CLUSTER ANALYSIS

1 What is Cluster Analysis?

Cluster Analysis = searching for groups (clusters) in data, in such a way that objects of the same cluster resemble each other, whereas objects in different clusters are dissimilar.

- 2 or 3 dimensions: clusters can be recognized by eye.
- Otherwise: some kind of computer algorithms needed.

Partitioning Algorithms: methods that divide the data set into k clusters, where the integer k needs to be specified by the user.
Examples: `pam`, `clara`, and `fanny`.

Hierarchical Algorithms: methods yielding an entire hierarchy of clusterings of the data set.
Examples: `agnes`, `diana`, and `mona`.

Data sets for clustering can have either of the following structures:

1. $n \times p$ data matrix:
$$\begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix}$$
where rows stand for objects and columns stand for variables.

2. $n \times n$ dissimilarity matrix:
$$\begin{pmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & & & \ddots & \\ d(n,1) & & \cdots & & 0 \end{pmatrix}$$
where $d(i,j) = d(j,i)$ measures the "difference" or *dissimilarity* between the objects i and j.

2 Dissimilarity Matrices

2.1 Definition

The dissimilarity between two objects measures "how different" they are.

But : dissimilarity $\neq$ metric.

Often only the following 3 axioms of a metric are satisfied:

- $d(i,i) = 0$,
- $d(i,j) \geq 0$,
- $d(i,j) = d(j,i)$,

but not the triangle inequality.

2.2 Computation

depends on the type of the original variables.

INTERVAL-SCALED VARIABLES

= continuous measurements on a (roughly) linear scale.
Examples: temperature, height, weight, ...

In this case: dissimilarity = actual metric.
Examples:

$$d(i,j) = \sqrt{\sum_{f=1}^{p} (x_{if} - x_{jf})^2} \quad \text{(Euclidean distance)}$$

or

$$d(i,j) = \sum_{f=1}^{p} |x_{if} - x_{jf}| \quad \text{(Manhattan distance).}$$

Note: the choice of measurement units strongly affects the resulting clustering. The variable with the largest dispersion will have the largest impact on the clustering. If all variables are considered equally important, the data need to be standardized first:

$$z_{if} = \frac{x_{if} - m_f}{s_f} \qquad \text{(z-scores)}$$

where

$$m_f = \frac{1}{n} \sum_{i=1}^{n} x_{if} \qquad \text{and} \qquad s_f = \frac{1}{n} \sum_{i=1}^{n} |x_{if} - m_f| \ .$$

s_f = *mean absolute deviation* $\Rightarrow$ more robust than usual standard deviation.

Example. In Figure 1, age and height of 4 people are plotted, height is measured in centimeters in Figure 1a, and in feet in Figure 1b. Both variables are standardized in Figure 1c.

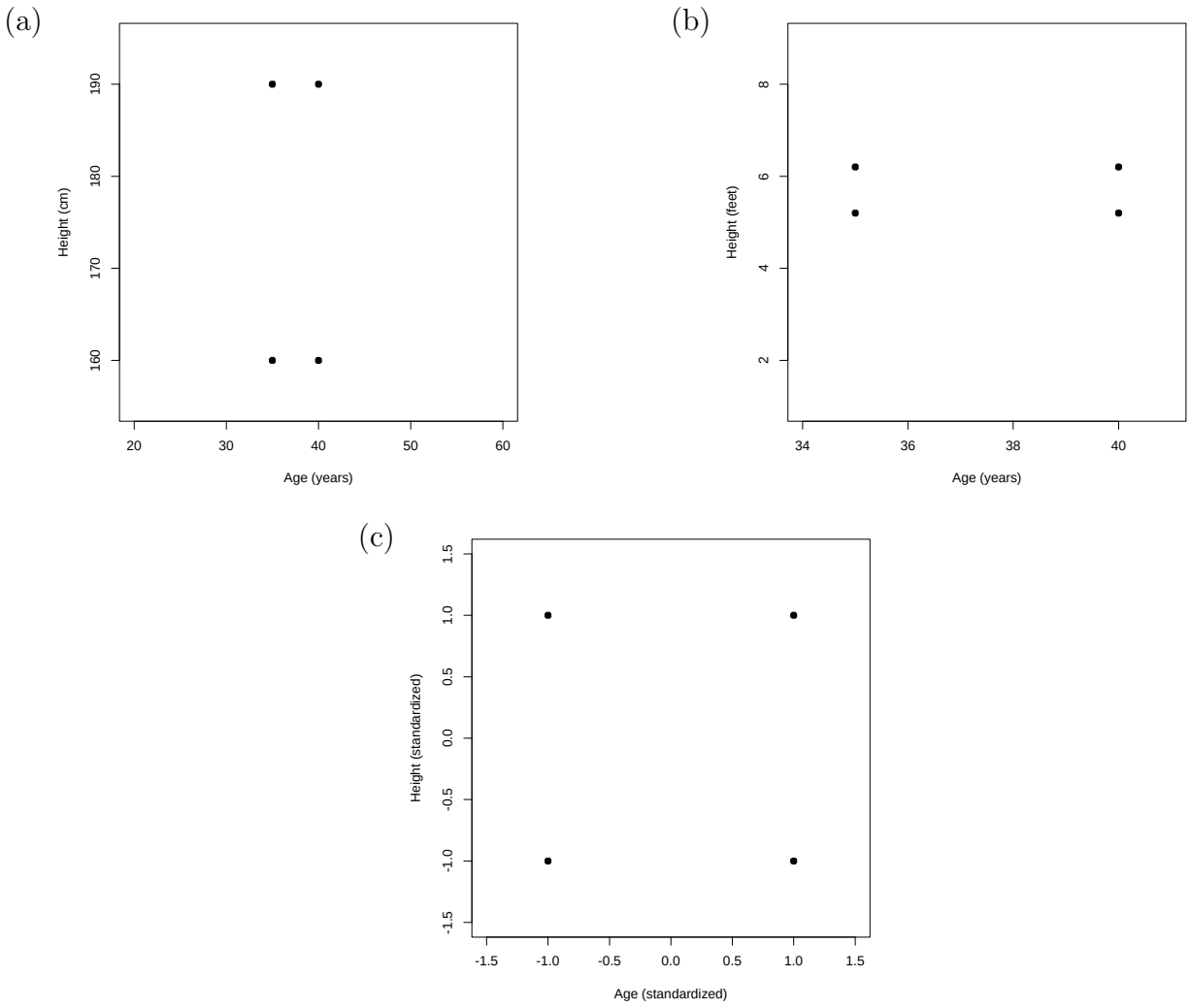


Figure 1: Effect of measurement units on clustering structure.

