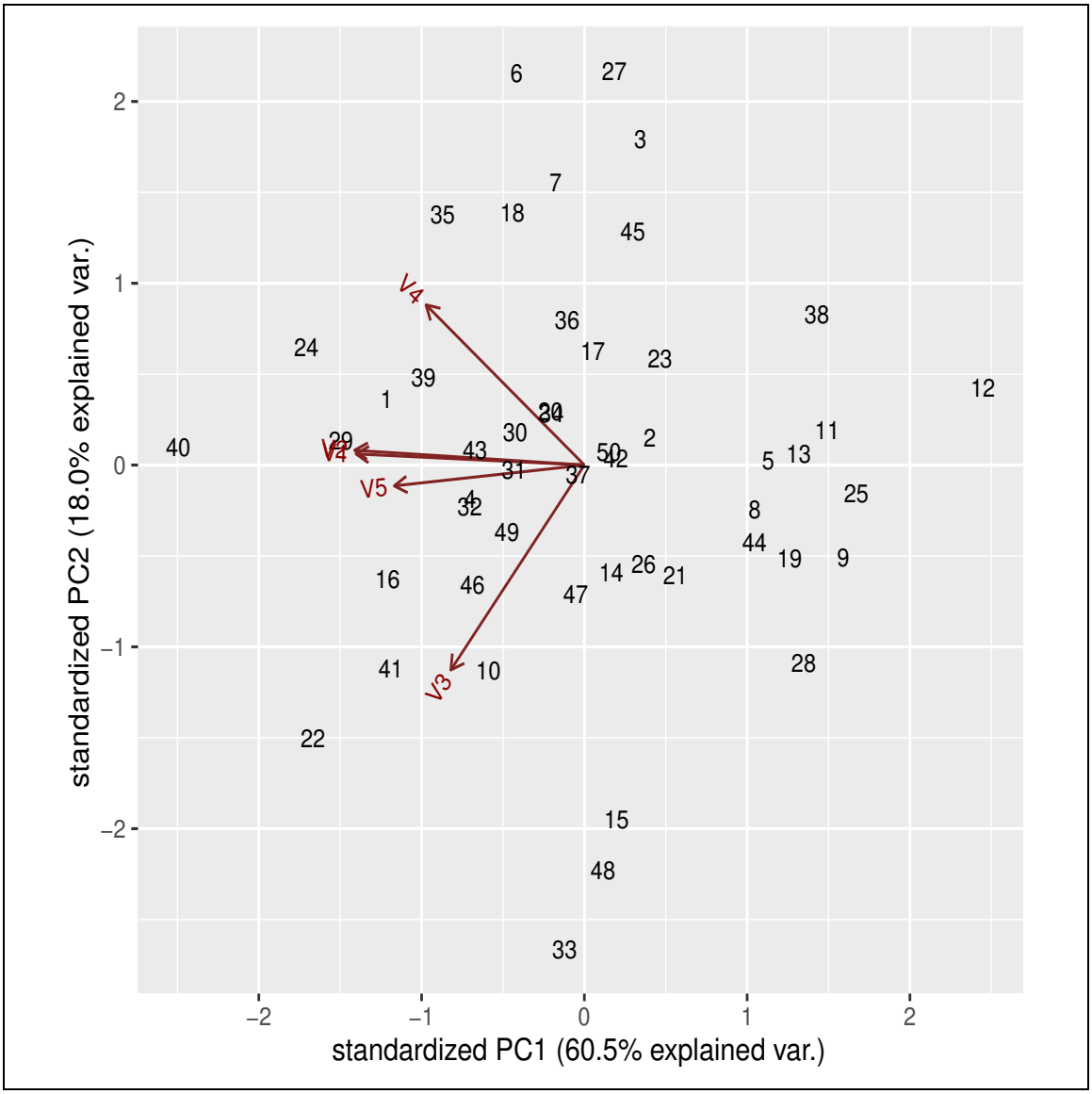

Abbildung 12: Biplot der dat-Daten

Der Abbildung können auf Grund der obigen erfolgten Clusterbildung Ellipsen und Farben zugeordnet werden:

```
ggbiplot(res,ellipse=TRUE, labels=rownames(dat), groups=datDf$Cluster)
```

und man erhält die Abbildung 13

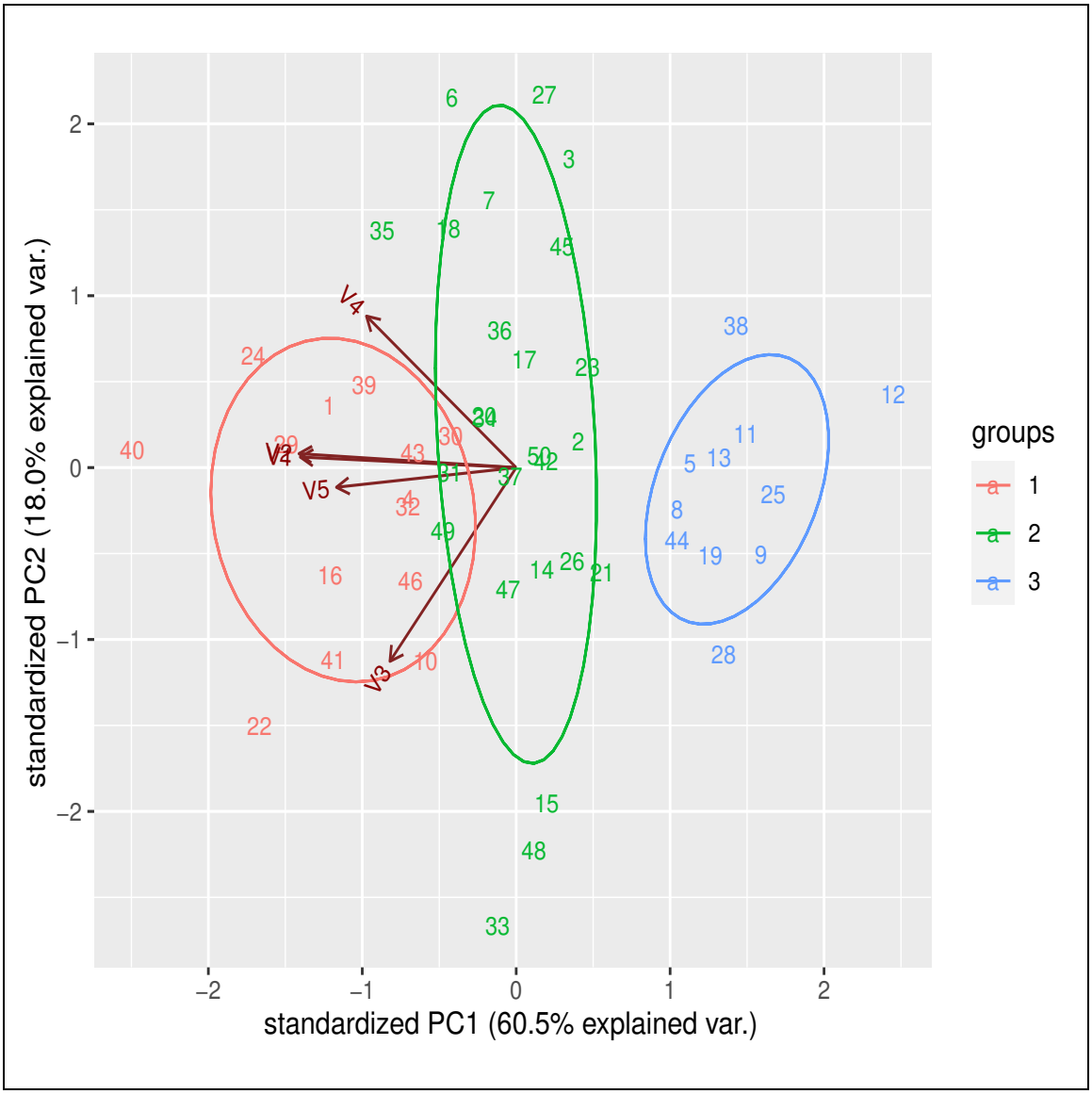

Abbildung 13: Biplot der dat-Daten

6 Weitere Beispiele

6.1 Europäische Währungs- und Wirtschaftsuniton EWU 1997

EWU-Daten (X1=Inflationsrate 1997 in %, X2=langfristiger Zinssatz 1997 in %, X3=öffentliche Neuverschuldung 1997 in % des BIP 1997, X4=öffentlicher Schuldenstand 1997 in % des BIP).

```
EWU=data.frame(
Staat=c("B","DK","D","Fin","F","GR","IRL","I","L","NL","A","P","S","E","GB"),
X1=c(1.4,1.9,1.4,1.3,1.2,5.2,1.2,1.8,1.4,1.8,1.1,1.8,1.9,1.8,1.8),
X2=c(5.7,6.2,5.6,5.9,5.5,9.8,6.2,6.7,5.6,5.5,5.6,6.2,6.5,6.3,7.0),
X3=c(2.1,-0.7,2.7,0.9,3.0,4.0,-0.9,2.7,-1.7,1.4,2.5,2.5,0.8,2.6,1.9),
X4=c(122.2,65.1,61.3,55.8,58.0,108.0,66.3,121.6,6.7,72.7,66.1,62.0,76.6,68.8,53.4),
Beitritt=c(1957,1973,1957,1995,1957,1981,1973,1957,1957,1957,1995,1986,1995,1986,1973))
EWUdat= scale(EWU[,2:5])
```

