

## Kapitel: Clusteranalyse (Version: 01.11.2021)

(Anregungen und Rückfragen an: 📧 Prof. Dr. Markus Pospeschill ✉ [pospeschill@mx.uni-saarland.de](mailto:pospeschill@mx.uni-saarland.de))

### Inhaltsverzeichnis

Agglomerative hierarchische Clusteranalyse .....	2
Funktion hclust() .....	3
Funktion agnes() .....	4
Divisive hierarchische Clusteranalyse .....	6
Funktion mona() .....	6
Unschärfe partitionierende Clusteranalyse (fuzzy clustering) .....	7
Funktion fanny() .....	7

Clusteranalytische Verfahren gehören zu den strukturentdeckenden Verfahren innerhalb der multivariaten Statistik. Sie dienen dazu, Elemente anhand mehrerer Merkmale zu (möglichst) homogenen bzw. ähnlichen Gruppen zu klassifizieren (zu „clustern“). Die Elemente können dabei Fälle (Personen), unbelebte Objekte (z. B. Güter oder Produkte) oder auch Aggregate bzw. komplexere Einheiten (z. B. Institutionen oder Organisationen) sein. Die Clusteranalyse wird in dieser Funktion vornehmlich als exploratives Verfahren eingesetzt, um ggf. in einem zweiten Analyseschritt gefundene gegenüber a priori gesetzten (oder theoretisch angenommenen) Gruppenzuordnungen zu vergleichen (z. B. mittels einer Diskriminanzanalyse).

Ziel der Clusteranalyse ist es (ähnlich einer Faktorenanalyse) eine Datenreduktion durch Strukturierung bzw. Typologisierung der Daten zu erzielen (allerdings hier seltener über die Messvariablen, als vielmehr über die Elemente). Dabei werden die Datensätze durch Ähnlichkeits- bzw. Unähnlichkeitszuordnungen (Distanzen) strukturiert. Das Ziel dieses Vorgehens ist es, die Zuordnungen der Elemente so vorzunehmen, dass sich die Elemente eines Clusters möglichst ähnlich sind (dies entspricht einer hohen Intra-Cluster-Homogenität), während sich die Elemente in unterschiedlichen Clustern möglichst unähnlich sein sollen (dies entspricht einer geringen Inter-Cluster-Homogenität).

Typisch ist weiter, dass die Ergebnisse von Clusteranalysen nicht im klassischen Sinne inferenzstatistisch abgesichert, sondern bestenfalls mittels Gütemaßen oder außerstatistischen Erwägungen in ihrer Lösung beurteilt werden können. Gütemaße bestimmen dazu, wie gut das ermittelte statistische Modell mit den empirischen Daten übereinstimmt („fittet“). Daneben kann aber auch ein Abgleich oder eine Validierung mit theoretischen Annahmen oder Vorhersagen (z. B. Vorab-Klassifikationen) erfolgen.

Pospeschill, M. (2021/22). *Ergänzungen zu R*. Saarbrücken: Universität des Saarlandes.

Hinter dem Begriff der Clusteranalyse verbergen sich allerdings verschiedene statistische Verfahren. Eine Gruppe sind deterministische Clusteranalysen, bei der die Elemente nur einem Cluster zugeordnet werden können, in dem sie auch nach Zuordnung verbleiben. Alternativ sind probabilistische Clusterverfahren zu nennen, bei denen jedes Element Schätzungen zu Wahrscheinlichkeiten für die Zuordnung zu den Clustern erhält.

Voraussetzung für die Durchführung einer Clusteranalyse ist die Transformation der Rohdaten anhand von Proximitätsmaßen in eine Ähnlichkeits- oder Distanz-Datenmatrix durch paarweisen Vergleich der Elemente. Dabei ist zuvor zu entscheiden, welches Skalenniveau den Daten unterliegt. Distanzmaße finden Anwendung bei metrischen Daten, da nur solche Daten eine geometrisch exakte Erfassung von Abständen zulassen; hier ist die Distanz eines Elementes zu sich selbst 0. Bei nominalen Daten hingegen werden zumeist Ähnlichkeitsmaße verwendet, bei denen die Ähnlichkeit eines Elements zu sich selbst 1 ist. Zur genauen Berechnung der verschiedenen Ähnlichkeits- oder Distanzmaße siehe u. a.: Pospeschill, M. (2018). *SPSS für Einsteiger und Fortgeschrittene*. 11. Auflage, Hannover: Leibniz Universität IT-Services oder Wentura, D. & Pospeschill, M. (2015). *Multivariate Statistik*. Wiesbaden: Springer.