CONTINUOUS ORDINAL VARIABLES

= continuous measurements on an unknown scale, or such that only the ordering is known but not the actual magnitude.

1. Replace the x_{if} by their rank $r_{if} \in \{1, \ldots, M_f\}$.
2. Transform the scale to $[0, 1]$ as follows:
$$z_{if} = \frac{r_{if} - 1}{M_f - 1}.$$
3. Compute the dissimilarities as with interval-scaled variables.

RATIO-SCALED VARIABLES

= positive continuous measurements on a nonlinear scale, e.g. an exponential scale.
Example: growth of a bacterial population (growth function $f(t) = Ae^{Bt}$), equal time intervals multiply the population by the same ratio.

- Treat as interval-scaled variables: not recommended because this distorts the measurement scale.
- Treat as continuous ordinal data.
- First transform the data, e.g. by taking logarithms, and treat the result as interval-scaled.

DISCRETE ORDINAL VARIABLES

= M possible values (scores), which are ordered. ($M \in \mathbb{N}$)

Dissimilarities are computed in the same way as for continuous ordinal variables.

NOMINAL VARIABLES

= M possible values, which are not ordered. ($M \in \mathbb{N}$)

$$d(i,j) = \frac{\#\text{variables taking different values for } i \text{ and } j}{\text{total number of variables}}.$$

This is called the *simple matching coefficient*.

SYMMETRIC BINARY VARIABLES

= 2 possible values, coded 0 and 1, which are equally important.
Example: male / female.

Symmetric binary variables are nominal variables, hence we again use the simple matching coefficient. Let us also consider the contingency table of the objects i and j:

$i \backslash j$	1	0
1	a	b
0	c	d

We can then rewrite the simple matching coefficient as:

$$d(i,j) = \frac{b+c}{a+b+c+d}.$$

ASYMMETRIC BINARY VARIABLES

= 2 possible values, one of which carries more importance than the other. The most meaningful outcome is coded as 1, and the less meaningful outcome as 0. Typically, 1 stands for the presence of a certain attribute (e.g. a particular disease), and 0 for its absence.

$$d(i,j) = \frac{\#\text{variables taking different values for } i \text{ and } j}{\text{total number of meaningful comparisons}}.$$

Using the contingency table again, this becomes

$$d(i,j) = \frac{b+c}{a+b+c},$$

which is called the *Jaccard coefficient*.

VARIABLES OF MIXED TYPES

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}} \qquad \in [0,1]$$

where

$d_{ij}^{(f)}$ = contribution of variable f to $d(i,j)$, which depends on its type:

- f binary or nominal: $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$, and $d_{ij}^{(f)} = 1$ otherwise,

- f interval-scaled: $d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{\max_h x_{hf} - \min_h x_{hf}}$,
- f ordinal or ratio-scaled: compute ranks r_{if} and $z_{if} = \frac{r_{if}-1}{M_f-1}$ and treat these z_{if} as interval-scaled,

and

$\delta_{ij}^{(f)}$ = weight of variable f:

- $\delta_{ij}^{(f)} = 0$ if x_{if} or x_{jf} is missing,
- $\delta_{ij}^{(f)} = 0$ if $x_{if} = x_{jf} = 0$ and variable f is asymmetric binary,
- $\delta_{ij}^{(f)} = 1$ otherwise.

Example. 15 countries of the EU in 1994, 2 interval-scaled variables: the gross national product and the percentage of the people employed in agriculture (see also Sections 3 and 5).

```
  B  0
 DK  5.2  0
  D  2.2  3.1  0
 GR 21.2 22.6 22.1  0
  E 11.1 14.3 12.7 11.2  0
  F  2.4  4.3  2.5 19.6 10.2  0
IRL 12.1 13.9 13.1  9.1  3.8 10.6  0
  I  6.4  9.4  7.8 14.9  4.9  5.3  5.9  0
  L  9.8  5.3  7.6 27.5 19.6  9.5 19.1 14.7  0
 NL  1.4  5.8  3.2 19.8  9.7  1.8 10.7  5.0 10.7  0
  A  4.4  3.6  3.6 19.0 10.7  2.1 10.3  6.0  8.9  3.9  0
  P 14.5 17.7 16.1  9.0  3.5 13.7  5.2  8.4 23.0 13.2 14.1  0
FIN  6.4  8.4  7.3 14.9  5.9  4.7  5.8  1.7 13.7  5.0  4.8  9.3 0
  S  0.5  5.2  2.3 20.7 10.7  1.9 11.6  5.9  9.9  1.0  3.9 14.2 5.9 0
 UK  4.4  9.6  6.6 20.0  8.9  5.7 11.2  5.6 14.1  3.9  7.7 12.1 6.7 4.4 0
     B    DK   D    GR   E    F    IRL  I    L    NL   A    P    FIN S   UK
```

3 Partitioning Around Medoids (pam)

Idea:

1. Compute k *representative objects*, called *medoids*, $m_1, \ldots, m_k$, one for each cluster. They should minimize the sum of the dissimilarities of all objects i to their nearest medoid:

$$\text{objective function} = \sum_{i=1}^{n} \min_{t=1,\ldots,k} d(i, m_t). \tag{3.1}$$

2. Assign each object to the cluster corresponding to the nearest medoid. That is, object i is put into cluster v_i when medoid m_{v_i} is nearer than any other medoid m_w :

$$d(i, m_{v_i}) \leq d(i, m_w) \text{ for all } w = 1, \ldots, k.$$

The algorithm proceeds in two steps:

1. Step BUILD.

   Construct initial 'medoids':

   - m_1 is the object with the smallest $\sum_{i=1}^{n} d(i, m_1)$;
   - m_2 decreases the objective (3.1) as much as possible;

     $\vdots$

   - m_k decreases the objective (3.1) as much as possible.