Mit

```
PCAEWU=prcomp(EWUdat,scale=T)
PCAEWU$sdevˆ2
PCAEWU$rotation
summary(PCAEWU)
```

erhält mit PCAEWU$sdevˆ2 die Eigenwerte: [1] 2.56392387 0.93370674 0.44438531 0.05798409
Proportion of Variance: 0.641 0.2334 0.1111 0.0145
Cumulative Proportion: 0.641 0.8744 0.9855 1.0000
Die Eigenvektoren mit PCAEWU$rotation.
Die Hauptkomponenten mit PCAEWU$x.
Die standardisierten Hauptkomponenten erhält man, indem man die Werte der Hauptkomponenten durch $\sqrt{\lambda_i}$, die Standardabweichung teilt:

```
PCAEWU$x%*%solve(diag(PCAEWU$sdev))
```

oder „scale(PCAEWU$x)“ rechnet. Für die Ladungsmatrix (Korrelationsmatrix) rechnet man

```
t(cor(PCAEWU$x,EWUdat))
```

(oder mit dem obigen Befehl Komm) und erhält:

	PC1	PC2	PC3	PC4
X1	0.89	-0.42	0.03	-0.17
X2	0.89	-0.42	-0.02	0.17
X3	0.69	0.55	0.47	0.01
X4	0.71	0.52	-0.48	-0.01

Die Xi korrelieren alle mit PC1. PC1 ist damit eine Variable, welche eine allgemeine Tendenz aller Variablen misst: höhere Schulden, höhere Inflationsraten und höhere Zinsen hängen zusammen und PC1 drückt diesen Zusammenhang aus. Man sieht aber auch, dass der Zusammenhang zwischen PC1 und X3 sowie X4 schwächer ist als zwischen PC1 und den ersten zwei Variablen. Bei der zweiten Hauptkomponenten PC2 ergeben sich positive und negative Korrelationen. Diese Variable unterscheidet zwischen Schulden einerseits und Inflationsrate sowie Zinsen andererseits.

Gemäss obiger Theorie könnte man für die Ladungsmatrix auch rechnen:

```
(vorPCAEWU=PCAEWU$rotation%*%diag(PCAEWU$sdev))
```

Diese letzte Berechnungsart ist auch möglich, wenn nur die Korrelationsmatrix vorliegt – d.h. wenn die Daten nicht vorliegen. Die Kommunalitäten berechnet man für zwei Hauptkomponenten mittels

```
vor2PCAEWU=vorPCAEWU^2
rowSums(vor2PCAEWU[,1:2])
```

(oder für alle Hauptkomponenten mit dem obigen Befehl Komm) und erhält

X1	0.9699710
X2	0.9707598
X3	0.7835760
X4	0.7733237

Man sieht, dass die ersten zwei Variablen X1 und X2 durch die ersten zwei Hauptkomponenten sehr gut erfasst werden, die Variablen X3 und X4 weniger gut. Bei nur einer Hauptkomponente erhält man etwa mit

```
vor2PCAEWU[,1]
```

X1	0.7927737
X2	0.7913999
X3	0.4810714
X4	0.4986789

Man sieht, dass die zweite Hauptkomponente beträchtlich zum Ansteigen der Kommunalitäten beiträgt. Mit dem Scree-Plot 14 erhält man

```
screeplot(PCAEWU,type="l",main="Screeplot")
```

die Abbildung 14

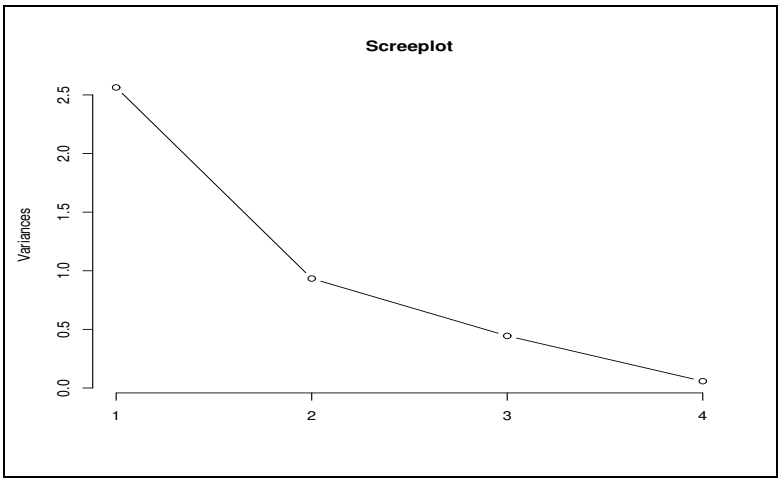

Abbildung 14: Screeplot der EWU-Daten

Der Bartlett- und der Levene-Test scheiden keine Gruppe von Variablen aus, deren Varianzen sich nicht signifikant unterscheiden.

Man hätte also eine oder zwei Hauptkomponenten zu berücksichtigen. Wegen des beträchtlichen Zuwachses bei der Kommunalität bei der Berücksichtigung der zweiten Hauptkomponenten, ist es sicher gerechtfertigt, mit zwei Hauptkomponenten zu arbeiten.

Instruktiv kann auch die Darstellung der Punkte sein, die durch die zwei ersten Hauptkomponenten bestimmt sind. Im Beispiel erhält man für die standardisierten Hauptkomponenten mittels

```
HKpunkteEWU=PCAEWU$x%*%solve(diag(PCAEWU$sdev))
plot(HKpunkteEWU[,1:2],main="Plot der Punkte der ersten zwei Hauptkomponenten",
xlab="erste HK",ylab="zweite HK")
abline(c(0,0));lines(c(0,0),c(-2.2,2))
HKpunkteEWUt1=as.data.frame(HKpunkteEWU[,1:2])
names(HKpunkteEWUt1)=c("HK1","HK2")
HKpunkteEWUt1$Staat= EWU$Staat
HKpunkteEWUt1[11,1]=-0.53; HKpunkteEWUt1[11,2]=0.7; HKpunkteEWUt1[12,2]=.1
text(EWU$Staat,x=HKpunkteEWUt1[,1]+0.1,y=HKpunkteEWUt1[,2]+0.1)
```

die Abbildung (s. Abbildung 15)

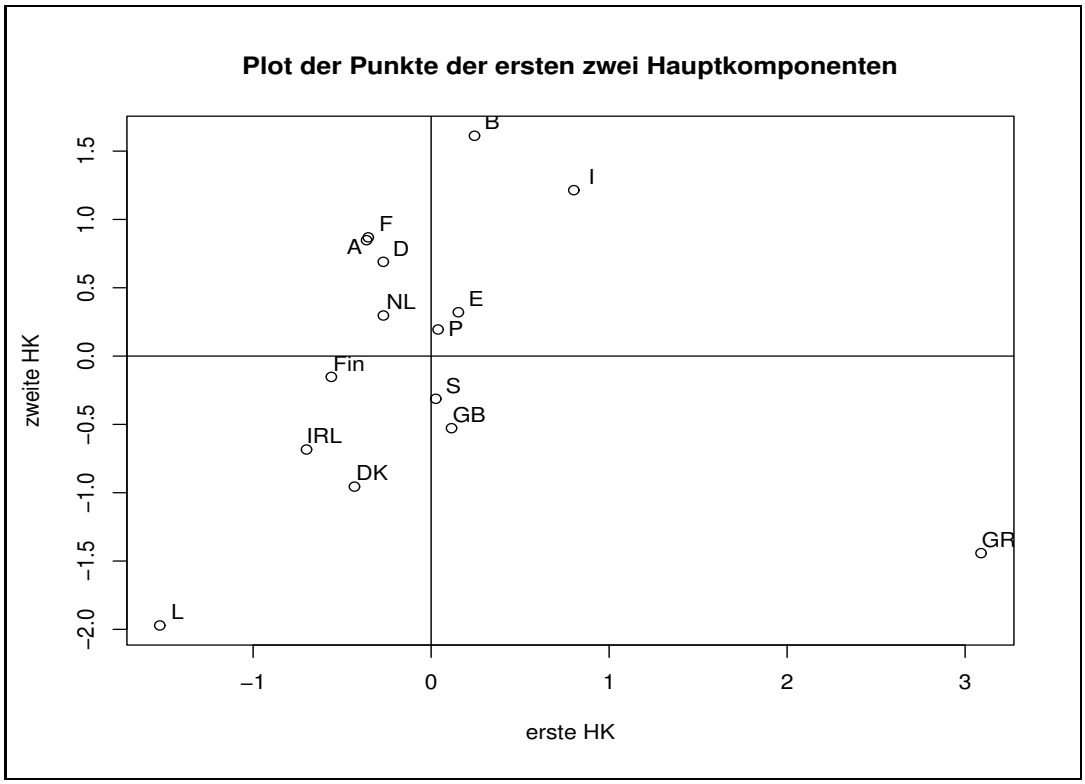

Abbildung 15: Länderpunkte, die durch die zwei Hauptachsen gegeben sind (EWU-Daten)