Als weitere Entscheidung wird die Spezifizierung des Fusionierungsverfahrens benötigt, das festlegt, wie die Elemente den Clustern zugeordnet werden. Auch hier stehen verschiedene Verfahren zur Wahl. Zur Gruppe der hierarchischen Clusterverfahren gehört das agglomerative Verfahren, das zunächst einzelne Elemente in einen gemeinsamen Cluster fusioniert und dann sukzessive weitere Elemente diesen oder weiteren Clustern zuordnet, bis sich alle Elemente in einem gemeinsamen Cluster befinden. Das divisive Verfahren kehrt dieses Vorgehen um, in dem es zunächst alle Elemente in einen Cluster ablegt und dann sukzessive in mehrere Cluster trennt. Als weitere Gruppe werden partitionierende Verfahren unterschieden, die eine vorgegebene Gruppeneinteilung verwenden und versuchen durch Umgruppierung einzelner Objekte eine bestehende Lösung zu optimieren.

Der Fusionierungsprozess hierarchischer Clusterverfahren kann durch Dendrogramme („Strukturbaum“) oder Banner-Plots visualisiert werden, diese zeigen die einzelnen Schritte der Zusammenführung der Elemente zu den Clustern.

## Agglomerative hierarchische Clusteranalyse

Die erste Klasse von deterministischen Cluster-Methoden verwendet eine hierarchische Klassifikation von Daten, die aus einer Abfolge von Unterteilungen der einzelne Elemente zu Clustern besteht, solange bis sich alle Elemente in einem Cluster befinden. Bevor dieser Prozess allerdings gestartet werden kann, sind die Daten in eine Distanz- oder Ähnlichkeitsmatrix zu überführen, wie z. B. eine Matrix der Euklidischen Distanzen. Liegt diese Matrix vor beginnt der Fusionierungsprozess, der nach jeder Zuordnung eines Elementes die Distanzen zwischen den verbleibenden Elementen und Clustern neu bestimmt. Für die Bestimmung dieser neuen Distanzen besteht wiederum die Auswahl zwischen verschiedenen Agglomerierungsverfahren.

Pospeschill, M. (2021/22). *Ergänzungen zu R*. Saarbrücken: Universität des Saarlandes.

Zur Durchführung dieser Clusteranalyse können die Funktionen `hclust()` aus dem Paket `stats` und `agnes()` aus dem Paket `cluster` verwendet werden. Alternativ kann aus dem letztgenannten Paket die Funktion `diana()` für eine divisive hierarchische Clusteranalyse oder `mona()` bei binären Daten verwendet werden. Zur Visualisierung stehen die Funktion `dendrogram()` aus dem Paket `stats` oder `plot()` aus dem Paket `cluster` zur Verfügung.

Weitere Funktionen stellt das Paket `cluster` bereit. Die Installation des Paketes erfolgt mit: `install.packages("cluster")`

## Funktion `hclust()`

Zunächst soll die Funktion `hclust()` aus dem Paket `stats` am Beispiel erläutert werden.

Die Funktion `hclust()` erwartet als Datensatz eine Distanz-/Unähnlichkeitsmatrix, die (falls noch nicht vorhanden) mit dem Argument `dist()` direkt aus den Daten erzeugt werden kann. Mit dem Argument `method` können verschiedene Agglomerierungsverfahren gewählt werden: `"single"`, `"complete"`, `"average"`, `"mcquitty"`, `"median"`, `"centroid"` sowie `"ward.D"` und `"ward.D2"` (D2 berücksichtigt dabei ein besonderes Cluster-Kriterium, dass die Distanzen vor jedem Cluster-Update quadriert).

