Clusteranalyse mit R

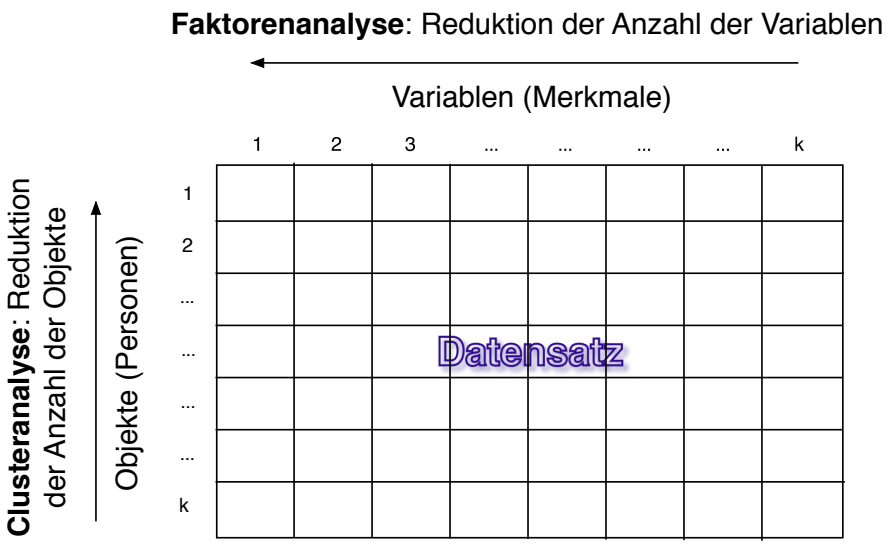
R User Group, Köln

Stephan Holtmeier

kibit GmbH, stephan@holtmeier.de

12. April 2013

Clusteranalyse: Um was es geht...



Die Clusteranalyse ist...

...ein strukturentdeckendes Verfahren. Ziel ist es, einander *ähnliche* Objekte (hier z.B. Führungskräfte, Abteilungen, Fragebogenitems, ...) den selben Clustern zuzuordnen.

Gefahr

In den selben Dingen können ganz unterschiedliche Muster erkannt werden.

Grundprinzip

Das Vorgehen bei einer Clusteranalyse ist im Prinzip sehr einfach:

1. Variablen/Merkmale festlegen, die zur Clusterung herangezogen werden sollen
2. Distanz-/Ähnlichkeitsmatrix (=**Proximitätsmatrix**) berechnen: Entscheiden, nach welchen Kriterien (Un-)ähnlichkeit definiert sein soll.
3. **Clusteralgorithmus** auf die Proximitätsmatrix anwenden.

Grundprinzip

Das Vorgehen bei einer Clusteranalyse ist im Prinzip sehr einfach:

1. Variablen/Merkmale festlegen, die zur Clusterung herangezogen werden sollen
2. Distanz-/Ähnlichkeitsmatrix (=**Proximitätsmatrix**) berechnen: Entscheiden, nach welchen Kriterien (Un-)ähnlichkeit definiert sein soll.
3. **Clusteralgorithmus** auf die Proximitätsmatrix anwenden.

Grundprinzip

Das Vorgehen bei einer Clusteranalyse ist im Prinzip sehr einfach:

1. Variablen/Merkmale festlegen, die zur Clusterung herangezogen werden sollen
2. Distanz-/Ähnlichkeitsmatrix (=**Proximitätsmatrix**) berechnen: Entscheiden, nach welchen Kriterien (Un-)ähnlichkeit definiert sein soll.
3. **Clusteralgorithmus** auf die Proximitätsmatrix anwenden.