2. Step SWAP.

   Repeat until convergence:

Example. Agriculture data, 2 clusters.

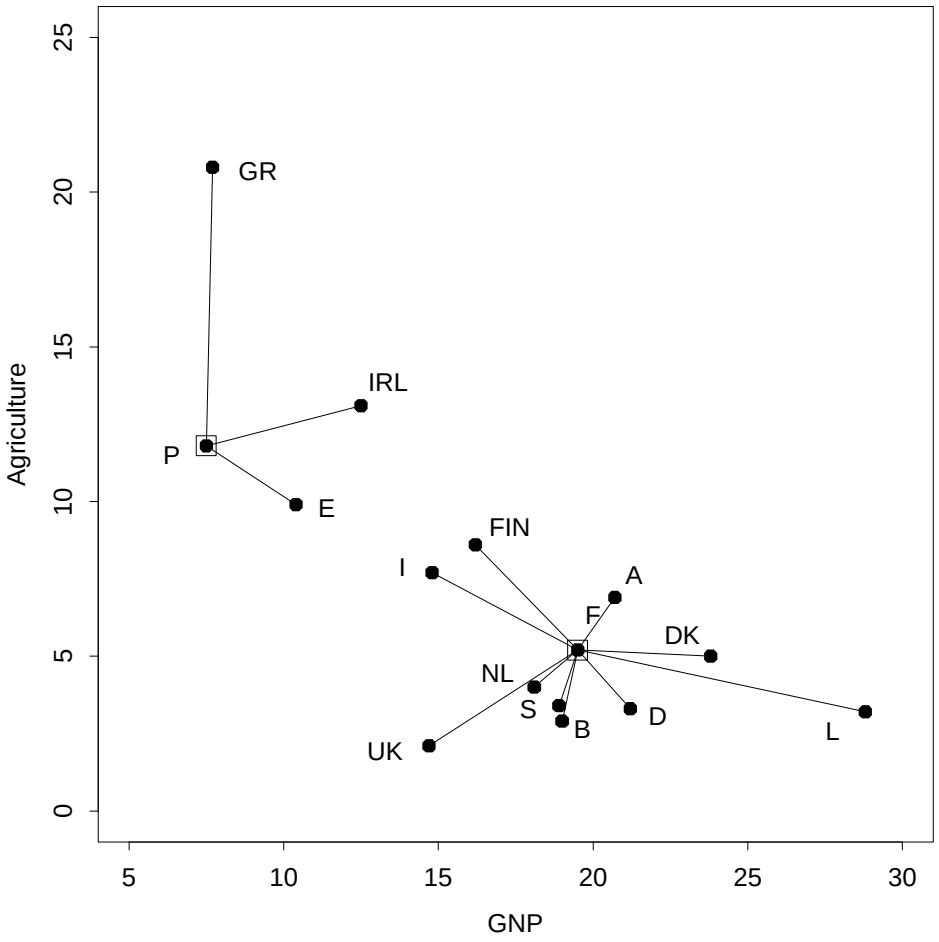


Figure 2: Agriculture data set about 15 European countries. Applying the function `pam` with $k = 2$ yields two medoid objects, indicated by squares. Each object is then assigned to its nearest medoid, yielding clusters with 4 and 11 objects.

Consider all pairs of objects (i, j) with

$$i \in \{m_1, \ldots, m_k\} \qquad \text{and} \qquad j \notin \{m_1, \ldots, m_k\},$$

and make the $i \leftrightarrow j$ swap (if any) which decreases the objective most.

Graphical representation: *silhouette plot.*

For each object i the silhouette value $s(i)$ is computed, showing how well object i belongs to cluster v_i:

1. For each object i we denote by A the cluster to which it belongs, and compute

$$\begin{aligned} a(i) &:= \frac{1}{|A|-1} \sum_{j \in A, j \neq i} d(i,j) \\ &= \text{average dissimilarity of } i \text{ to all other objects of } A. \end{aligned} \tag{3.2}$$

2. Now consider any cluster C different from A and put

$$\begin{aligned} d(i,C) &:= \frac{1}{|C|} \sum_{j \in C} d(i,j) \\ &= \text{average dissimilarity of } i \text{ to all objects of } C. \end{aligned} \tag{3.3}$$

After computing $d(i, C)$ for all clusters $C \neq A$ we take the smallest of those:

$$b(i) := \min_{C \neq A} d(i, C).$$

The cluster B which attains this minimum [that is, $d(i, B) = b(i)$] is called the *neighbor* of object i. This is the second-best cluster for object i.

3. The *silhouette value* $s(i)$ of the object i is defined as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}.$$

Clearly $s(i) \in [-1, 1]$.

The value $s(i)$ may be interpreted as follows:

$s(i) \approx 1$	$\Rightarrow$	object i is well classified (in A);
$s(i) \approx 0$	$\Rightarrow$	object i lies intermediate between two clusters (A and B);
$s(i) \approx -1$	$\Rightarrow$	object i is badly classified (closer to B than to A).

4. The silhouette of a cluster is a plot of the $s(i)$, ranked in decreasing order, of all its objects i. The entire silhouette plot shows the silhouettes of all clusters next to each other, so the "quality" of the clusters can be compared. The *overall average silhouette width* of the silhouette plot is the average of the $s(i)$ over all objects i in the data set.

Hint: run `pam` several times, each time for a different k, and compare the resulting silhouette plots. Then select that value of k yielding the highest average silhouette width, which is called the *silhouette coefficient*. Experience has led to the following subjective interpretation of the silhouette coefficient (SC). This interpretation does not depend on the number of objects.

