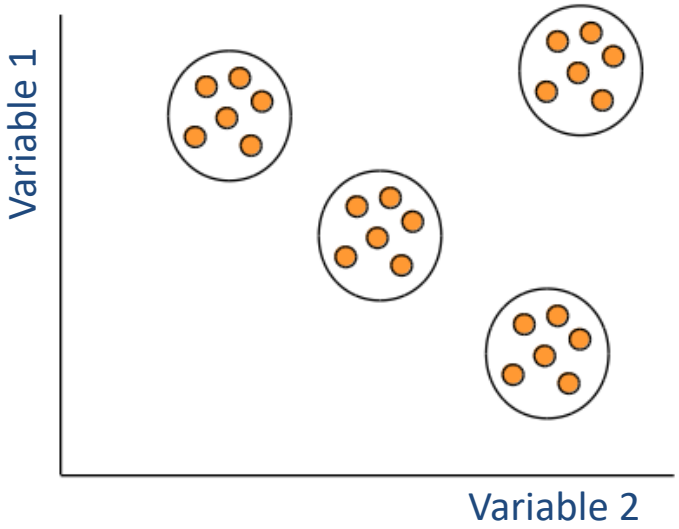
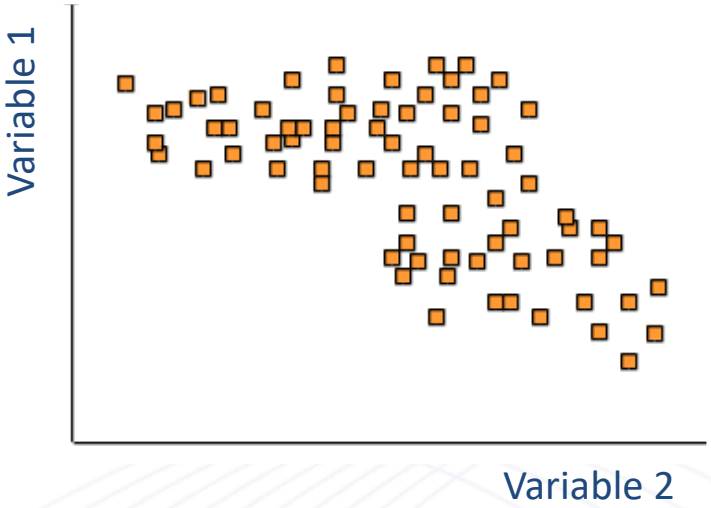
9. Methods for segmentation: Cluster analysis

Based on Backhaus, K.; Erichson, B.; Plinke, W.; Weiber, R.: Multivariate Analysis, Berlin. Some images may be under fair use guidelines (educational purposes).

9.1 Problem definition

Wish versus reality





Application areas for cluster analyses

- **Cluster analysis** is used to find homogeneous (sub)groups (clusters) in a large number of observations.
- Typical question: Which clusters are there within the data - and how do the clusters differ?
- The clusters should be formed in such a way that **different elements of a cluster** are as **similar** as possible, while **elements of different clusters** are as **different** as possible.

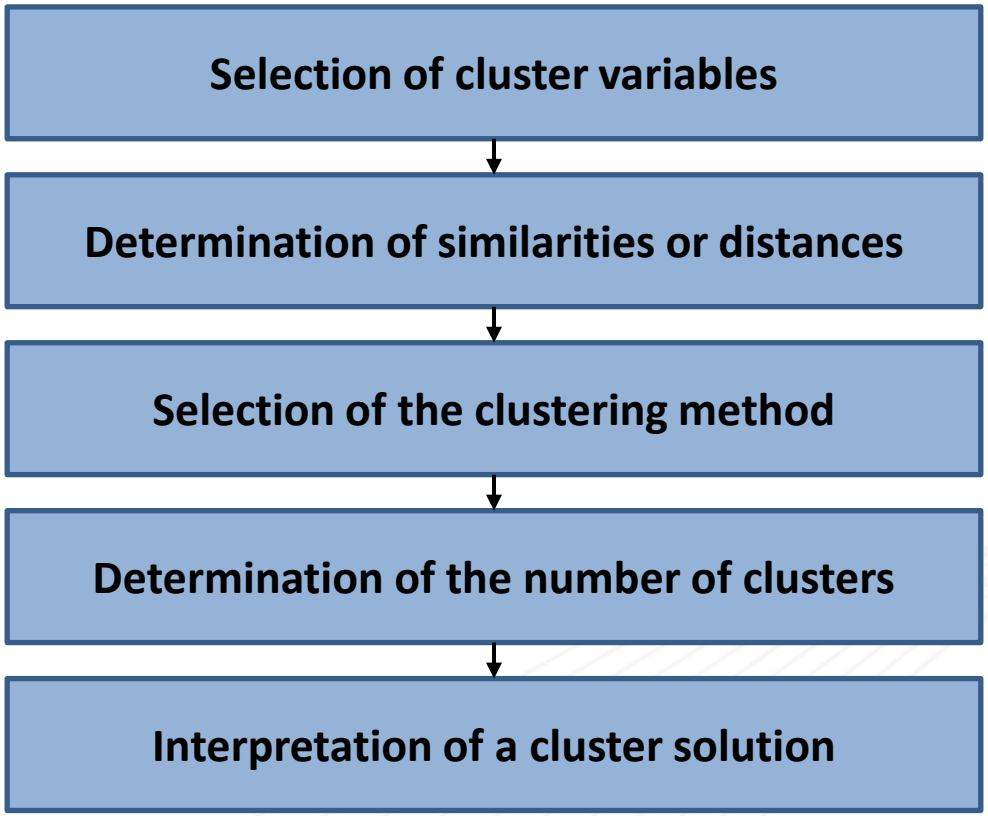
Application examples for cluster analyses

Discipline	Exemplary research questions
Agriculture	Which plants show a similar growth behavior and should therefore be cultured in a similar manner?
Biology	Are there unexplored genetic relationships between certain animal species?
Finance	Which creditworthiness levels can be distinguished based on the payment behavior of bank customers?
Marketing	How can an overall market be broken down into homogenous market segments on the basis of consumer behavior?
Medicine	How can patients be divided into different groups on the basis of laboratory values in order to develop more tailored therapies?
Meteorology	Can regions with similar climatic conditions be identified to develop an early warning system for each group of regions?
Pharmacy	Can drugs with similar side effects be identified in order to derive recommendations for the best possible therapy?

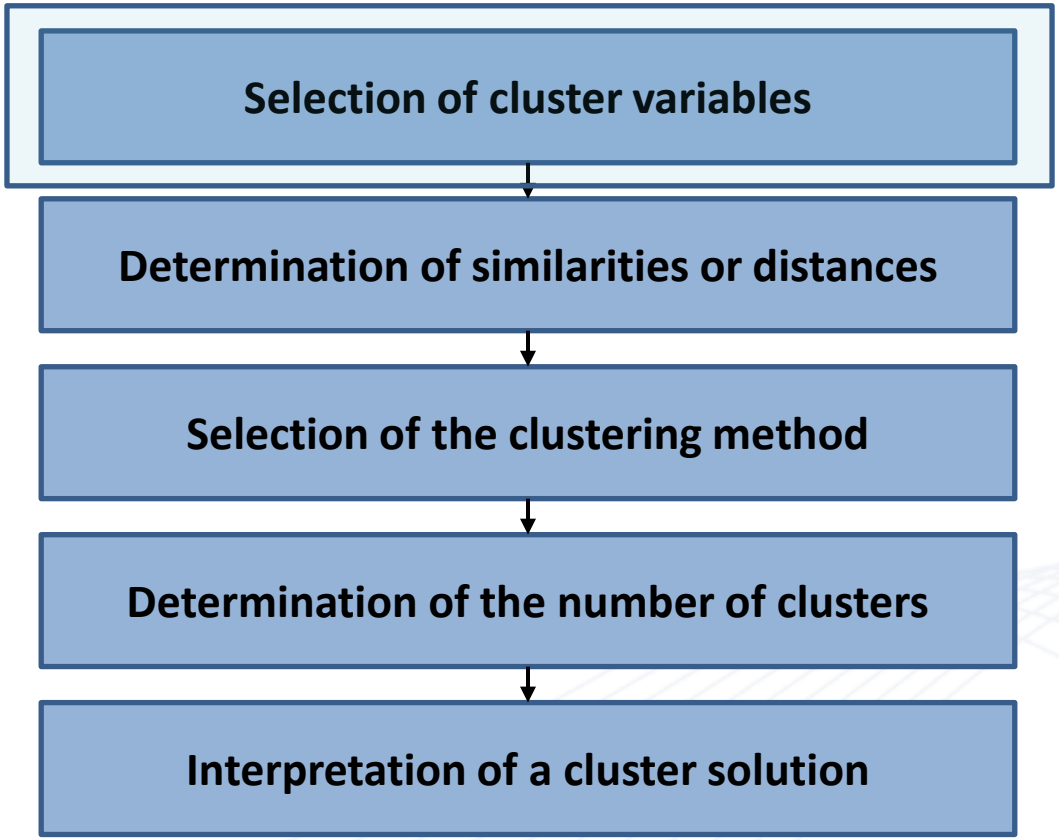
Backhaus et al. (2023), p. 455.

9.2 Carrying out a cluster analysis

Procedure


Selection of cluster variables
Determination of similarities or distances
Selection of the clustering method
Determination of the number of clusters
Interpretation of a cluster solution

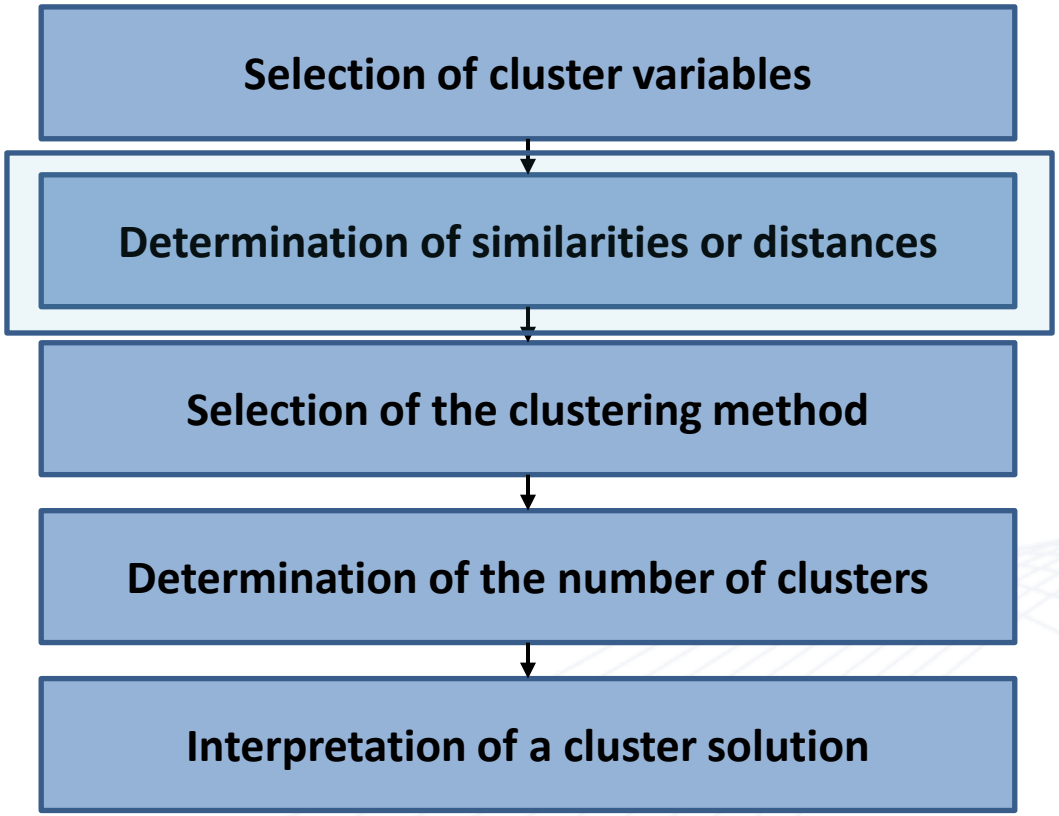
Procedure


Selection of cluster variables
Determination of similarities or distances
Selection of the clustering method
Determination of the number of clusters
Interpretation of a cluster solution

Selection of cluster variables

- The homogeneity of a cluster is defined by the variables that are used to form the cluster.
- The following properties should be fulfilled by cluster variables:
  - relevance for the grouping,
  - independence,
  - measurability,
  - comparability of the measurement dimensions,
  - controllable (may be influenced),
  - high separating power,
  - representativeness,
  - cluster stability.