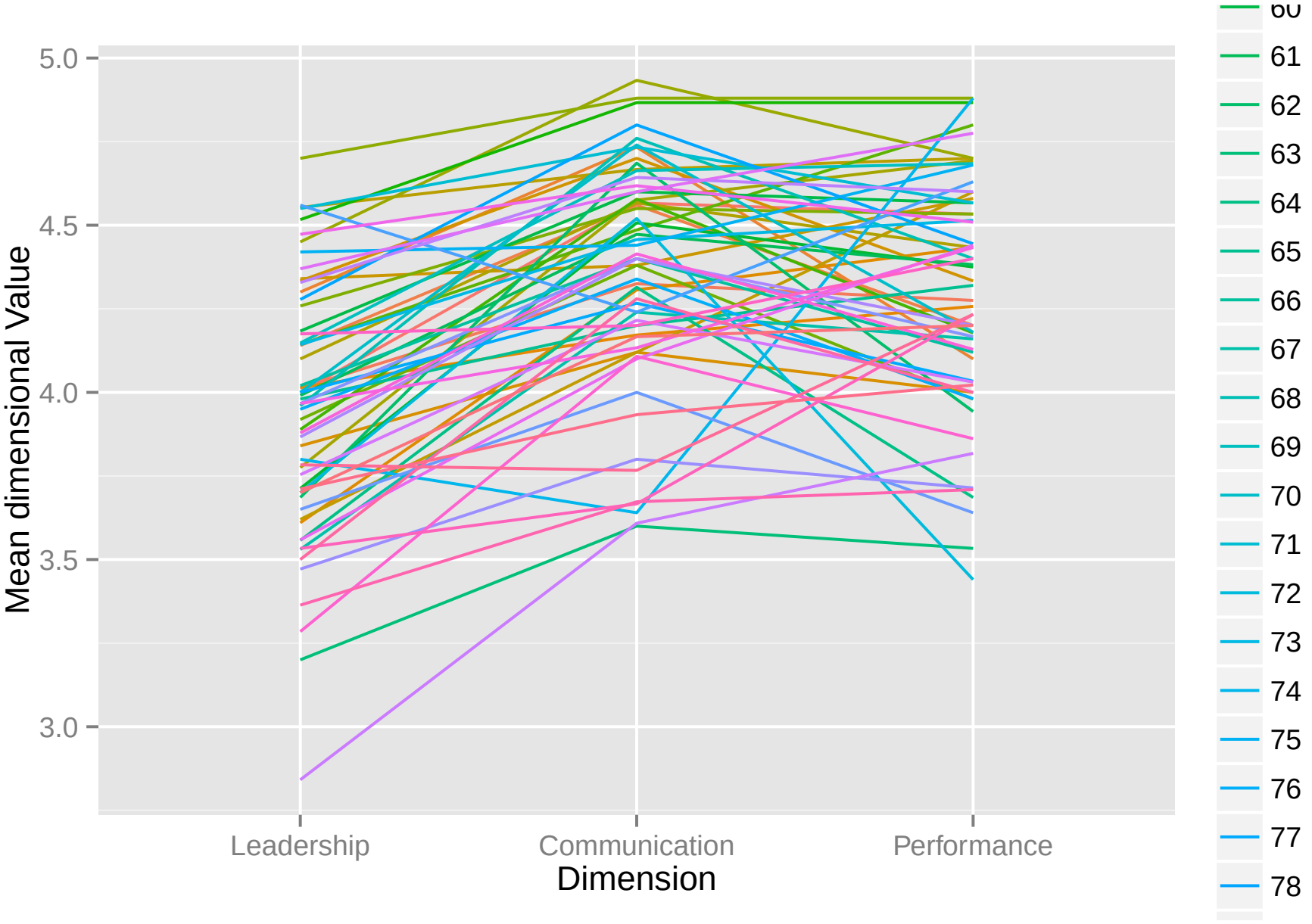
Beispieldatensatz: 360° Feedback

	Leadership	Communication	Performance
11	3.87	4.23	4.80
12	4.11	4.25	4.58
13	3.70	4.23	3.70
14	4.18	4.62	4.44
15	4.12	4.37	4.03

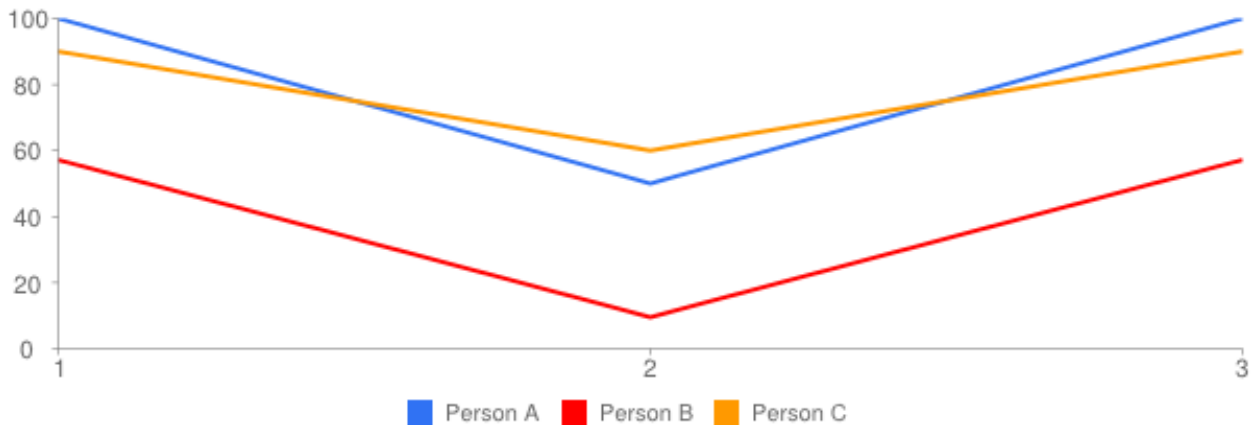
Tabelle : Beispieldatensatz (gekürzt)