Einige dieser Verfahren sind eher konservativ in der Anwendung (insbesondere Zentroid, Median und Ward), so dass eine Verwendung nur bei (metrischen) Distanzmaßen sinnvoll ist. Linkage-Verfahren (wie Single, Complete und Average) hingegen sind bei der Verwendung von Proximitätsmaßen flexibler. Bei der Wahl gilt besonders die Ward-Methode als verlässlich, vorausgesetzt, die Variablen besitzen ein metrisches Skalenniveau, sind unkorreliert und weisen keine Ausreißer auf. Ward sollte zudem mit der Annahme verknüpft sein, dass in etwa gleiche große Cluster mit gleicher Varianz vorliegen. Complete-Linkage ist ein dilatierendes Verfahren und neigt zur Bildung etwa gleich großer Cluster, die insgesamt aber kleiner bleiben. Single-Linkage verwendet einen kontrahierenden Algorithmus und neigt zur Bildung zunächst weniger großer Cluster, denen viele kleine Cluster gegenüberstehen; dabei kann eine Aneinanderreihung oder Kettenbildung die Folge sein. Kontrahierende Verfahren sind vor allem zur Identifizierung von Objekten mit extremen Merkmalen (die den Fusionierungsprozess verzerren) sinnvoll. Diese werden auf späteren Stufen des Fusionierungsprozesses zu bestehenden Clustern hinzugefügt (vgl. Wentura, D. & Pospeschill, M. (2015). *Multivariate Statistik*. Wiesbaden: Springer).

Um das eigentliche Ergebnis einer Clusteranalyse zu sehen, muss das Resultat an die Funktion `plot()` übergeben werden. Diese erzeugt ein Dendrogramm, dass den Fusionierungsprozess visualisiert und die zusammengefassten Cluster identifizierbar macht. Durch das optionale Argument `hang = -1` besteht dabei die Möglichkeit, die Äste mit den Objektbeschriftungen auf eine Höhe zu bringen.

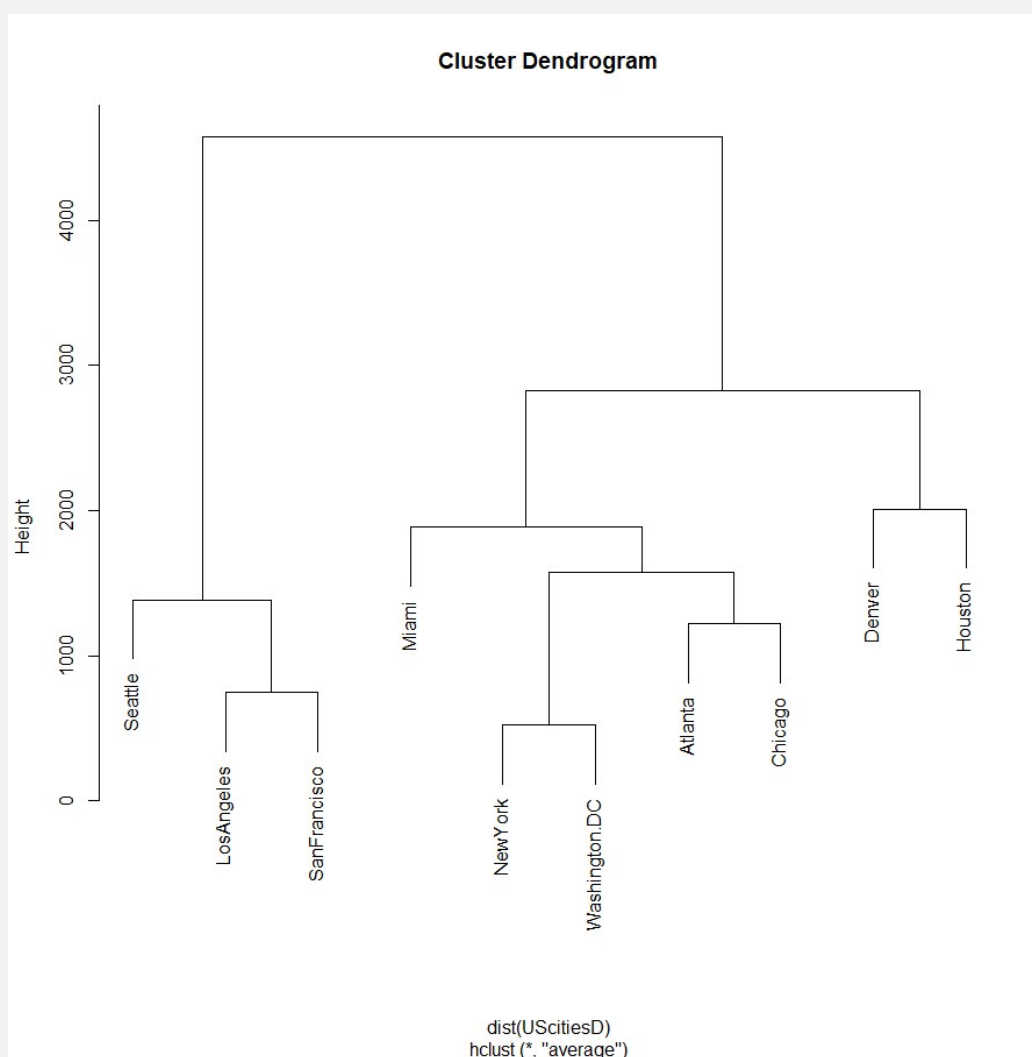
Zur Erleichterung der Identifizierung einzelner Cluster kann optional die Funktion `rect.hclust()` verwendet werden. Hiermit wird eine bestimmte Anzahl von Clustern im Dendrogramm farblich eingerahmt, z. B. zur Identifizierung von 3 Clustern aus dem vorherigen Beispiel: `rect.hclust(hc, k = 3)`. Alternativ kann über das Argument `h` auch die seitliche Skalierung (*Height*) als Marker für die Unterteilung gewählt werden, z. B. mit: `rect.hclust(hc, h = 2000)`.