Procedure


Selection of cluster variables
Determination of similarities or distances
Selection of the clustering method
Determination of the number of clusters
Interpretation of a cluster solution

Determination of similarities

- For all observations, it is determined pairwise how similar they are with regard to all cluster variables.
- For this purpose, the (n x k) data matrix is converted into an (n x n) similarity matrix:

	Var 1	Var 2	…	Var k
Obs. 1				
Obs. 2				
…				
Obs. n				

	Obs. 1	Obs. 2	…	Obs. n
Obs. 1				
Obs. 2				
…				
Obs. n				

- The (n x n) similarity matrix contains metrics that indicate how similar two observations are with respect to their characteristics in all k variables.
- Measures that quantify the similarity or distance between the observations are referred to as **proximity measures**.

Similarity and distance measures

There exist two types of measures:

- **Similarity measures** reflect the similarity between two observations: the larger the value of a similarity measure becomes, the more similar two observations are.
- **Distance measures** measure the dissimilarity between two observations: The greater the distance becomes, the more dissimilar two observations are. If two observations are identical, the distance is zero.

Similarity and distance measures are **not complementary**, i.e. the following does not always apply: the greater the similarity, the smaller the distance.

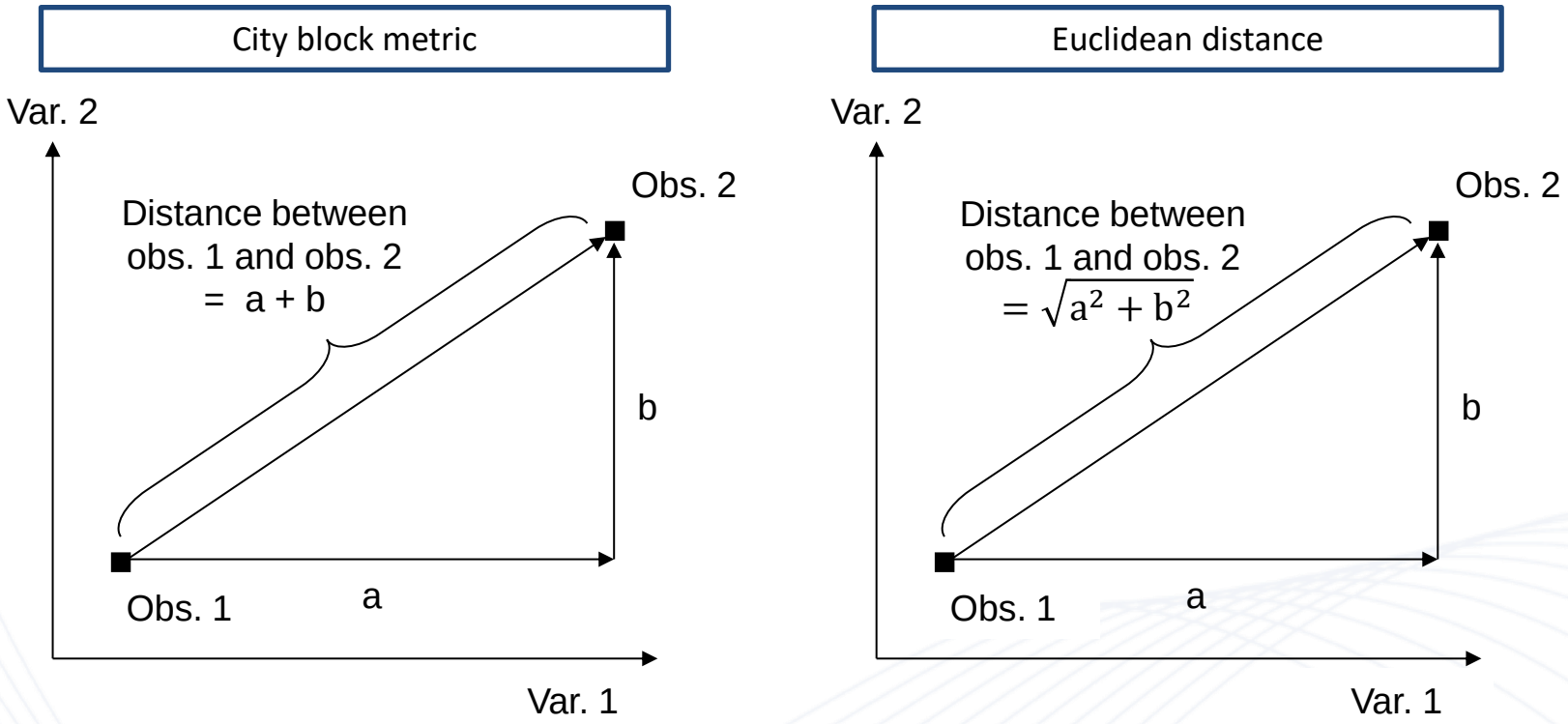
Depending on the scale level of the variable, different proximity measures are available.

Selected proximity measures

	Scale level of the attributes		
	Metric data (interval)	Metric data (interval)	Count data (frequency)
Similarity measures	• Cosine • Pearson correlation	• Simple matching (M-coefficient) • Phi 4-point correlation • Lambda (Goodman & Kruskal) • Dice • Jaccard • Rogers and Tanimoto • Russel and Rao	
Distance measures	• Euclidean distance • Squared Euclidean distance • Chebychev • City block metric • Minkowski	• Euclidean distance • Squared Euclidean distance • Size difference • Pattern difference • Variance • Dispersion • Lance and Williams	• Chi-square measure • Phi-square measure

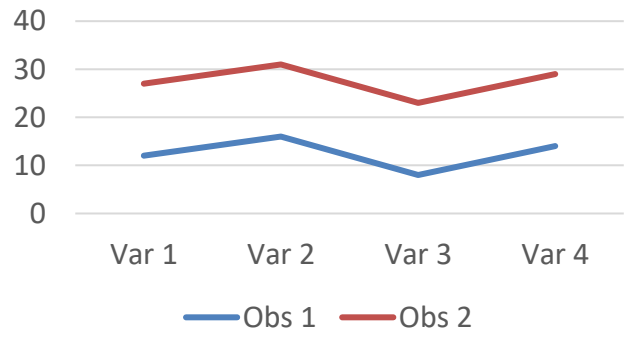
Backhaus et al. (2023), p. 462.

City block metric and Euclidean distance



Difference between similarity and distance measures

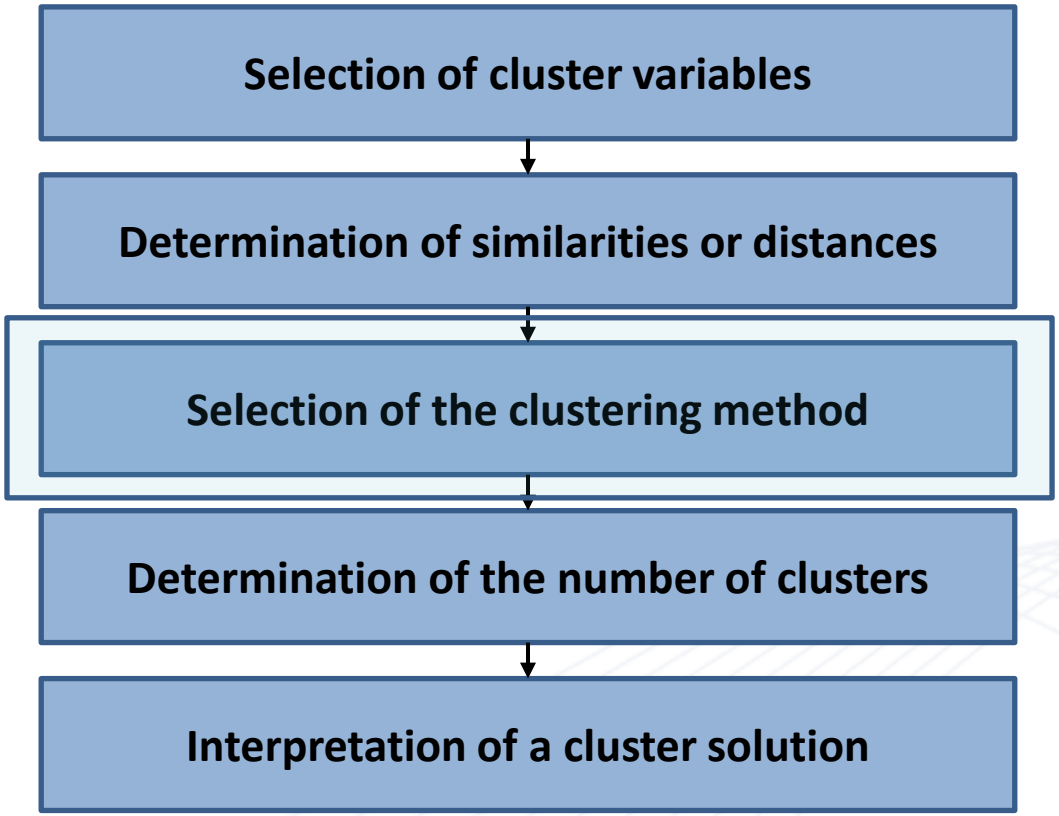
	Var 1	Var 2	Var 3	Var 4
Obs. 1	12	16	8	14
Obs. 2	27	31	23	29