Man sieht, dass es zwei bis vier Ausreisser hat. Manchmal kann man in diesen Darstellungen auch Gruppen ausmachen, indem man nach einer nominalen Variable die Punkte verschiedene Farben oder Formen aufweisen lässt (z.B. nach Geschlecht). Man sieht dann, ob eine Hauptkomponente die Gruppen unterscheidet. Dazu braucht es Sachverstand. Man könnte sich bezüglich der obigen Datenpunkt z.B. fragen, ob die Lage vom EU-Beitrittsdatum abhängt, von den Regierungsparteien (links, mitte, rechts) oder von der geographischen Lage (Süden oder Norden, Westen oder Osten). Zudem kann man einen Clusterbaum erstellen. Mit

```
PCAEWU=prcomp(EWUdat,scale=T)
clustewu=hclust(dist(PCAEWU$x), method = "ward.D2")
plot(clustewu)
```

erhält man die Abbildung 16:

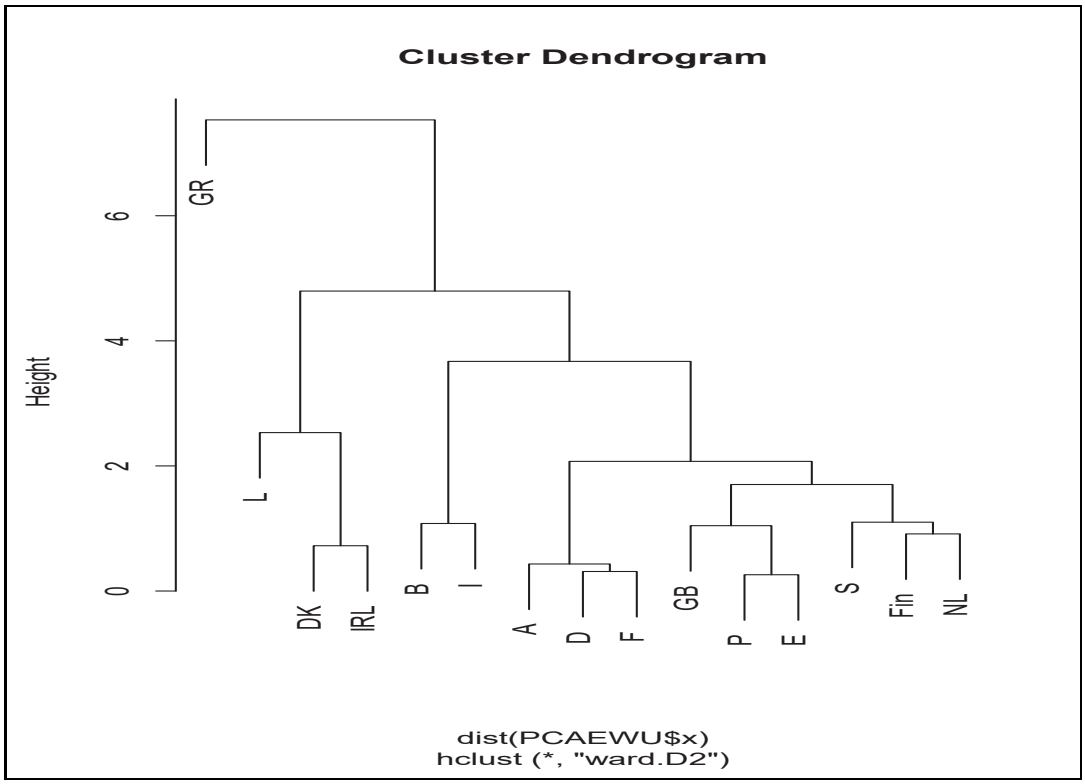

Abbildung 16: Cluster-Dendrogramm der ersten zwei EWU-Hauptkomponenten

Die graphische Darstellung der Länder-Punkte mit Ellipsen sieht mit

```
dat3Clusters = cutree(clustewu, k = 3)
ewu_clust = data.frame(PCAEWU$x, "Cluster- factor(dat3Clusters))
library(ggplot2); library(ggrepel); library(ggforce)
ggplot(ewu_clust,aes(x=PC1, y=PC2), pch=Cluster) +
geom_text_repel(aes(label = rownames(EWUdat))) +
theme_classic() +
geom_point(aes(pch = Cluster), alpha = 0.55, size = 3) +
xlab("PC1") + ylab("PC2") + xlim(-5, 6) +
geom_ellipse(aes(x0 = -1.6, y0 = -1.3, a = 1.2, b = 0.5, angle = 10)) +
geom_ellipse(aes(x0 = 5.1, y0 = -1.3, a = 0.5, b = 0.5, angle = 10)) +
geom_ellipse(aes(x0 = 0.3, y0 = 0.6, a = 1.5, b = 1.5, angle = 10)) +
ggtitle("Plot von EWU-Daten mit drei Clustern")
```

wie folgt aus (ggbiplot liefert hier nur eine Ellipse − zu wenig Daten pro Cluster. Deshalb werden die Ellipsen hier von Hand eingezeichnet − mit Hilfe des Paketes „ggforce“) (s. Abbildung 17):

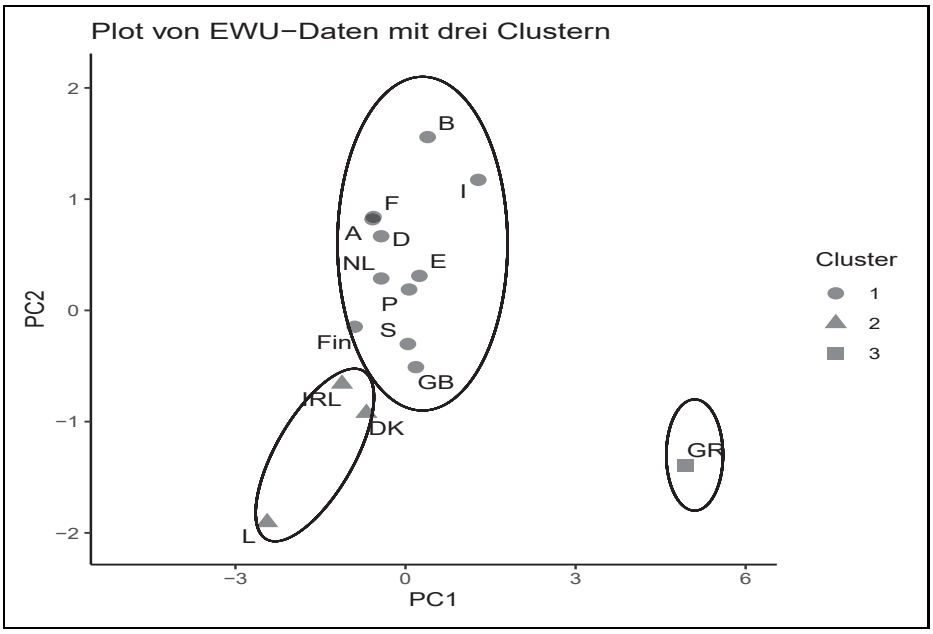

Abbildung 17: Länderpunkte mit Clusterung

Inhaltlich ist die Interpretation der Clusterung ohne genügend volkswirtschaftlichen Sachverstand schwierig. So sind z.B. Länder Südeuropas mit Ländern Nordeuropas im gleichen Cluster. Luxemburg, Dänemark und Irland liegen zusammen in einem weiteren Cluster. Dass Griechenland isoliert dasteht, macht noch am ehesten Sinn. Man könnte es mit vier statt drei Clustern versuchen, um eventuell eine inhaltlich besser interpretierbare Darstellung zu erhalten.