Pospeschill, M. (2021/22). *Ergänzungen zu R*. Saarbrücken: Universität des Saarlandes.

```
# Datensatz UScitiesD aus dem Paket stats

# Syntax
hc <- hclust(dist(UScitiesD), method = "average")
plot(hc)

# Ausgabe
## Cluster method      : average
## Distance            : euclidean
## Number of objects: 10
```



## Funktion agnes()

Ähnliche Funktionen stehen im Paket `cluster` zur Verfügung. Eine hierarchische Clusterbildung kann mittels der Funktion `agnes()` (*agglomerative nesting*) vorgenommen werden. Dazu wird entweder eine bereits erzeugte Distanz-Matrix oder ein Datensatz übergeben, der dann in eine entsprechende Unähnlichkeitsmatrix transformiert wird. Bei

Pospeschill, M. (2021/22). *Ergänzungen zu R*. Saarbrücken: Universität des Saarlandes.

einem numerischen Datensatz sollte jede Zeile mit einer Beobachtung und jede Spalte mit einer Variable korrespondieren.

Um eine Distanz-Matrix zu erzeugen stellt die Funktion mittels des Arguments **metric** Euklidische Distanzen ("**euclidean**") oder Manhattan-Distanzen ("**manhattan**") zur Verfügung. Euklidische Distanzen entstehen aus der Quadratwurzel der Quadratsummen der Differenzen, während Manhattan-Distanzen (auch City-Block-Metrik genannt) aus der Summe der absoluten Differenzen entstehen. Weitere Distanz-Matrizen lassen sich über die gesonderte Funktion **dist** (aus dem Paket stats) oder **daisy** (aus dem Paket cluster) erzeugen.