SC	Proposed Interpretation
0.71–1.00	A strong structure has been found.
0.51–0.70	A reasonable structure has been found.
0.26–0.50	The structure is weak and could be artificial, try additional methods.
≤ 0.25	No substantial structure has been found.

Table 1: Interpretation of the silhouette coefficient for partitioning methods.

Example. Agriculture data, 2 clusters.

```
> pam(agri96,2)
Medoids:
   gnp agriculture
F 19.5         5.2
P  7.5        11.8
Clustering vector:
 B DK D GR E F IRL I L NL A P FIN S UK
 1  1 1  2 2 1   2 1 1  1 1 2   1 1  1
Objective function:
    build     swap
 3.863585 3.863585
```

Silhouette plot: medoids have high $s(i)$, boundary objects have low $s(i)$.

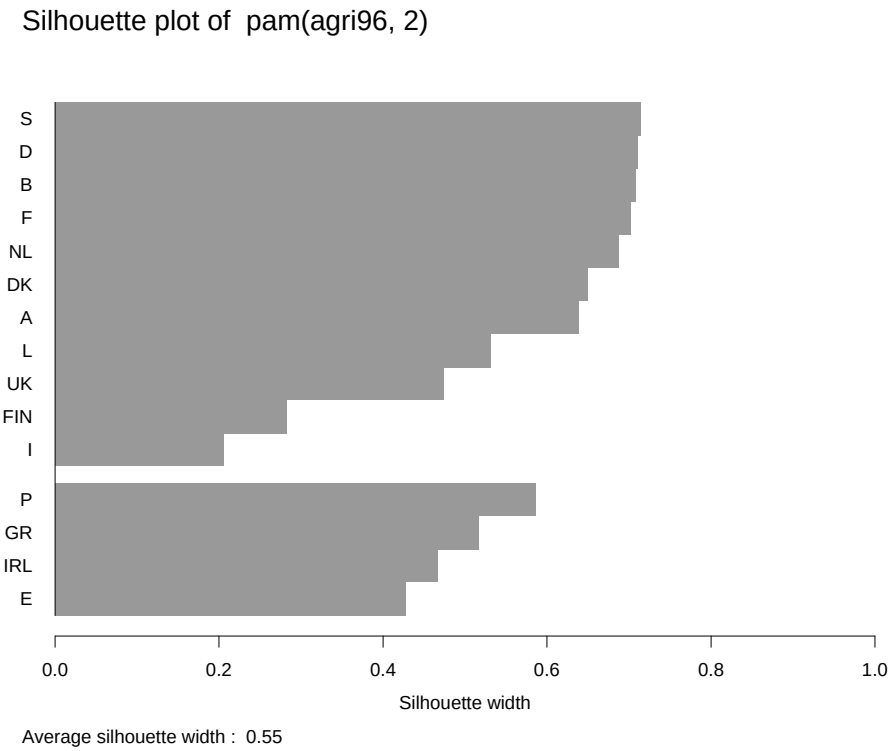


Figure 3: Silhouette plot of the `pam` clustering of Figure 2.

Graphical representation: *clusplot*.

Example. Ruspini data, 75 objects, 2 variables.

The 2-dimensional clusplot of the data set is given in Figure 4, and the silhouette plot constructed by `pam` in Figure 5.

