Manag. Data Science
Cluster Analysis

Matthias Temple

Institute for Competitiveness and Communication

University of Applied Sciences and Arts Northwestern Switzerland
School of Business

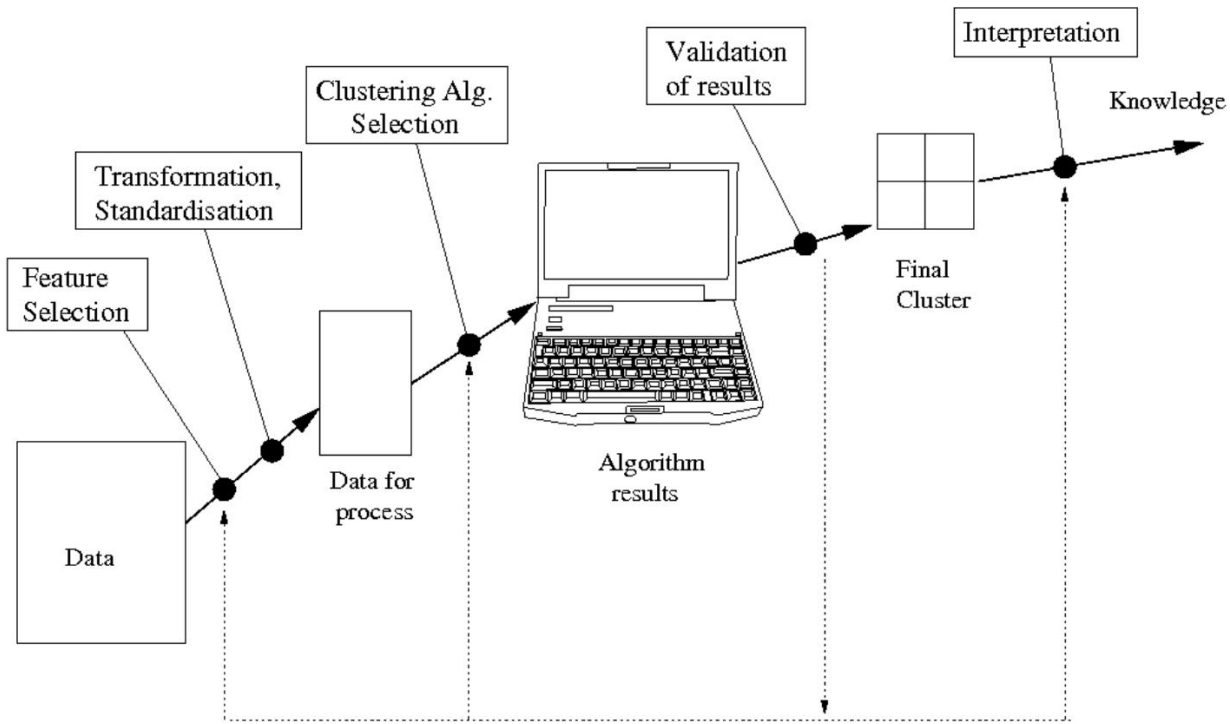
Important from before

ÿ Distance measure (the basis for most cluster methods) ÿ Transformation, standardization ÿ Introduction to cluster analysis including Mussel example ÿ Comments on variable clustering (also called Q-mode Clustering)
ÿ Comments on missing values and variable selection

Review: what is cluster analysis?

- **Exploratory** tool to find similar observations in data find.

Slightly simplified process of a cluster analysis



Different approaches

ÿ **Visual:** with parallel coordinate plot, the heatmap
Distances, MDS and Tours
ÿ **Hierarchical:** in each step, clusters become larger
(agglomerative) or smaller (divisive).
ÿ **Partitioning:** each observation becomes exactly one cluster
allocated
ÿ **Fuzzy clustering** and **model-based clustering:** each
Observations fall into each cluster with a certain level of
membership. ÿ **Density-**
based methods: can be thought of as propagating
Introduce epidemic
ÿ etc.

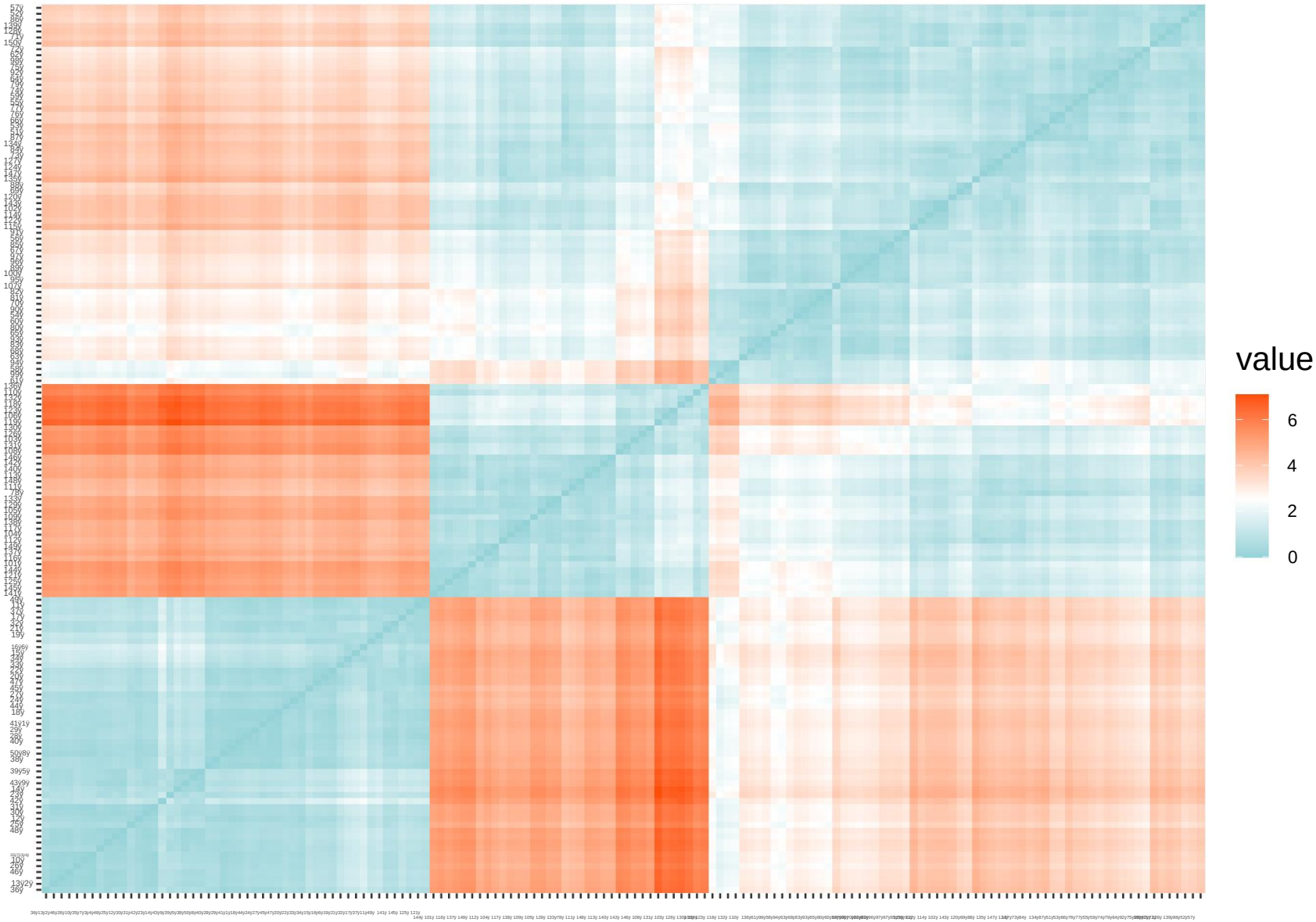
Distances again: Heatmap

The (sorted) distance matrix already gives an idea of whether there is a good cluster structure. Using the iris data it looks like this (plot: next page)

```
library(factoextra)
distance <- get_dist(iris[, 2:5]) # or dist() fviz_dist(distance,
order = TRUE, lab_size = 1, gradient =
                  list(low =
                  "#00AFBB", mid = "white", high =
                                       "#FC4E07"))
```

If structures are noticeable, this is an indication that a good cluster structure is present.

Distances again: Heatmap



Again distances

To try it out live (as it is interactive)

```
library(heatmaply)
heatmaply(as.matrix(distance))
```

The dendrograms on the right and above will be discussed next.

Your cooperation is required

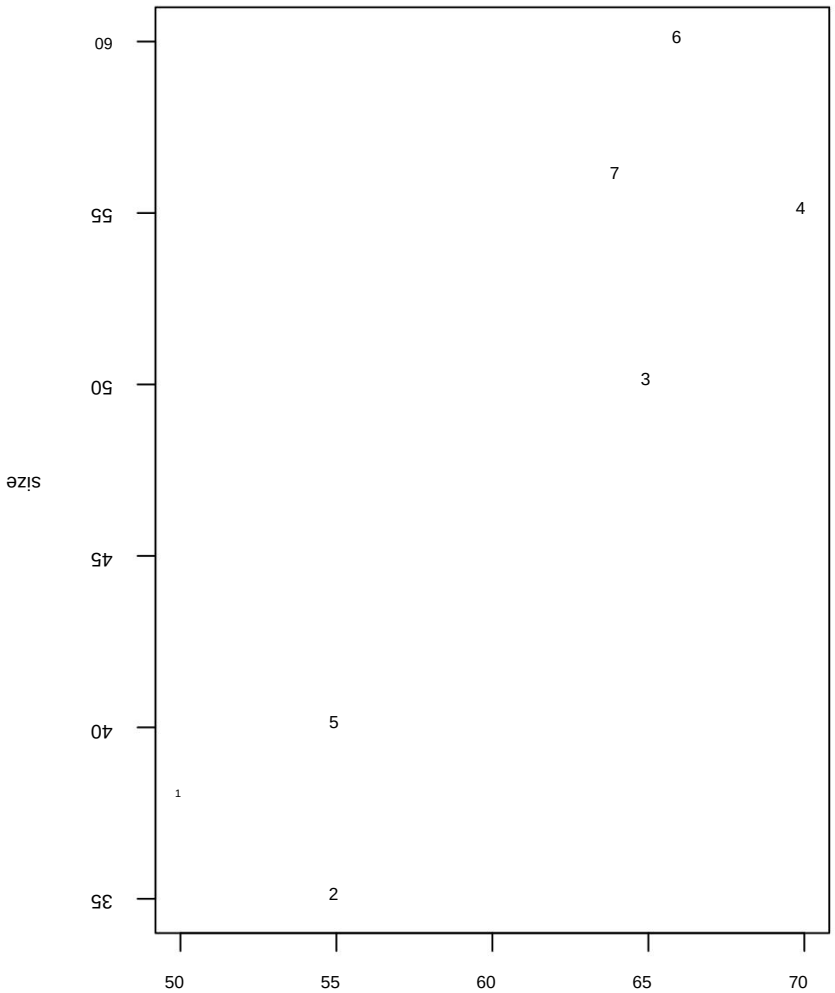
all_exercises-cluster_nolsg.pdf

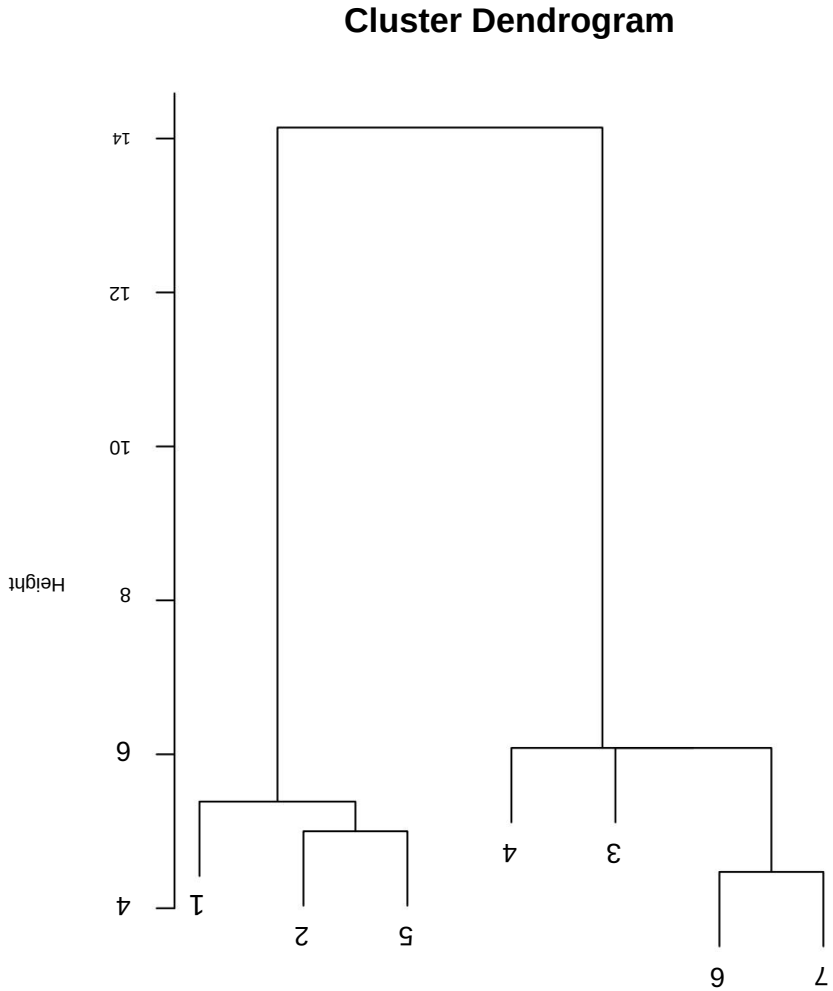
Clustering

- Task 1: Visualize the distances between observations

Hierarchical clustering

Dendrogram: example Mussels, Height expresses the dissimilarity





Hierarchical clustering

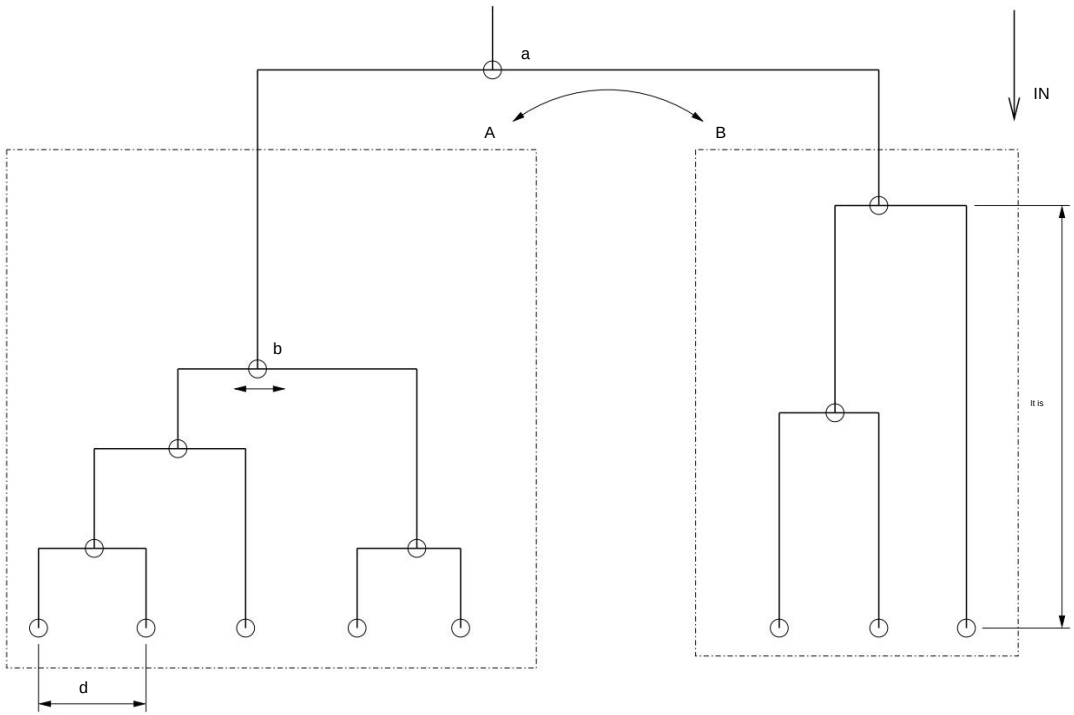
Sequence of cluster partitions visualized with a clustering tree, the **dendrogram.**

dendrogram:

ÿ First, each data point is a cluster, these are then successively combined. ÿ The fusion of different clusters is marked with a horizontal line in the dendrogram. ÿ The y-axis of the dendrogram shows the heterogeneity removed within the clusters, it increases with increasing cluster size. ÿ Areas with long vertical lines in the dendrogram indicate a large increase in heterogeneity. ÿ Different methods can be used to measure heterogeneity/dissimilarity.