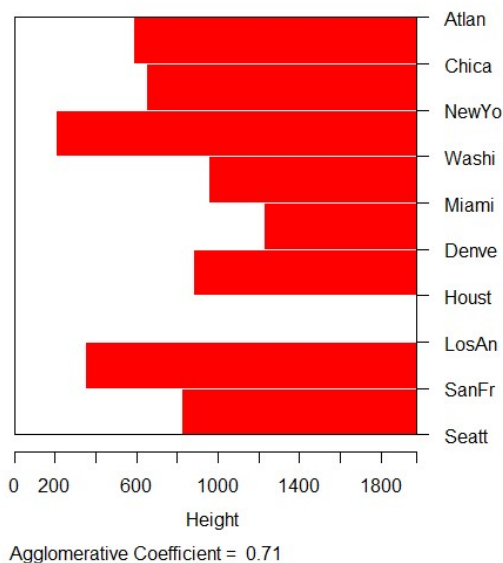
```
library("cluster")

# Datensatz UScitiesD aus dem Paket stats

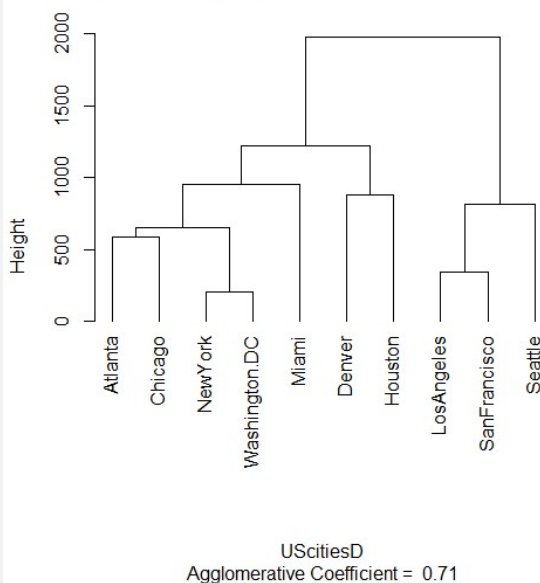
# Syntax
hc <- agnes(UScitiesD, method = "average")
plot(hc, hang = -1)

# Ausgabe
## Call:      agnes(x = UScitiesD)
## Agglomerative coefficient:  0.7060197
## Order of objects:
## [1] Atlanta Chicago NewYork Washington.DC Miami Denver ...
## Height (summary):
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 205.0   587.0   818.5   848.5   951.8   1975.0
```

**Banner of agnes(x = UScitiesD)**



**Dendrogram of agnes(x = UScitiesD, method = "average")**



Pospeschill, M. (2021/22). *Ergänzungen zu R*. Saarbrücken: Universität des Saarlandes.

Ähnlich zur Funktion `hclust()` können mit dem Argument `method` verschiedene Agglomerierungsverfahren ausgewählt werden: "average" (default), "single", "complete", "ward", "weighted", "flexible" (eher mit Vorsicht anzuwenden) sowie "gaverage".

Der Banner-Plot stellt ein horizontales Balken-Diagramm dar, dass die (agglomerative oder divisive) Clusterung aus der Dendrogramm-Struktur zeigt. Dieser lässt sich auch separat mit der Funktion `bannerplot()` aus dem Paket `cluster` erzeugen.

Der Agglomerationskoeffizient (als besonderes Feature der Funktion `agnes()`) stellt einen Durchschnittswert nach dem Average-Linkage-Verfahren, bezogen auf Gruppen dar. Er ermittelt sich aus der Unähnlichkeit eines Objektes zum ersten zugeordneten Cluster, dividiert durch die Unähnlichkeit der finalen Fusionierung in der Clusteranalyse. Gemittelt über alle Objekte beschreibt der Koeffizient damit die Eindeutigkeit einer Clusterstruktur. Niedrige Werte deuten auf eine kompakte Clusterbildung, größere Werte auf weniger ausgeformte Cluster hin.

## Divisive hierarchische Clusteranalyse

Alternative hierarchische Clusteranalysen sind divisive Verfahren wie sie in `diana()` (Divise Analysis Clustering) oder `mona()` (Monothetic Analysis Clustering of Binary Variables) implementiert sind. Divisive Verfahren starten mit einem gemeinsamen Cluster aller Elemente und unterteilen diese dann so lange, bis am Ende jedes Element einen eigenen Cluster bildet.