Als nächstes stellen wir die Korrelationen zwischen den berücksichtigen Hauptachsen und den ursprünglichen, standardisierten Variablen als Punktediagramm dar. Mit

```
EWUdat= EWU[,2:5]
rownames(EWUdat)=EWU$Staat
mod=prcomp(EWUdat,scale=T)
lad=Korr_Y_PC(mod)
ladeHK1=lad[,1]
ladeHK2=lad[,2]
plot(ladeHK1,ladeHK2,xlim=c(0.6,1),ylim=c(-1,1),lwd=0.5,
 xlab="Korrelationen erste Hauptkomponente",
 ylab="Korrelationen zweite Hauptkomponenten",
 main="Korrelationen Hauptkomponenten mit Variablen")
text(ladeHK1+0.04,ladeHK2+c(0.1,-0.1,0.1,-0.1),labels=c("Inflationsrate",
"Zinssatz","Neuverschuldung","Schuldenstand"), cex=0.5)
```

erhält man die Abbildung 18:

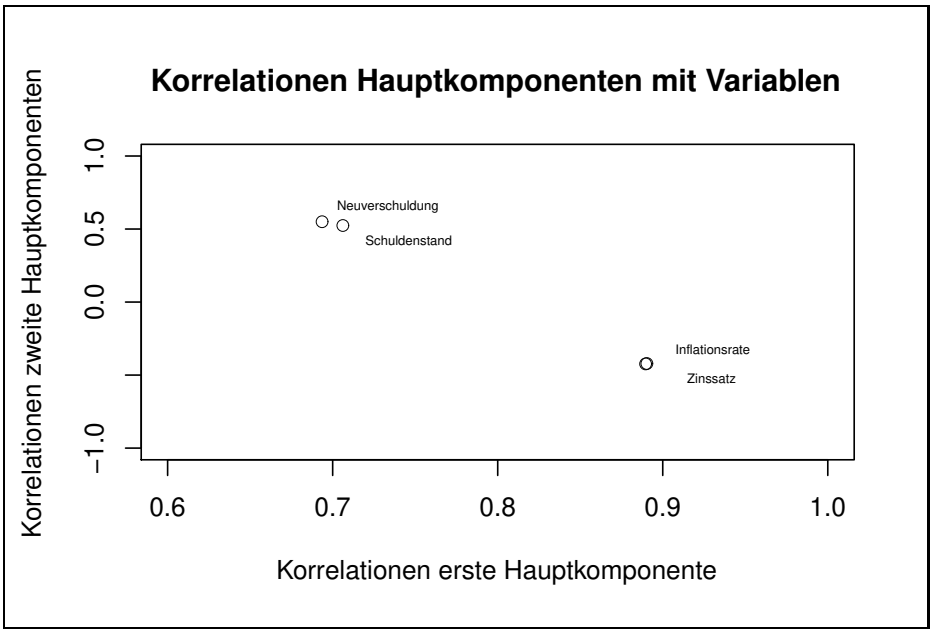

Abbildung 18: Korrelationen der standardisierten Ursprungsvariablen mit den ersten zwei Hauptkomponenten (Ladungen)

Man sieht deutlich zwei Gruppen von Variablen. Alle Variablen weisen eine positive Korrelation mit den ersten Hauptkomponenten auf. Zwei Variablen weisen eine positive Korrelation mit der zweiten Hauptkomponente (Neuverschuldung, Schuldenstand) und zwei Variablen eine negative Korrelation mit der zweiten Hauptkomponenten auf (Inflationsrate, Zinssatz). Die Frage ist, wie auf diesem Hintergrund die Hauptkomponenten inhaltlich zu interpretieren sind.
Die Variante mit den Pfeilen sieht wie folgt aus. Mit

```
plot(ladeHK1,ladeHK2,type="n",xlim=c(0,1),ylim=c(-1,1),lwd=0.5,
 xlab="Korrelationen mit erster Hauptkomponente",
 ylab="Korrelationen mit zweiter Hauptkomponente",
 main="Korrelationen Hauptkomponenten mit Variablen")
text(ladeHK1+0.04,ladeHK2+c(0.1,-0.1,0.1,-0.1),labels=c("Inflationsrate",
"Zinssatz","Neuverschuldung","Schuldenstand"), cex=0.5)
null=rep(0,4)
abline(h=0)
arrows(null,null,ladeHK1,ladeHK2)
```

ergibt sich fürs Beispiel (s. Abbildung 19):

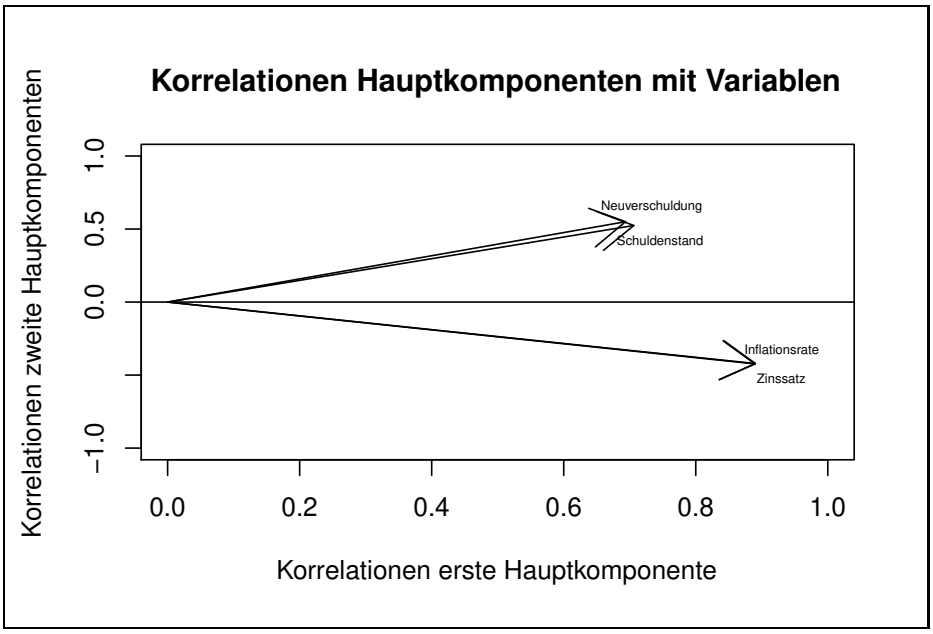

Abbildung 19: Darstellung der Korrelationen zwischen den Hauptkomponenten und den standardisierten Ursprungsvariablen (Ladungen) mit Pfeilen

Man sieht, welche Variablen nahe beieinanderliegen. Die Variablen Neuverschuldung und Schuldenstand korrelieren stark miteinander, ebenso die Variablen Inflationsrate und Zinssatz. Die Variablen Neuverschuldung und Schuldenstand korrelieren stark mit der 1. Hauptkomponenten, aber weniger stark als die anderen zwei Variablen.

Schliesslich kann man den Biplot betrachten: Mit

```
library(ggbiplot)
ggbiplot(PCAEWU,ellipse=TRUE,obs.scale = 1, var.scale = 1,
labels=rownames(EWUdat), groups=ewu_clust$Cluster) +
scale_colour_manual(name="Cluster", values= c("gray60", "black", "gray88"))+
ggtitle("PCA of EWU-dataset")+
theme_minimal()+
theme(legend.position = "bottom"+
geom_point(shape=23, size=3))
```

erhält man die Abbildung 20:

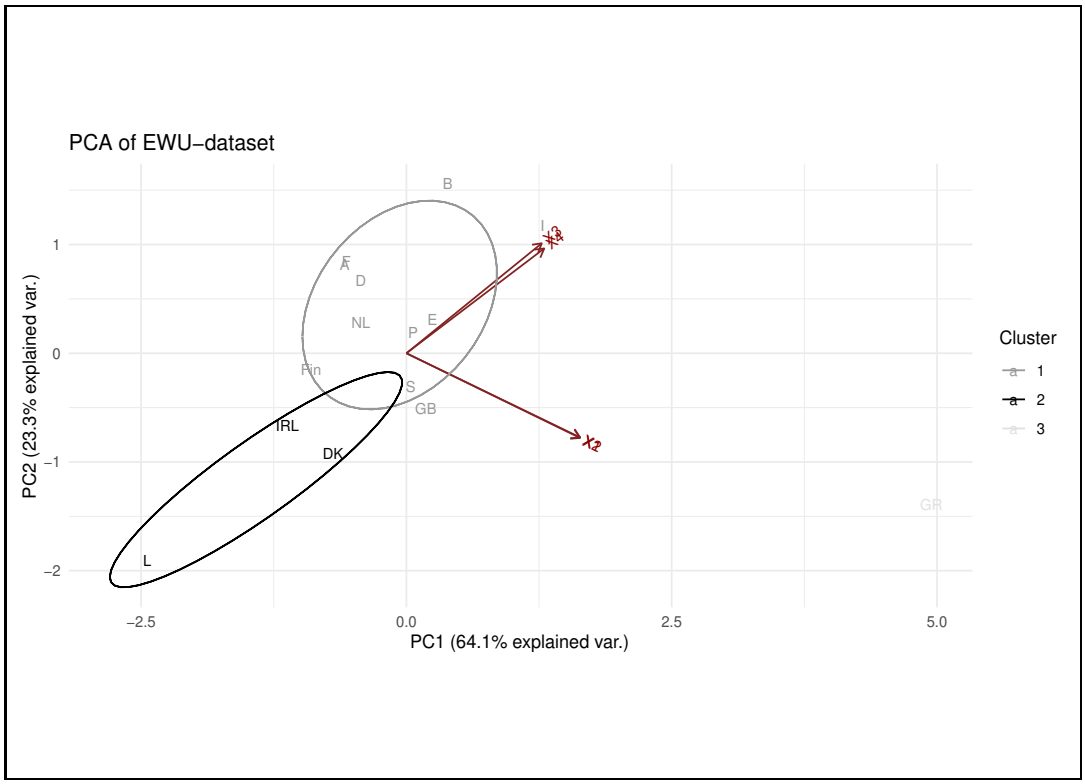

Abbildung 20: Biplot der EWU-Daten

Man sieht wiederum, dass die Variablen 1 und 2 stark korrelieren, ebenso, wenn auch etwas weniger, die Variablen 3 und 4. ggbiplot zeichnet zwei Ellipsen. Griechenland steht weit weg rechts unten.