Zwecks Komplexitätsreduktion wurde vorab eine Faktorenanalyse durchgeführt! Für jeden **Feedbackempfänger** (=Zeile) liegen drei uns drei **Dimensionsmittelwerte** (=Spalte) vor: Leadership, Communication und Performance

Beispieldatensatz (gekürzt) visualisiert



Welches Proximitätsmaß verwenden wir?



Das verwendete **Proximitätsmaß** ist abhängig vom *Skalenniveau* (i.d.R. haben wir bei kibit metrische Skalen) sowie von inhaltlichen Überlegungen (Korrelation vs. Distanz). Wir verwenden meist:

1. Q-Korrelationskoeffizient (Ähnlichkeit)
2. Euklidische Metrik (Distanz)

Proximitätsmaß berechnen

Euklidische Distanz

```
library(cluster)
dist<-daisy(bsp[,seq(1,3)], stand=TRUE, metric='euclidean')
```

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Q-Korrelation / Produktmomentkorrelation

```
qkorr<-round(1-abs(cor(t(bsp[,seq(1,3)]))),digits=3)
qkorr<-qkorr[lower.tri(qkorr)]
attr(qkorr,'class')<-'dist'
attr(qkorr,'Size')<-nrow(bsp)
```

Clusteralgorithmus auswählen

Es gibt verschiedene Clusterverfahren, die zu mehr oder weniger unterschiedlichen Ergebnissen führen.