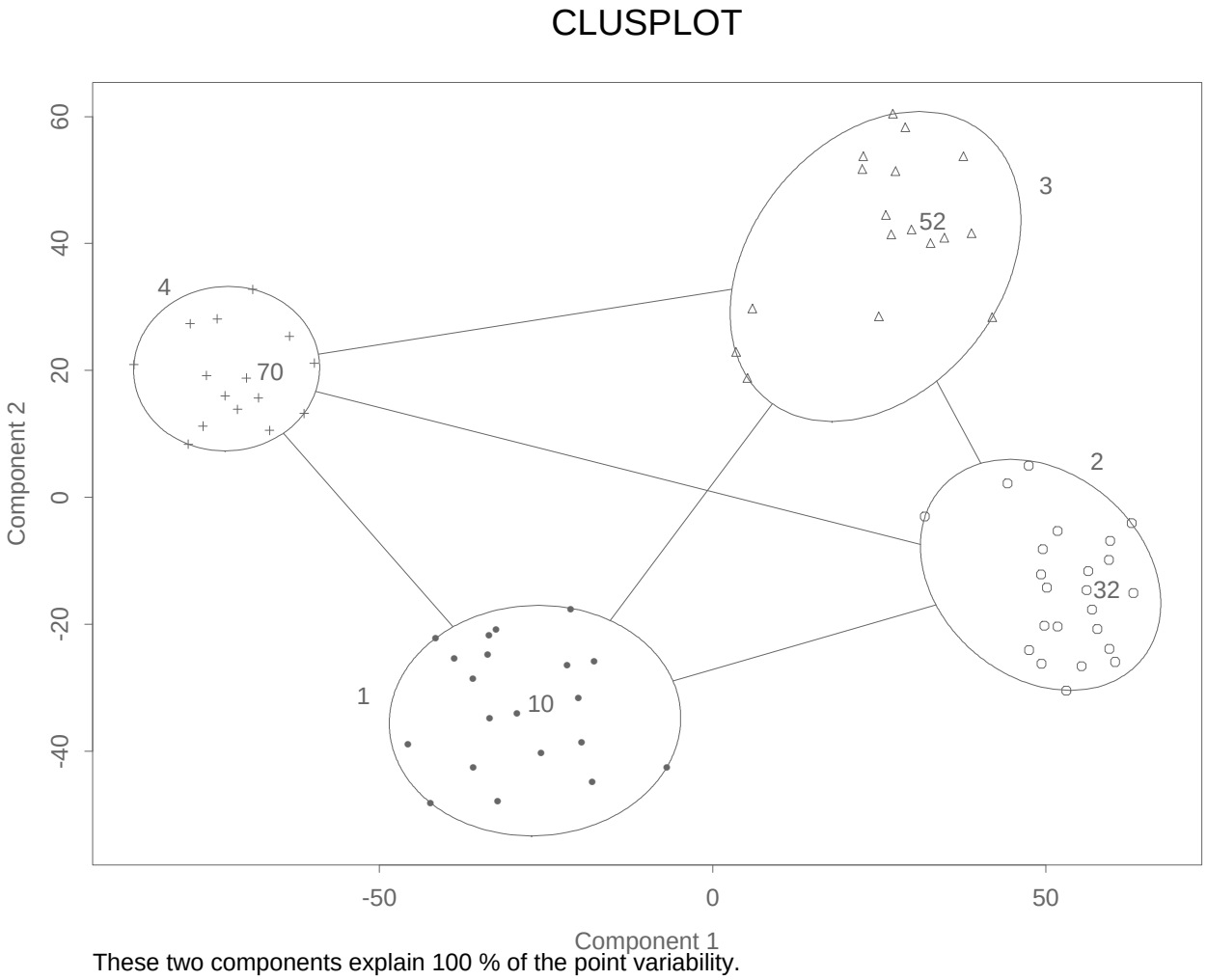


Figure 4: Two-dimensional plot of the Ruspini data, the medoids found by `pam` are indicated.

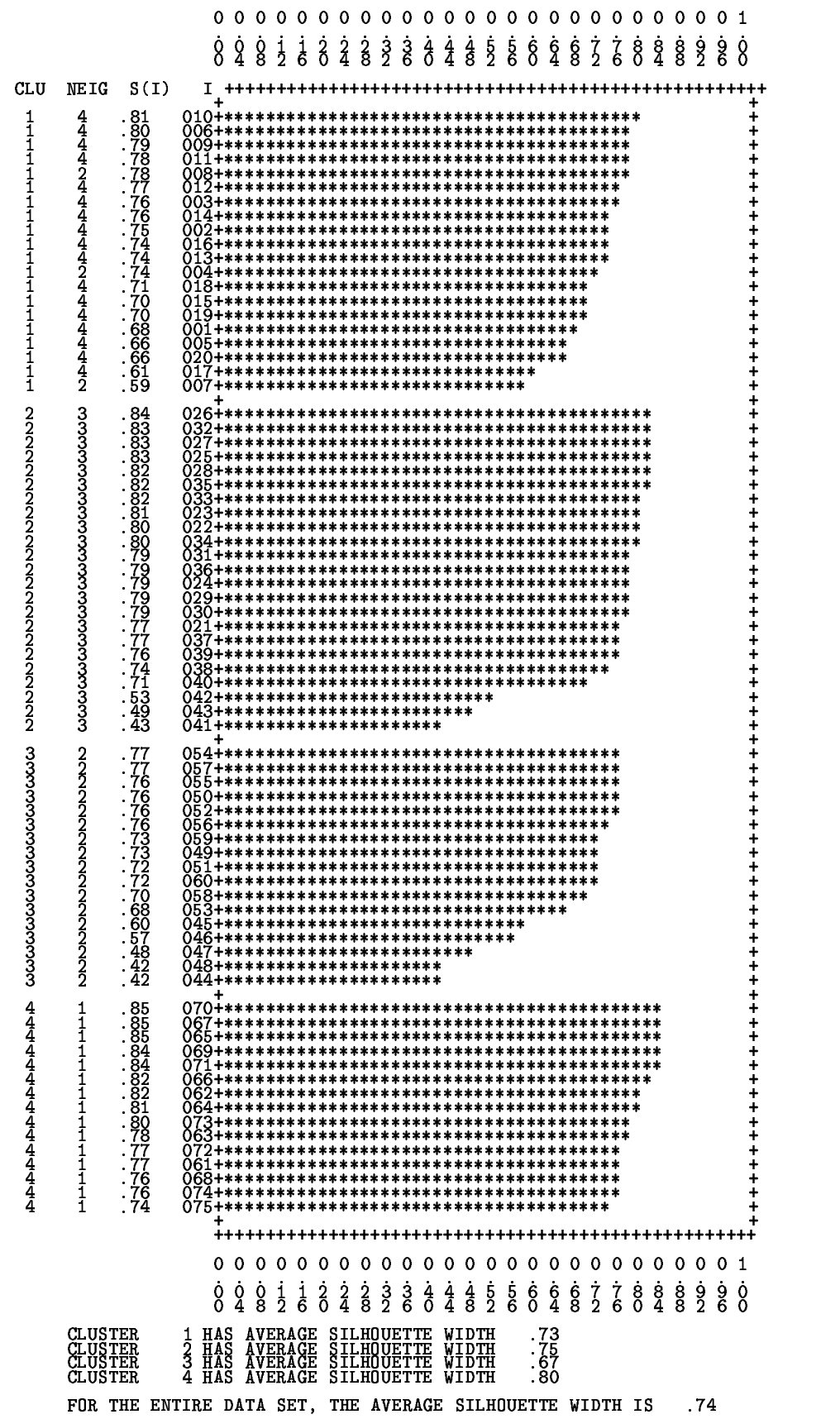
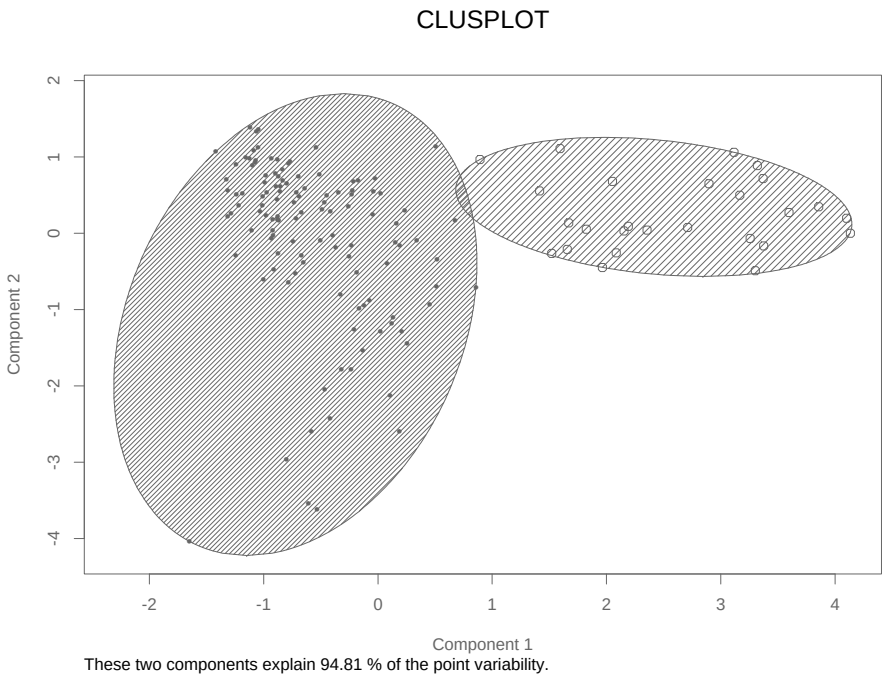