6.2 Schulnoten

Die folgende Tabelle sei die Korrelationsmatrix einer Erhebung von 220 Schülern (aus (Lawley & Maxwell, 1971)). Es wurde die Leistung in verschiedenen Fächern gemessen. Die zur Berechnung der Matrix verwendeten Daten sind vermutlich nicht metrisch skaliert und die PCA wäre dann nicht angemessen.

	Gälisch	Englisch	Geschichte	Arithmetik	Algebra	Geometrie
Gälisch	1					
Englisch	0.44	1				
Geschichte	0.41	0.35	1			
Arithmetik	0.29	0.35	0.16	1		
Algebra	0.33	0.32	0.19	0.59	1	
Geometrie	0.25	0.33	0.18	0.47	0.46	1

Es lohnt sich, jeweils die Korrelationsmatrix zu begutachten. Man sieht, dass alle Variablen positiv miteinander korreliert sind. Dies lässt vermuten, dass die erste Hauptkomponente

mit allen Fächern positiv korreliert ist. Zudem sieht man, dass Sprachen und Geschichte stärker miteinander korreliert sind als mit den formalen Fächern. Analoges gilt für die formalen Fächer. Da die Daten nicht gegeben sind, kann man nicht mit „prcomp“ arbeiten. Man erhält (marks sei die Korrelationsmatrix) mit

```
marksPCA=eigen(marks)
```

die Eigenwerte mittels marksPCA$values: 2.7286835, 1.1287922, 0.6152914, 0.6028089, 0.5225144, 0.4019097. Die Hauptkomponenten erfassen die folgenden Anteile an der Gesamtvarianz:

Komponenten	erklärt in %	erklärt kumulativ in %
1	45.48	45.48
2	18.81	64.29
3	10.25	74.55
4	10.05	84.59
5	8.71	93.30
6	6.70	100.00

Der folgende Screeplot (Abbildung 21), erzeugt mittels

```
plot(marksPCA$values,type="l",main="Screeplot Leistungen",
xlab="Komponenten",ylab="Eigenwerte")
points(marksPCA$values)
```

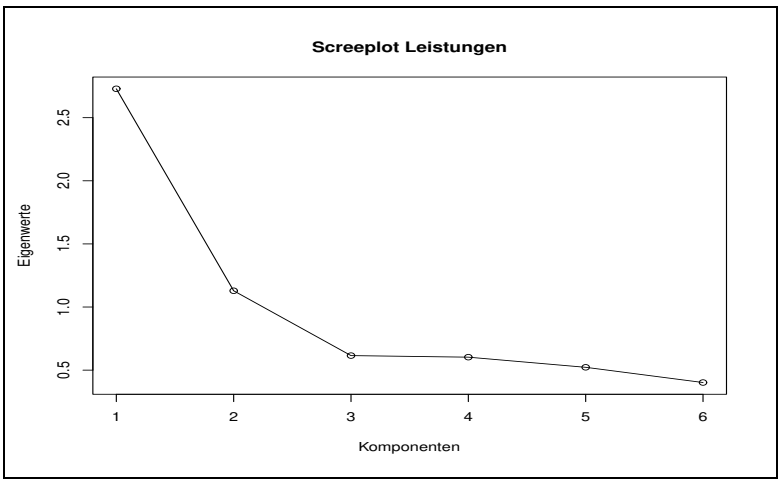

Abbildung 21: Screeplot der Schulleistungsbeispiels

führt vermutlich zum Schluss, dass man die ersten zwei Komponenten berücksichtigen sollte. Die Ladungsmatrix für die ersten zwei Komponenten ist mit (das Vorzeichen wird hinzugesetzt, um dieselben Resultate wie (Bartholomew et al., 2008, S. 127) zu erhalten):

```
-marksPCA$vectors %*% diag(sqrt(marksPCA$values))[,1:2]
```

Fächer	a^*_{i1}	a^*_{i2}
Gälisch	0.66	0.44
Englisch	0.69	0.29
Geschichte	0.52	0.64
Arithmetik	0.74	-0.42
Algebra	0.74	-0.37
Geometrie	0.68	-0.35

Man kann auch mit der oben vorgestellten Funktion rechnen:

```
Komm(marks,korr=T)
```

und eventuell die Vorzeichen entsprechend anpassen.

Alle Variablen korrelieren positiv mit der ersten Hauptkomponente. Man kann diese als eine Variable deuten, welche allgemein schulische Fähigkeiten misst. Je grösser die Werte dieser Variable, desto bessere Noten hat der betroffene Schüler. Die zweite Hauptkomponente korreliert positiv mit den Sprachen und Geschichte, negativ mit den mathematischen Fächern. Es handelt sich also um eine Variable, die zwischen eher formalen versus eher sprachlichen Fähigkeiten unterscheidet. Dies kann auch die folgende Abbildung 22 veranschaulichen:

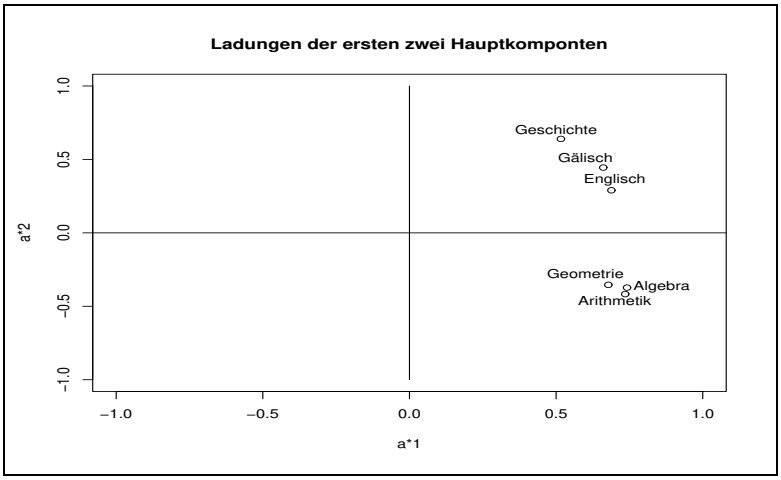

Abbildung 22: Ladungen des Schulleistungsbeispiels (zwei Hauptkomponenten)

Die Punkte liegen bezüglich der ersten Achse alle auf derselben Seite. Die zweite Achse trennt die zwei Punktwolken formale versus sprachliche Fähigkeiten. Da von der Korrelationsmatrix ausgegangen wurde und die ursprünglichen Daten nicht vorliegen, kann man die Punkte, die durch die beiden ersten standardisierten Hauptkomponenten gegeben sind, nicht zeichnen und es ist nicht möglich, einen Biplot (mit Clustern) zu zeichnen.

Es ergeben sich folgende Kommunalitäten für die erste und die ersten zwei Hauptkomponenten: Mit

```
ladm2=(-marksPCA$vectors %*% diag(sqrt(marksPCA$values)))^2
ladm2[,1]
rowSums(ladm2[,1:2])
```

	1. Hauptkomponente	erste zwei Hauptkomponenten
1	0.44	0.63
2	0.47	0.56
3	0.27	0.68
4	0.54	0.72
5	0.55	0.69
6	0.46	0.59

Man sieht, dass die ersten zwei Hauptkomponenten noch ziemlich viel Kommunalität für die restlichen Hauptkomponenten übrig lassen.

6.3 R-Datensatz mtcars

Der Datensatz mtcars wird von R mitgeliefert. Auf dem Internet findet man etliche PCA-Analysen dieses Datensatzes. Er weist folgende Variablen auf:

- cyl: Number of cylinders: more powerful cars often have more cylinders
- mpg: Fuel consumption (Miles per (US) gallon): more powerful and heavier cars tend to consume more fuel.
- disp: Displacement (cu.in.): the combined volume of the engine's cylinders
- hp: Gross horsepower: this is a measure of the power generated by the car
- drat: Rear axle ratio (Hinterachsübersetzung): this describes how a turn of the drive shaft (Antriebsachse) corresponds to a turn of the wheels. Higher values will decrease fuel efficiency.
- wt: Weight (1000 lbs): pretty self-explanatory!
- qsec: 1/4 mile time: the cars speed and acceleration
- vs: Engine block: this denotes whether the vehicle's engine is shaped like a "V", or is a more common straight shape.
- am: Transmission: this denotes whether the car's transmission (Übertragung, vermutlich Gangschaltung) is automatic (0) or manual (1).