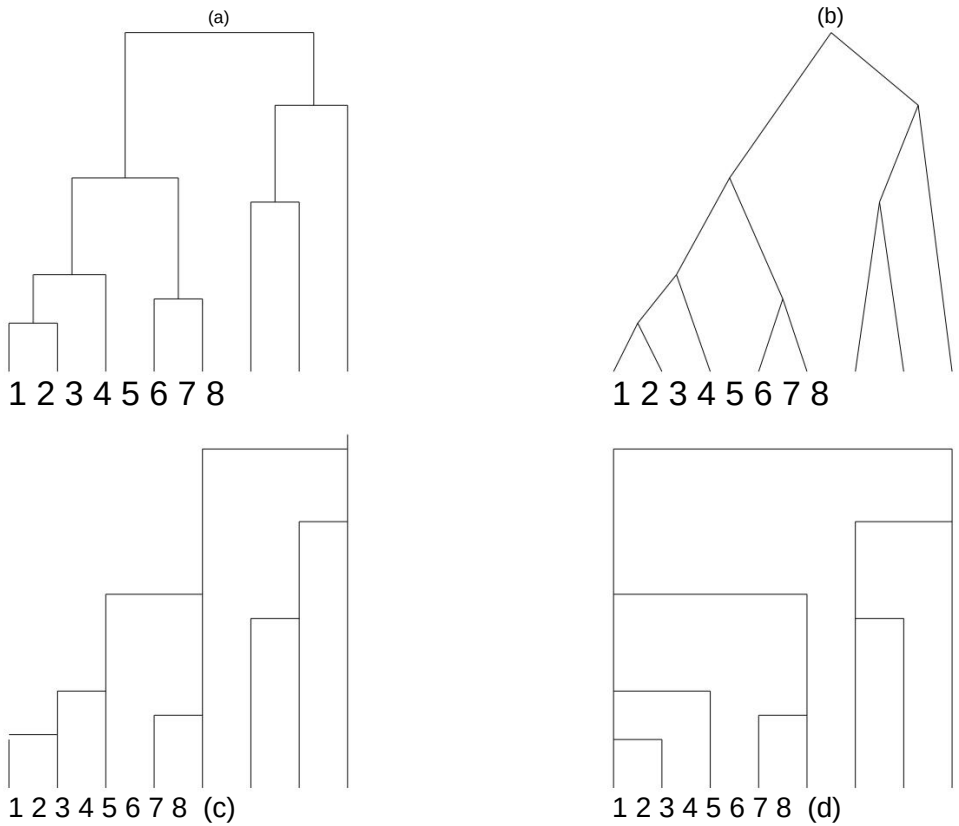
Hierarchical clustering

Some free design options: A and B can be swapped, b can be chosen arbitrarily, likewise d. c corresponds to the dissimilarity of clusters.



Hierarchical clustering

Four possible dendrograms containing the same info:



Hierarchical clustering (agglomerative)

More precisely:

Start:

- ÿ Each object is its own cluster ÿ n different ones Cluster.

Step-by-step procedure:

- ÿ In each step, the number of clusters is reduced by 1, with the two most similar classes being combined. ÿ Dissimilarity can be measured in different ways (single linkage, complete linkage, average linkage, Ward, . . .).

- ÿ A height is assigned to the newly obtained cluster ÿ In the end there is only one cluster left, all objects are in this cluster.

General concept (only for overview!)

ÿ Let Ci and Cj be two clusters and let the dissimilarity measure (dissimilarity) between these clusters be d(Ci , Cj).

ÿ As soon as these two clusters are linked, the following generalized scheme applies regarding the dissimilarity between Ci ÿ Cj and another cluster Ck

$$d(Ci \text{ ÿ } Cj , Ck) = \text{ÿ}i\,d(Ci , Ck) + \text{ÿ}j\,d(Cj , Ck) + \text{ÿ}d(Ci , Cj) + \text{ÿ}|d(Ci , Ck) \text{ ÿ } d(Cj , Ck)|,$$

mit ÿi , ÿj , *ÿ, ÿ* ÿ R.

General concept (only for overview!)

The most common choice of parameters:

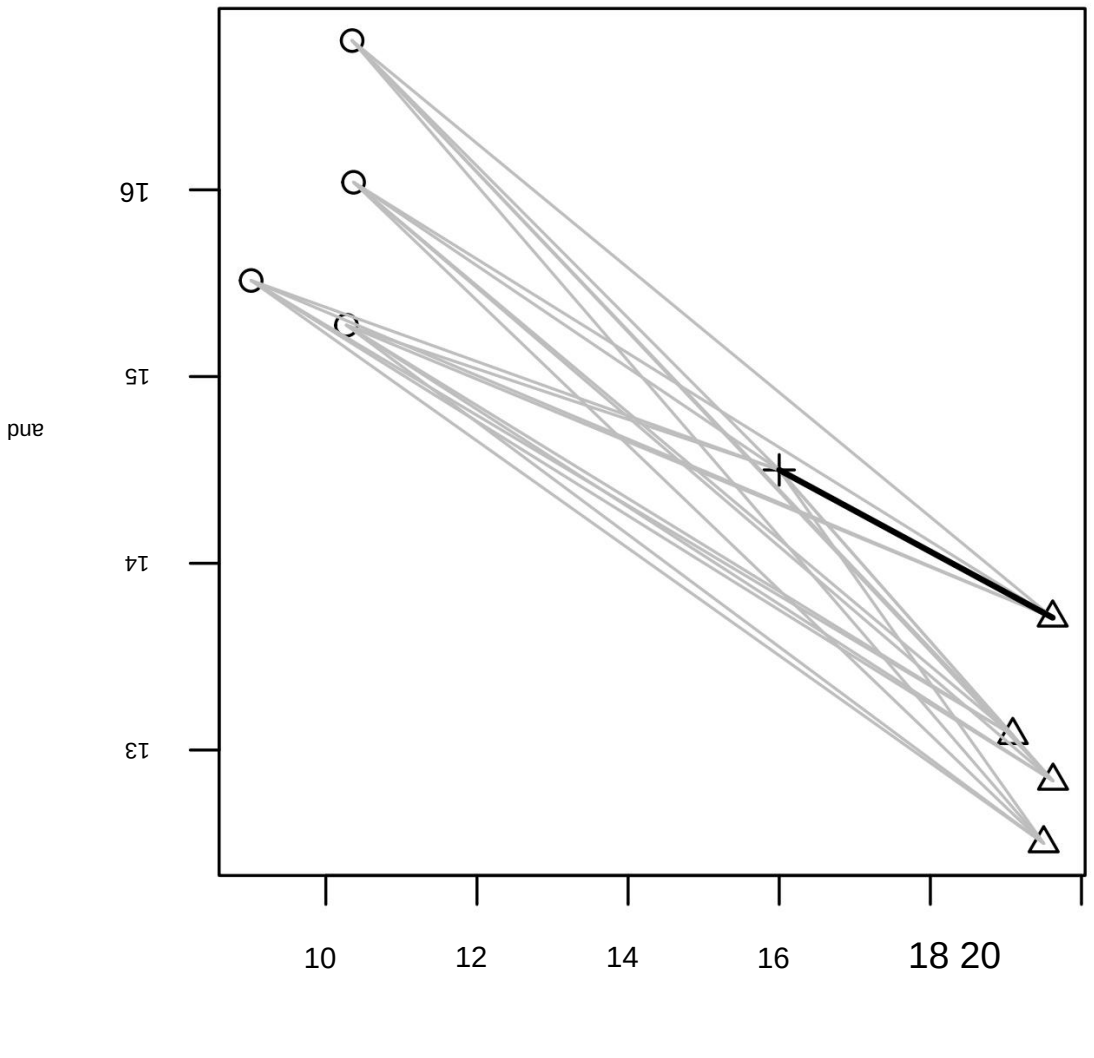
Clustering criterion	ÿi , ÿj	b	c
Single linkage	–	0	$\frac{1}{}$
Complete linkage	–	0	$\frac{}{2}$
Average linkage	–	0	1 2 0
Centroid linkage	$\frac{1}{2\ 1\ 2}$	$\frac{}{(in+nj)^2}$	0
Ward's method	$\frac{1\ 2\ ni}{ni+nj\ ni+nk\ ni+nj+nk}$	$\frac{-nk}{ni+nj+nk}$	0

nl is the number of observations in cluster Cl (l = *i, j,* k)

Easier to understand: the following illustrations

Single linkage (2-dim example)

To which cluster (observations with ÿ or with ÿ) is observation + assigned?



Single Linkage

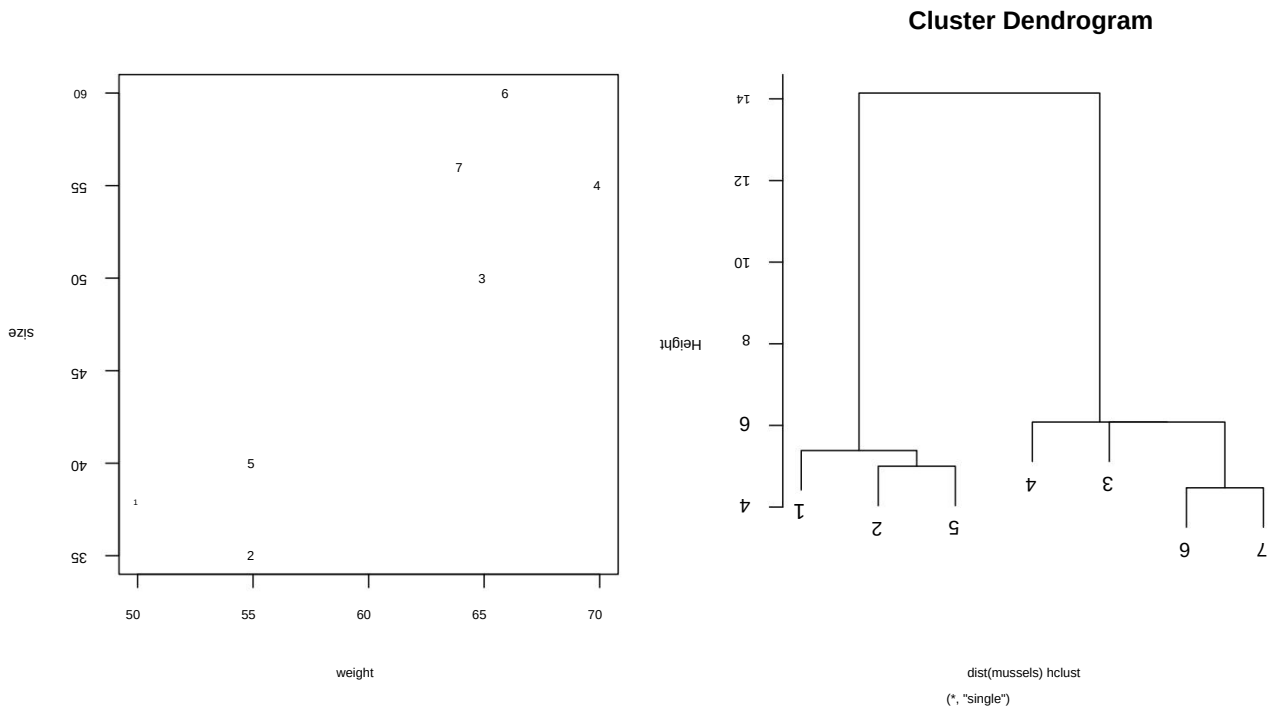
ÿ Tends to clusters of different sizes due to large clusters rather quickly put together. ÿ

Therefore also sometimes used for outlier detection. ÿ Can be calculated very efficiently.

Single linkage (2-dim example, data: mussels)

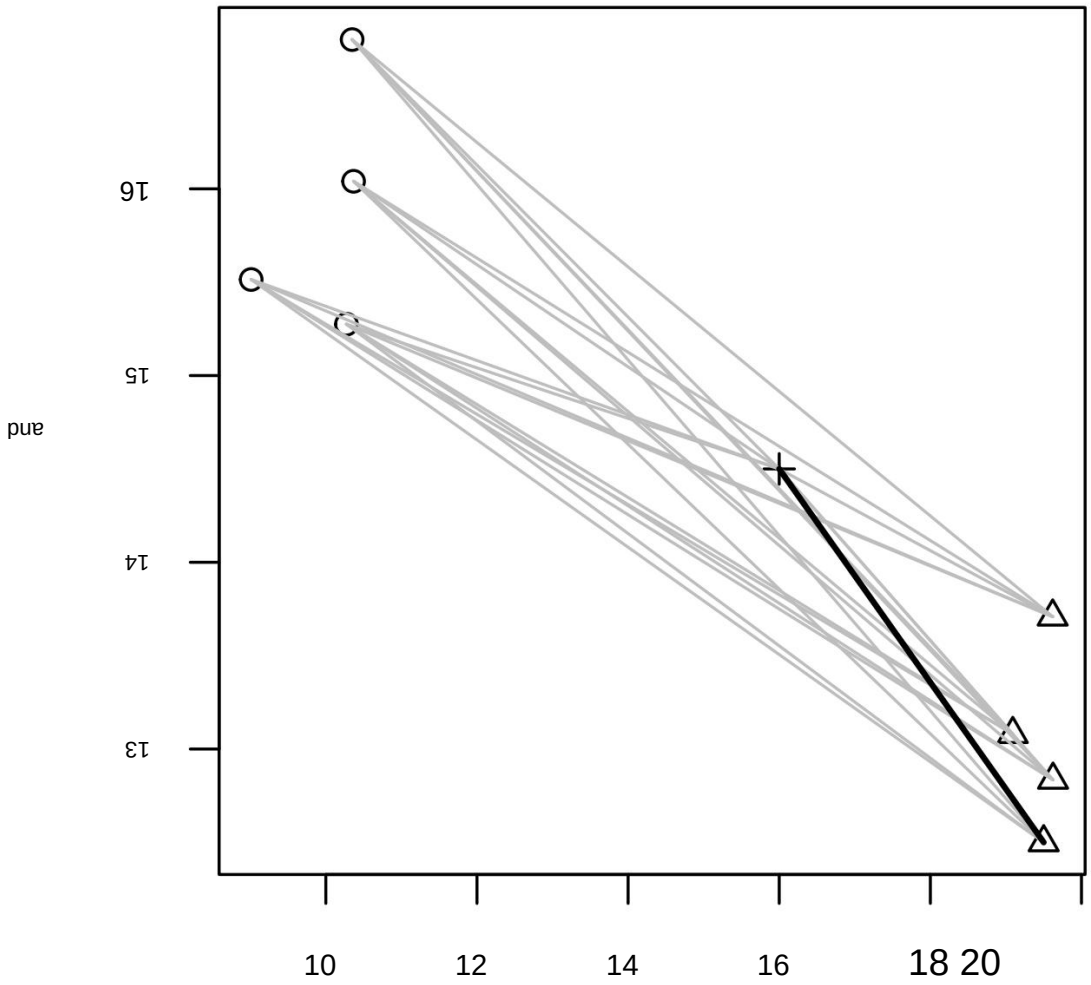
Repeat, this time with code:

```
par(mfrow = c(1,2), pty = "s")
plot(mussels, type = "n");text(mussels, rownames(mussels))
plot(hclust(dist(mussels ), method = "single"))
```



Complete Linkage (2-dim example)

The most distant observation from each cluster, to which cluster is this distance the smallest. . .



Complete linkage and other linkage criteria

Complete Linkage:

ÿ leads to clusters of more or less the same size. ÿ Low computing time

Average Linkage:

ÿ takes medium distances instead of minimum (single linkage) or maximum (complete linkage), but the computing time increases somewhat.