### Funktion `mona()`

Dabei sind die bisher behandelten hierarchischen Methoden allesamt polythetisch, d. h., sie berücksichtigen bei der Clusterbildung alle Variablen gleichzeitig. Monothetische Verfahren wie z. B. die Funktion `mona()` hingegen berücksichtigen bei der Unterteilung immer nur eine gezielt ausgesuchte Variable. Dabei werden auf jeder Stufe alle Cluster in Abhängigkeit des Wertes einer Variable unterteilt. Der `mona`-Algorithmus erlaubt zudem als Datensatz nur Variablen (Merkmale der Elemente) mit zwei (binären) Ausprägungen (0/1-Werte).

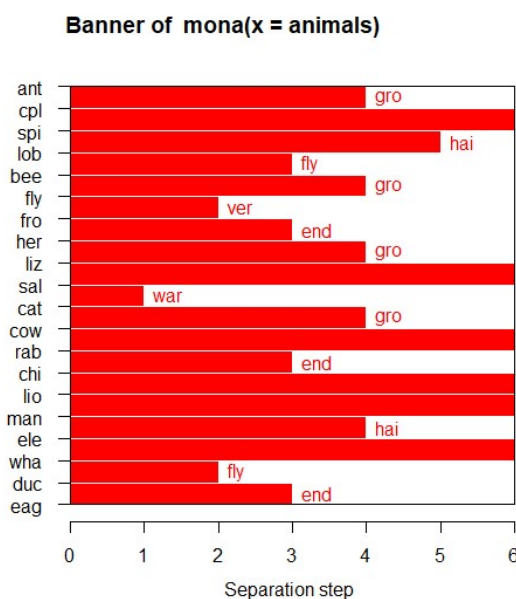
```
library("cluster")

# Datensatz animals aus dem Paket cluster

# Syntax
ma <- mona (animals)
plot(ma)

# Ausgabe
```

Pospeschill, M. (2021/22). *Ergänzungen zu R*. Saarbrücken: Universität des Saarlandes.



## Unschärfe partitionierende Clusteranalyse (fuzzy clustering)

Beim sog. fuzzy clustering, ein probabilistisches Clusterverfahren, wird davon ausgegangen, dass sich jedes Element über mehrere Cluster erstreckt, allerdings mit einem unterschiedlichen Grad der Zugehörigkeit. Der Grad der Zugehörigkeit (*membership*) ist dabei eine nicht-negative Größe und über alle angenommenen Cluster hinweg in der Summe 1 (bzw. 100 %).

### Funktion `fanny()`

Ähnlich zu den vorherigen Clusteranalysen verwendet die Funktion `fanny()` (*Fuzzy Analysis Clustering*) Daten aus einer Distanz-Matrix, erwartet aber als partitionierendes Verfahren eine Angabe zur (erwarteten oder angenommenen) Anzahl der Cluster.

Bei Rohdaten stehen mittels des Arguments `metric` Euklidische Distanzen ("`euclidean`"), Manhattan-Distanzen ("`manhattan`") sowie quadrierte Euklidische Distanzen ("`SqEuclidean`") zur Wahl.

Optional kann über den Parameter `memb.exp` ein *Membership*-Exponent ( $>1$ , Default: 2) angegeben werden, der für das Fit-Kriterium verwendet wird. Dieser führt dazu, dass die Clusterzugehörigkeit eindeutiger ( $<2$ ) oder weniger eindeutig (*more fuzzy*) wird ( $>2$ ). In der Ausgabe wird unter dem *Membership*-Koeffizienten entsprechend in Prozent die Zugehörigkeit jedes Elementes zu jedem Cluster angezeigt.

Schließlich kann bei Bedarf auch die Anzahl der durchgeführten Iterationen (Default: `maxit` = 500) und die Toleranz für die relative Konvergenz des Fit-Kriteriums (Default: `tol` =  $1e-15$ ) eingestellt werden.