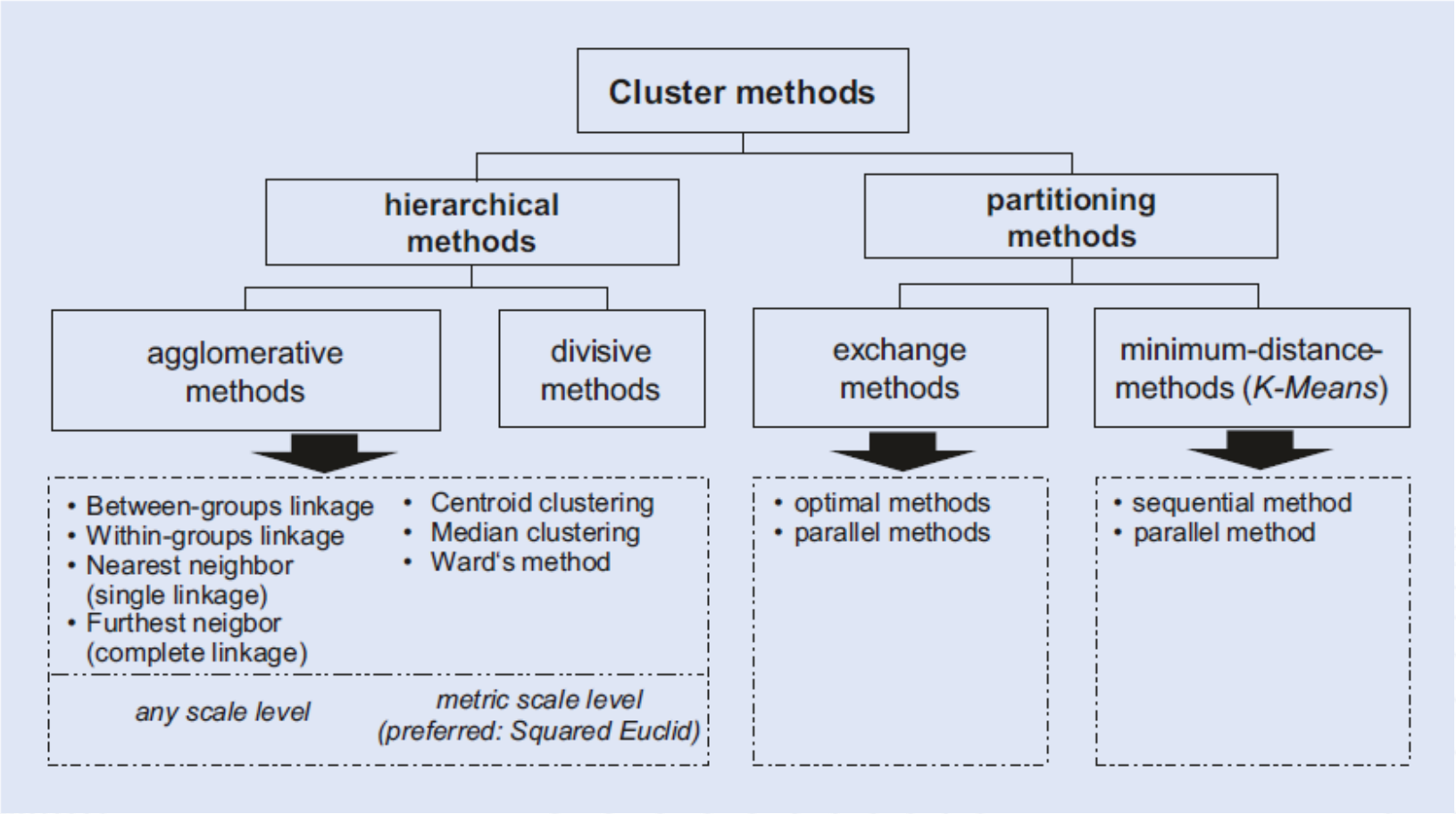
Observation 1 and observation 2 show both a high degree of similarity and a great distance.

- **Distance measures** Distance measures consider the absolute distance between objects, and the dissimilarity is greater if two objects are further away from each other according to the considered variables.
- **Similarity measures** based on correlation values consider how similar the profiles of two objects are, regardless of the specific values of the objects according to the considered variables.

Procedure


Selection of cluster variables
Determination of similarities or distances
Selection of the clustering method
Determination of the number of clusters
Interpretation of a cluster solution

Selected clustering methods



Hierarchical agglomerative methods are the most relevant.

Backhaus et al. (2023), p. 470.

Hierarchical agglomerative methods

Start
- Every observation represents a cluster (finest partition), i.e. for n observations, there are n different clusters.
- Selection of the proximity measure.

Schritt 1 Calculation of pairwise distances or similarities between all clusters.

Schritt 2 Combine the two cluster with the smallest distance or the biggest similarity and form a new cluster. The total number of clusters is reduced by 1.

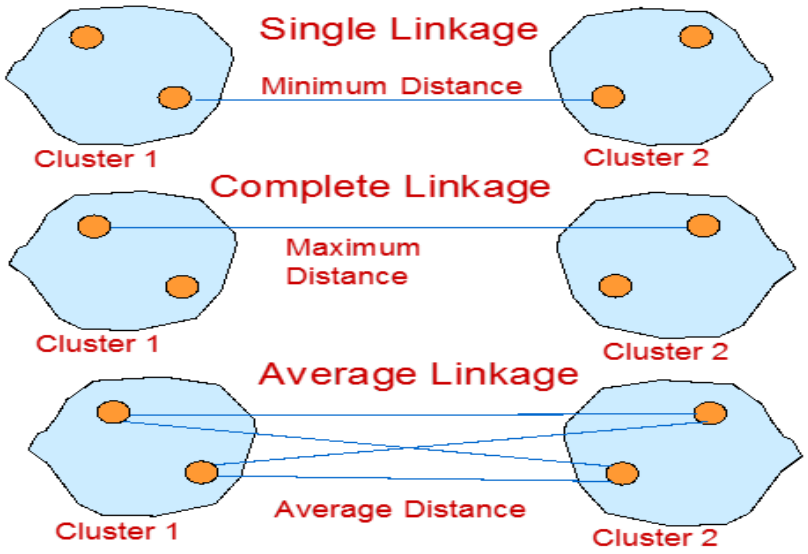
Schritt 3
- Selection of a cluster method.
- Calculation of pairwise distances or similarities between the new cluster and all other remaining clusters. This results in a reduced distance matrix or similarity matrix.

Repeat step 2 and step 3, until there is only 1 cluster left. With n total initial objects, this results in n-1 fusion steps.

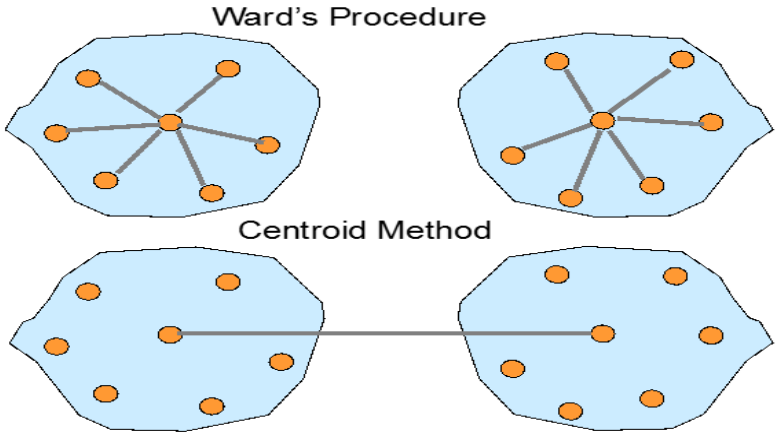
Cluster methods

If a cluster consists of several observations, it must be determined where the starting point for measuring the distance to other clusters should be set.