Ward Method:

ÿ in each step, every possible union with a Assessed information loss criterion, therefore more computationally intensive

ÿ the information loss criterion is usually the squared one Distance to the cluster means

ÿ Selection of that union with minimum increase in information criterion

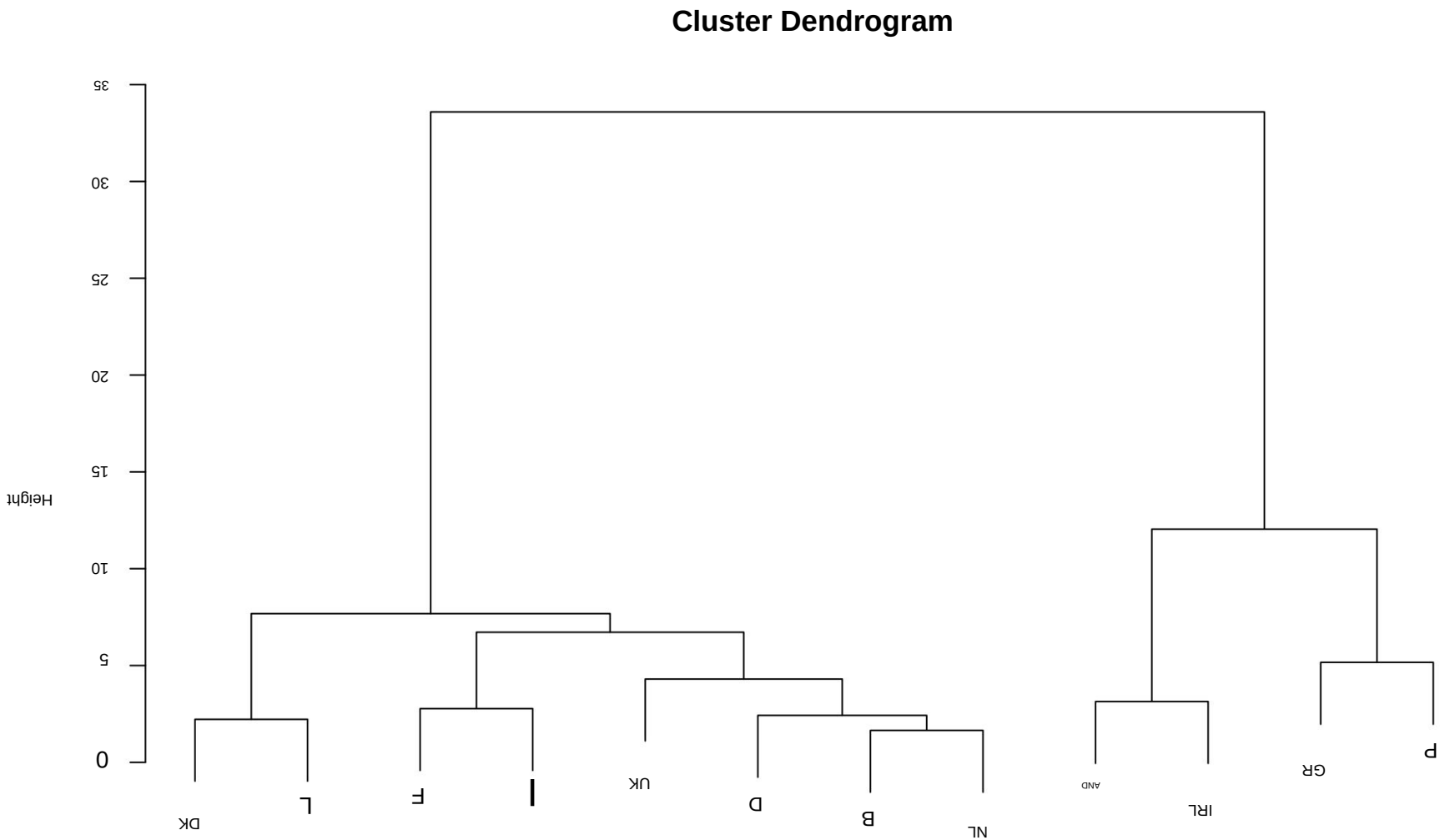
Example of hierarchical clustering

```
data(agriculture, package = "cluster") # ?agriculture
colnames(agriculture)
<- c("GDP", "Agriculture") agriculture
```

```
##              GDP Agriculture ##
B 16.8  2.7
## DK 21.3                 5.7
## D 18.7                  3.5
## GR 5.9                 22.2
## E 11.4                 10.9
## F 17.8                  6.0
## IRL 10.9               14.0
## I 16.6                  8.5
## L 21.0                  3.5
## NL 16.4                 4.3
## P          7.8         17.4
## UK 14.0
```

Example hierarchical clustering (Ward)

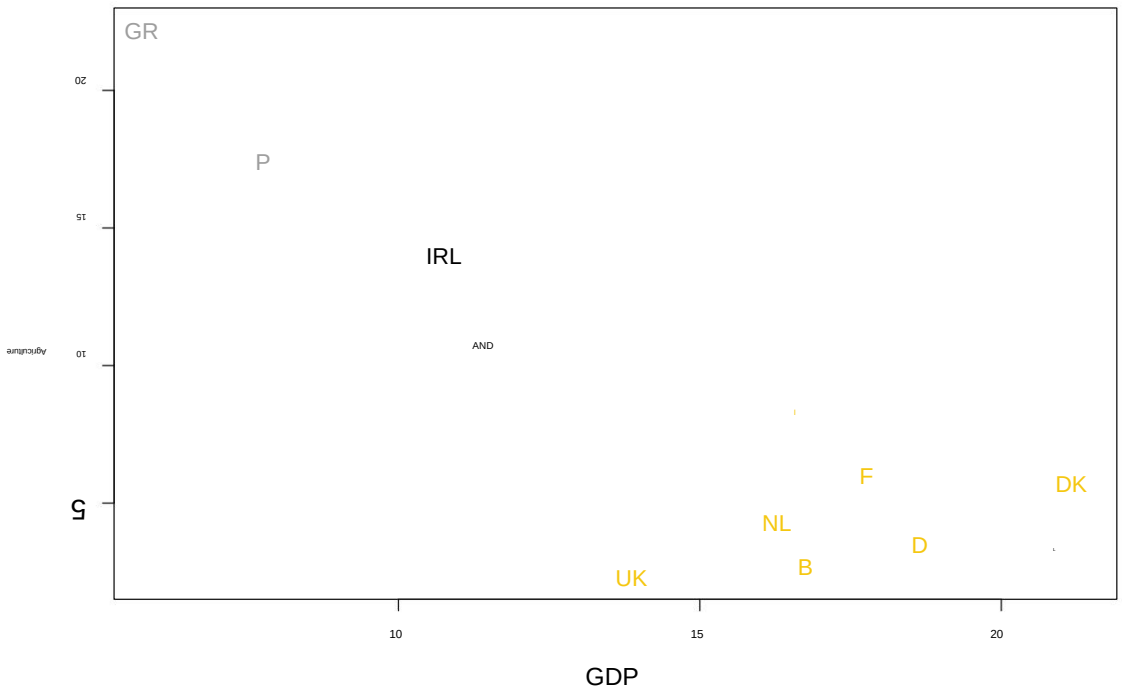
```
cl <- hclust(dist(agriculture), method = "ward.D2") plot(cl)
```



Example of hierarchical clustering: cutree

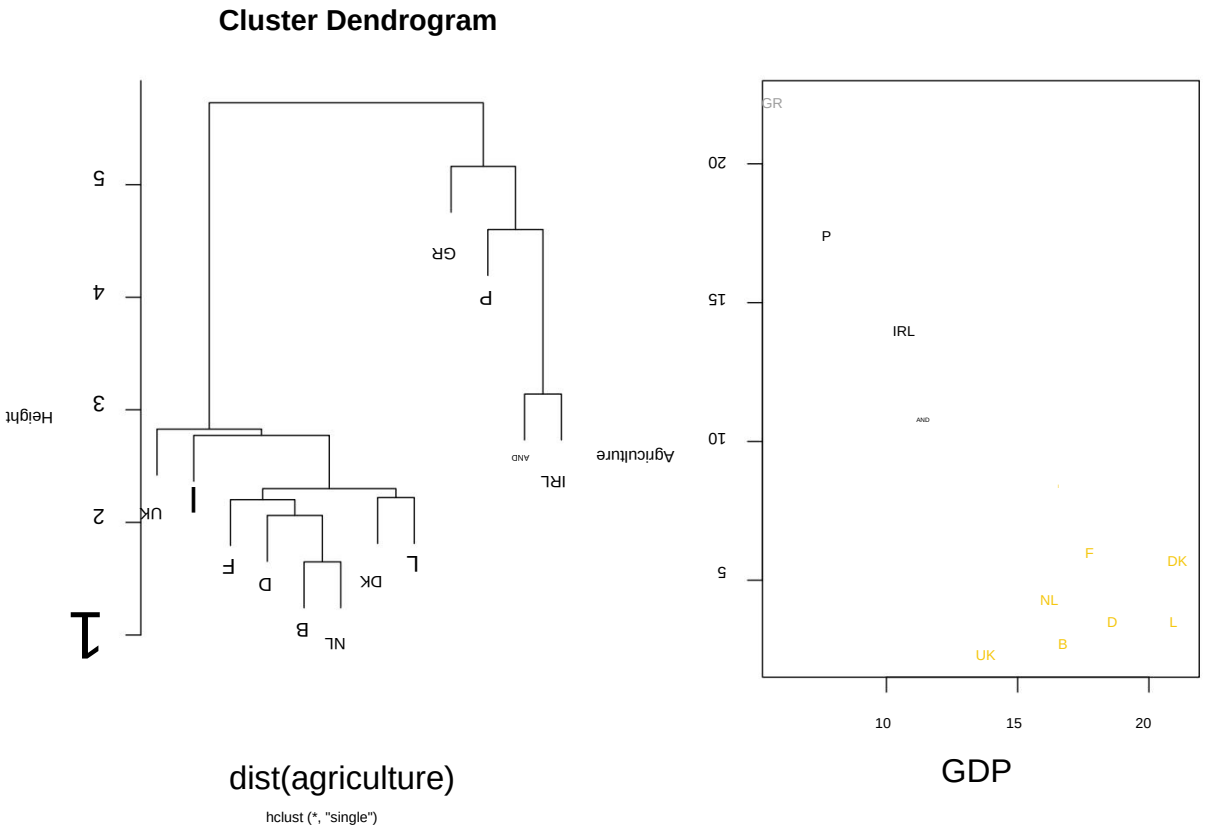
Cut through the histogram with the cutree function at that height so that, for example, exactly 3 clusters are created.

```
ct <- cutree(cl, 3)
plot(agriculture, type = "n", cex.lab = 1.5) text(agriculture,
rownames(agriculture),
        col = ct+6, cex = 2) # plot mit agriculture
```



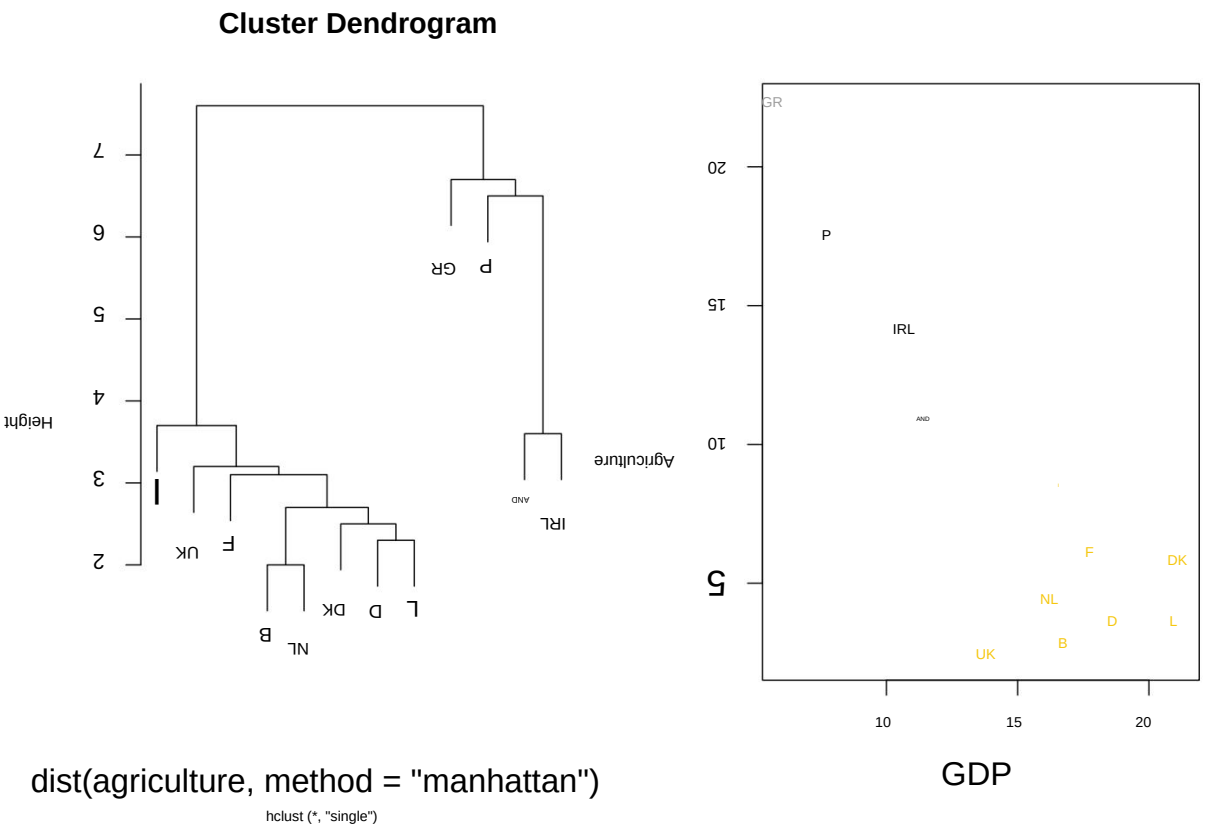
Example of hierarchical clustering

```
cl <- hclust(dist(agriculture), method = "single") ct <- cutree(cl,
3) par(mfrow = c(1, 2),
cex.lab = 1.5, cex = 1.2) plot(cl); plot(agriculture, type =
"n", cex = 2) text(agriculture, rownames(agriculture), col =
ct+6)
```



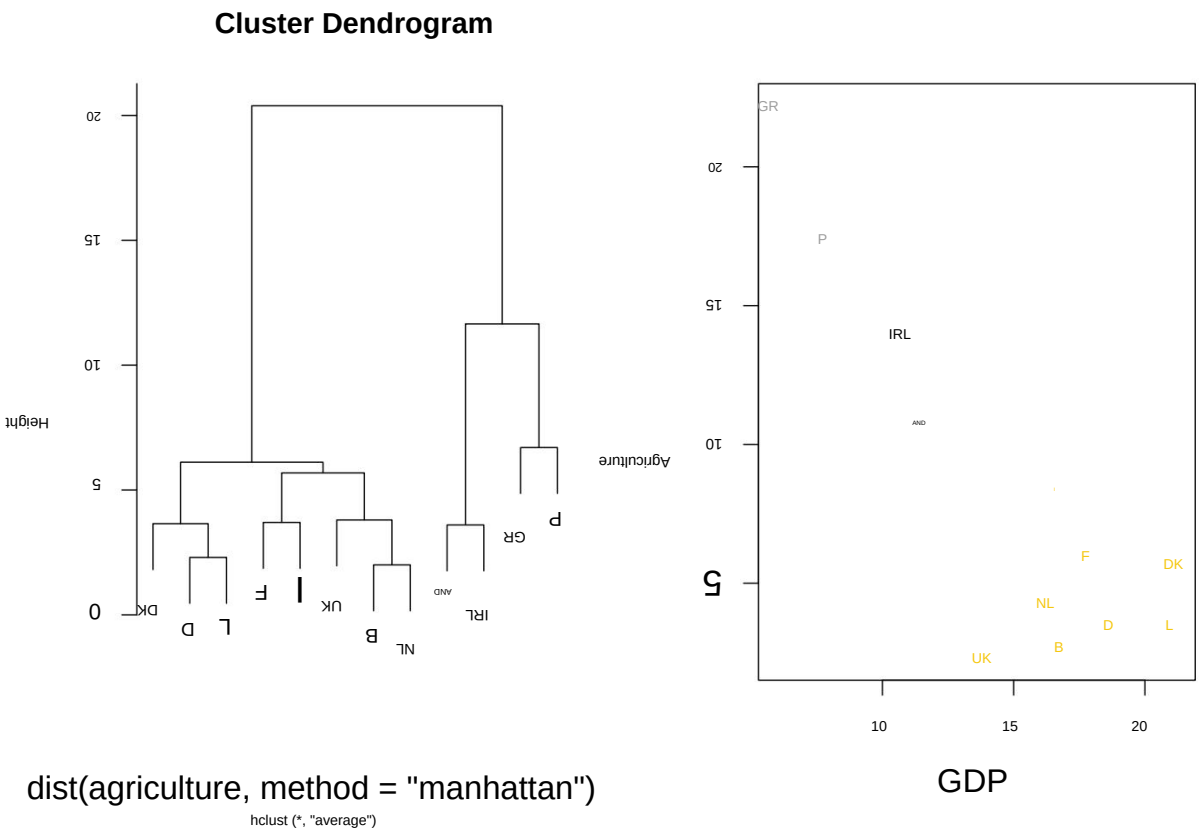
Example of hierarchical clustering

```
cl <- hclust(dist(agriculture, method = "manhattan"), method =
                    "single"); ct <- cutree(cl, 3) par(mfrow = c(1, 2),
cex.lab = 1.5, cex = 1.2) plot(cl); plot(agriculture, type =
"n", cex = 2) text(agriculture, rownames(agriculture), col =
ct+6)
```