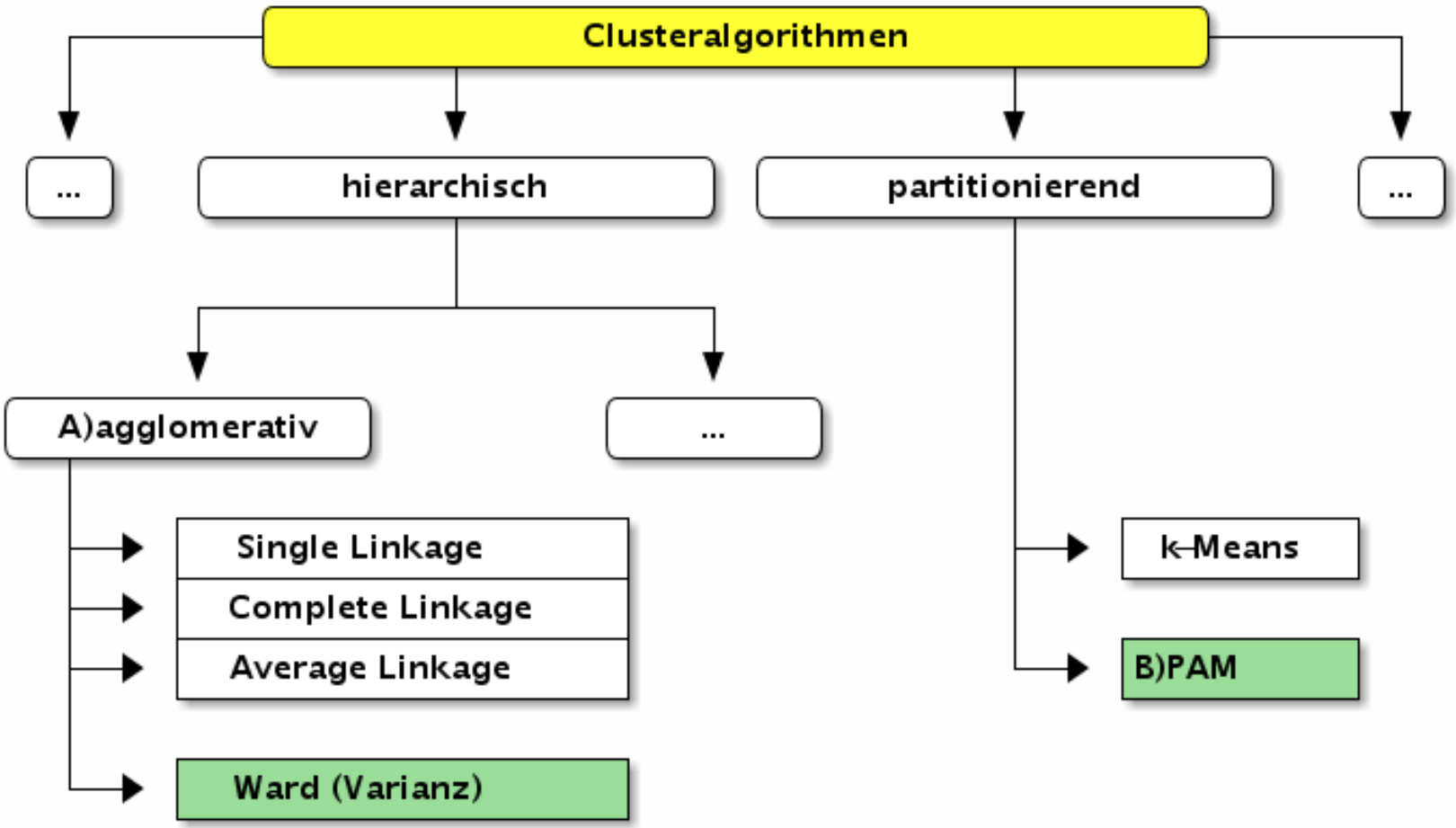
1. **Hierarchische Verfahren** gehen von der gröbsten („agglomerativ“) bzw. feinsten Partition aus. Durch Aufteilen bzw. Zusammenfassen werden Cluster gebildet. Einmal gebildet, können einzelne Elemente nicht mehr getauscht werden. Die Anzahl der Cluster wird zum Schluss festgelegt.
2. **Partitionierende Verfahren** verwenden eine gegebene Aufteilung und ordnen die Elemente durch Austauschfunktionen um, bis die verwendete Zielfunktion ein Optimum erreicht. Die Anzahl der Cluster wird zu Beginn festgelegt.

Clusteralgorithmus auswählen

Es gibt verschiedene Clusterverfahren, die zu mehr oder weniger unterschiedlichen Ergebnissen führen.

1. **Hierarchische Verfahren** gehen von der gröbsten („agglomerativ“) bzw. feinsten Partition aus. Durch Aufteilen bzw. Zusammenfassen werden Cluster gebildet. Einmal gebildet, können einzelne Elemente nicht mehr getauscht werden. Die Anzahl der Cluster wird zum Schluss festgelegt.
2. **Partitionierende Verfahren** verwenden eine gegebene Aufteilung und ordnen die Elemente durch Austauschfunktionen um, bis die verwendete Zielfunktion ein Optimum erreicht. Die Anzahl der Cluster wird zu Beginn festgelegt.

Übersicht Clusterverfahren



Hierarchisch agglomerativ (hclust) - Analyse

Ward-Algoritmus auf Basis der Euklidischen Distanz

```
wardclust<-hclust(dist,method='ward')
```