Feasible for all proximity measures

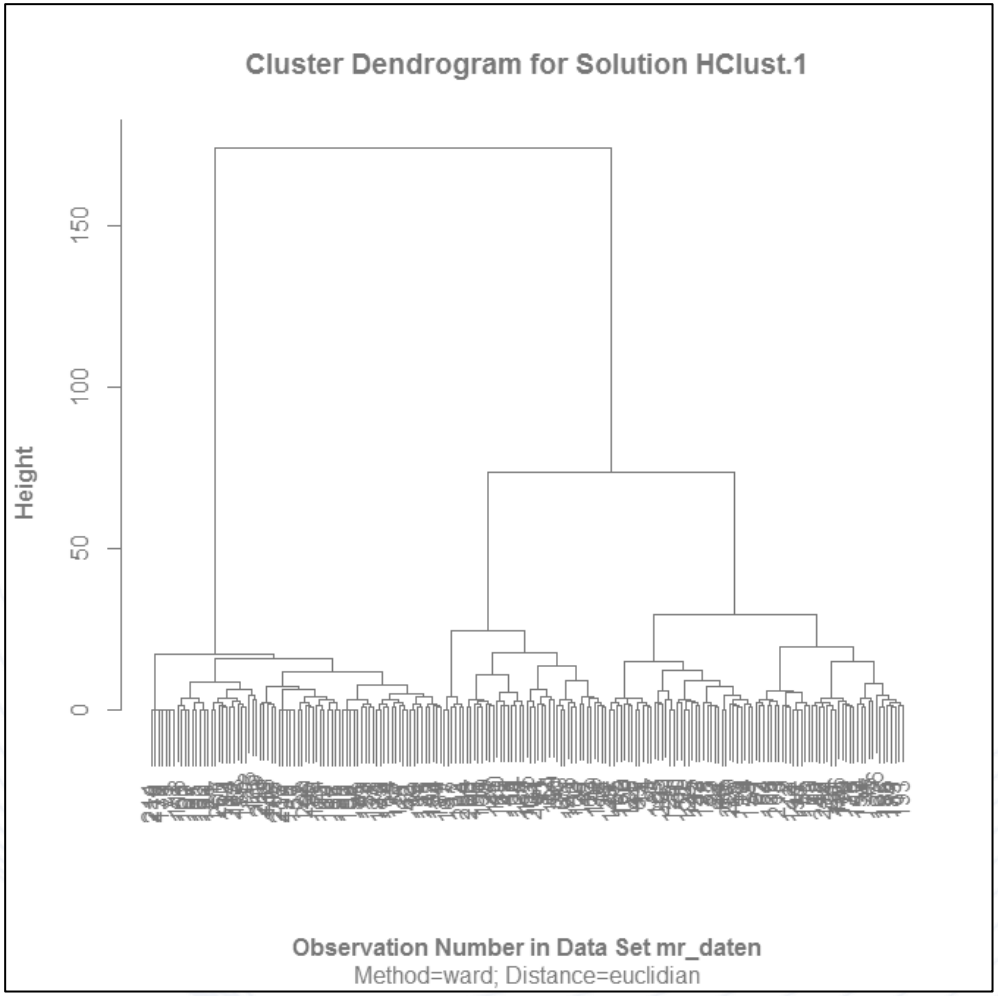


Feasible only for distance measures



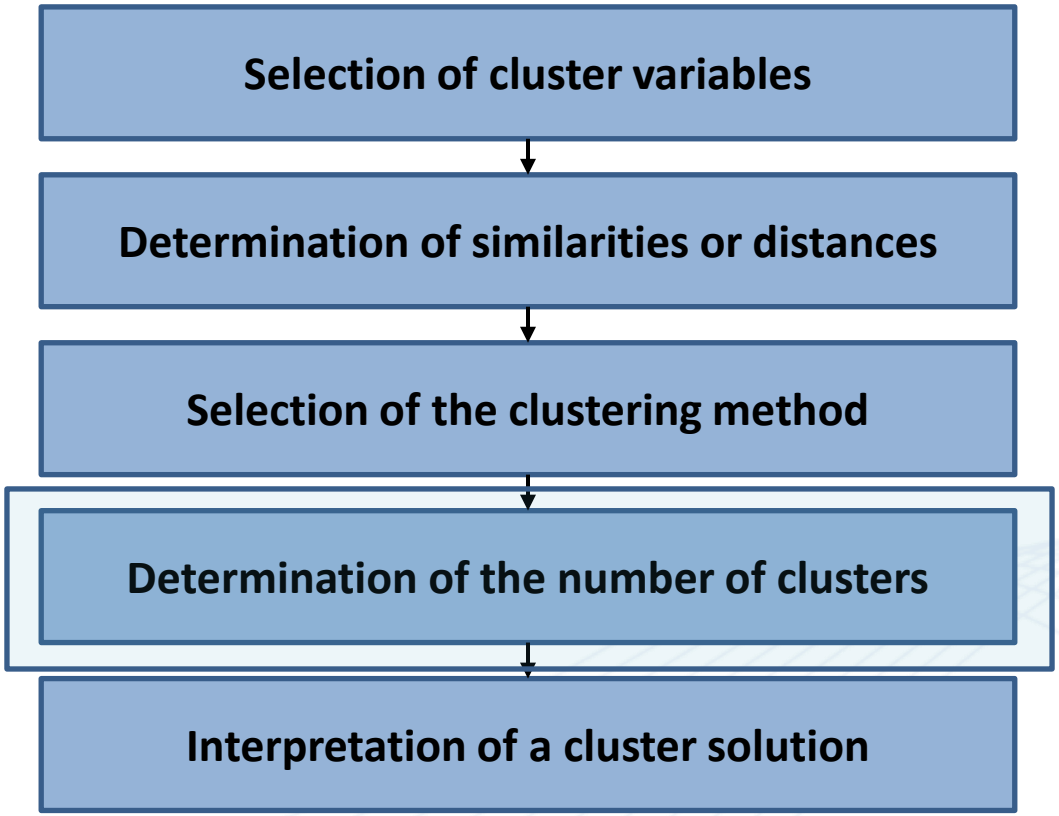
The Ward method summarizes those clusters with the smallest increase of the variance (error sum of squares) within the clusters.

Graphical representation of the fusion process - Dendrograms



- **Dendrograms** vizualise the distance between the two clusters that are merged in the respective step.
- If only slight increases in these distances are observed, the choice to fewer clusters is justifiable.
- If the increase is significant, it is appriopriate to stop the fusion process.

Procedure

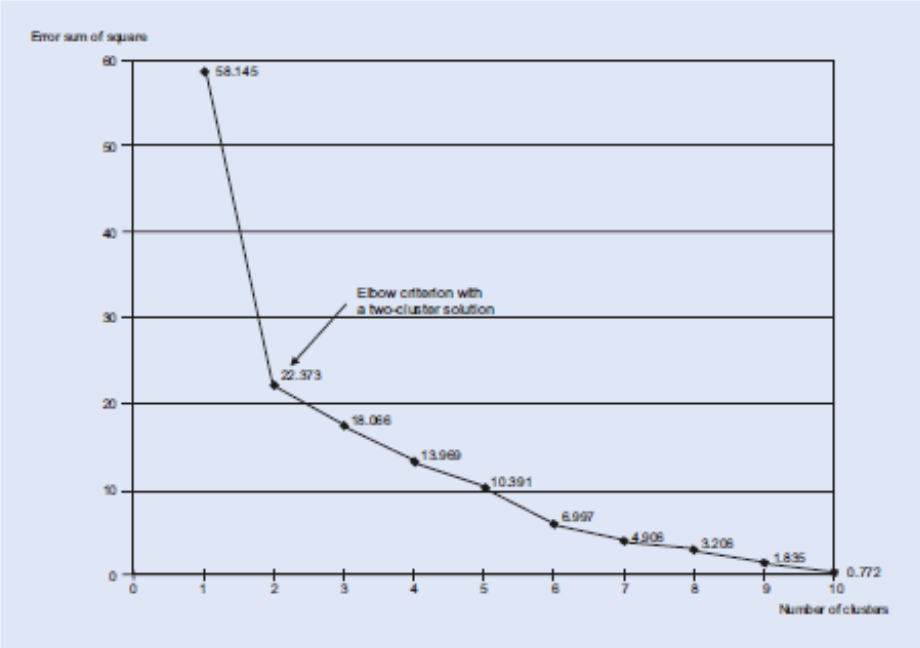

Selection of cluster variables
Determination of similarities or distances
Selection of the clustering method
Determination of the number of clusters
Interpretation of a cluster solution

Approaches for determining the number of clusters

- When deciding on the number of clusters, there is a conflict of objectives between
  - the homogeneity requirement for the cluster solution and
  - the manageability of the cluster solution.
- The decision on the number of clusters should primarily be based on statistical key figures, but content-related and logical considerations can also be used.
- Possible criteria are
  - the scree plot and elbow criterion,
  - the Calinski/Harabasz rule and
  - the Mojena test.
- The cluster solution found can then be optimized with a K-Means cluster analysis.

Scree plot and elbow criterion

- In the scree plot, the number of clusters is plotted on the x-axis and the value for the change in the respective proximity measure when the two most similar clusters are merged is plotted on the y-axis.
- Example: Scree plot using the error sum of squares as proximity measure.
- If there is an “elbow” in the curve in the scree plot, then the merging of the two corresponding clusters led to a particularly high increase in the proximity measure.



- The optimum cluster number according to the elbow criterion is then a two-cluster-solution.

Backhaus et al. (2023), p. 502.

The Calinski-Harabasz index

- One disadvantage of the elbow criterion is that the decision for the number of clusters is very subjective.
- The Calinski-Harabasz index, on the other hand, is based on a statistical decision criterion in which the variance between the groups (SS_b) is set in relation to the variance within the groups (SS_w):

$$CHI_k = \frac{\frac{SS_b}{k-1}}{\frac{SS_w}{n-k}}.$$

with:

k	number of clusters
n	number of observations (objects)
CHI_k	Calinski-Harabasz index for k clusters

- This index is calculated for every possible number of clusters.
- The optimum number of clusters is then the number at which the specified index takes on its maximum value, i.e. CHI → max.

Caliński/Harabasz (1974), p. 10.

Procedure from Mojena

- Let α be the fusion coefficient, i.e. the value for the change in the respective proximity measure when the two most similar clusters are merged.
- Now calculate the standardized fusion coefficients $\tilde{\alpha}_i$:

$$\overline{\alpha} = \frac{1}{n-1}\sum_{i=1}^{n-1}\alpha_i \quad \ldots s_\alpha = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n-1}(\alpha_i - \overline{\alpha})^2} \quad \ldots \quad \tilde{\alpha}_i = \frac{\alpha_i - \overline{\alpha}}{s_\alpha}$$

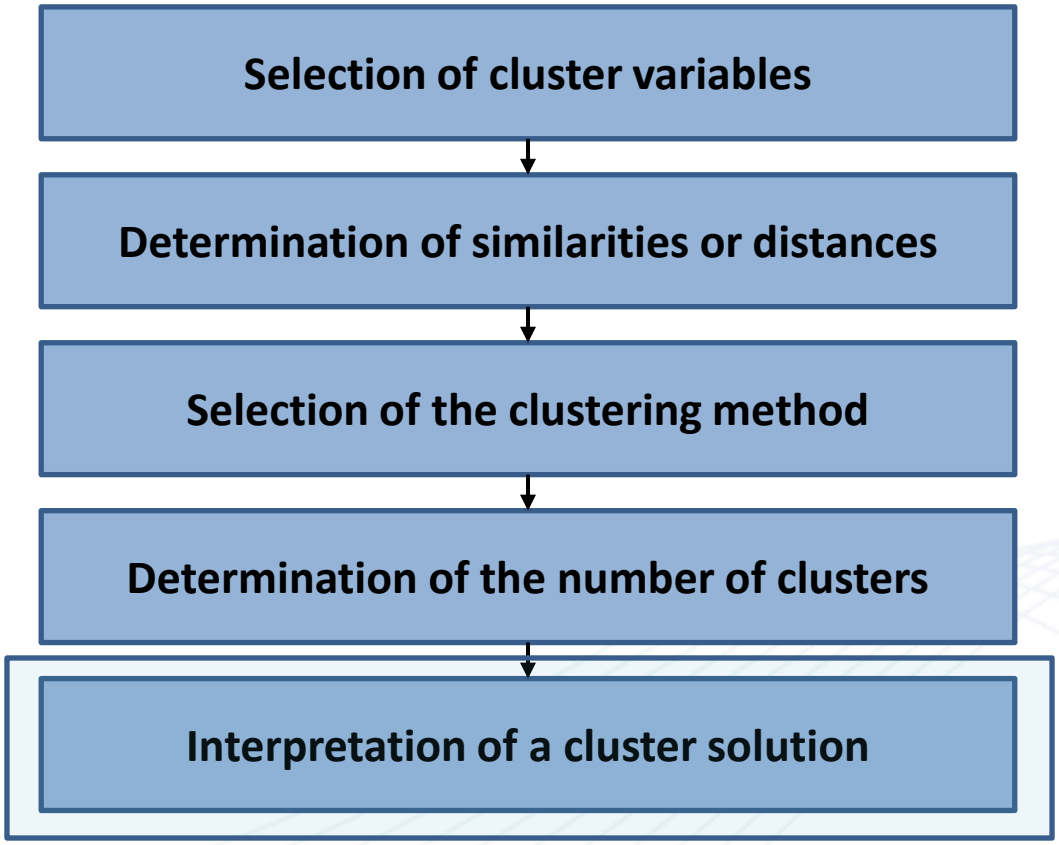
- A indicator for a good cluster solution is the largest cluster number at which a specified value of the standardized fusion coefficient is exceeded for the first time.
- There are various recommendations in the literature for the specified value, ranging from 1.25 to 2.75.

Mojena (1977), p. 359.

Optimization of a cluster solution with k-means

- Once the optimum number of clusters has been found, the assignment of the individual observations in the data set to the individual clusters can be optimized using a k-means cluster analysis.
- The k-means cluster analysis follows the following steps:
  1. Random selection of k cluster centers from the n observations and assignment of the observations to the next cluster center.
  2. Calculation of the cluster center of the assigned observations (e.g. via mean value).
  3. Assignment of the observations to the next cluster center.
  4. If the assignment has changed, continue with step 2, otherwise end.

Procedure


Selection of cluster variables
Determination of similarities or distances
Selection of the clustering method
Determination of the number of clusters
Interpretation of a cluster solution

Description of the clusters

- The interpretation of a cluster solution should be based on the characteristics of the cluster variables in the identified clusters.
- It is useful to compare the individual clusters with the entire sample.
- A descriptive approach can be taken by determining frequencies and distribution parameters of the variables of interest for the individual clusters.
- It is also helpful to calculate t values and F values.
- A discriminant analysis can provide information on which characteristics are decisive for cluster formation (not part of this lecture).