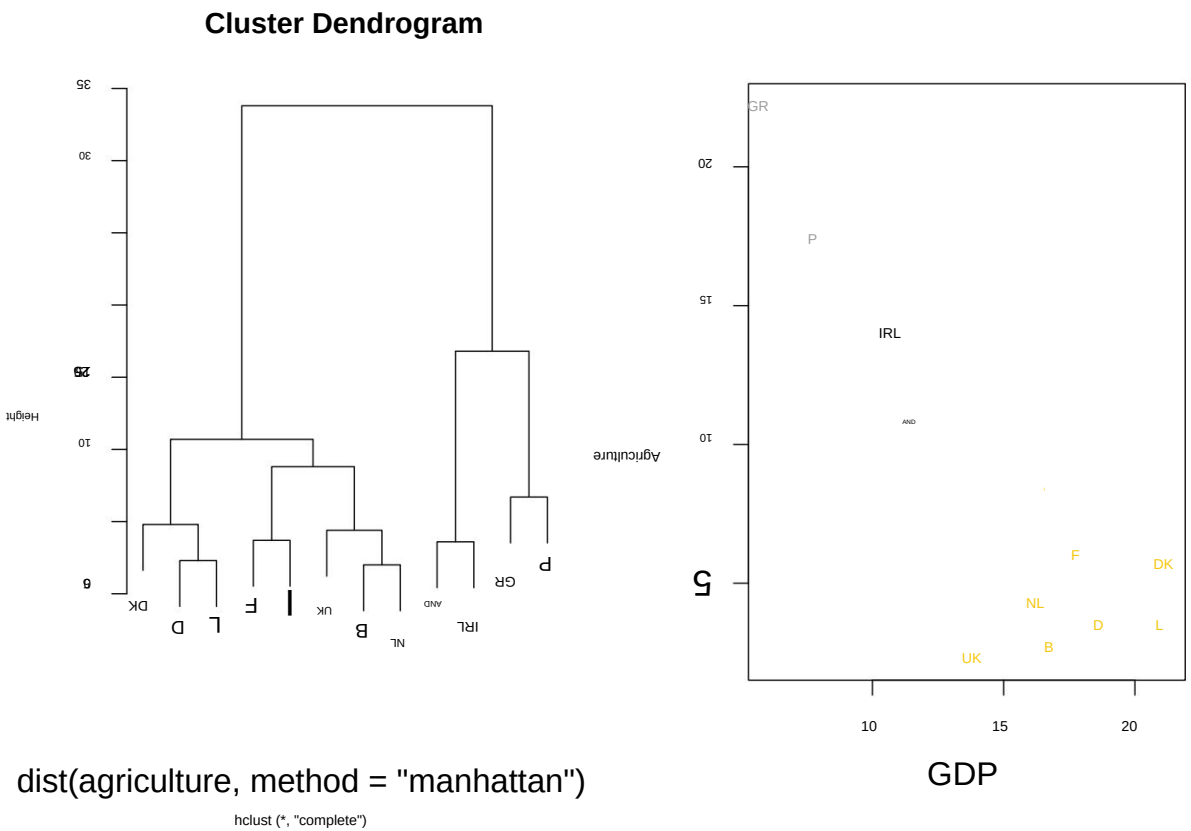
Example of hierarchical clustering

```
cl <- hclust(dist(agriculture, method = "manhattan"), method =
                    "average"); ct <- cutree(cl, 3) par(mfrow = c(1,
2), cex.lab = 1.5, cex = 1.2) plot(cl); plot(agriculture,
type = "n", cex = 2) text(agriculture, rownames(agriculture),
col = ct+6)
```



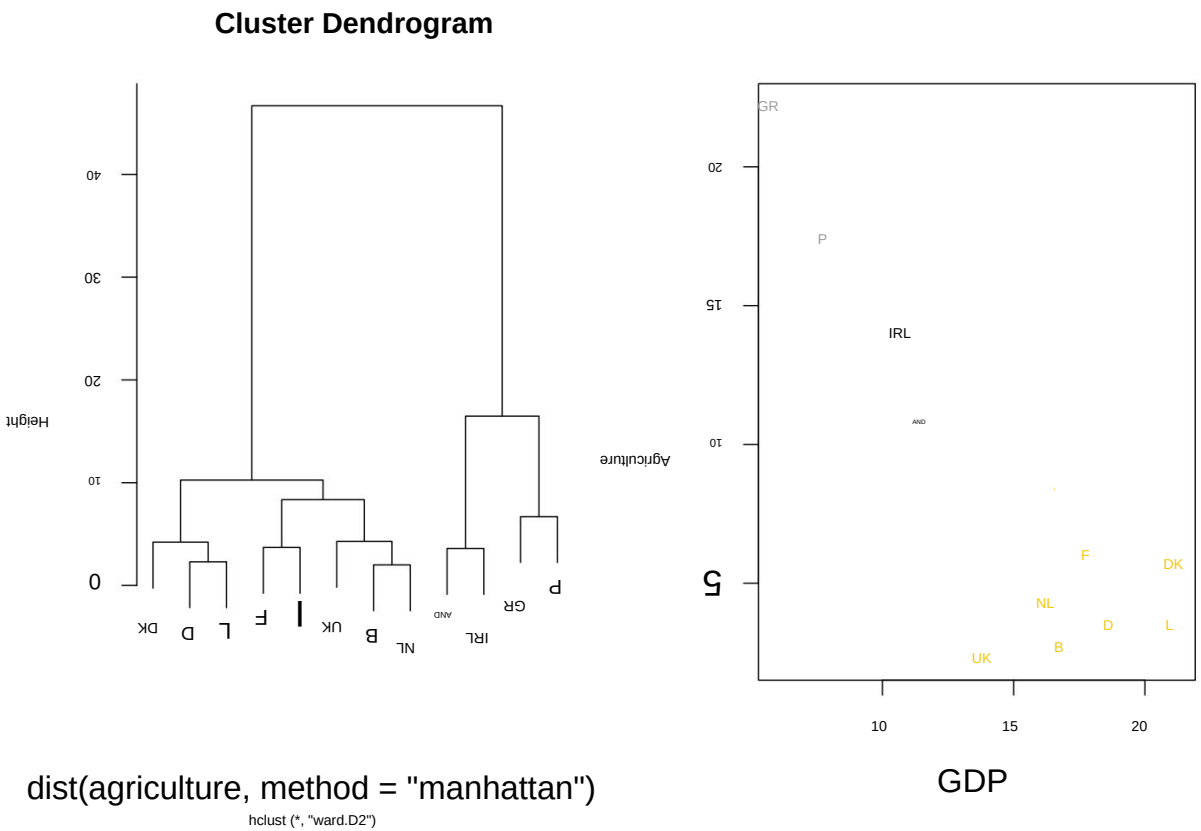
Example of hierarchical clustering

```
cl <- hclust(dist(agriculture, method = "manhattan"), method =
                    "complete"); ct <- cutree(cl, 3) par(mfrow = c(1,
2), cex.lab = 1.5, cex = 1.2) plot(cl); plot(agriculture,
type = "n", cex = 2) text(agriculture, rownames(agriculture),
col = ct+6)
```



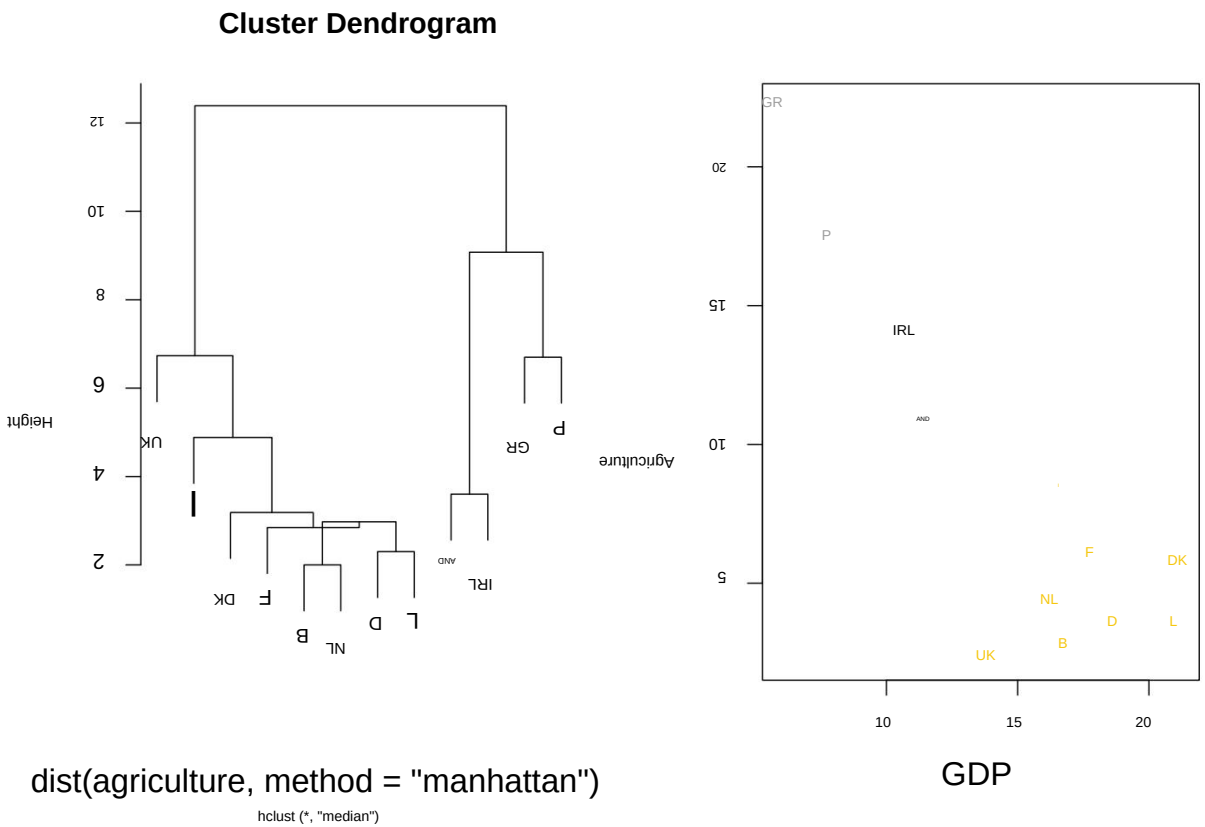
Example of hierarchical clustering

```
cl <- hclust(dist(agriculture, method = "manhattan"), method =
                                "ward.D2"); ct <- cutree(cl, 3) par(mfrow = c(1,
2), cex.lab = 1.5, cex = 1.2) plot(cl); plot(agriculture,
type = "n", cex = 2) text(agriculture, rownames(agriculture),
col = ct+6)
```



Example of hierarchical clustering

```
cl <- hclust(dist(agriculture, method = "manhattan"), method =
                    "median"); ct <- cutree(cl, 3) par(mfrow = c(1,
2), cex.lab = 1.5, cex = 1.2) plot(cl); plot(agriculture,
type = "n", cex = 2) text(agriculture, rownames(agriculture),
col = ct+6)
```



Your cooperation is required

all_exercises-cluster_nolsg.pdf

Clustering

- Task 2: Hierarchical clustering and Dendrogram
- Task 3: Q-mode clustering (Variablenclustering)

partitioning methods

1st step: Fix number of clusters 2nd step: Apply a partitioning cluster method

Restrictions:

- ÿ Each observation falls into exactly one cluster
- ÿ Each cluster contains at least one observation

The best known and by far the most frequently used

The method used is the E(xpection)M(aximum) algorithm **k-means.**

We will get to know (mostly) better methods later.

k-means. mean vectors

ÿ Starting point is data matrix **X** with n observations and p Variables.

ÿ Goal: Assign observations to the nc clusters

{C1, C2, . . . , Cnc } such that clusters Ck total n(k) has members and each observation is assigned to exactly one cluster.

ÿ The p components of the mean vector (centroid, center or also called prototype) vk of a cluster Ck can be calculated as follows.

$$v_k (\ddot{y} \; R^p) = \ddot{y} \frac{1}{n(n)} \sum_{i=1}^{n(n)} x_{i1}^{(k)}, \ldots, \frac{1}{n(n)} \sum_{i=1}^{n(n)} x_{ip}^{(k)} \ddot{y}^T$$

x $_i^{(k)} = (x_{i1}^{(k)}, \ldots, x_{ip}^{(k)})^{\ddot{y}}$ where $i_1, \ldots, i_p$ denotes the i-th observation assigned to cluster Ck . For each cluster C1, . . . , Cnc become the centroids v1, . . . , vnc calculated.

k-means. EM steps

If the number of clusters nc was previously initialized , the start location (the centroids) of the clusters nc is also initialized.

The algorithm now iterates, always assigning the observations to the nearest centroids:

1) Fix an initial partition with nc clusters.

2) E-step: (re)compute the centroids with the current cluster affiliations (memberships).

3) M-step: Assign each object to the closest cluster centroid ÿ new affiliations.