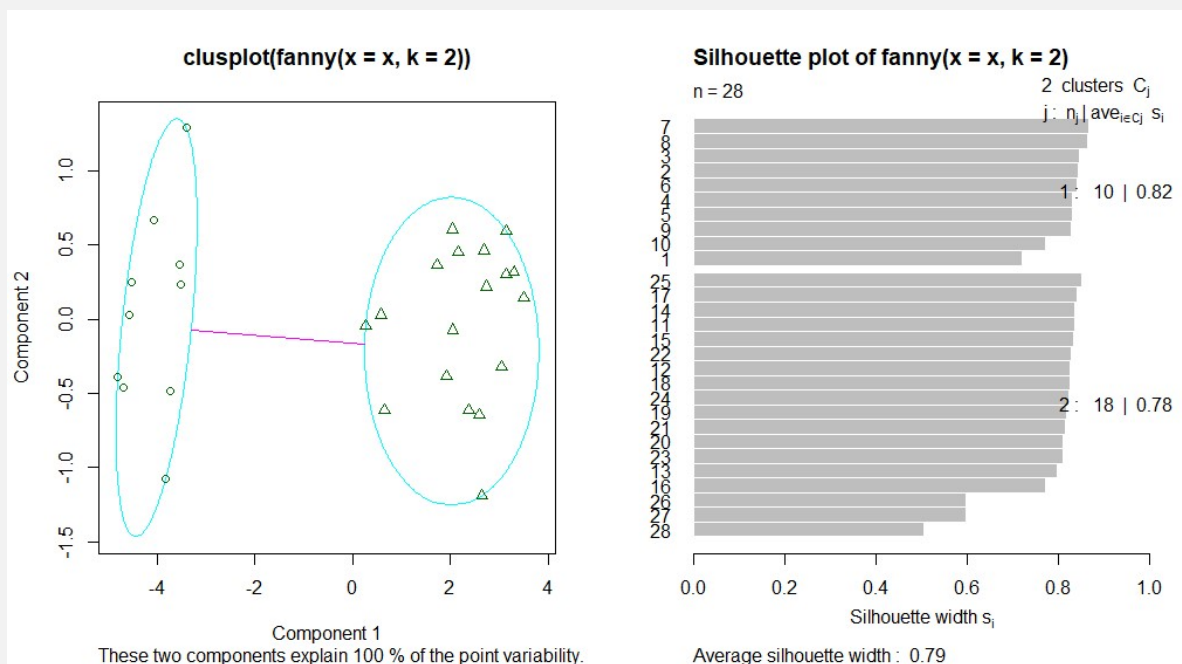
Pospeschill, M. (2021/22). *Ergänzungen zu R*. Saarbrücken: Universität des Saarlandes.

```
Min. 1st Qu. Median Mean 3rd Qu. Max.
0.1277 1.1562 2.7518 3.7124 6.4895 8.3248
```

Metric : euclidean

Number of objects : 28

...



Besonderheit ist die Ausgabe eines Silhouetten-Plots und entsprechender Koeffizienten. Damit wird für jedes Element und durchschnittlich für jeden Cluster angegeben, wie gut die Zuordnung eines Elements oder aller Elemente zu den beiden nächstgelegenen Clustern gelingt. Die Angaben stellen damit ein von der Anzahl der Cluster unabhängiges Gütemaß dar.

Die Koeffizienten lassen sich per Daumenregel wie folgt nach dem Grad der Strukturierung (Zuordnung eines Elementes zum Cluster  $x$ ) interpretieren: Werte zwischen 0,75 und 1 gelten als „starke Zuordnung“, Werte zwischen 0,5 und 0,75 als „mittlere Zuordnung“, Werte zwischen 0,25 und 0,5 als „schwache Zuordnung“ und Werte zwischen 0,25 und 0 als „ohne Zuordnung“ bzw. zwischen zwei Clustern liegend. Im Gegensatz zu den *Membership*-Koeffizienten können hier negative Silhouetten (Werte  $< 0$ ) auftauchen. In diesem Fall ist das Element zum nächstgelegenen Clustern näher, als zum aktuellen Cluster. Dies ist zumeist ein Hinweis auf eine verbesserungswürdige Lösung, da die Elemente offensichtlich nicht korrekt zugeordnet sind. Negative Werte können auch dafür sprechen, weitere Cluster für eine korrekte Abbildung in das Modell aufzunehmen.

Der Silhouetten-Plot zeigt sämtliche Silhouetten des Datensatzes. Dabei werden alle Elemente, die zu einem Cluster gehören als waagerechte Linie dargestellt. Je besser zwei Cluster durch die Daten getrennt sind, desto besser gelingt die Zuordnung der Datenpunkte.