Determination of t values

- Analogous to a test for differences in means, t values can be calculated for each variable j in each cluster g as follows:

$$t_{gj} = \frac{\bar{x}_{gj} - \bar{x}_j}{s_j}$$

- Where:
  - $\bar{x}_{gj}$ mean of variable j in cluster g (g = 1, …, G).
  - $\bar{x}_j$ mean of variable j in the entire sample (j = 1, …, J).
  - s_j standard deviation of variable j in the entire sample.
- The t values represent standardized values.
  - Negative t values indicate that a variable is underrepresented in the group under consideration compared to the entire sample.
  - Positive t values indicate that a variable is overrepresented in the group under consideration compared to the entire sample.
- Caution: t values say nothing about the quality of a cluster solution.

Determination of F values

- To assess the homogeneity of a cluster compared to the entire sample, the F value for each variable j in each cluster g can be calculated as follows, analogous to the F test for each group:

$$F_{gj} = \frac{s_{gj}^2}{s_j^2}$$

- Where
  - s_{gj}^2 variance of variable j in cluster g (g = 1, …, G).
  - s_j^2 variance of variable j in the entire sample (j = 1, …, J).
- The smaller the F value, the smaller the variance of this variable in a group compared to the entire sample.
- The F value should not exceed 1, as in this case the corresponding variable in the group has a greater spread than in the entire sample.

Procedure of a hierarchical agglomerative cluster analysis

The considerations so far lead to the following four steps for carrying out a hierarchical agglomerative cluster analysis:

1. Application of the single linkage method (nearest neighbor) to identify outliers.
2. Elimination of outliers and subsequent application of a further agglomerative method (e.g. Ward) to the reduced data set. The agglomerative method must be selected against the background of the respective application situation and the fusion properties of the clustering method.
3. Optimization of the cluster solution found in step 2 using a k-means cluster analysis.
4. Assessment of the robustness of a cluster analysis and interpretation of the content of the results.

9.3 Cluster analysis using 

Required packages and example data set

R Script

```
# Required packages
library(fpc) # calinhara()
library(cluster) # clusplot()
library(mosaic) # favstats()
library(psych) # describe()
```

The sample data set contains 11 observations (objects) that are to be clustered with regard to 10 variables (properties).

R Script

```
# Load and describe sample data set
load("BackhausClusterDaten.RData")
mydatc8_case$Flavor # objects
colnames(mydatc8_case)[-1] # properties
```

```
> mydatc8_case$Flavor
 [1] "Milk"       "Espresso"   "Biscuit"    "Orange"     "Strawberry" "Mango"
 [7] "Cappuccino" "Mousse"     "Caramel"    "Nougat"     "Nut"
> colnames(mydatc8_case)[-1]
 [1] "Price"      "Refreshing" "Delicious"  "Healthy"    "Bitter"     "Light"
 [7] "Crunchy"    "Exotic"     "Sweet"      "Fruity"
>
```

Calculation of the proximity measure

There are various functions for calculating the proximity measure. The standard proximity measure is the Euclidean or quadratic Euclidean distance. The R command ?dist() can be used to display all distance measures that can be calculated with the dist() function. For further proximity measures, e.g. the daisy() function from the cluster package can be used.

R Script

```
# Proximity measure: Squared Euclidean distances matrix of the eleven
# chocolate types
distdata <- dist(mydatc8_case[,-1], method = "euclidean")^2
distdata
```

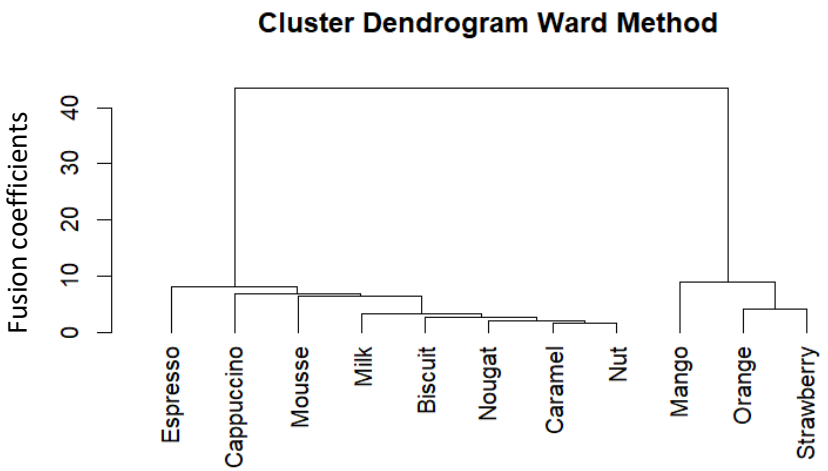
The result is a lower triangular matrix containing the pairwise distances for all 11 observations (objects).

Hierarchical cluster analysis using the Ward method

A hierarchical cluster analysis can be carried out for the matrix with the proximity measures using the hclust() function and the result can be displayed graphically as a dendrogram using the plot() function

R Script

```
hc_w <- hclust(distdata, method = "ward.D2")
hc_w$labels <- mydatc8_case$Flavor
plot(as.dendrogram(hc_w),
        ylab = "Fusion coefficients",
        main = "Cluster Dendrogram Ward Method")
```



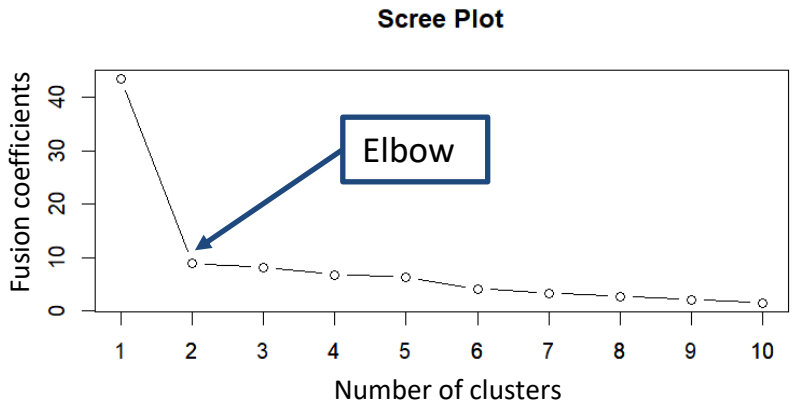
The dendrogram suggests a 2-cluster solution.

Number of clusters – Scree plot

R Script

```
n <- nrow(clusdata)-1
nclust <- rev(seq_along(n:1))
step <- seq(1:n)
clust_steps_data <- data.frame(hc_w[2:1], nclust, step)
tail(clust_steps_data,10)

# Draw the scree plot
plot(clust_steps_data$nclust,clust_steps_data$height, type = "b",
        main = "Scree Plot",
        xlab = "Number of clusters",
        ylab = "Fusion coefficients")
axis(side = 1, at = seq(1, 10, 1))
```

The scree plot results in a 2-cluster solution



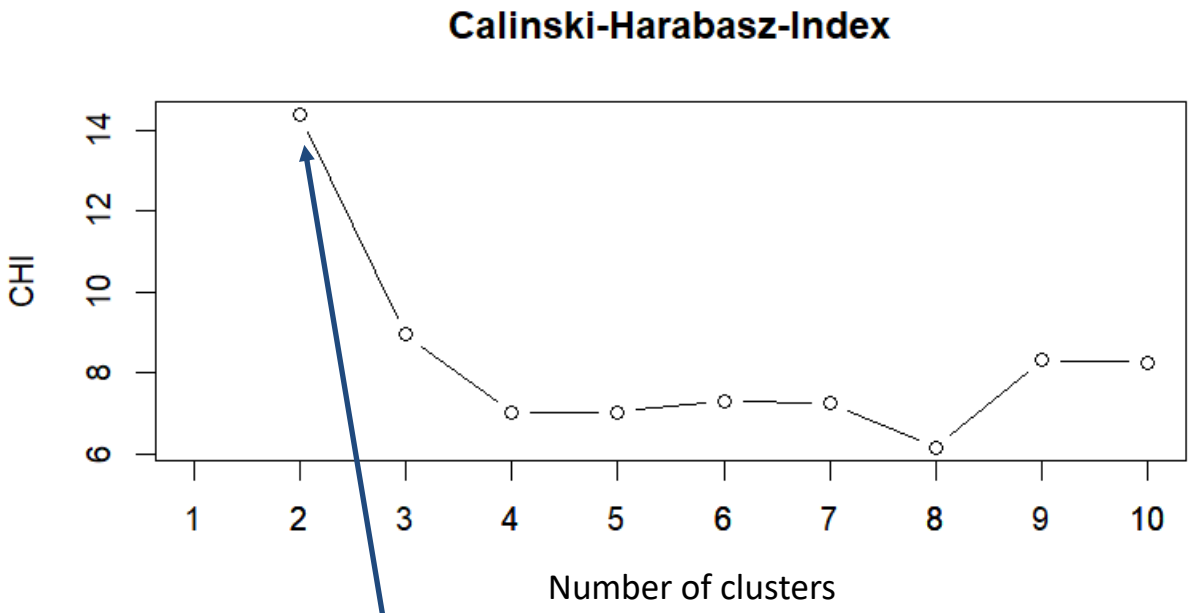
Number of clusters – Calinski-Harabasz index (CHI)

The CHI must be calculated for each number of clusters. As operational applications rarely require more than 10 clusters, the number of clusters is limited to 10. The sums of squares required for the calculation are calculated using a k-means cluster analysis.

R Script

```
nclus <- min(10, nrow(mydatc8_case)-1)
k <- list()
for(i in 1:nclus){k[[i]] <- kmeans(mydatc8_case[,-1], centers=i)}

# Calculate Calinski-Harabasz index for each number of clusters.
calinhara <- list()
for(i in 1:nclus){calinhara[[i]] <-
                calinhara(mydatc8_case[,-1],k[[i]]$cluster)}

# Graphical display of the Calinski-Harabasz indexes
plot(1:nclus, calinhara, type = "b",
       main = "Calinski-Harabasz index",
       xlab = "Number of clusters (k)", ylab = "CHI")
axis(side = 1, at = seq(1, 10, 1))
# --> Optimum number of clusters is 2
```

Number of clusters – Calinski-Harabasz index (CHI)



The CHI takes on its highest value with 2 clusters, so the optimum number of clusters is 2.