4) Go to 2) until the memberships, and therefore the centroids, do not change by more than a very small constant.

k-means. objective function

k-means clustering thus optimizes the objective function

$$\mathbf{J(X,}\ V,\ \mathrm{U}) = \sum_{k=1}^{nc} \sum_{i=1}^{n} \mathrm{ok}\ d^{\,2}\ \mathbf{(xi\ ,}\ \mathrm{vk}\ ), \qquad (1)$$

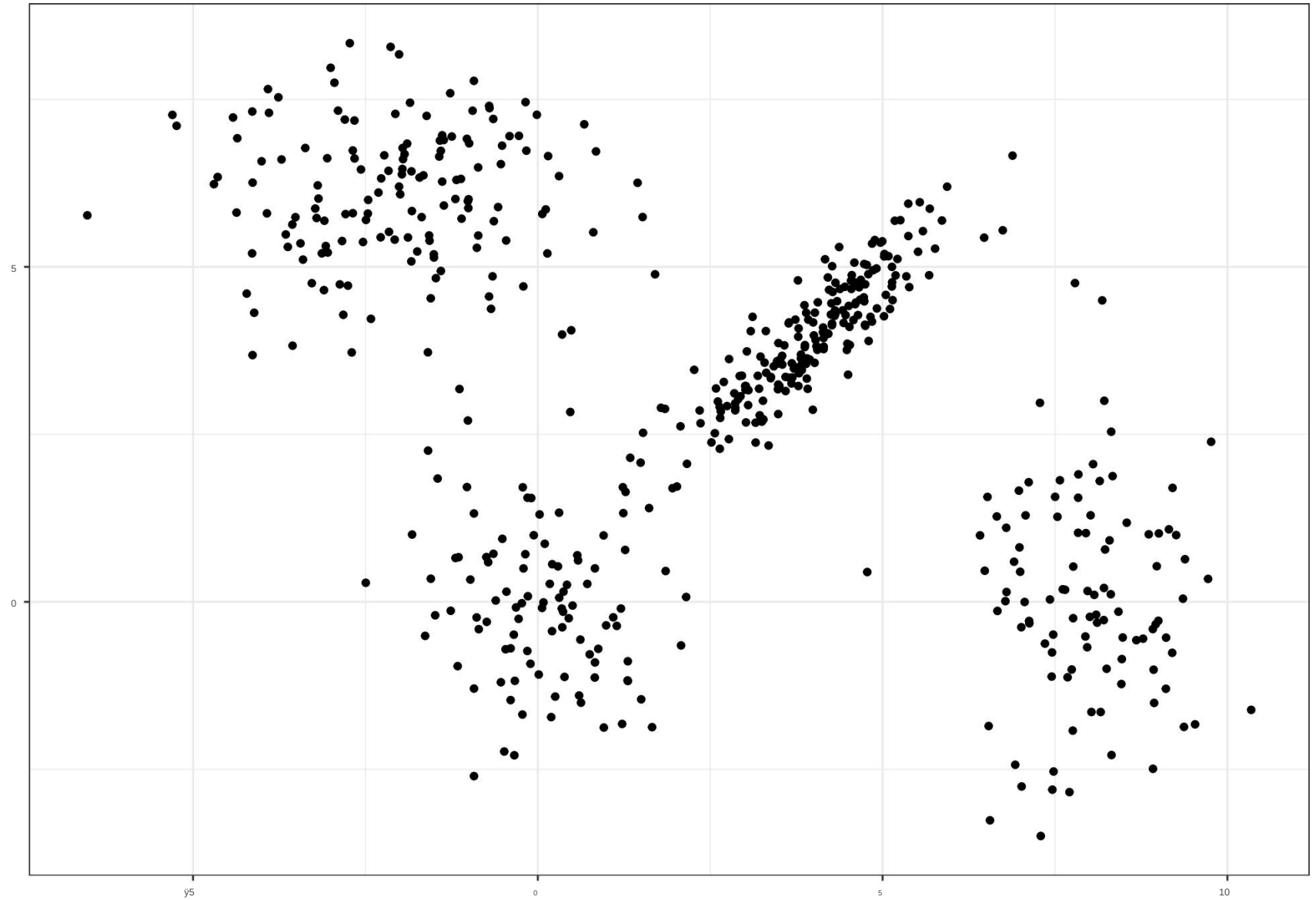
with

- **V** = (v1, . . . , vnc) the matrix of the cluster centers of the dimension p × nc and
- **U** = (uik) is an n × nc matrix of membership coefficients uik for observation xi to a cluster Ck .
- The Euclidean distance d measures the distance between Observations and cluster centers.

k-means. Remarks

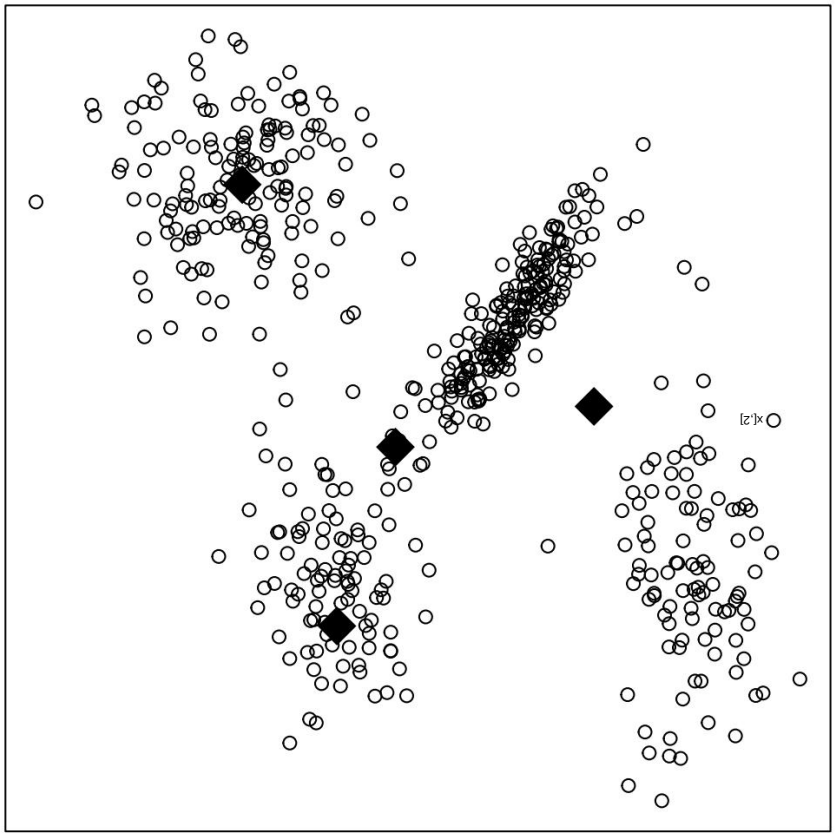
ÿ the E-Step is the estimation step (cluster centers are calculated), the M-Step is the assignment step. ÿ Iteration between E- and M-Step gradually improves the solution, **J(X,** *V,* U) becomes smaller with each iteration. ÿ the algorithm is very fast, can also be parallelized and is also suitable for relatively large data sets.

k-means. Functionality visual

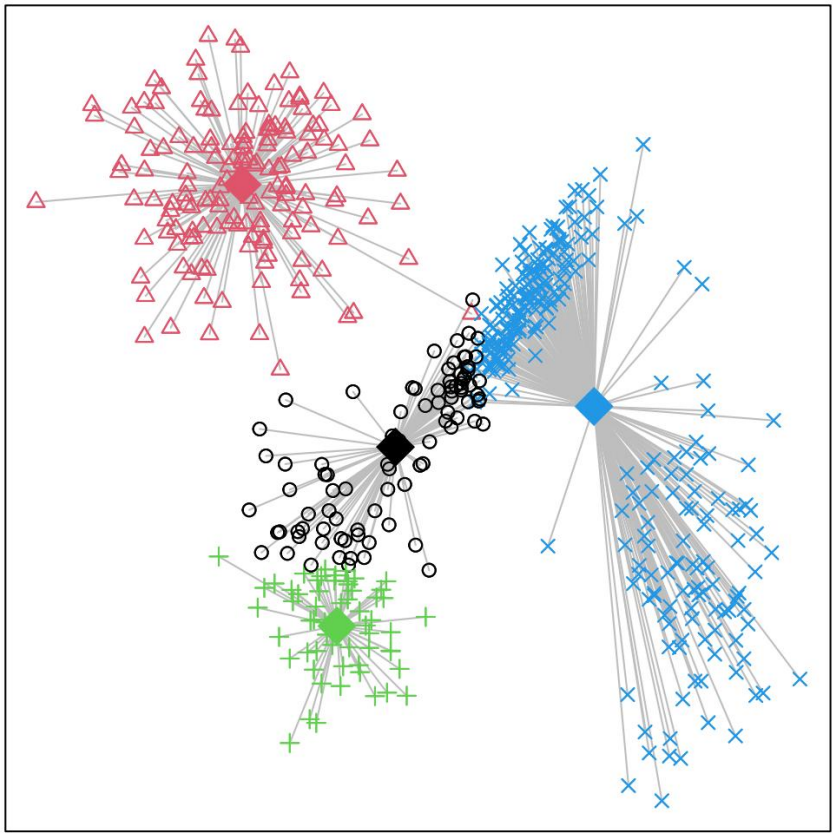


k-means. Functionality visual

Eÿstep (1)

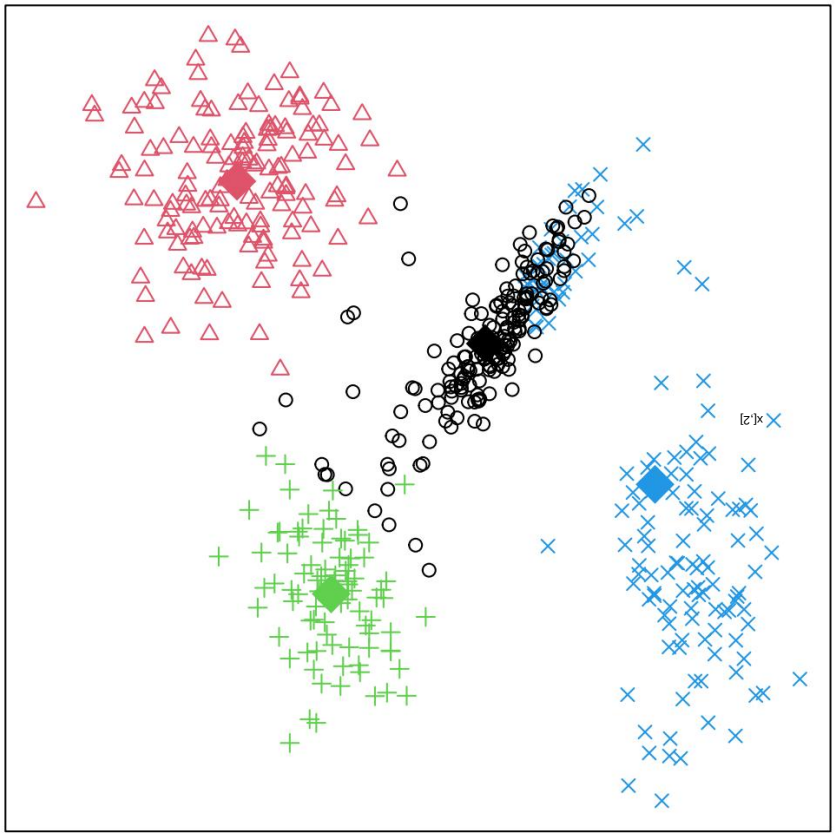


Mÿstep (1)

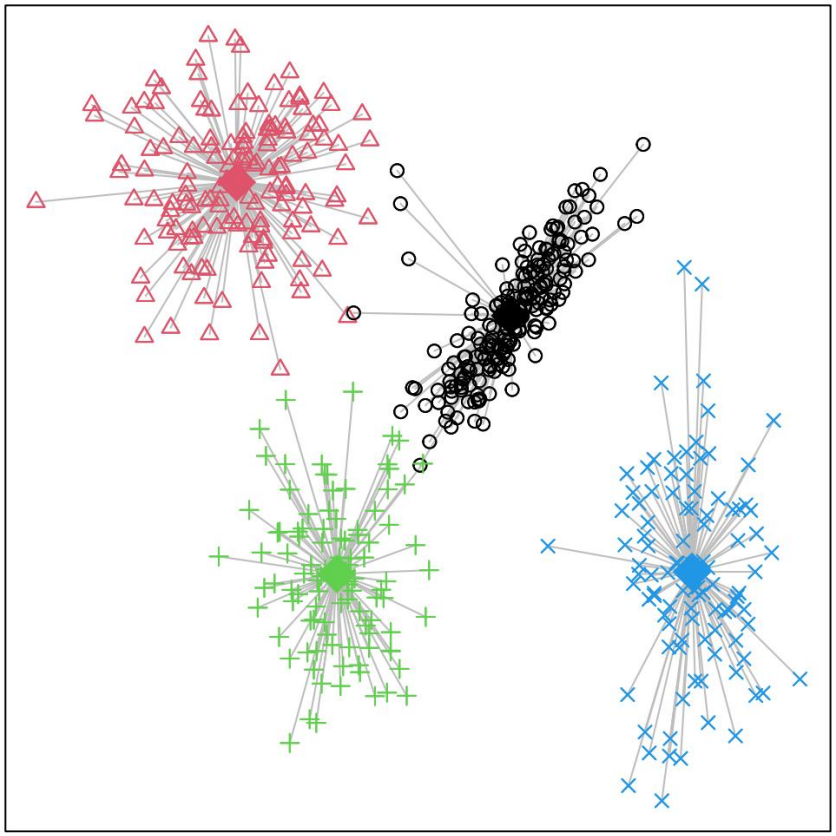


k-means. Functionality visual

Eÿstep (2)

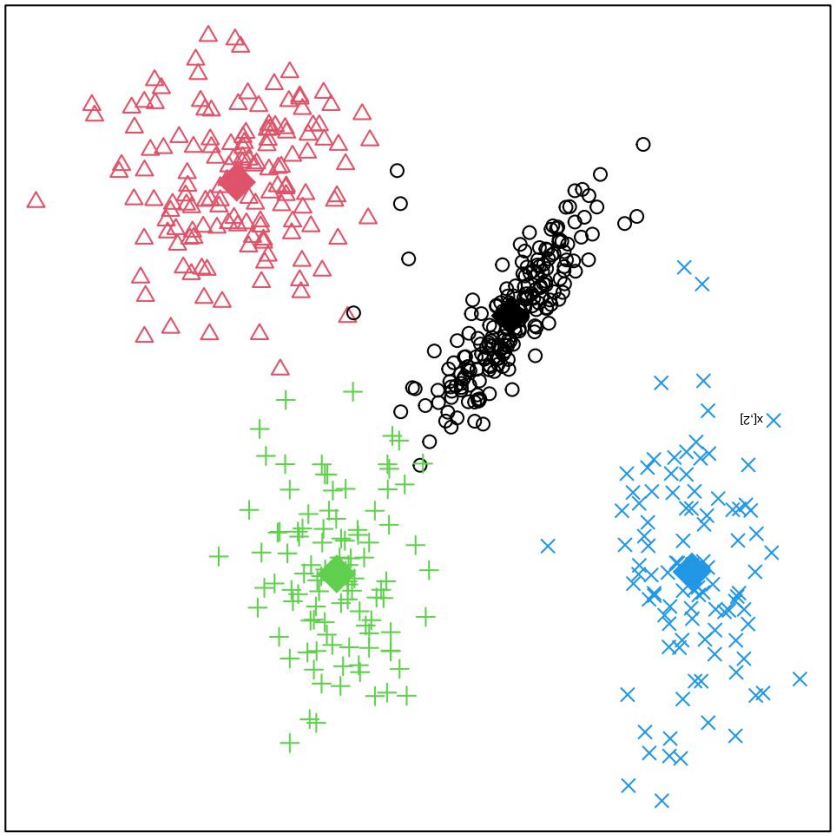


Mÿstep (2)

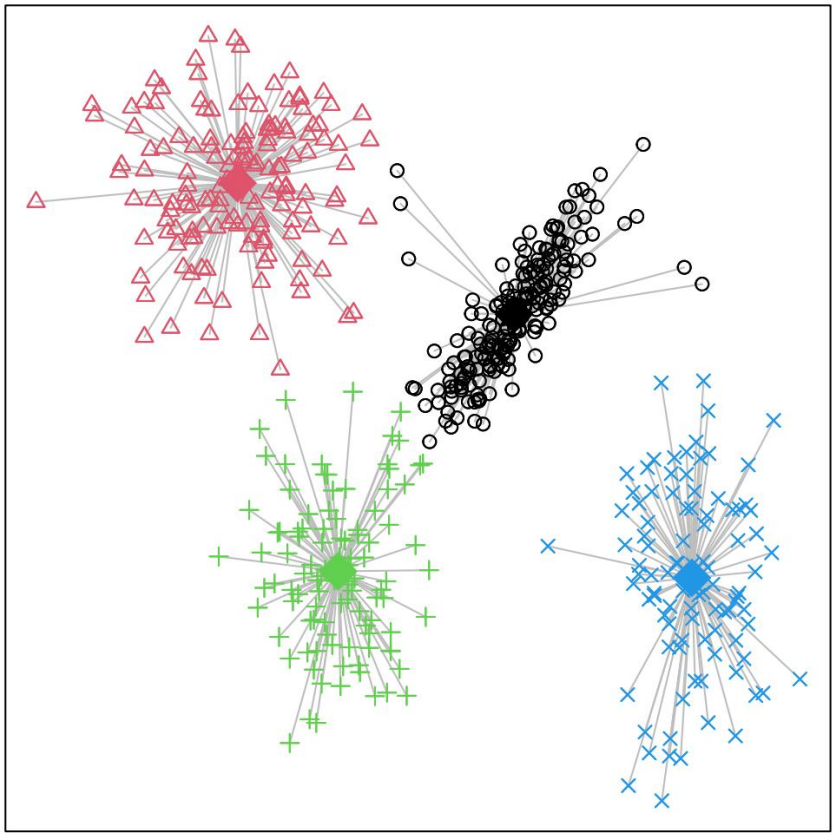


k-means. Functionality visual

Eÿstep (3, converged)



Mÿstep (3, converged)



k-means. modifications

ÿ Instead of Euclidean distances, other distances can be used be used (e.g. Manhattan). But then it is no longer called kmeans, but kmedians or cluster::pam (partioning around medoids) ÿ Very large data sets can be sampled cleverly (function cluster::clara - but this is a kmedians algorithm) ÿ The standard function in R is kmeans ÿ Lots of R packages offering kmeans

k-means. Disadvantages

kmeans is used very often, but it is a bad method.