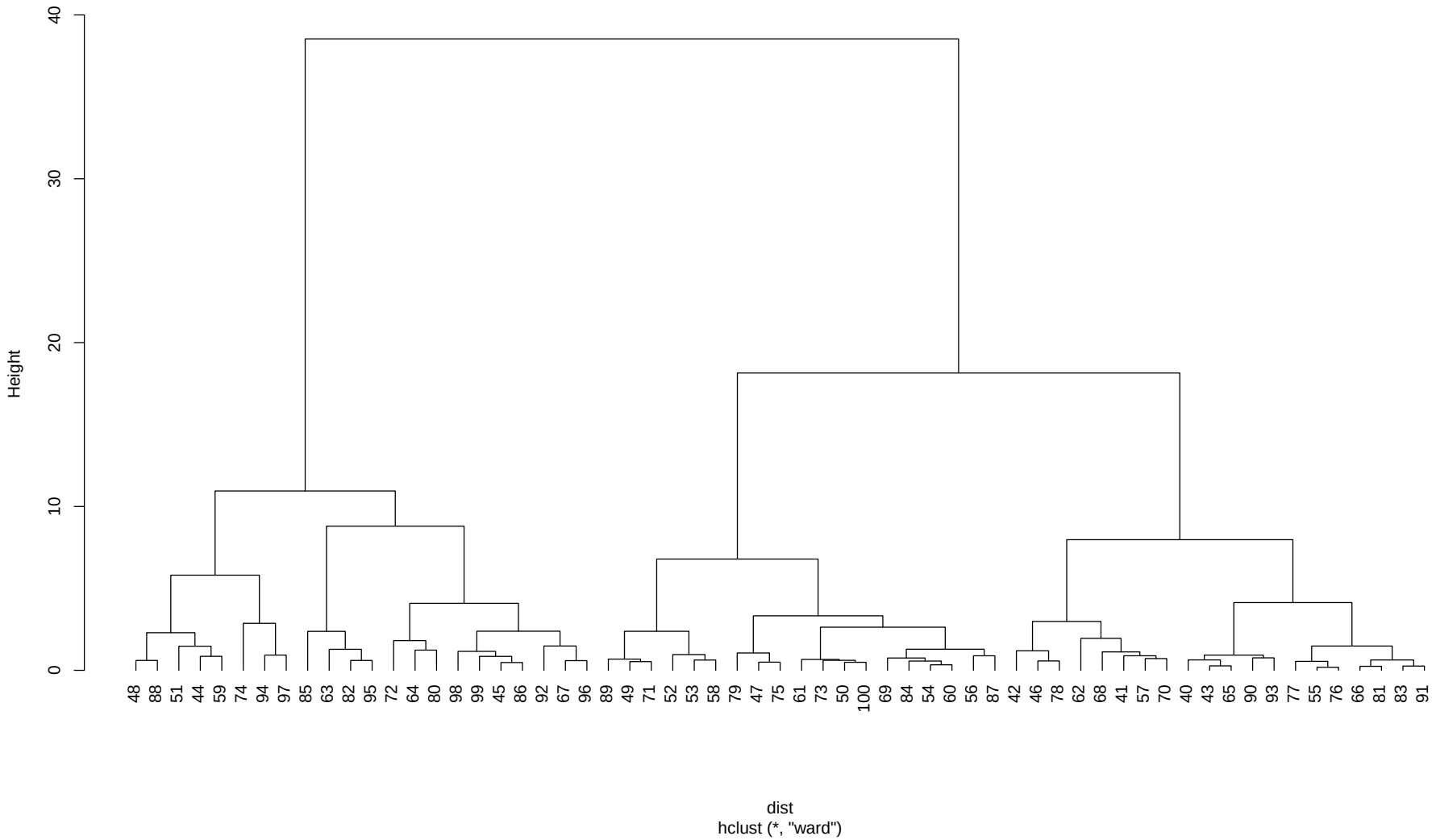
Der Ward-Algoritmus ist oft die beste Wahl, aber...

Alternative Algoritmen: **average**, **complete**, **single**. Ausreißer können gut via single-Linkage-Verfahren identifizieren und ggf. vorab eliminiert werden.