- gear: Number of forward gears (Getriebe): sports cars tend to have more gears.
- carb: Number of carburetors (Vergaser): associated with more powerful engines

Note that the units used vary and occupy different scales.
Die Variablen 8 und 9 sind für die PCA ungeeignet, da nicht metrisch skaliert. Mit

```
matcars=mtcars[,-c(8:9)]
```

erhält man mit

```
prcomp(matcars,scale=T)
```

die Standardabweichungen:
[1] 2.3782219 1.4429485 0.7100809 0.5148082 0.4279704 0.3518426 0.3241326 0.2418962 0.1489644
und die Eigenvektoren:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
mpg	-0.393	0.028	-0.221	-0.006	-0.321	0.720	-0.381	-0.125	0.115
cyl	0.403	0.016	-0.252	0.041	0.117	0.224	-0.159	0.810	0.163
disp	0.397	-0.089	-0.078	0.339	-0.487	-0.020	-0.182	-0.064	-0.662
hp	0.367	0.269	-0.017	0.068	-0.295	0.354	0.696	-0.166	0.252
drat	-0.312	0.342	0.150	0.846	0.162	-0.015	0.048	0.135	0.038
wt	0.373	-0.172	0.454	0.191	-0.187	-0.084	-0.428	-0.198	0.569
qsec	-0.224	-0.484	0.628	-0.030	-0.148	0.258	0.276	0.356	-0.169
gear	-0.209	0.551	0.207	-0.282	-0.562	-0.323	-0.086	0.316	0.047
carb	0.245	0.484	0.464	-0.214	0.400	0.357	-0.206	-0.108	-0.320

Mit

```
summary(prcomp(matcars,scale=T))
```

erhält man

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
Stabw	2.3782	1.4429	0.7101	0.5148	0.4280	0.3518	0.3241	0.2419	0.1490
Anteil Var	0.6284	0.2313	0.0560	0.0294	0.0204	0.0138	0.0117	0.0065	0.0025
Kum. An. Var.	0.6284	0.8598	0.9158	0.9453	0.9656	0.9794	0.9910	0.9975	1.0000

Mit screeplot(prcomp(matcars,scale=T),type="l") erhält man die Abbildung 23:

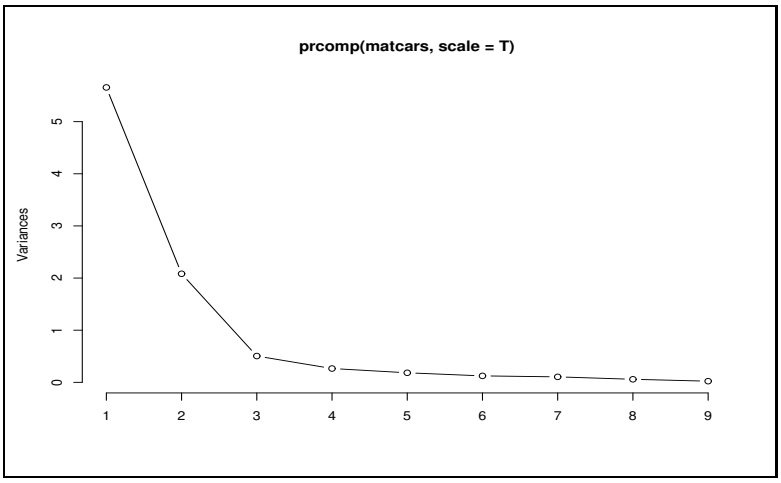

Abbildung 23: Screeplot der mtcars-Daten-Beispiels

Gemäss Screeplot würde man die ersten zwei oder drei Hauptkomponenten berücksichtigen. Die ersten zwei Hauptkomponenten weisen Eigenwerte grösser als 1 auf. Die ersten drei weisen einen kumulierten Varianzanteil von 90% auf. Mit

```
(Komm(prcomp(matcars,scale=T))[[1]])[,1:3]
```

erhält man:

	PC1	PC2	PC3
mpg	-0.935	0.040	-0.157
cyl	0.957	0.023	-0.179
disp	0.945	-0.128	-0.056
hp	0.873	0.389	-0.012
drat	-0.742	0.493	0.106
wt	0.888	-0.248	0.322
qsec	-0.534	-0.698	0.446
gear	-0.498	0.795	0.147
carb	0.582	0.699	0.330

Die Hauptkomponente weist negative und positive Korrelationen mit den Variablen auf. Mit

```
(Komm(prcomp(matcars,scale=T))[[2]])[,1:3]
```

ergibt sich

	PC1	PC2	PC3
mpg	0.874	0.876	0.900
cyl	0.917	0.917	0.949
disp	0.893	0.909	0.913
hp	0.762	0.913	0.913
drat	0.550	0.793	0.804
wt	0.789	0.850	0.954
qsec	0.285	0.773	0.971
gear	0.248	0.880	0.901
carb	0.338	0.827	0.935

Der Barlett- und der Levenetest unterscheiden nicht Gruppen von Variablen (Befehl s. obige Funktion: bartlett.pca(prcomp(matcars,scale=T)$x);
bartlett.pca(prcomp(matcars,scale=T)$x,T)

Insgesamt (% Erfassung Varianz, Screeplot, Kommunalitäten) wird man zum Schluss kommen, dass man die ersten zwei Hauptkomponenten verwenden wird. Wir erstellen eine Biplot:

```
mtcars.pca=prcomp(matcars,scale=T)
library(ggbiplot)
ggbiplot(mtcars.pca)
```

und erhalten die Abbildung 24:

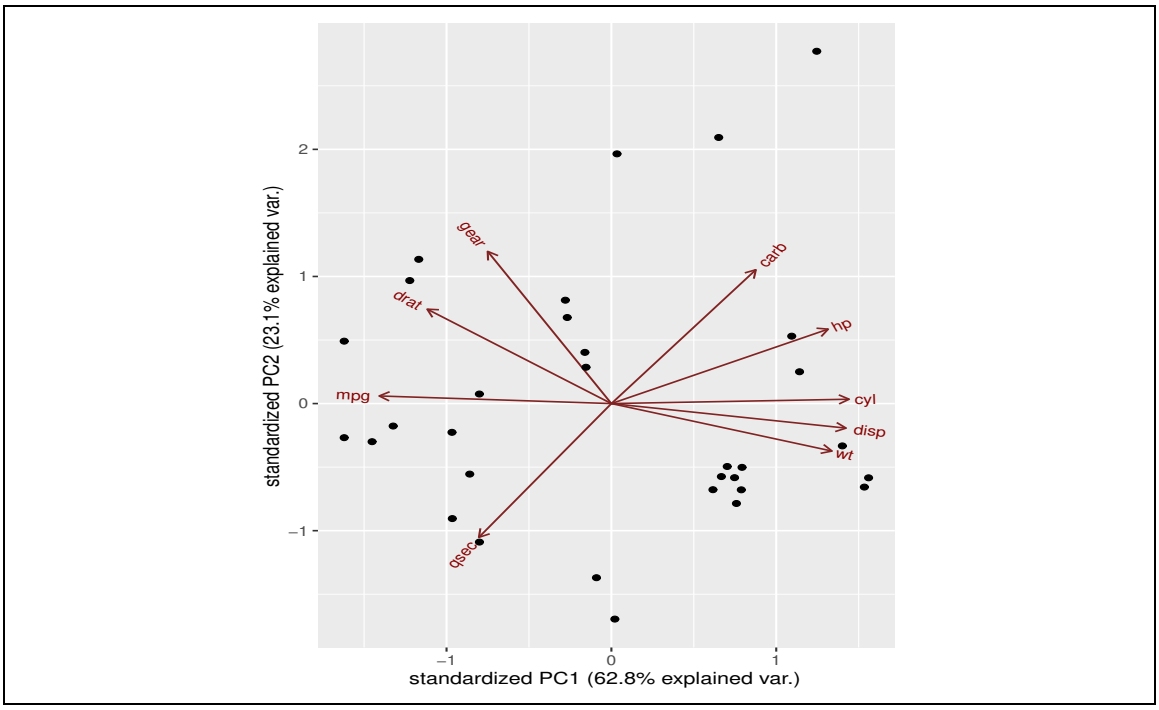

Abbildung 24: Biplot des mtcars-Beispiels

Hier sieht man, dass die Variablen hp, cyl und disp alle zu PC1 beitragen, wobei höhere Werte in diesen Variablen die Stichproben in der Darstellung nach rechts schieben. Ohne zu sehen, um welche Automarken es sich handelt, ist die Darstellung aber nicht sehr informativ. Deshalb fügt man am besten die Namen der Automarken hinzu. Mit

```
ggbiplot(mtcars.pca, labels=rownames(mtcars))
```