ÿ Arithmetic means are highly non-robust. Better: kmedians - instead of artith. Means, use medians and instead of Euclidean distances, Manhattan distances, implemented eg in cluster::pam ÿ Outliers are also assigned to centers.

ÿ Way out: trimmed kmeans (eg in R packages tclust or trimcluster)

ÿ kmeans/ kmedians only assigns 0/1 memberships (in the respective cluster or not). ÿ Way out: fuzzy clustering (or model-based clustering) ÿ Based exclusively on distances, so one discovers (only) spherical structures!

ÿ Way out: model-based clustering

Your cooperation is required

all_exercises-cluster_nolsg.pdf

Clustering

- Task 4: kmeans clustering

Model-based Clustering

ÿ The theory of model-based clustering is very sprawling and in described in detail in the literature. We will only focus on practical aspects.

ÿ A statistical model is used to describe the shape of the clusters.

ÿ Standard form: Multivariate normal distribution, ie assumption: distribution of a cluster j has the density of a multivariate normal distribution, but we do not know µj and ÿj .

Given nc clusters from multiv. Normal densities with expectation µj and covariances ÿj (for j = *1, . . . ,* nc)

ÿ The cluster size in proportions are given as mixing coefficients ÿ1, . . . , ÿnc where ÿ1 + . . . + ÿnc = 1.

ÿ All these parameters *(µj ,* ÿj , ÿj) are calculated with the EM algorithm estimated.

Model-based Clustering

In the case of p variables, the covariance matrix of each individual cluster is of dimension p × p. If p is large (but also nc), a large number of parameters have to be estimated, which can lead to instabilities. Therefore, the cluster models are often simplified by restrictions.

The simplest constraint is

ÿ ÿj = $\ddot{y}^2$ I, for j = *1, . . . ,* where **I** is the identity matrix and nc , the $_{2p}$ parameter of variance.

ÿ ÿ all clusters are spherical with certain radii. Therefore 2 only needs more $\ddot{y}$ to be appreciated.

A less restricted covariance structure is

ÿ ÿj = $\ddot{y}_j^2$ ÿ In I, for j = *1, . . . ,* nc .

this case: Clusters are still spherical in nature, but whose size varies with respect to their variances $\ddot{y}_j^2$, which must be estimated .

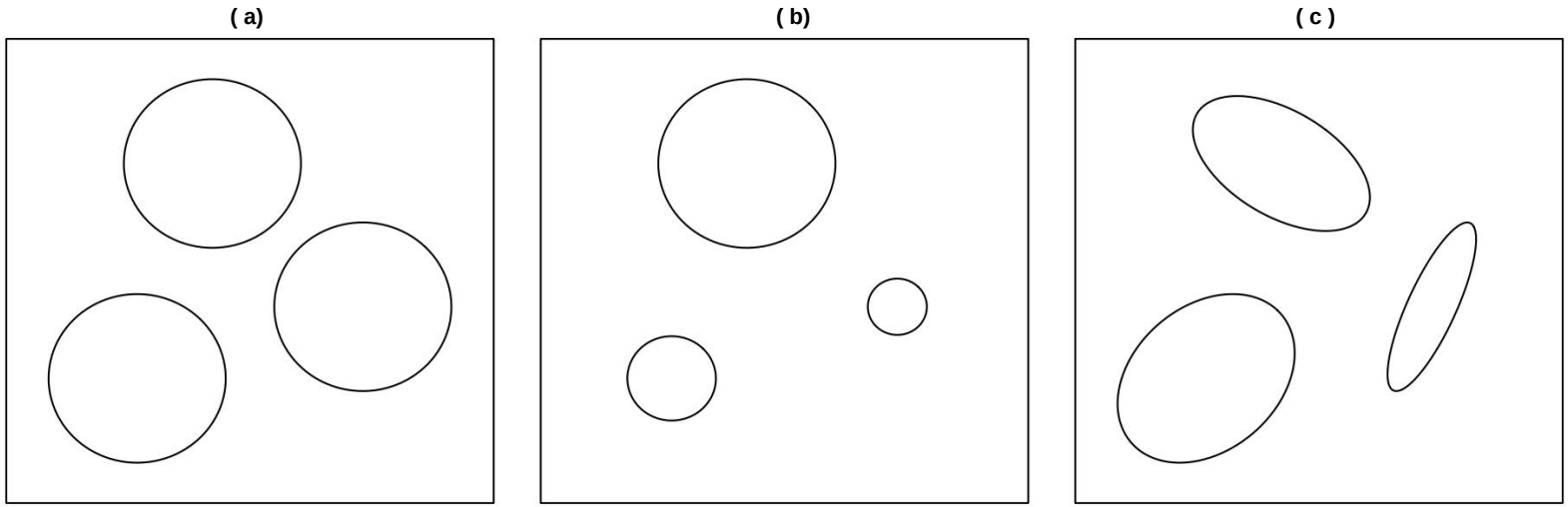
Different covariances

The following figure illustrates different covariance structures of the three clusters

a. ÿ1 = ÿ2 = ÿ3 = *ÿ* 2 ^{2} I;

b. ÿj = *ÿc* . $_{j}$ I, for j = *1, 2,* 3;

all ÿj are different and no special spherical ones
Structure



Optimal model

An optimal model can be determined using the BIC (Bayesian Information Criterion).

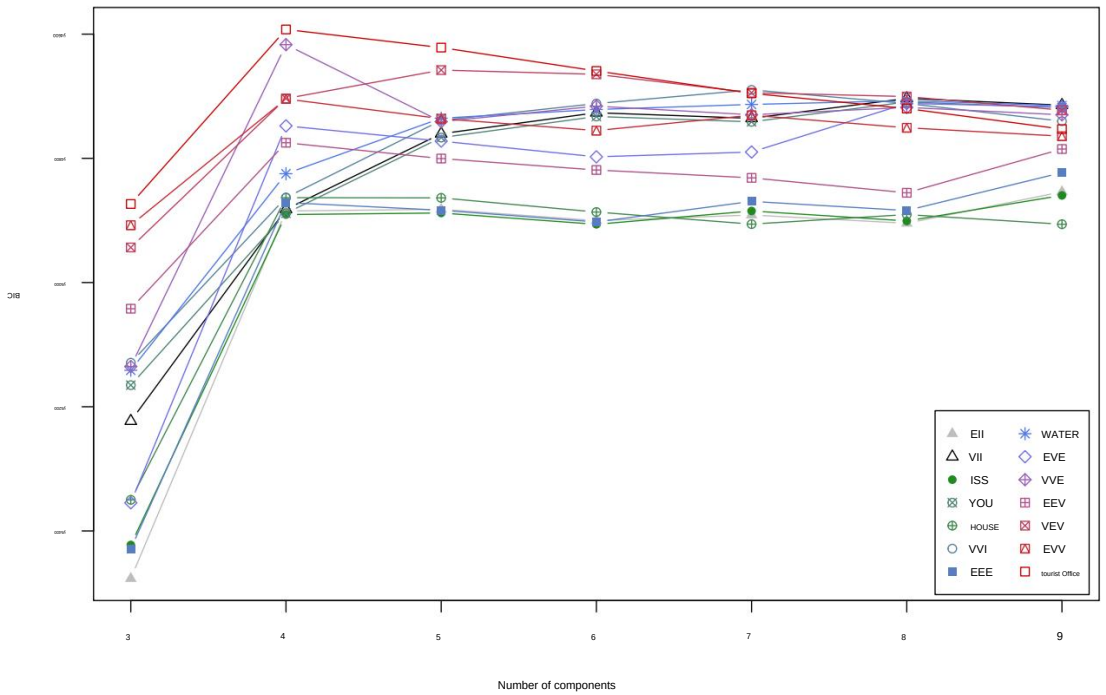
BIC:

- ÿ Theoretically, this would get too much out of hand to explain and derive it precisely
- ÿ ÿ 2 * log(avg likelihood of clusters) minus penalty from too many clusters.
  - ÿ Quite compact clusters are searched for ÿ However, this would result in clusters that are too small without penalty ÿ Optimal BIC: Balance between compactness and not too many clusters
- ÿ The bigger, the better.

Optimal model with BIC in R

Various cluster structures are searched for by default, see ? mclust::mclustModelNames (mclust must of course be installed once)

```
library("mclust"); data(Nclus, package = "flexclust") #
3 to 9 mixture components:
res <- Mclust(Nclus, G = 3:9, verbose = FALSE)
plot(res, what = "BIC")
```



Optimal model

ÿ Different covariance structures and different numbers

on mixing components were tested. ÿ Maximum

BIC indicates optimal model. ÿ 4 clusters seem optimal

ÿ The VVV model is the best model

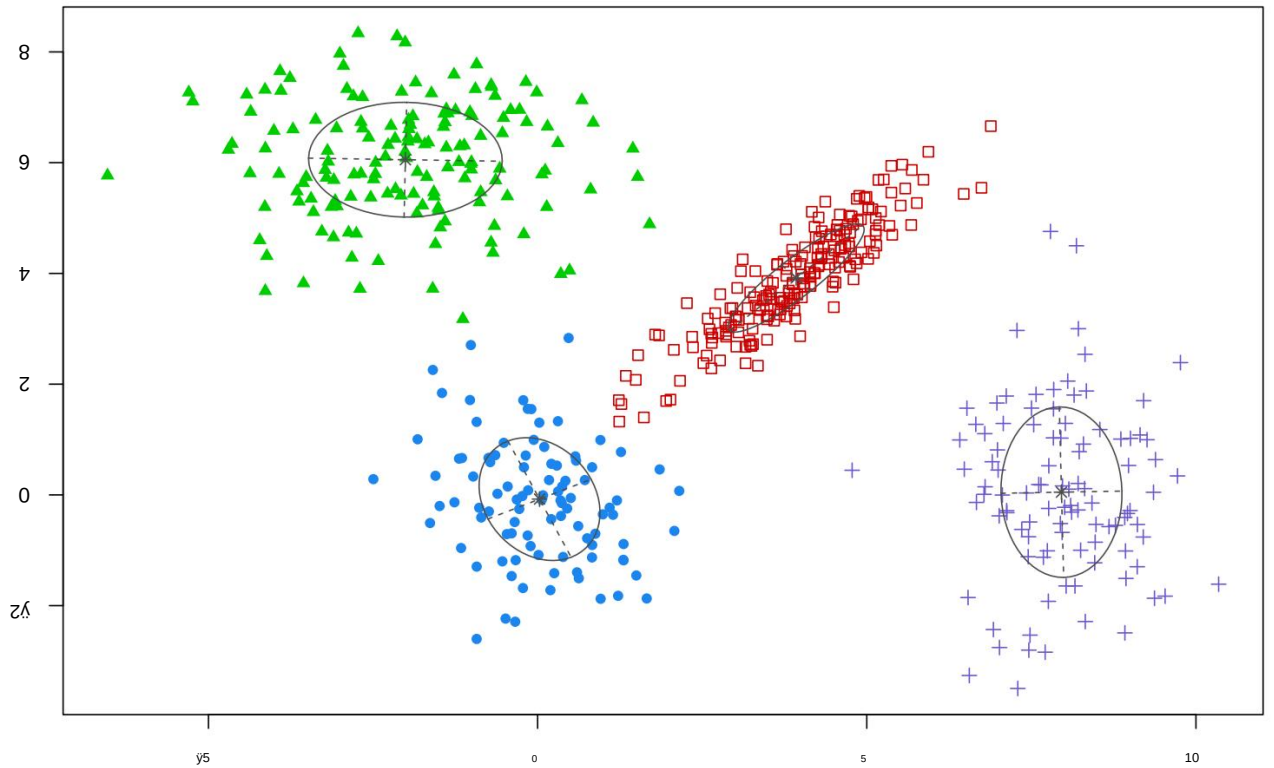
ÿ Explanations in ?mclustModelNames

(package mclust) ÿ “VVV" means that the covariances of the clusters are different (ellipsoidal, varying volume, shape, and orientation)

Optimal model

Result of the clustering including the covariance structure of the best Solution VVV:

```
plot(res, what = "classification")
```



Compared to kmeans

- In model-based clustering, the EM algorithm must also estimate the mixing coefficient and the covariance(s) in each step. For kmeans only the cluster means.
  So Mclust is much more computationally intensive.
- kmeans apparently assigned some of the oblong cluster observations to another cluster. Mclust does it better.
- kmeans recognizes more spherically symmetric clusters, Mclust is more flexible here.

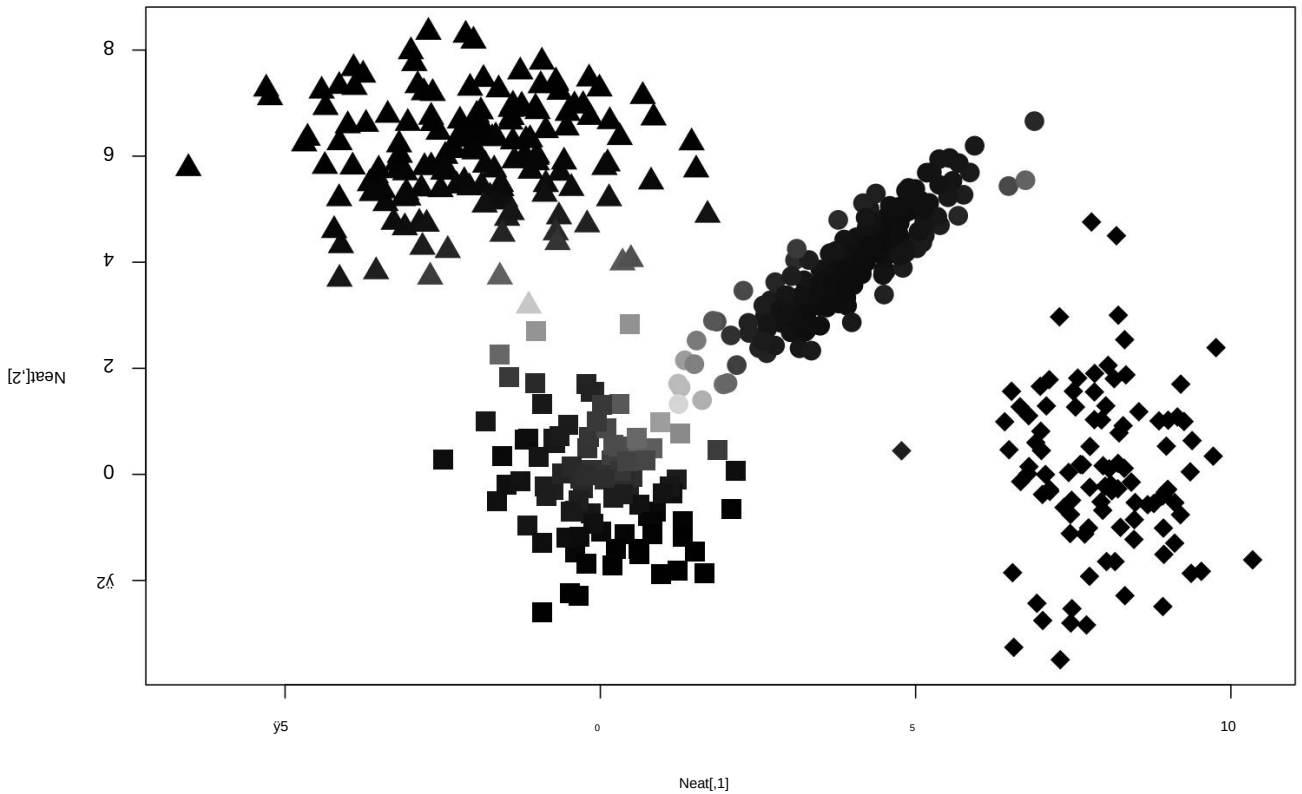
Even if this seems unknown in common practice (elsewhere):

If you do not cluster very large data, model-based clustering is definitely preferable to kmeans.