Visualisierung der Clusterdaten - Dendrogramm

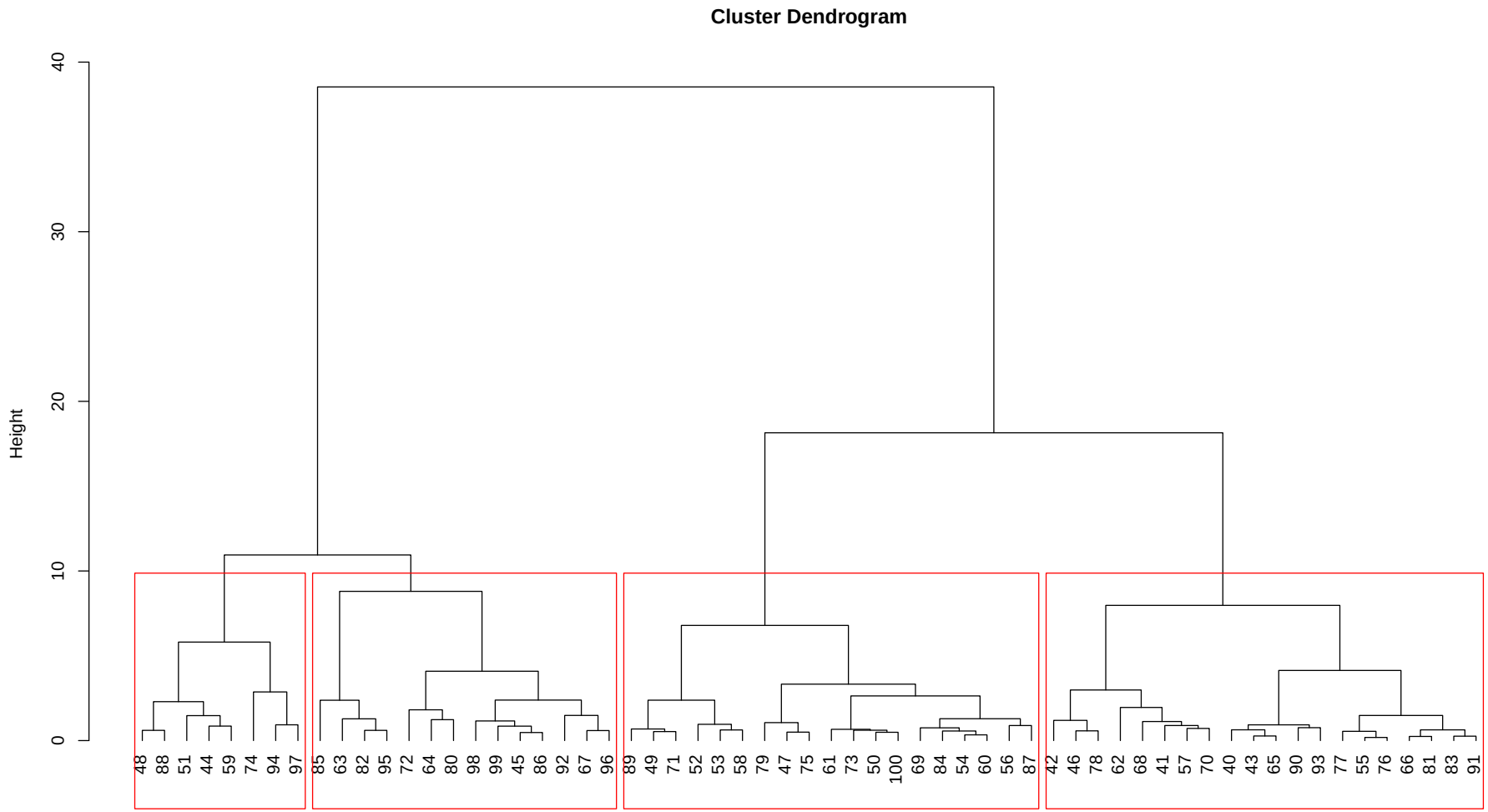
```
plot(wardclust,hang=-0.01)
```

Hierarchisch agglomerativ (hclust) - Dendrogramm



Hierarchisch agglomerativ (hclust) - Clusteranzahl?

```
rect.hclust(wardclust, k = 4)
```



Partitionierend (PAM) - Analyse

PAM-Algoritmus auf Basis der Euklidischen Distanz

```
pamclust<-pam(dist,4,diss=TRUE)
```