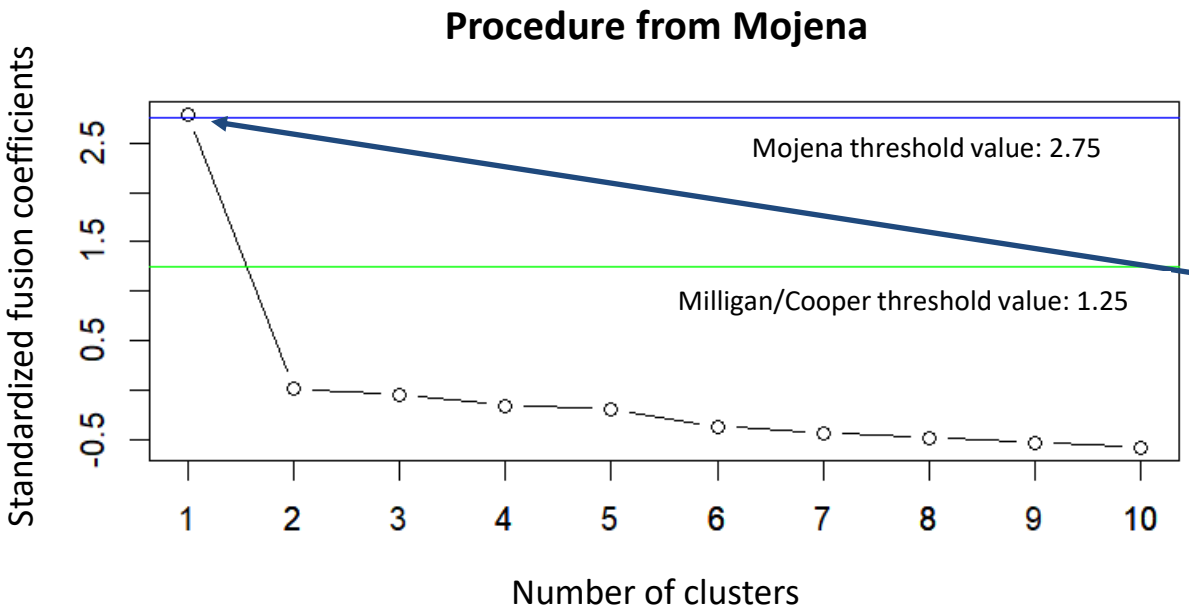
Number of clusters – Procedure from Mojena

The resulting object of the hclust() function contains the fusion coefficients in the variable height: hc_w$height. Standardization is performed with the scale() function.

R Script

```
# Graphical representation of the standardized fusion coefficients and
# the procedure from Mojena
# Calculate standardized fusion coefficients
std_height <- scale(hc_w$height)
# Take only the last nclus standardized fusion coefficients
std_height <- tail(std_height,nclus)

plot(nclus:1, std_height, type = "b",
        main = "Procedure from Mojena",
        xlab = "Number of clusters",
        ylab = "Standardized fusion coefficients")
axis(side = 1, at = seq(1, 10, 1))
abline(h=1.25, col="green")
text(x=8, y=1.05, cex=.8, "Milligan/Cooper threshold value: 1.25")
abline(h=2.75, col="blue")
text(x=8, y=2.55, cex=.8, "Mojena threshold value: 2.75")
```

Number of clusters – Procedure from Mojena



Since only one standardized fusion coefficient is greater than the two limit values, the optimal cluster number is 1.

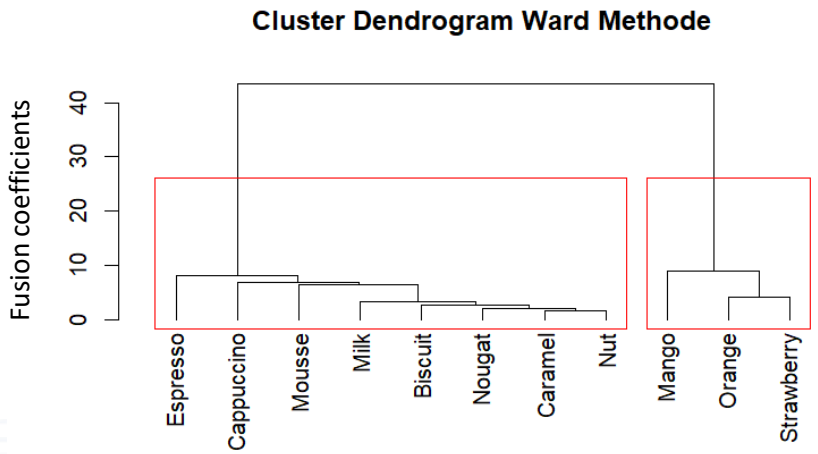
Taken everything together (Scree plot, Calinski-Harabasz index and procedure from Mojena) yields that a 2-cluster solution is best.

Display cluster allocation in the dendrogram

The rect.hclust() function can be used to color-code the allocation of objects to clusters in the dendrogram.

R Script

```
# Display cluster allocation in the dendrogram
hc_w <- hclust(distdata, method = "ward.D2")
hc_w$labels <- mydatc8_case$Flavor
plot(as.dendrogram(hc_w),
        main = "Cluster Dendrogram Ward Methode",
        ylab = "Fusion coefficients")
rect.hclust(hc_w, k = 2, border = "red")
```



Optimization of the allocation of objects to the clusters

The centers= option in the kmeans() function denotes the number of clusters. It is useful for further analyses to add the allocation of the individual objects to the clusters (variable cluster in the result object of kmeans()) to the original data set.

R Script

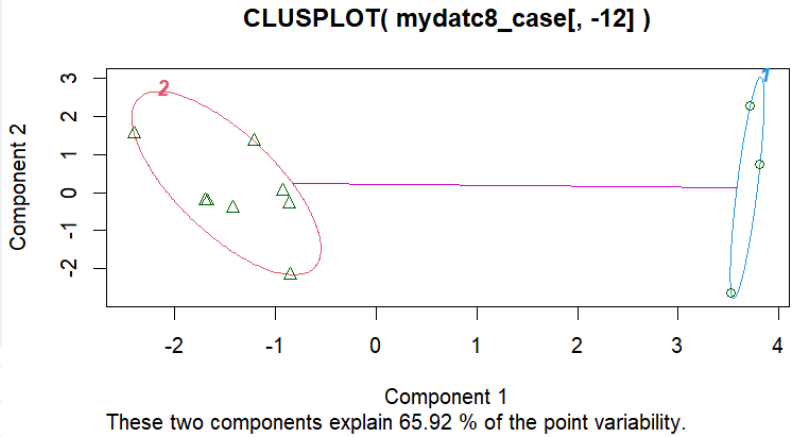
```
# Application of the k-means cluster analysis for 2 clusters
fit_kmeans <- kmeans(mydatc8_case[,-1],centers = 2)

# Results of the k-means cluster analysis
summary(fit_kmeans)
fit_kmeans
fit_kmeans$centers

# Add objects to cluster allocation to the data set
mydatc8_case <- data.frame(mydatc8_case,fit_kmeans$cluster)
colnames(mydatc8_case)[ncol(mydatc8_case)] <- "Cluster"

# Display objects allocated to the two clusters
mydatc8_case[mydatc8_case$Cluster == 1,1]
mydatc8_case[mydatc8_case$Cluster == 2,1]
```

Descriptive statistics for and graphical display of the 2 clusters

For the descriptive statistics, well-known functions such as describe() from the psych package or favstats() from the mosaic package can be used. The cluster solution can also be displayed graphically using the clusplot() function from the cluster package.

R Script

```
# Descriptive statistics for the 2 clusters
describe(mydatc8_case[mydatc8_case$Cluster == 1,])
describe(mydatc8_case[mydatc8_case$Cluster == 2,])

favstats(~Price | Cluster, data=mydatc8_case)

# Graphical representation of the
# 2 clusters in a biplot
clusplot(mydatc8_case[,-12],
         fit_kmeans$cluster,
         color=TRUE,
         shade=FALSE,
         labels=4)
```



Interpretation of the cluster solution

With the help of t values, the properties of each cluster can be set in relation to the entire data set. F values provide information on the homogeneity of the clusters.

R Script

```
# Calculation of t and F values from mean values and standard
# deviations
clusres <- data.frame(colnames(mydatc8_case)[-c(1,12)],
        describe(mydatc8_case[,2:11])$mean,
        describe(mydatc8_case[,2:11])$sd,
        describe(mydatc8_case[mydatc8_case$Cluster == 1,2:11])$mean,
        describe(mydatc8_case[mydatc8_case$Cluster == 1,2:11])$sd,
        describe(mydatc8_case[mydatc8_case$Cluster == 2,2:11])$mean,
        describe(mydatc8_case[mydatc8_case$Cluster == 2,2:11])$sd)
colnames(clusres) <- c("characteristic","MeanTotal","SdTotal",
        "MeanClus1","SdClus1","MeanClus2","SdClus2")

# Add t and F values
clusres$tClus1 <- (clusres$MeanClus1-clusres$MeanTotal)/clusres$SdTotal
clusres$tClus2 <- (clusres$MeanClus2-clusres$MeanTotal)/clusres$SdTotal
clusres$FClus1 <- (clusres$SdClus1 / clusres$SdTotal)**2
clusres$FClus2 <- (clusres$SdClus2 / clusres$SdTotal)**2
```

Interpretation of the cluster solution

R Script

```
# Print t and F values
clusres
```

Interpretation:

T < 0	The corresponding property is below average in this cluster.
T > 0	The corresponding property is above average in this cluster.
F < 1	The heterogeneity (dispersion of individual values) in this cluster is lower than in the entire data set.
F > 1	The heterogeneity (dispersion of individual values) in this cluster is greater than in the entire data set.

Interpretation of the cluster solution

```
> clusres[,-c(2:7)]
   characteristic     tClus1     tClus2    FClus1     FClus2
1           Price  0.5144286 -1.3718096 0.3031869 0.05750611
2      Refreshing -0.3068630  0.8183013 0.2809712 2.63551583
3       Delicious  0.3706978 -0.9885273 0.8487954 0.01376937
4         Healthy  0.2738572 -0.7302858 0.1637163 3.32702590
5          Bitter -0.2898467  0.7729246 0.9292713 0.51538727
6           Light -0.4516895  1.2045054 0.5432088 0.10642577
7         Crunchy -0.4564086  1.2170897 0.5232009 0.11360069
8          Exotic -0.5212711  1.3900562 0.2008149 0.31186970
9           Sweet -0.1840473  0.4907929 1.0820114 0.71614983
10          Fruity -0.5269019  1.4050719 0.1558665 0.38262435
```

Above-average price of objects in the first cluster, below-average price of objects in the second cluster.

Disproportionately high heterogeneity of the sweetness of the objects in the first cluster, disproportionately low heterogeneity of the sweetness of the objects in the second cluster.

Exercise

Exercise the cluster analysis.

10. Text mining: A short overview

Based on Manderscheid, K. (2022), Text Mining, in: Baur, N.; Blasius, J. (Hrsg.), Handbuch Methoden der empirischen Sozialforschung, 3. Auflage, Springer, Wiesbaden, S. 1719-1732. Some images may be under fair use guidelines (educational purposes).