Comments Model-based clustering, uncertainty

ÿ Additional information: Allocation to clusters probabilistic

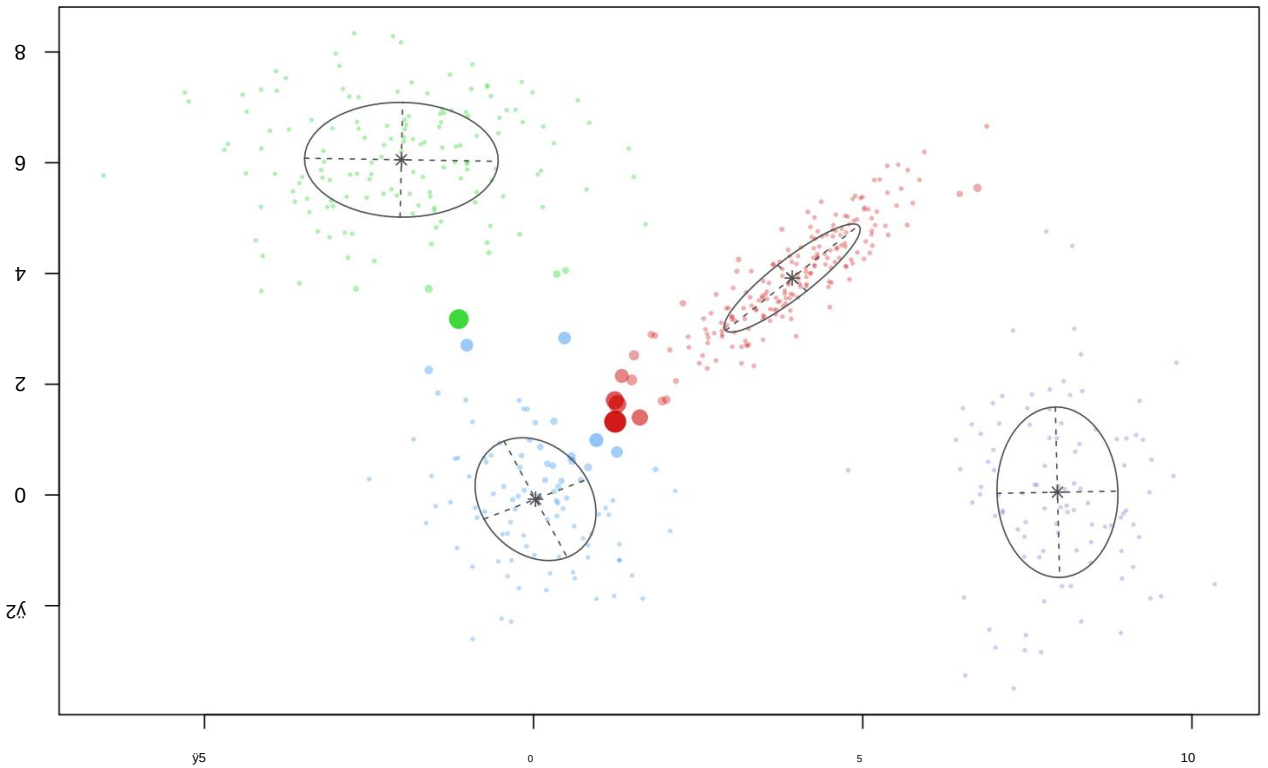
```
# see names(res)
plot(Nclus, col = grey((res$uncertainty)^(1/4)), pch =
        14+res$classification, cex = 2)
```



Comments Model-based clustering, uncertainty

Or easier with

```
plot(res, what = "uncertainty")
```



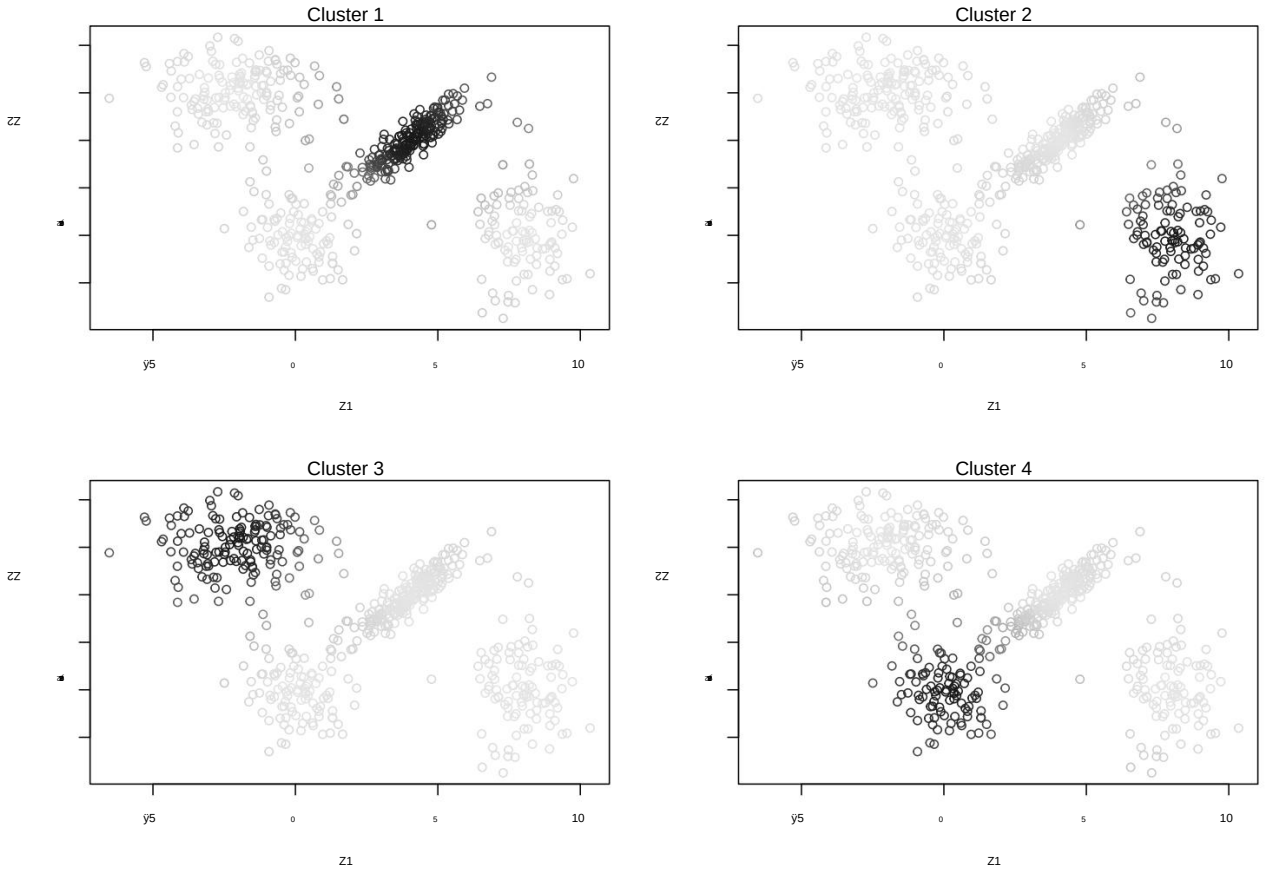
Fuzzy Clustering

We only want to reproduce the essential ideas here and largely do without theory.

ÿ Observations are assigned to all clusters. ÿ A membership coefficient uik expresses the degree of membership of the observation i to the k-th cluster (i = *1, . . . ,* n; k = *1, . . . ,* nc), with uik ÿ 0 and ui1 + . . . + uinc = 1, for all i. Note: for kmeans the uik are only 0 or 1. ÿ For a fixed number of clusters nc the solution is given by Minimization of an objective function found. ÿ The default implementations in R have numeric problems. The cmeans function from the e1017 package is therefore recommended. ÿ Attention: random start of the algorithm. New call may return different results.

Fuzzy Clustering

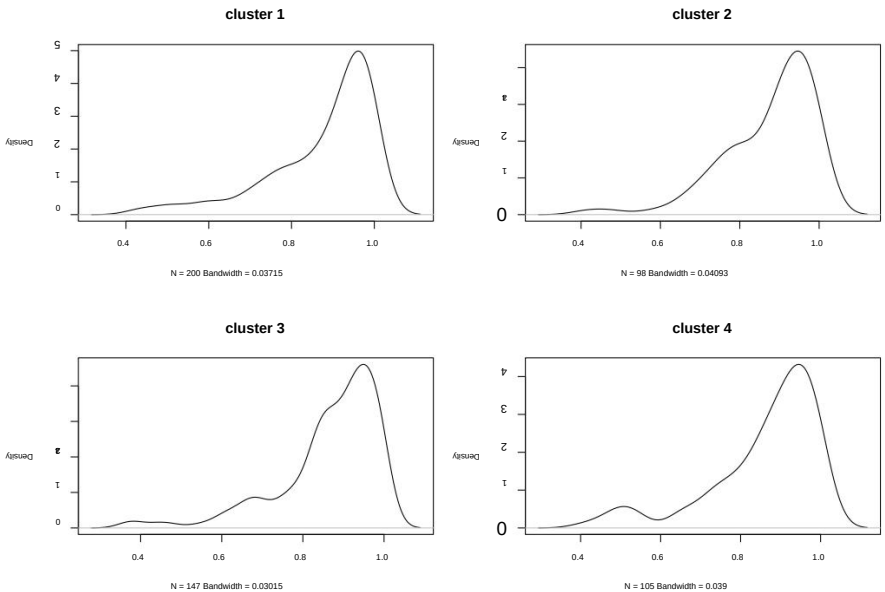
```
library("e1071"); groups <- 4 res <-
cmeans(Nclus, groups) for(i in
seq_along(1:groups)){ plot(Nclus, col
    = gray(1 - res$membership[, i]))}
```



Fuzzy Clustering

What about the quality of the clusters? A quick way: to visualize the membership distribution par(mfrow = c(2,2)) **for(i in** 1:4)

```
{ plot(density(res$membership[res$cluster == i,i]), main =
        paste("clusters", i))
}
```



Variablenclustern (Q-mode Clustering)

Instead of clustering the observations, cluster the variables.

- either one again measures distances as usual or uses the correlation between variables (more common)

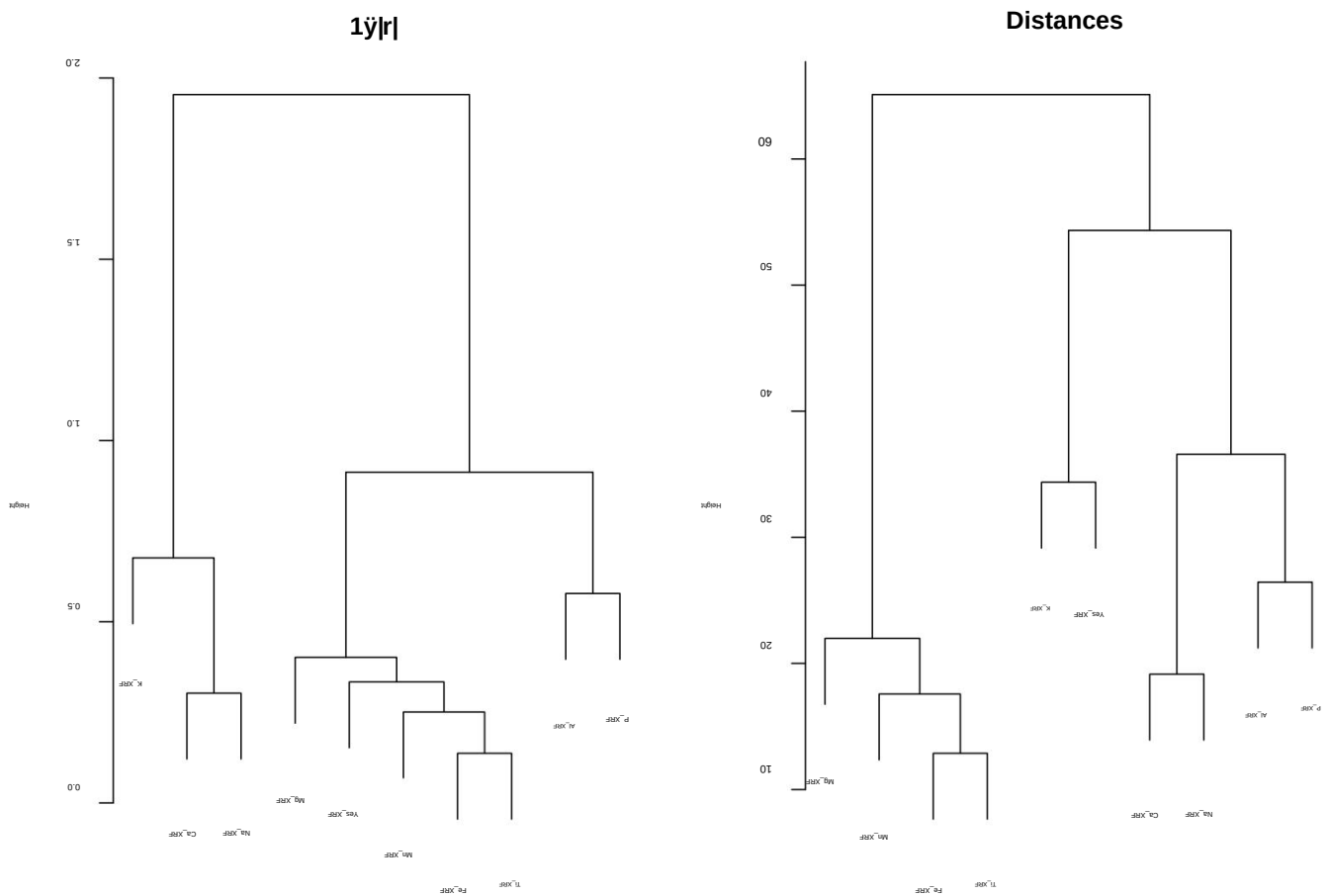
Variablenclustern (Q-mode Clustering)

Example Kola data (more details on the data later in **exercise 04_cluster_xkola)**

```
library(mvoutlier)
data(chorizon) ##
the (chemical) main elements, log-scaled: x <-
scale(log(chorizon[, 101:110]))
# Q-mode Clustering (1- |Korrelationen|) v.cor <- as.dist(1
- abs(cor(x)))
# about distances:
v.d <- dist(t(as.matrix(x)))
```

Variablenclustern (Q-mode Clustering)

```
par(mfrow = c(1,2), mar = c(0,4,3,2))
plot(hclust(v.cor,method="ward.D"),xlab="",main="1-|r|")
plot(hclust(v.d,method="ward.D"),xlab="",main="Distances")
```



Your cooperation is required

all_exercises-cluster_nolsg.pdf

Clustering

- ÿ Task 5: Fuzzy clustering ÿ
- Task 6: Model-based clustering ÿ
- Task 7: Another example of fuzzy and model-based clustering

Evaluation of the clusters

- How many clusters are optimal?
- Which cluster method?
- What transformation/standardization of the data?
- Which parameter settings for a cluster method?

Basically, it should be true that a cluster is as homogeneous as possible and the clusters are as heterogeneous as possible to one another.

Different types of quality criteria

We differentiate fundamentally

- ÿ **internal** cluster validation measure: you evaluate it
  Cluster result ÿ
- **external** cluster validation measure: one compares one
  Cluster result with known grouping ÿ **relative**
- cluster validation mass: one compares two
  Clusterresultate

Internal cluster validation measures include one or more of these criteria:

1. Compactness of a cluster (homogeneity): how close are obs. of a cluster to each other. The within-cluster mass.
2. Separation (heterogeneity): how separated is a cluster from the other clusters. The between-cluster mass.
3. Connectivity: Is the nearest neighbor of an observation in the same cluster?

Evaluation of the clusters. heterogeneity

One can measure heterogeneity in the following way

$$\text{Bnc} = \sum_{k=1}^{nc} \mathbf{ÿvk}\, ÿ\, v\bar{}ÿ2, \qquad (2)$$

with ÿ · ÿ the Euclidean norm, vk the k-th cluster center (k = *1, . . . ,* nc), and

$$v\bar{} = \frac{1}{nc} \sum_{k=1}^{nc} vk$$

the overall mean of the cluster centers.

This measure is known as between cluster sum of squares.