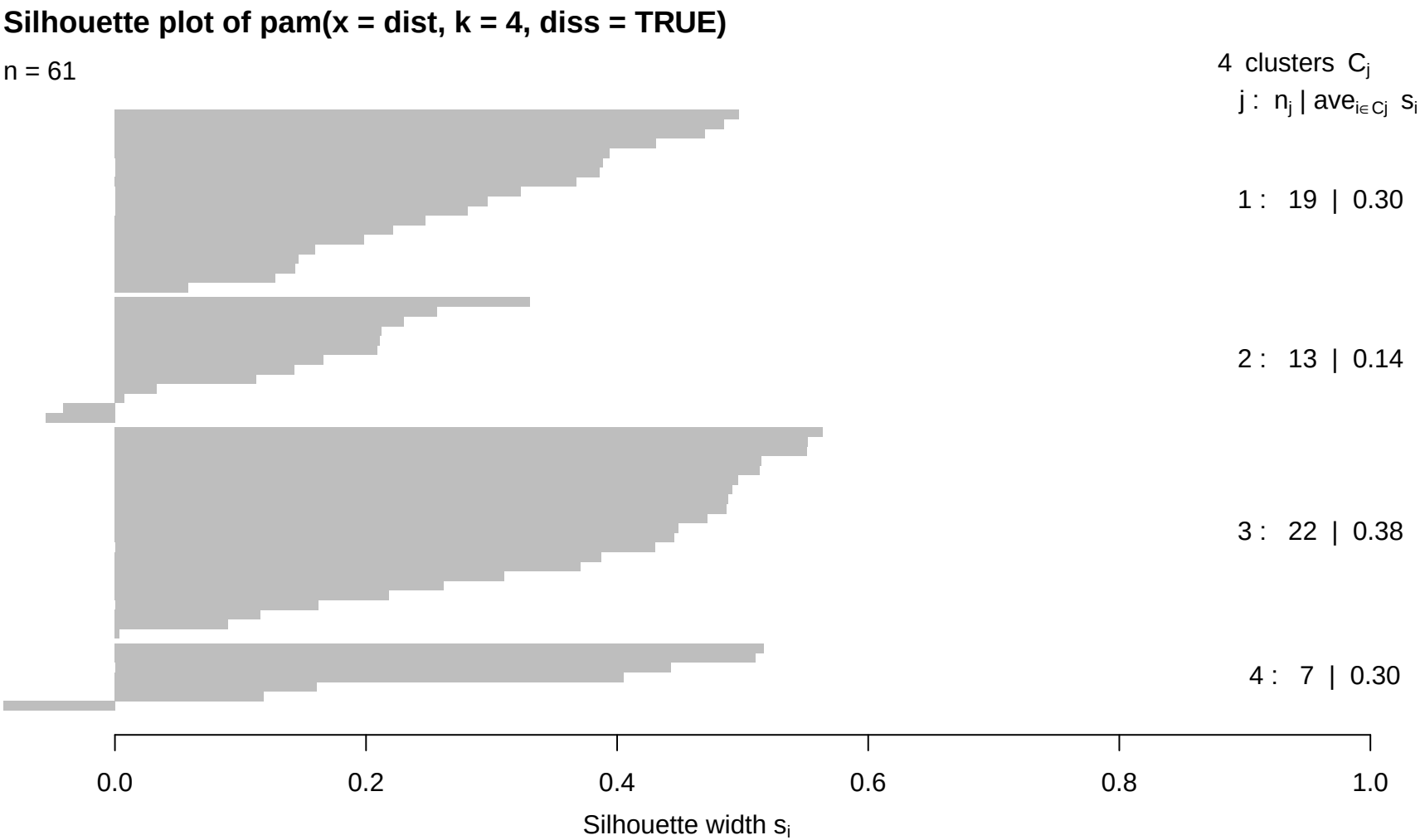
PAM = Partinioning Around Medoids

robustere Alternative zu k-Means; vergl. Hellbrück (2009). Im Unterschied zu den hierarchisch agglomativen Verfahren muss hier die Anzahl der Cluster vorab festgelegt werden.

Visualisierung der Clusterdaten - Silhouette-Plot

```
plot(pamclust)
```

Partitionierend (PAM) - Silhouette-Plot



Average silhouette width : 0.29

Interpretation eines Silhouette-Plot

ASW[1]	Interpretation
0.71-1.0	super Clusterstruktur!
0.51-0.70	gute Clusterstruktur
0.26-0.50	schwache Clusterstruktur, evtl. artifiziell
< 0.25	unzureichende Clusterstruktur

Auch für hclust() kann der Silhouetteplot angefordert werden

```
plot(silhouette(cutree(wardclust,4),dist))
```