erhält man die Abbildung 25

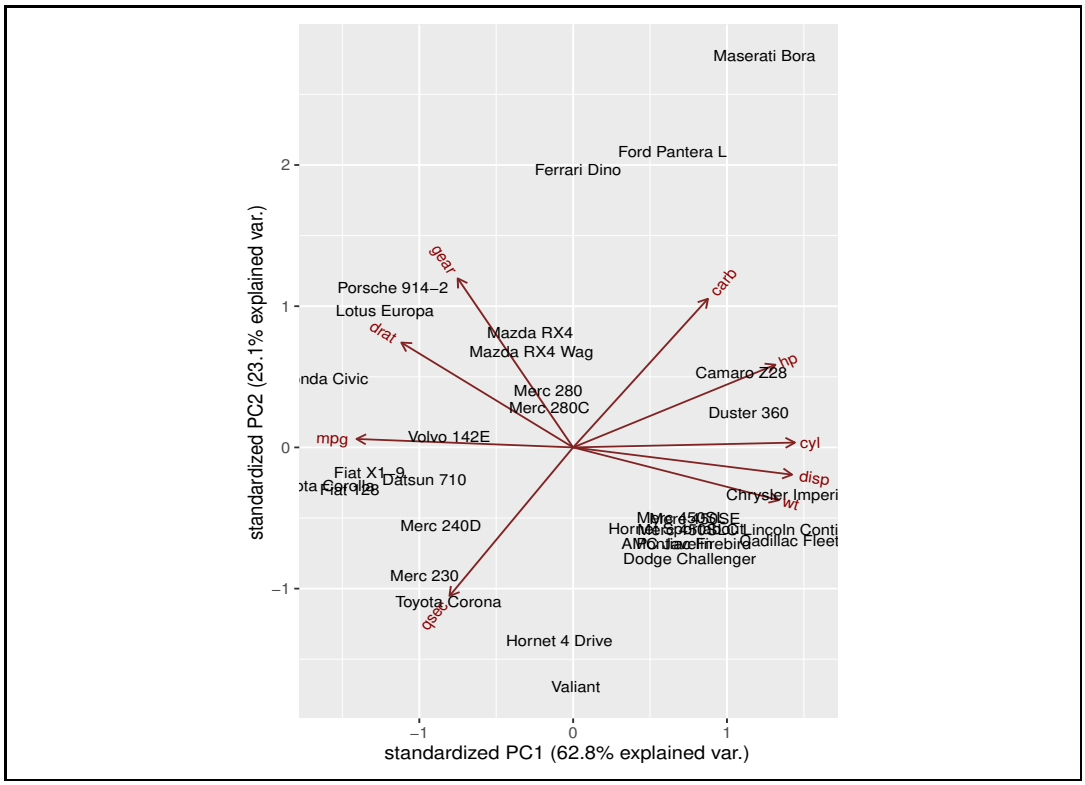

Abbildung 25: Biplot des mtcars-Beispiels mit Benennung der Automarken

Kennt man sich mit Automarken aus (z.B. Entwicklungsstandorte, Sport- versus Gebrauchswagen, etc.) so kann man es mit Clusterung versuchen. Man wisse, ob die Autos in den USA, in Europa oder in Japan entwickelt würden. Da diese Variable nicht im Datensatz vorkommt, kann man sie hinzufügen (Beispiel aus: https://www.datacamp.com/tutorial/pca-analysis-r):

```
mtcars.country = c(rep("Japan", 3), rep("US",4), rep("Europe", 7),
rep("US",3), "Europe", rep("Japan", 3), rep("US",4), rep("Europe", 3),
"US", rep("Europe", 3))
```

Mit

```
ggbiplot(mtcars.pca,ellipse=TRUE, labels=rownames(mtcars),
groups=mtcars.country)
```

erhält man die Abbildung 26:

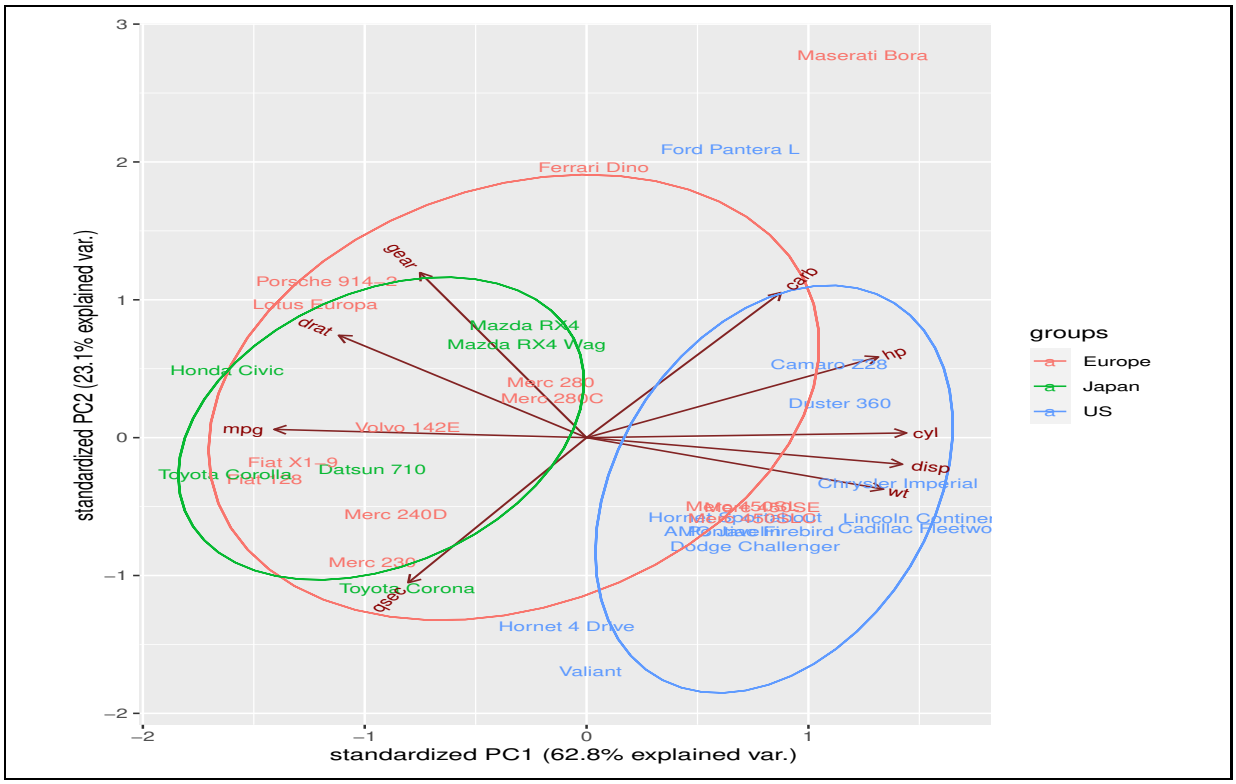

Abbildung 26: Biplot des mtcars-Beispiels mit Clusterbildung gemäss Entwicklungsstandort

Die US-amerikanischen Autos bilden eine deutliche Gruppe auf der rechten Seite. Man sieht, dass die US-amerikanischen Autos durch hohe Werte für Anzahl Zylinder (cyl), Gesamtvolumen der Zylinder (disp) und Gewicht (wt) gekennzeichnet sind. Die japanischen Fahrzeuge zeichnen sich dagegen durch einen hohen Benzinverbrauch (mpg) aus. Die europäischen Autos liegen eher in der in der Mitte und liegen weniger dicht beieinander als beide anderen Gruppen. Möchte man graphisch die Ellipsen verändern (z.B. eine Ellipse mit Punkten, eine mit Strichen und eine mit durchgezogener Linie, müsste man wohl mit ggplot arbeiten, d.h. die Ellipsen mit den entsprechenden Graphik-Parametern von Hand einzeichnen).

7 Rotation

Manchmal werden die Hauptachsen rotiert, um die Hauptkomponenten interpretierbarer zu machen (s. (Fahrmeir et al., 1996, S. 677 ff.)). Es werden orthogonale und oblique (schiefe) Transformationen unterschieden. Bei der orthogonalen Rotation werden die Hauptachsen und die Ladungen mittels einer orthogonalen Matrix rotierte. Die erklärten Varianzanteile der Variablen bleiben dadurch unverändert und die Hauptkomponenten bleiben unabhängig. Es gibt unterschiedliche Verfahren. Beim Varimaxverfahren (Kaiser, 1958) geht

es darum, die Unterschiede bei den Ladungen (Korrelationen zwischen den Hauptkomponenten und den standardisierten Ausgangsvariablen) möglichst gross werden zu lassen. Die Interpretation von „möglichst gross “ besteht bei der Varimax-Methode darin, die Varianz dieser Korrelationen möglichst gross werden zu lassen. Dies erreicht man dadurch, dass Gruppen von Variablen möglichst stark und andere Gruppen möglichst schwach geladen sind. Das Resultat kann die Interpretation der Hauptkomponenten mittels der Ursprungsvariablen erleichtern.

Es wird mit einem iterativen Verfahren gearbeitet. Mit R: varimax(). Einzugeben ist die Matrix der Korrelationen zwischen den Hauptkomponenten und den standardisierten Ausgangsvariablen (Ladungsmatrix), die in diesem Zusammenhang auch in R „matrix of loadings“ genannt wird. Es wird die neue Ladungsmatrix geliefert und die Rotationsmatrix (Transformationsmatrix). Im EWU-Beispiel erhält man mit

```
vEWUlad=Komm(prcomp(EWUdat,scale=T))[[1]]
varimax(vEWUlad)
```

die Ergebnisse (nichts steht in der Ladungsmatrix, wenn die Ergebnisse < 0.1 sind, s. Abbildung 27):

```
Loadings:
   PC1    PC2    PC3    PC4
X1  0.955  0.183 -0.159 -0.171
X2  0.955  0.151 -0.189  0.170
X3  0.192  0.942 -0.274
X4  0.203  0.276 -0.939

                 PC1   PC2   PC3   PC4
SS loadings    1.903 1.021 1.018 0.058
Proportion Var 0.476 0.255 0.254 0.015
Cumulative Var 0.476 0.731 0.985 1.000
```

Abbildung 27: Rotierte Ladungsmatrix

Man sieht, dass die Ladungen der rotierten Ladungsmatrix für die ersten zwei Variablen und die erste Hauptkomponenten grösser geworden sind, die zwei letzten Ladungen jedoch kleiner. Die erste Hauptkomponente kann also durch die ersten zwei Variablen interpretiert werden. Bezüglich der zweiten Hauptkomponente wurde die Ladung für die Variable 3 markant vergrössert. Man kann also versuchen, die zweite Hauptkomponente durch die dritte Variable zu verstehen. Die Nähe der dritten und vierten Variable verschwindet. Die vierte Variable korreliert nun negativ und hoch mit der dritten Hauptkomponente.