Figure 5: Silhouette plot of the `pam` clustering of the Ruspini data.

Example. Diabetes data, 145 objects, 3 variables.

Figure 6 and 7 use the first two principal components to represent the data in a two-dimensional space.

(a)



(b)

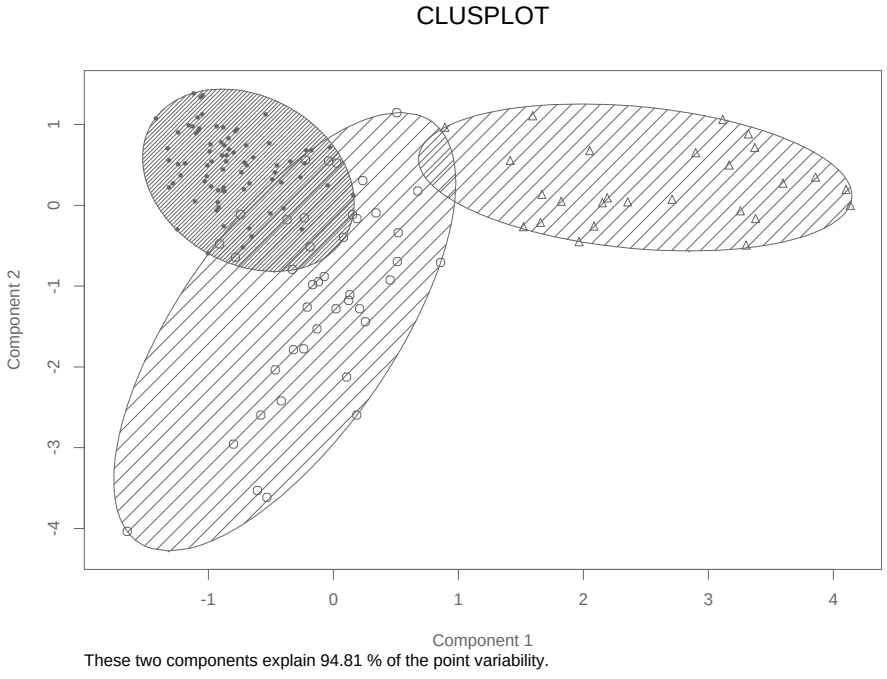


Figure 6: Plot of the diabetes data set and its partition into (a) $k = 2$ clusters; and (b) $k = 3$ clusters.

Example. Abbot-Perkins data, 20 objects, dissimilarity data.

Figure 7 represents the data in a two-dimensional space by means of a multidimensional scaling technique.