10.1 Introduction

Text mining and differentiation from qualitative content analysis

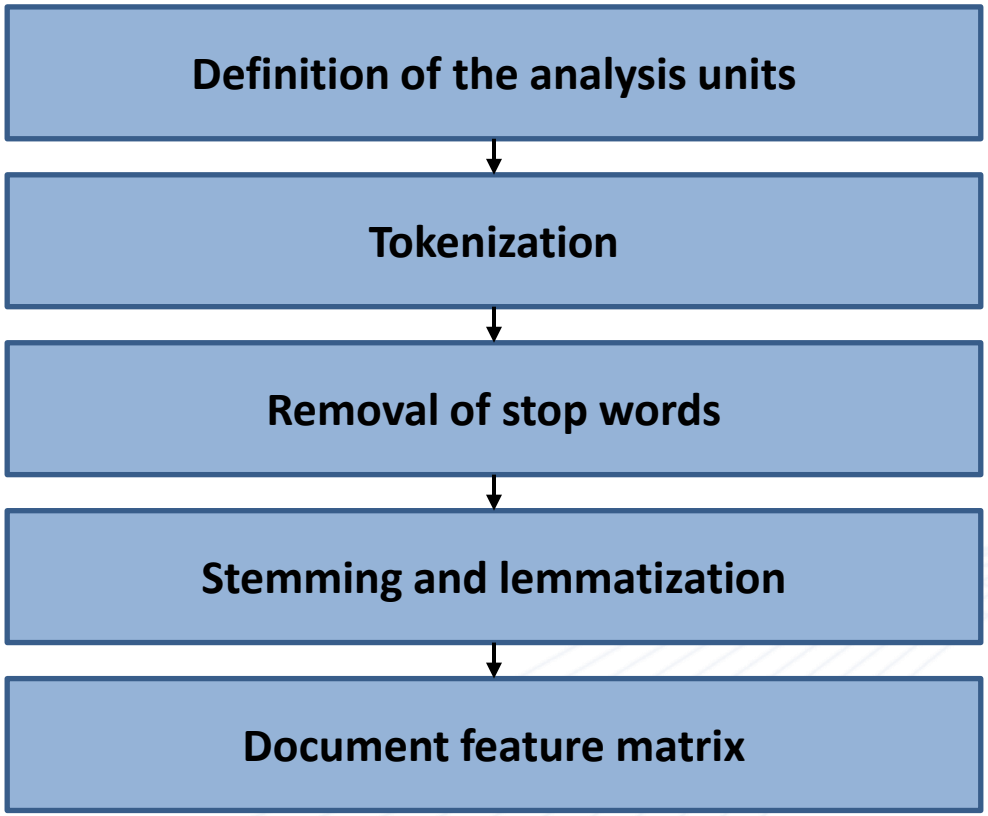
- The term “**text mining**” refers to computer-aided and algorithm-based evaluation methods for extracting information from large digital text collections.
- The evaluation methods of text mining include statistical, lexical and linguistic methods.
- Text mining is becoming increasingly important as more and more texts of all kinds (e.g. books, magazines and other documents, but also websites and social media posts) are available in digitized form.
- While the first applications of text mining were mostly limited to counting word occurrences, current evaluations are aimed at identifying argumentation structures and contexts of meaning.
- Text mining aims to discover structures in the text data and not to work out individual pieces of information, types or specific logics of argumentation on the basis of a small number of texts, as is the case with qualititative content analysis.

Special features of texts as a data source

- Texts are usually always available as secondary data, which means that data generation is not controlled by the researcher.
- This makes it necessary to deal with the context of the text productions.
- Before the actual data analysis, texts must first be converted into a format suitable for data analysis.
- The first step is always to define the **text data corpus**, i.e. the text material to be analyzed, on the basis of the following criteria:
  - Narrowing down the text sources: Books, newspapers, social media platforms, etc.
  - Temporal limitation: Time period for the publication date.
  - Spatial limitation: Geographical publication locations or language used.
  - Subject limitation: Content selection of the text data.
- The nature of this selection in turn determines the informative value and transferability of the results to research.

10.2 Preparation of text data

Procedure


Definition of the analysis units
Tokenization
Removal of stop words
Stemming and lemmatization
Document feature matrix

Definition of the analysis units

- The **analysis units** must be defined on the basis of the research question.
- Units of analysis can be: Newspaper articles, book chapters, paragraphs, individual sentences, Twitter threads, statements by individuals.
- The text material is then split into these units. Together they form the **text data corpus** (analogous to cases in a classic data matrix).
- In addition, metadata, e.g. information on the medium, date, etc. of the text producers can be included.
- The text data corpus therefore contains the analysis units and their content as well as the metadata as a third dimension.

Tokenization

- In the next step, the analysis units are broken down into word units, so-called "**tokens**" (characters, words).
- **White-space tokenization** is usually used for this, i.e. the word units are identified from the chains of individual characters strung together using spaces and punctuation marks.
- Example: The sentence "The text material must be prepared" is broken down into the tokens "The", "text material", "must", "prepared" and "will".
- It may make sense to treat terms such as "text mining" as one token rather than several.
- Names can be identified with the help of the "JRC-Names" database. The JRC-Names database is provided by the European Commission as a scientific resource for language analysis (https://ec.europa.eu/jrc/en/language-technologies/jrc-names#Download%20JRC-Names).
- Sometimes it is also helpful to determine the word parts of the tokens, i.e. whether they are nouns or verbs.

Removal of stop words

- In the next step, the so-called “stop words” are removed from this analysis unit, which now consists of a number of single tokens.
- **Stop words** are single words such as articles, pronouns and prepositions.
- These occur in large numbers in texts, but are of little importance for the analysis.
- In addition, punctuation marks are usually removed and all capital letters are converted to lower case.

Stemming and lemmatization

- Both methods trace all words back to their word stem (their lemma).
- **Stemming** reduces words to their common letter sequences.
- Example: The verb form “researched” or “researching” becomes “research”.
- The stemming process often produces character strings that do not correspond to the linguistic word stem and are therefore not always directly understandable or are ambiguous. An example for that is the past form of “go”: “went”.
- **Lemmatization**, on the other hand, is about recognizing the basic linguistic form of a word.
- Example: “goes” becomes “go”, “went” becomes “go”.
- Extensive electronic dictionaries are usually used for lemmatization. This is therefore the more complex procedure.
- The method used in each case determines the vocabulary, i.e. the (word) types contained in a text corpus.

Document feature matrix

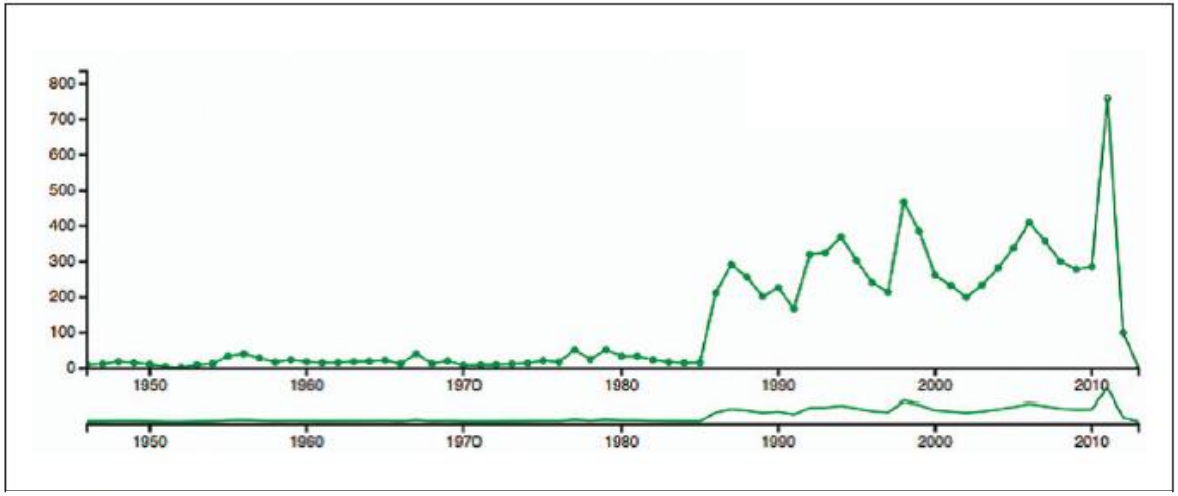
- The **document feature matrix (DTM)** contains a row for each analysis unit of the text data corpus and a column for each token.
- This makes the DTM the basis for further text statistical analyses.
- The texts per document or unit are represented as a set of words, disregarding the order in which they appear or the grammar.
- Example (without stemming and lemmatization!):
  - Analysis unit 1: „I always go by car.“
  - Analysis unit 2: „I never go by car.“
  - Document feature matrix:

	I	always	never	go	by	car
AU 1	1	1	0	1	1	1
AU 2	1	0	1	1	1	1

10.3 Text mining methods

The frequency analysis

- The simplest form of evaluating text data is **frequency analysis**.
- This involves counting the absolute and relative frequencies of all words or selected key terms in the text data corpus.
- This provides insights into patterns of term usage and their development over time.
- Example: In a study on the nuclear energy discourse, a frequency analysis was carried out on the term “nuclear energy”:
- From the second half of the 1980s, there were more articles on nuclear energy than before.



Frequency analysis for the term “nuclear energy”, n = 8,840 documents.

Manderscheid (2022), p. 1724.

Visualization of frequency analyses: The ngram

- **Ngrams** are suitable for displaying frequencies of words and word combinations over time or in comparison between different locations or media.
- N stands for the number of units such as characters or words.
- A monogram contains one unit, a bigram a word sequence with two units, etc.
- Example: Ngram viewer from Google: https://books.google.com/ngrams/. Generates Ngrams of one to five words from the books scanned into Google Books.
- Ngrams visualize the frequencies or probabilities of certain word combinations over time.

Visualization of frequency analyses: The word cloud

- Word frequencies can be visualized as a **tag cloud** or **word cloud**.
- All relevant keywords in the text are displayed over a large area, the size of the area is proportional to the frequency of their occurrence.
- In this way, the information contained in a text unit is visualized in a quantitatively condensed form.
- Example: Word Cloud for an excerpt from Durkheims Regeln der soziologischen Methode (Durkheim's Rules of Sociological Method).

Word cloud for an excerpt from Durkheim: Die Regeln der soziologischen Methode.

Manderscheid (2022), p. 1726.

The co-occurrence analysis

- In linguistics, **co-occurrence** refers to the joint occurrence of two lexical units, e.g. two words, within a context unit.
- Context units can be a sentence, paragraph or document, or a fixed number, right or left neighbor of a fixed term.
- A distinction is made between sentence co-occurrences - the terms are in a sentence, but not directly next to each other - and neighbor co-occurrences - the terms follow each other directly.
- This form of evaluation is based on the assumption that the analyzed lexical units, if they frequently occur together, belong together.
- **Co-occurrence analyses** thus make it possible to identify contexts of meaning in texts.

The sentiment analysis

- **Sentiment analysis** (also known as **opinion mining**) attempts to identify the mood or attitude expressed in a text.
- A distinction is made between positive or negative opinions, emotions, evaluations, beliefs or feelings.
- For this purpose, dictionaries with terms that express positive and negative emotions or evaluations are created or existing dictionaries are used.
- The ratio of positive and negative terms is then used to classify documents, sentences or other text units.
- Example: “The weather was a nightmare, but it was a great match and a super atmosphere ☺ ☺”. 4 positive and 1 negative expression → positive sentiment.
- The problem here is recognizing irony (“What the customer service did – just great”).

10.4 Text mining using 

Required packages and example data set

R Script

```
# Required packages
library(quanteda)
library(quanteda.textstats) # textstat_frequency()
library(quanteda.textplots) # textplot_wordcloud()
```

The example data set stems from kaggle.com and contains tweets from X from the early days of the Covid19 pandemic in March and April 2020.