Evaluation of the clusters. homogeneity

Homogeneity in clusters is defined as

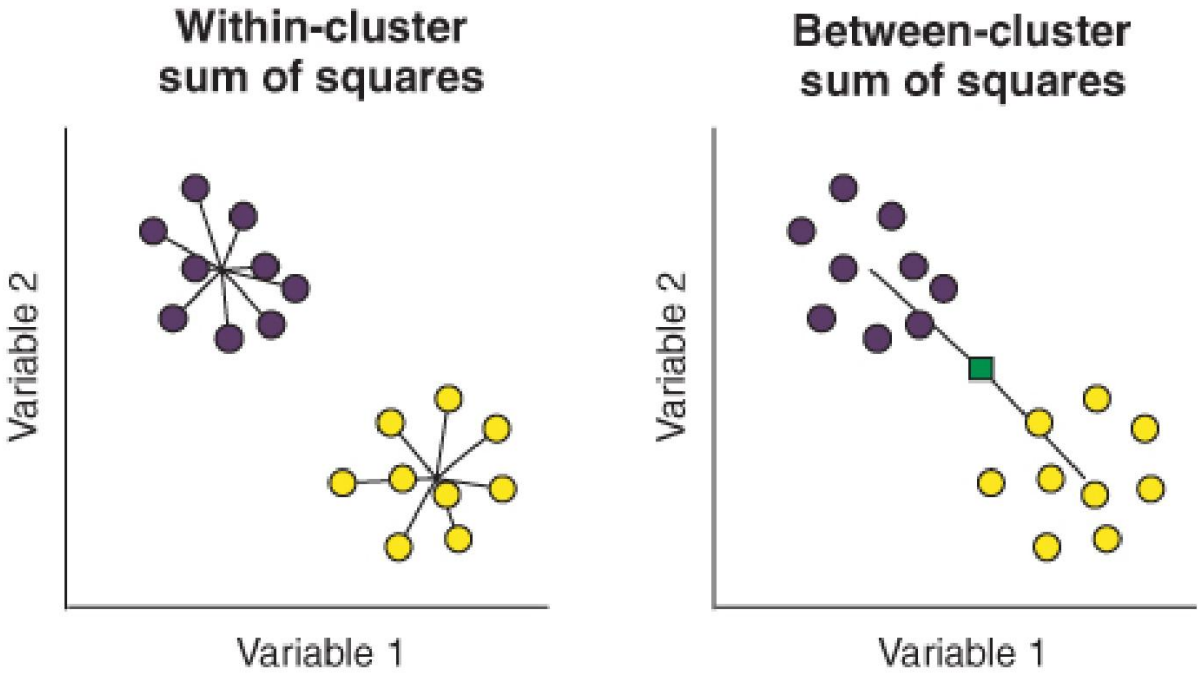
$$Wnc = \sum_{k=1}^{nc} \sum_{i \ddot{y} Ck} \ddot{y}xi \ddot{y} vk \ddot{y}^{2}, \qquad (3)$$

mit day , i = *1, . . . ,* n.

This measure is known as within cluster sum of squares because it takes into account Euclidean distances of the observations with their respective cluster centers.

- ÿ Bnc should be as large as possible ÿ
- Wnc should be as small as possible ÿ
- Both depend on the number of clusters (nc) to which should therefore be taken into account.

B and W visualized



(aus Machine Learning with R, the tidyverse, and mlr (Ryhs, 2020))

Evaluation of the clusters. Calinski-Harabasz Index

Calinski-Harabasz index : (Optimum: Max-Wert)

$$CHnc = \frac{Bnc\ /(nc\ ÿ\ 1)}{Wnc\ /(n\ ÿ\ nc\ )}$$

Hartigan index: (Optimum: knee)

$$Hnc = \ln \frac{Bnc}{Wnc}$$

Practice:

1. Try different number of clusters and
   Use cluster method
2. Calculate quality mass
3. Where the measure of goodness is best: optimal number of
   Cluster.

R implementations of good mass

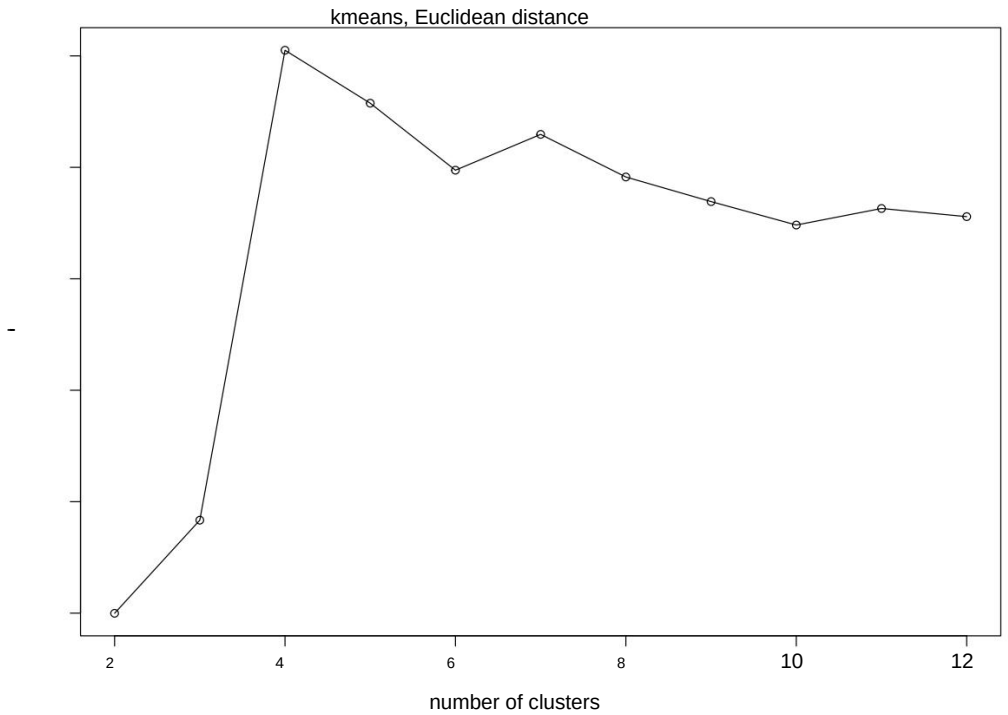
ÿ Package NbClust provides (many) measures of goodness, but only in terms of kmeans and some hierarchical clustering methods. ÿ Function fpc::cluster.stats also has (not quite as many) implemented measures of merit and it can be used in conjunction with (almost) all clustering methods. ÿ The clValid package provides some measures of merit for given clustering methods.

ÿ Package clusterSim, cclust, clv also offer quality criteria an.

Example of grade Hartigan and Calinski-Harabasz

Optimum number of clusters is 4 according to Calinski-Harabasz

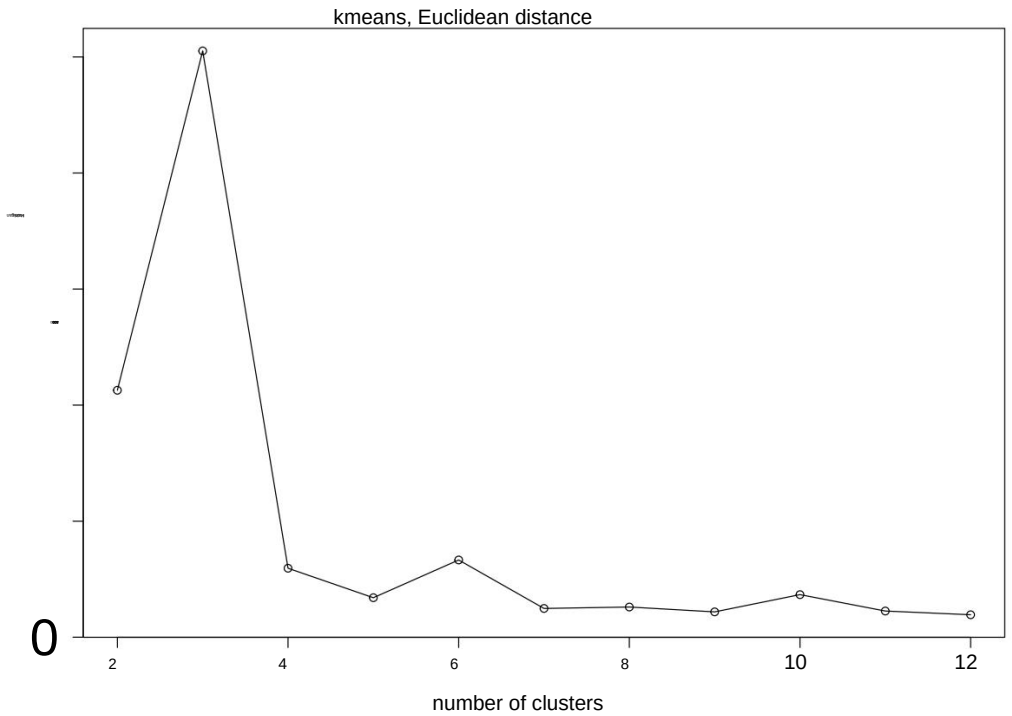
```
library(NbClust) stats <- NbClust(data = Nclus, min.nc = 2, max.nc =
                                  12, method = "kmeans", index = "ch")
plot(2:12, stats$All.index, type = "o",
       xlab = "number of clusters", ylab = "CH index")
```



Example of grade Hartigan and Calinski-Harabasz

Optimum number of clusters is 4 according to Hartigan

```
library(NbClust) stats <- NbClust(data = Nclus, min.nc = 2, max.nc =
                                  12, method = "kmeans", index = "hartigan")
plot(2:12, stats$All.index, type = "o", xlab =
       "Anzahl Cluster", ylab = "Hartigan Index")
```



Evaluation of the clusters. Silhouette Widths

Average dissimilarity of an observation xi belonging to the cluster Ck to all other observations **of the same** cluster:

$$di,Ck = \frac{1}{n(k) \ddot{y} 1} \sum_{i,j \ddot{y} Ck ,i \ddot{y}=j} 2\,d\,(\mathbf{xi}\ ,\ xj),$$

where n(k) is the number of observations in cluster Ck .

Average dissimilarity of xi to observations of **another** Clusters Cl :

$$of,Cl = d\ n(\sout{l)\ j \ddot{y} Cl} \quad 1 \quad 2 \quad (\mathbf{xi}\ ,\ xj).$$

Evaluation of the clusters. Average Silhouette Width

The smallest of these values is:

$$di,C = \text{mean}_{I} \; di,Cl \;,$$

thus has smallest dissimilarity of the i-th observations, not to its own, but to its **closest** of the remaining clusters.

The silhouette value is defined as:

$$\text{and} = \frac{in,C \text{ ÿ } in,Ck}{\max(\text{in,Ck} \;,\; in,C)}$$

ÿ this is normalized to [ÿ1, 1] ÿ If si close to 1: well classified ÿ If si close to 0: observation lies between 2 clusters ÿ If si close to -1: no good classification of the observation ÿ Negative si : observation is possibly in wrong cluster assigned

Evaluation of the clusters. Average Silhouette Width

The **Average Silhouette Width** is:

$$\frac{1}{n}\sum_{i=1}^{n} \quad \text{and ,}$$

The higher the value, the better the clustering.

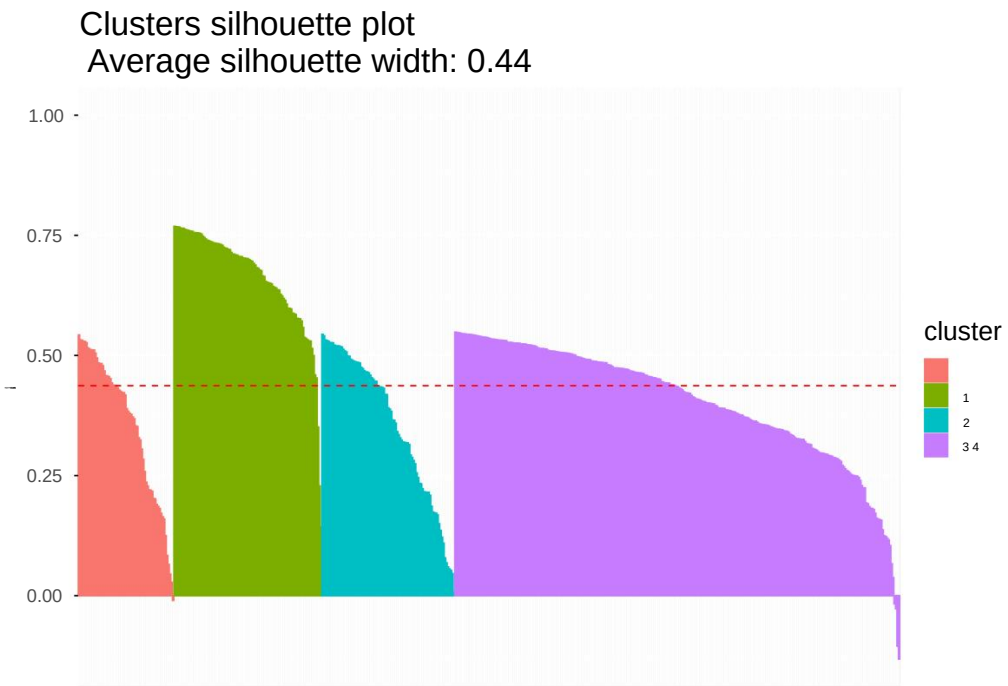
(very rough) rule of thumb:

- ÿ no cluster structure below 0.25 ÿ weak cluster structure between 0.25 and 0.5 ÿ good cluster structure between 0.5 and 0.75 ÿ very good cluster structure above 0.75

Example of Silhouette Widths

Eg with package factoextra and cluster::silhouette. Good clustering, regarding all clusters.

```
library(factoextra); library(cluster) cl1 <-
kmeans(Nclus, centers = 4) sil <-
silhouette(cl1$cluster, dist(Nclus)) fviz_silhouette(sil,
print.summary = FALSE)
```



Comment Silhouette Widths

ÿ Input (in silhouette or formula before) is the distance matrix and the classification of the observations into clusters. ÿ It is about **Euclidean distances** of an observation to all observations within the same cluster and to all observations of other clusters.

ÿ The measure is therefore not suitable for model-based clustering if elliptical structures are assessed using different covariances.

Comments on masses of goodness for practice

Apart from embellished (2-dimensional) teaching examples (= toy data), it is often difficult in practice to find the optimal number of clusters or to evaluate a clustering.

- ÿ Data with noise (this is the practice): ÿ
  - global measures of quality are to be taken with great care. Often
    - if you specify too few clusters ÿ
  - relative measures of quality are also affected here ÿ
  - often better than only individual ones with local measures of quality
    - Rate clusters
- ÿ It is often useful to look at the cluster results visually to assess whether clusters have been missed. . . Eg by parallel coordinate plot or Grand Tours or Guided Tours.

Therefore, we will look at a more complex example, which is also more complex than you (for examination) require (ÿ 04_cluster_kola.pdf)

Your cooperation is required

all_exercises-cluster_nolsg.pdf

Clustering

- Task 8: Validity of clusters

External good mass

External: Here you want to compare a cluster solution with a known partition. In other words, one has information about groups and evaluates to what extent this partition coincides with the cluster result. Let's assume iris[, 5] contains the **true partition.**

```
set.seed(123) cl <-
kmeans(iris[, 1:4], 3, nstart = 10) table(cl$cluster, iris$Species)
```

```
##
##          setosa versicolor virginica 0
## 1        50                0
## 2        0                 48                14
## 3        0                 2                 36
```

We see some misclassifications (2 and 14).

Externe Gütemasse: Corrected Rand Index

ÿ Corrected Rand Index measures a similarity between 2 partitions.

ÿ The correction is relatively complex, we do not want it here
carry out.

ÿ This index is between -1 (no match) to 1
(cluster division and partition match 100%)

```
library(fpc) stats
<- cluster.stats(dist(iris[, 1:4]), clustering = cl$cluster,
    alt.clustering =
    as.integer(iris$Species))
stats$corrected.rand
```

```
## [1] 0.7302383
```

Comment Clustering / Classification

ÿ With unsupervised learning methods (MDA, clustering, guided tours, . . .) there were NO target variables ÿ unsupervised learning, unsupervised groups were sought.

ÿ With methods of classification, there is a target variable which is classified ÿ supervised learning

ÿ In practice, clustering can be a preliminary step to classification if the target variable has to be found first

- ÿ Find groups using cluster analysis
- ÿ Train a classification method on these groups (= target variable) to classify future observations.

Summary of the most important functions in R

ÿ **dist** or **factoextra::get_dist** for distance calculation,
factoextra::fviz_dist for visualization of distances ÿ
hclust, cutree and **plot.hclust** for hierarchical clustering ÿ
kmeans, cluster::pam, cluster::clara for k-means and k-medians methods .

ÿ **e1071::cmeans** for fuzzy clustering ÿ
mclust::Mclust and **mclust::plot.Mclust** for model-based clusters

ÿ **NbClust::NbClust** i and **fpc::clusterstats** for quality mass ÿ **cluster::silhouette** and **factoextra::fviz_silhouette**