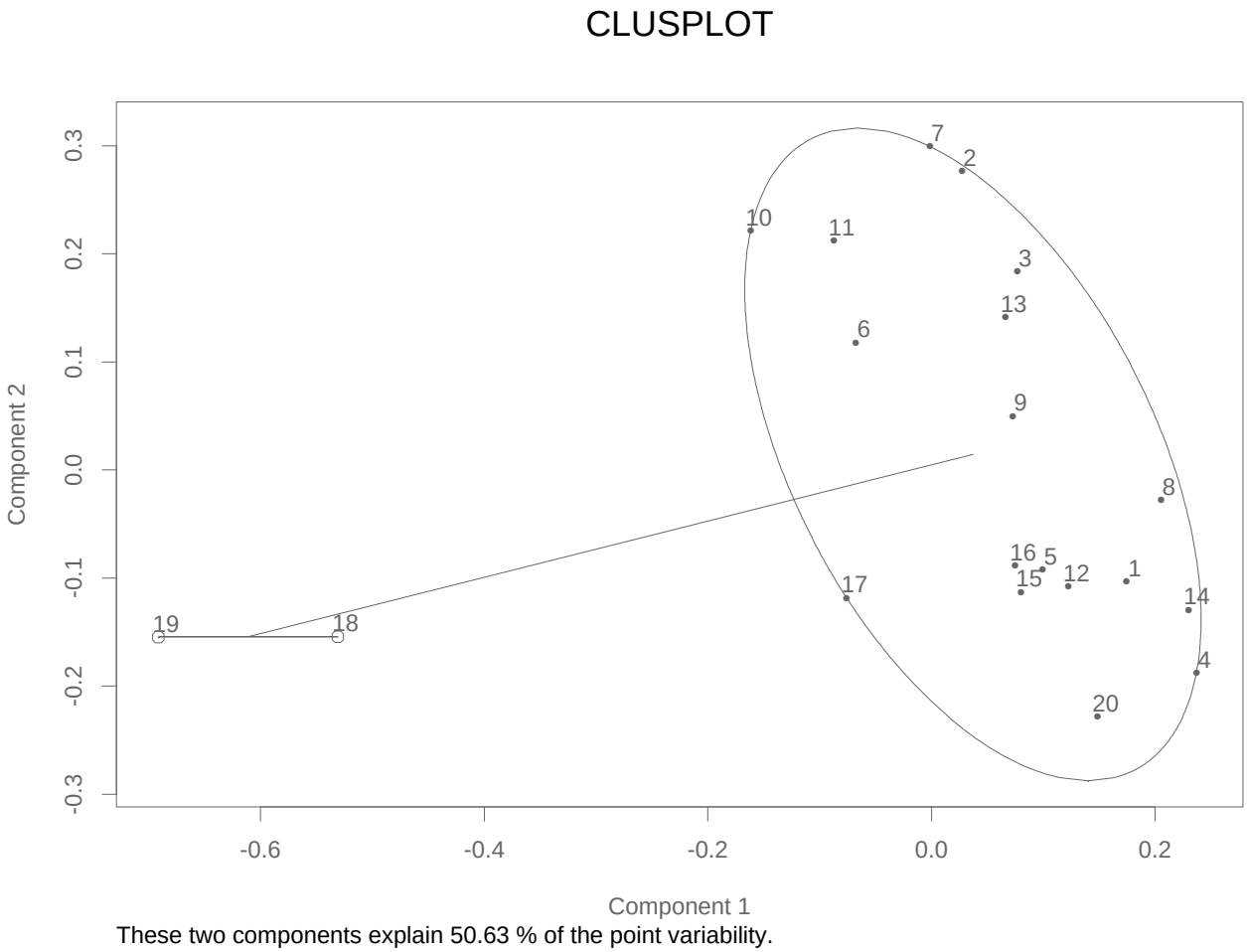


Figure 7: Clusplot of the Abbot-Perkins data.

4 Clustering Large Applications (`clara`)

Idea:

`pam` needs to store the entire dissimilarity matrix (which has O(n^2) entries) in central memory.
$\Rightarrow$ space complexity $O(n^2)$.
$\Rightarrow$ computation time $O(n^2)$.
$\Rightarrow$ less convenient for larger data sets (> 250 objects).
Solution: `clara` does not compute the entire dissimilarity matrix at once.

The algorithm proceeds as follows, with storage and time complexity indicated at the right:

1. Input of n cases with measurements. O(n)
2. Repeat '`samples`' times:
   (a) draw a subdata set of '`sampsize`' cases, fixed
   (b) applying `pam` to it yields medoids $\{m_1, \ldots, m_k\}$, fixed
   (c) compute objective $= \sum_{i=1}^{n} \min_{t=1,\ldots,k} d(i, m_t)$, O($n$)
   (d) retain $\{m_1, \ldots, m_k\}$ if objective $<$ currently best objective. fixed
3. Assigning all n cases to $\{m_1, \ldots, m_k\}$ yields the final partition. O(n)

The total storage and time are thus linear in n, instead of quadratic!
The default number of samples (`samples`) is 5, and the default sample size (`sampsize`) is $40 + 2k$, but these numbers can be adapted by the user.

The graphical representation of a clustering obtained by `clara` is a silhouette plot or a clusplot, as described in Section 3. Due to the size of the data set, the silhouette plot is given only for the best subdata set.

Example: 3000 objects.

Figure 8: clusplot of the partitioning into 3 clusters.

Figure 9a: scatter plot of the data.

Figure 9b: silhouette plot.

Overall average silhouette width = 0.74 ⇒ a strong clustering structure.

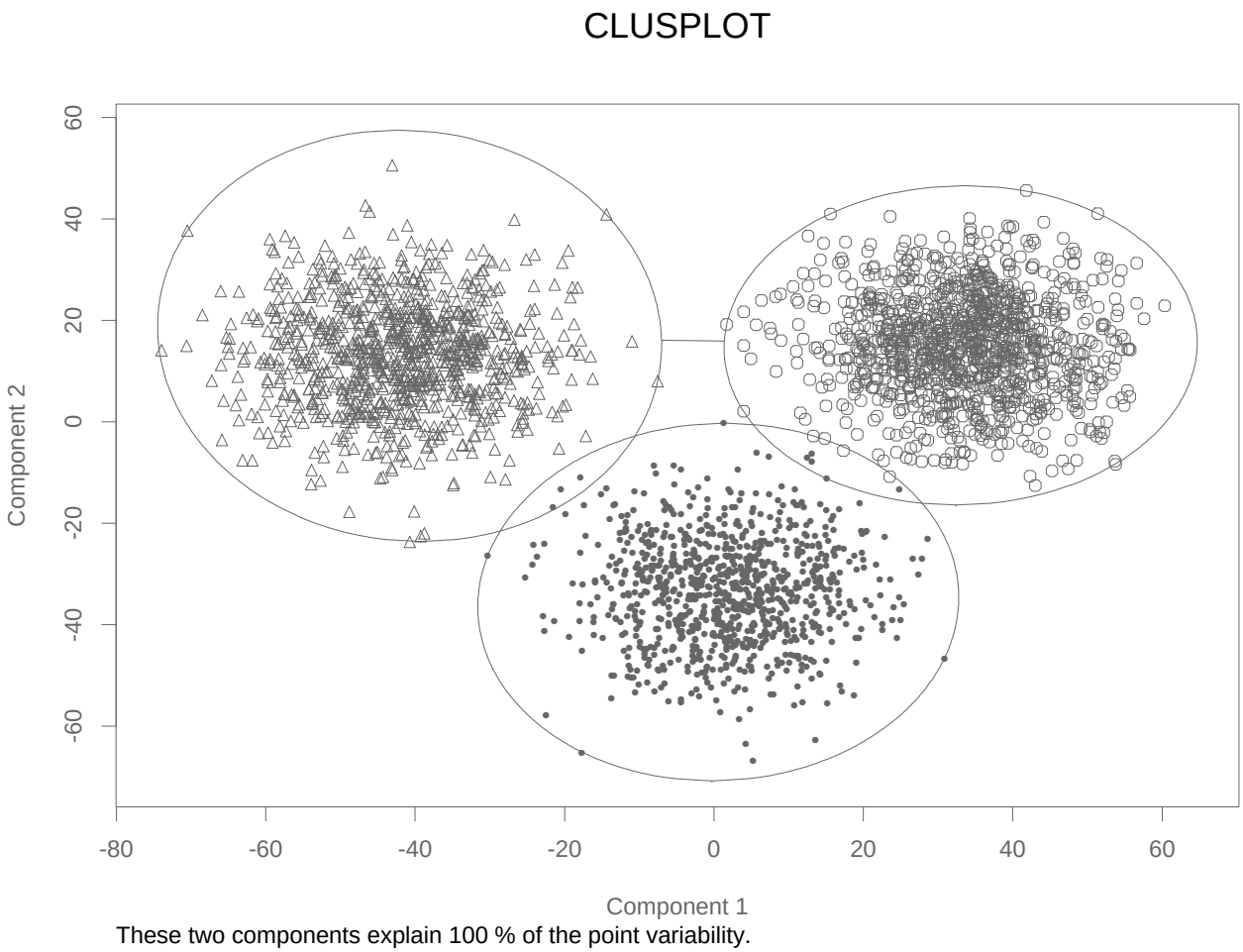
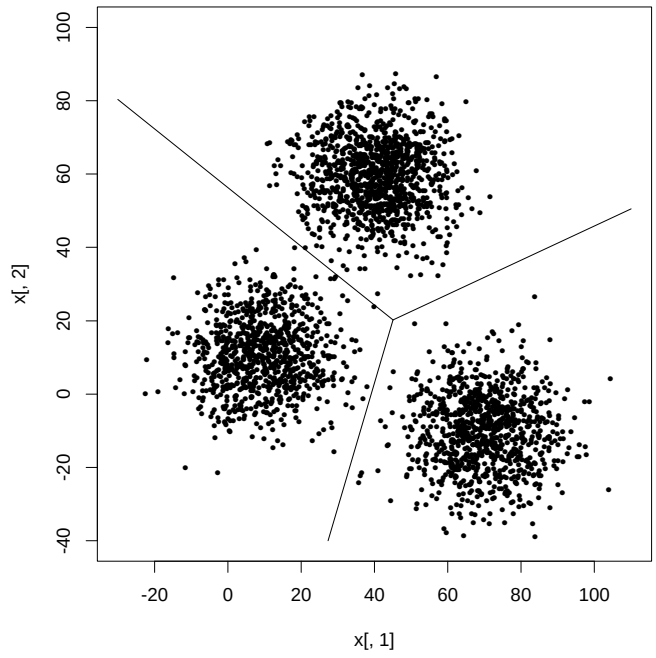