R Script

```
# Data set description
# https://www.kaggle.com/datasets/datatattle/covid-19-nlp-text-classification?resource=download
# Use the train data set

# Read in data
# Name of the data set: Corona_NLP_train.csv
tweets <- read.csv(file.choose())
```

Generation of the text corpus

Having the data read in with the readtext() function, the text corpus can be generated directly with the corpus() function. The option docid_field= specifies the column in which the analysis unit is specified, and the option text_field= specifies the column in which the texts to be analyzed are located.

R Script

```
# Generation of the text corpus
corpus_tweets <- corpus(tweets,
                        docid_field = "UserName",
                        text_field = "OriginalTweet")


# Viewing the corpus
head(corpus_tweets)
str(corpus_tweets)
summary(corpus_tweets, 10) # the first 10 rows
```

Frequency analysis for the number of tweets by day (1/2)

Strictly speaking, no special text mining functions are required for this analysis.

The table() function creates the frequency count.

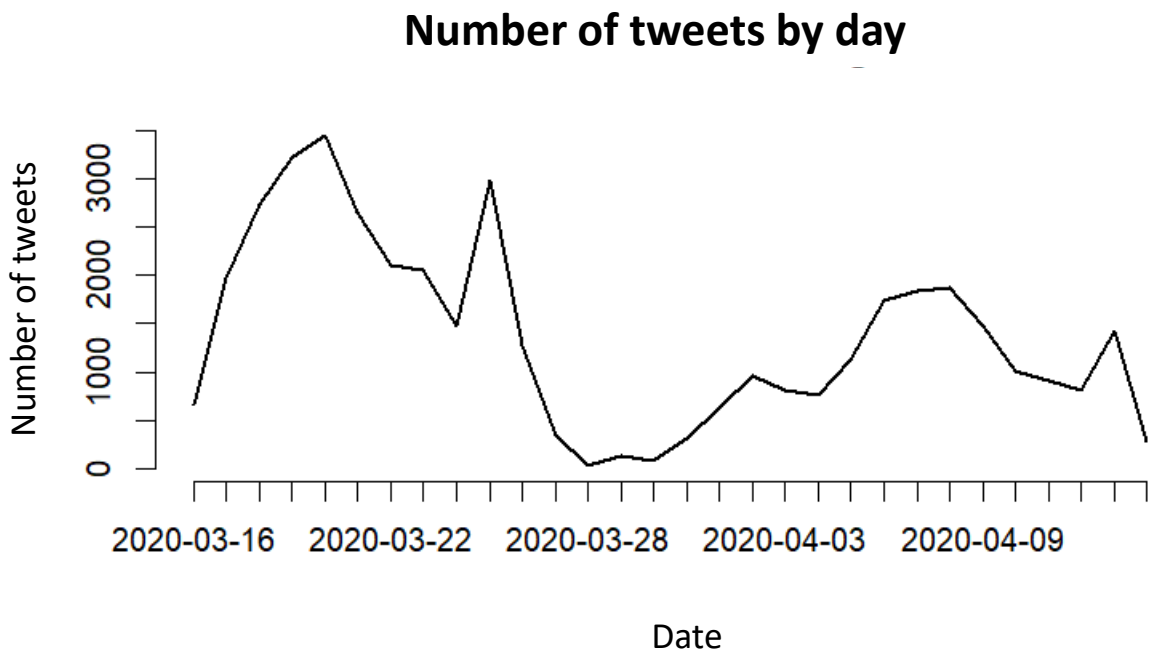
The function as.Date() with the format setting format="%d-%m-%Y" converts the dates into date format. If this is omitted, the dates on the x-axis are interpreted as text and not as dates and are arranged incorrectly.

The graphic option t="l" creates a line. If this is omitted, columns are created.

R Script

```
# Line plot for the number of tweets by day
plot(table(as.Date(tweets$TweetAt, format="%d-%m-%Y")),
                                   t="l",
                                   xlab = "Date",
                                   ylab = "Number of tweets",
                                   main = "Number of tweets by day")
```

Frequency analysis for the number of tweets by day (2/2)



Unsurprisingly, most of the tweets are from the very beginning of the pandemic.

Scanning tweets for individual terms and phrases

A text corpus can be scanned for individual words and phrases using the kwic() function. To do this, the corpus must first be converted into the object class “token” (with the tokens() function).

Example: Scanning the corpus for the phrases “panic” and “don't panic”.

R Script

```
# Scan text corpus for terms and phrases
kwic(tokens(corpus_tweets), pattern = "panic")
kwic(tokens(corpus_tweets), pattern = phrase("don't panic"))
```

Editing the text corpus

It is helpful to remove stop words and punctuation from a text corpus.

Stop words can be removed with the remove_tokens() function, punctuations (and any other characters and words) can be removed with the tokens() function.

R Script

```
# FYI: List of pre-defined English and German stop words
head(stopwords("en"), 20)
head(stopwords("de"), 20)

# Remove stop words and other characters
corpus_reduced <- tokens_remove(tokens(corpus_tweets),
pattern=stopwords("en"))

corpus_reduced <- tokens_remove(tokens(corpus_tweets, remove_punct =
TRUE), pattern=stopwords("en"))
```

Creating and analyzing the document feature matrix

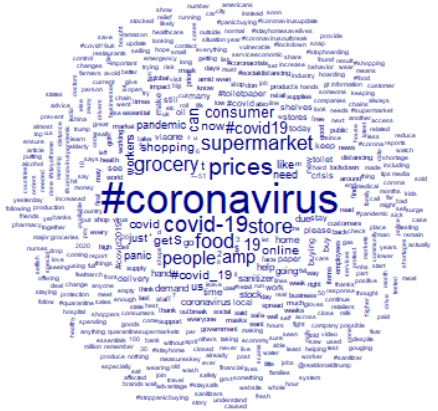
The document feature matrix enables analyses such as a word cloud.

R Script

```
# Create document feature matrix
dfm_tweet <- dfm(tokens(corpus_reduced))

# The 20 top most features in the tweets
topfeatures(dfm_tweet, 20)

# The 20 top most features in the tweets
textstat_frequency(dfm_tweet, 20)
# Reading example: The term "prices" appeared 7849 times in 7382
different tweets.

# Word cloud
textplot_wordcloud(dfm_tweet)
```

Analysis results for the document feature matrix

```
> textstat_frequency(dfm_tweet, 20)
        feature frequency rank docfreq group
1  #coronavirus     16021    1   15914   all
2        prices      7849    2    7382   all
3     covid-19      7591    3    7421   all
4          food      6833    4    5873   all
5         store      6820    5    6448   all
6   supermarket      6701    6    6575   all
7       grocery      6087    7    5853   all
8        people      5550    8    4871   all
9           amp      5197    9    3852   all
10           19      4946   10    4848   all
11     #covid19      4845   11    4826   all
12            s      4808   12    4196   all
13     consumer      4297   13    4121   all
```



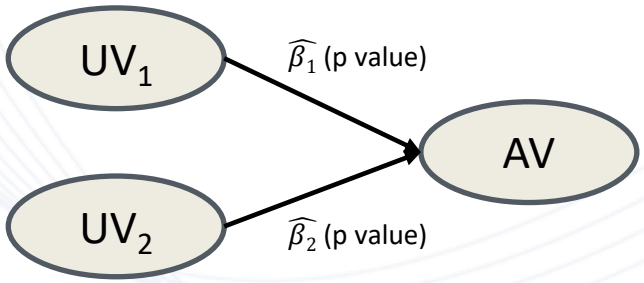
Reading example: The term "prices" appeared 7849 times in 7382 different tweets.

Appendix

Interpretation of statistical key figures in scientific papers

$\bar{X}; M; \hat{\mu}$	Sample mean
$s; sd; \hat{\sigma}$	Sample standard deviation
$l_{95}\ and\ u_{95}$	Lower and upper 95% confidence limits
$q_{25};\ q_{75}$	Lower and upper quartile
IQR	Inter quartile range
$r;\ \rho$ (rho)	(Pearson) correlation coefficient
$\hat{\beta}$ (beta)	Estimator for the regression coefficient / path coefficient
p; p-value	p value
SE	Standard error

Interpretation of key figures in model diagrams:



Note: Usually, standardized path / regression coefficients $\hat{\beta}$ are presented.

Literature

- Backhaus, Klaus; Erichson, Bernd; Gensler, Sonja; Weiber, Rolf; Weiber, Thomas (2023). Multivariate Analysis, 2nd. Edition, Springer Gabler.
- Baraldi, Amanda N.; Enders, Craig K. (2010). An introduction to modern missing data analyses. Journal of School Psychology (48), 5–37, https://www.researchgate.net/profile/Mansour-Abdullah-Alshehri/post/Are-there-any-suggested-criteria-for-dropping-questionnaires-in-quantitative-research-due-to-missing-responses/attachment/5a861c36b53d2f0bba52349f/AS%3A594446450176000%401518738486135/download/An+introduction+to+modern+missing+data+analyses.pdf, accessed 11.02.2025.
- Caliński, T.; Harabasz, J. (1974). A dendrite method for cluster analysis. Communications in Statistics - Theory and Methods, 3(1), 1-27, http://docode.techyoung.cn/calinski_harabasz/chi.pdf, accessed 29.01.2025.
- Davenport, Thomas H.; Patil, D.J. (2012). Data Scientist: The Sexiest Job of the 21st Century, Harvard Business Review, Oktober 2012, https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century, accessed 03.01.2025.
- Friendly, Michael (1994). Mosaic displays for multi-way contingency tables. *Journal of the American Statistical Association*, *89*, 190–200.
- Kleiber, Christian; Zeileis, Achim (2008). Applied Econometrics with R, Springer, New York.
- Kleiber, Christian; Zeileis, Achim (2024). AER: Applied Econometrics with R, https://CRAN.R-project.org/package=AER , accessed 07.01.2025.

Literature

- Manderscheid, Katharina (2022), Text Mining, in: Baur, N.; Blasius, J. (eds.), Handbuch Methoden der empirischen Sozialforschung, 3rd edition, Springer, Wiesbaden, pp. 1719-1732
- Mojena, R. (1977). Hierarchical clustering methods and stopping rules: An evaluation. The Computer Journal, 20(4), 359–363. https://academic.oup.com/comjnl/article-pdf/20/4/359/1108679/200359.pdf, accessed 16.01.2025.
- Olawale, Fatoki; Garwe, David (2010). Obstacles to the growth of new SMEs in South Africa: A principal component analysis approach. African Journal of Business Management Vol. 4(5), 729-738.
- Provost, Foster; Fawcett, Tom (2013). Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking, O'Reilly.
- Röver, Christian; Raabe, Nils; Luebke, Karsten; Ligges, Uwe; Szepannek, Gero; Zentgraf, Marc; Meyer, David (2023). klaR: Classification and Visualization. https://CRAN.R-project.org/package=klaR, accessed 07.01.2025.
- Shearer, Colin (2000). The CRISP-DM Model: The New Blueprint for Data Mining. Journal of Data Warehousing, 5(4), pp. 13-22.