[1] AWS=Average silhouette width

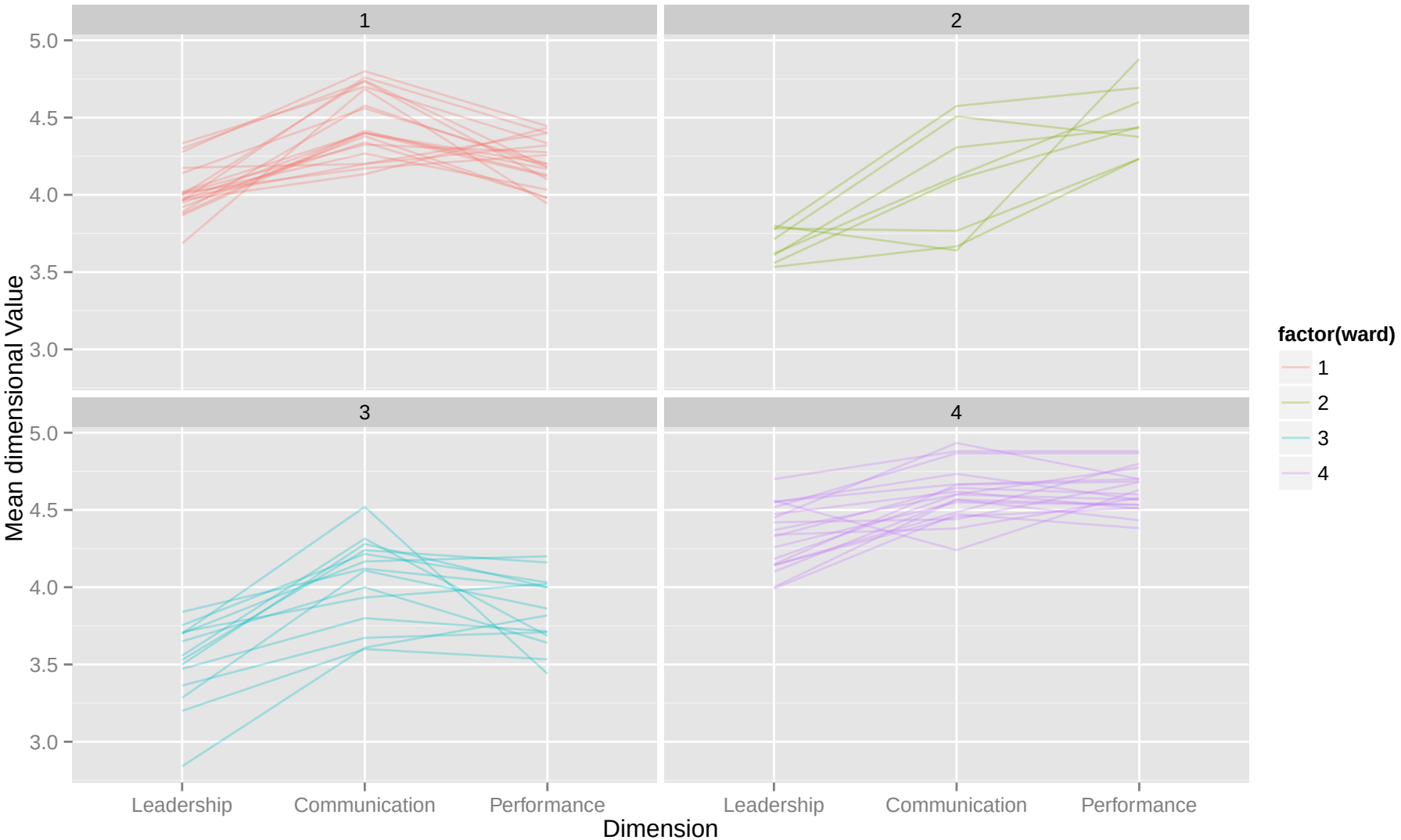
Cluster interpretieren I

```
library(reshape, ggplot2)

#Clusterzuordnung dem Datensatz hinzufügen:
bsp$ward<-factor(cutree(wardclust,k=4))
bsp$pam<-factor(pamclust$clustering)

long <- melt(data.frame(FEID=rownames(bsp),bsp),
    id.vars=c("FEID", "ward"), measure.vars=seq(2,4))
ggplot(long, aes(x=variable, y=value, color=factor(ward))) +
    geom_line(aes(group=FEID), alpha=.3) +
    labs(x = "Dimension", y = "Mean dimensional Value") +
    facet_wrap(~ ward)
```

Cluster interpretieren II



Empfehlungen zur Vertiefung

1. **pvclust()** berechnet p-Werte für hierarchische Cluster auf Basis von "Multiscale Bootstrap Resampling". Die Daten müssen transponiert werden. Sehr rechenintensiv! Cluster mit p-Wert größer .95 werden optisch hervorgehoben, denn sie erfahren starken Support.
2. Visualisierung mit **clusplot()** und **plotcluster()**
3. Und zum nachlesen: Das Kapitel "Clusteranalyseïm Backhaus (2003)
4. **cluster.stats()** aus dem **fpc**-Paket