Bei „SS loadings“ steht die Summe der quadrierten Ladungen pro Hauptkomponente

(also der Spalten, Berechnung von Hand mit

```
ergvEWUlad=varimax(vEWUlad)
colSums(ergvEWUlad$loadingsˆ2)
```

Bei „Proportion Var“ steht dann der Anteil dieser Summen. Im Beispiel für die erste Hauptkomponente:

$$\frac{1.903}{1.903 + 1.021 + 1.018 + 0.58} = 0.476$$

Mit R von Hand mittels:

```
colSums(ergvEWUlad$loadingsˆ2)/sum(colSums(ergvEWUlad$loadingsˆ2))
```

Bei „Cumulative Var“ werden diese Anteile von links nach rechts kumuliert. Mit R von Hand mittels:

```
cumsum(colSums(ergvEWUlad$loadingsˆ2)/sum(colSums(ergvEWUlad$loadingsˆ2)))
```

Schliesslich wird die Rotationsmatrix geliert:

	1	2	3	4
1	0.77	0.45	-0.45	-0.00
2	-0.64	0.56	-0.53	-0.00
3	0.02	0.70	0.72	-0.02
4	0.00	0.02	0.02	1.00

Die neue Ladungsmatrix erhält man auch durch die Multiplikation der alten Ladungsmatrix mit der Rotationsmatrix. Im Beispiel:

```
Komm(prcomp(EWUdat,scale=T))[[1]]%*%ergvEWUlad[[2]]
```

Um Graphiken zu erstellen, kann man die Hauptkomponenten (prcomp(EWUdat,scale=T)$x) mit der Rotationsmatrix multiplizieren, die neuen Daten in einer Graphik abbilden. Ebenso kann man die ersten zwei Spalten der neuen Ladungsmatrix graphisch darstellen (als Punkte oder mit Pfeilen). Schliesslich kann man wieder verschiedene Varianten von Biplots erstellen. Die Auswirkungen der Varimax-Rotation fürs Beispiel kann man mit Hilfe der folgenden Abbildung 28 veranschaulichen:

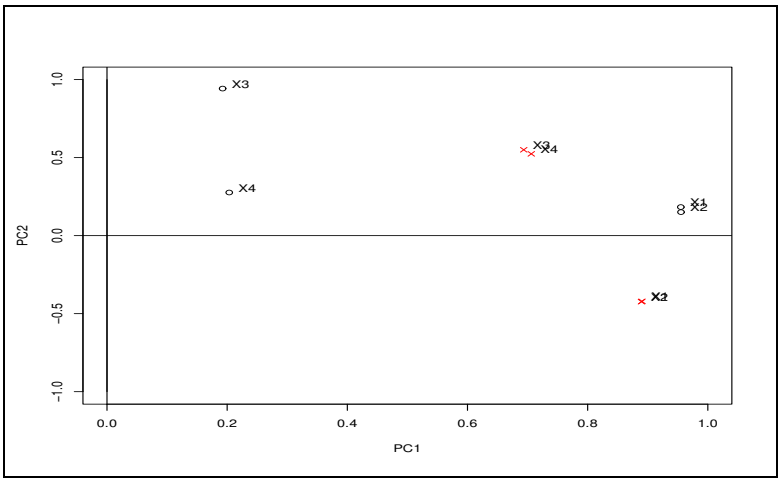

Abbildung 28: Ursprüngliche Ladungen bezüglich des alten Koordinatensystems (×) und neue Ladungen bezüglich des neuen Koordinatensystems (o)

Die Ladungen liegen nun näher an den neuen Hauptachsen. Durch die starke Korrelation in eine Richtung und die schwache in die andere, ist es manchmal leichter, die Hauptachsen mit Hilfe der Variablen zu interpretieren.

In der Beschreibung des varimax-Befehls von R wird noch die Beschreibung des promax-Befehls geliefert. Diese liefert eine oblique Transformation (also nicht orthogonal, s. (Hendrickson & White, 1964)). Die Hauptkomponenten sind nachher nicht mehr orthogonal und damit nicht mehr unabhängig.

Bemerkung 11. *Mit SPSS im PCA-Befehl in der Syntax /ROTATION VARIMAX hinzufügen (oder im Menu Entsprechendes anklicken). In der Ausgabe wird zusätzlich eine Tabelle mit dem Titel „Rotierte Summe der quadrierten Ladungen“ angeführt. Es werden die Ergebnisse von oben „SS loadings“, etc. geliefert, wobei die Anteile in % geliefert werden. Bei /ROTATION VARIMAX und „Faktorscores“ „Regression“ werden die standardisierten, rotierten Hauptkomponentenwerte in der Datentabelle geliefert. Mit R erhält man dies fürs EWU-Beispiel mittels:*

```
scale(prcomp(EWUdat,scale=T)$x)%*%ergvEWUlad[[2]]
```

In der „Ausgabe“ werden die neuen „Ladungen“ (Korrelationen) geliefert – unter dem Titel „Rotierte Komponentenmatrix“. Zudem wird die Rotationsmatrix geliefert: die orthogonale Matrix, welche die Hauptkomponenten und die Ladungen umrechnet (unter dem Titel „Komponententransformationsmatrix“).

8 Schlussbemerkungen

Die Hauptkomponentenanalyse hat einige Nachteile:

- Die Hauptkomponenten sind schwierig zu interpretieren.
- Die Interpretation ist subjektiv.
- Die Anzahl der auszuwählenden bedeutenden Hauptkomponenten ist nicht eindeutig.

Vorteile sind:

- Sie liefert Einsichten in die Korrelationsstruktur der Variablen
- Sie kann zu einer Reduktion der Anzahl Variablen führen.
- Die graphischen Darstellungen können zu Einsichten in die Daten führen (z.B. Cluster)
- Eliminiert irrelevante Information
- Vermindert die Gefahr der Konstruktion von Artefakten z.B. durch Ausreisser
- Das Verfahren ist numerisch stabiler als andere (z.B. ML-Faktoranalyse).

Die wesentlichen Schritte einer Hauptkomponentenanalyse sind:

- Man überprüft, ob die Beziehungen zwischen den Variablen affin sind (sonst kann man keine Pearson-Korrelationen zwischen den Variablen berechnen. Dies wurde in den obigen Beispielen nicht durchexerziert. Diese Bedingung wird bei nicht metrisch-skalierten Variablen nicht erfüllt sein!)
- Man berechnet die Korrelationsmatrix und überprüft, ob es offensichtliche Gruppen in den Variablen gibt. Wenn alle Korrelationen annähernd 0 sind, ist eine Hauptkomponentenanalyse nicht angebracht.
- Man entscheidet, welche Eigenwerte genügend gross sind. Die Zahl der zu berücksichtigenden Eigenwerte gibt die vermutliche Dimension des Raums an, in dem sich die Daten befinden.
- Man überprüft, ob die Hauptkomponenten Hinweise auf Gruppierungen der Variablen geben und versucht die Hauptkomponenten zu interpretieren (Analyse der Korrelationen zwischen den standardisierten Variablen und den Hauptkomponenten).
- Benutzung der als wichtig erachteten Hauptkomponenten in anderweitigen Analysen (Dimensionsreduktion durch Variablenreduktion).

Home-pages mit Interpretationsbeispielen:

- http://stat.cmu.edu/~cshalizi/350/lectures/10/lecture-10.pdf
- https://onlinecourses.science.psu.edu/stat505/node/54
- https://mmi.psycho.unibas.ch/r-toolbox/faktorenanalyse.html
- https://www.methodenberatung.uzh.ch/de/datenanalyse_spss/interdependenz/reduktion/faktor.html

Literatur

Bartholomew, D. J., Steele, F., Galbraith, J. & Moustaki, I. (2008). *Analysis of Multivariate Social Science Data* (2. Aufl.). London: Chapman and Hall/CRC.

Fahrmeir, L., Hamerle, A. & Tutz, G. (1996). *Multivariate statistische Verfahren* (2. Aufl.). Berlin: Walter de Gruyter.

Gabriel, K. R. (1971). The Biplot Graphic Display of Matrices with Application to Principal Component Analysis. *Biometrika*, *58* (3), 453–467.

Hendrickson, A. E. & White, P. O. (1964). Promax: A quick method for rotation to oblique simple structure. *British Journal of Statistical Psychology*, *17* (1), 65-70.

Kaiser, H. F. (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, *23*, 187 - 200.

Lawley, D. & Maxwell, A. (1971). *Factor analysis as a statistical method*. Butterworths.