Figure 8: Clusplot of the 3000 objects.

(a)



(b)

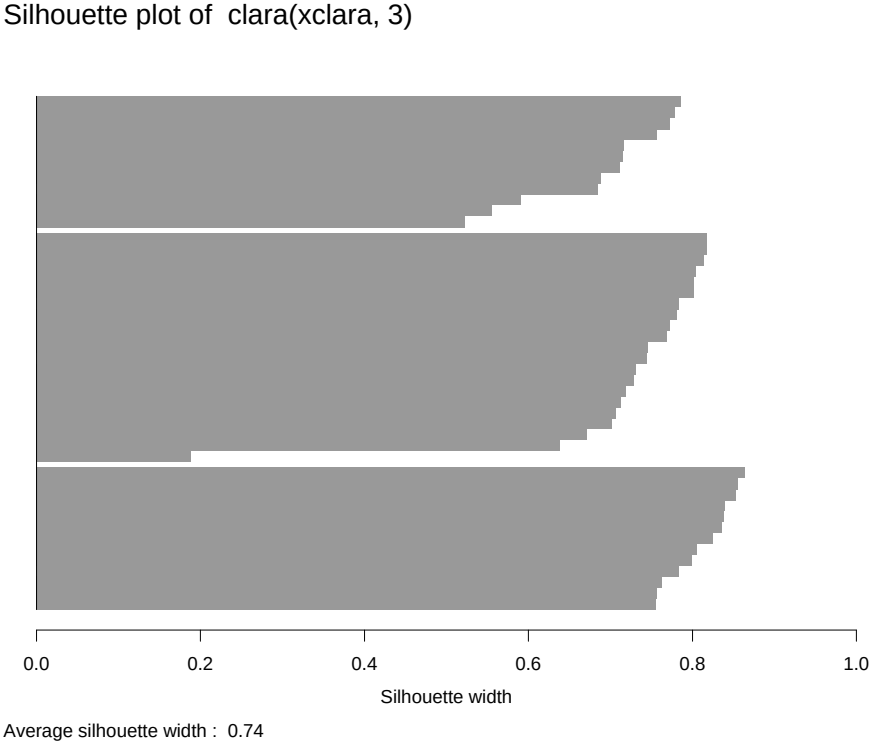


Figure 9: (a) Plot of 3000 observations and its partition into $k = 3$ clusters given by `clara`; and (b) Silhouette plot of the `clara` clustering of (a).

5 Fuzzy Analysis (`fanny`)

Idea:

Crisp Clustering Methods: each object of the data set is assigned to exactly one cluster. For instance, an object lying between two clusters will be assigned to one of them (as in Figure 2).

Examples: `pam` and `clara`.

Fuzzy Clustering Methods: each object is spread over the various clusters.

Example: `fanny`.

A fuzzy method will compute for each object i and each cluster v a *membership* u_{iv} which indicates how strongly object i belongs to cluster v. Memberships have to satisfy the following conditions:

- $u_{iv} \ \geq \ 0$ for all $i = 1, \ldots, n$ and all $v = 1, \ldots, k$;
- $\sum_{v=1}^{k} u_{iv} \ = \ 1 \ = \ 100\%$ for all $i = 1, \ldots, n$.

Summarizing, we can say that
crisp memberships $\in \{0, 1\}$ and fuzzy memberships $\in [0, 1]$.

Example. Agriculture data, 2 clusters.

Table 2 lists the memberships of all 15 countries, Figure 10 compares this clustering with the one found by `pam`. It is clear that intermediate and outlying objects receive more fuzzy memberships compared to objects which clearly belong to one of the clusters.

			crisp memberships		fuzzy memberships	
Country	x_i	y_i	u_{i1}	u_{i2}	u_{i1}	u_{i2}
B (Belgium)	19.0	2.9	1	0	.89	.11
DK (Denmark)	23.8	5.0	1	0	.80	.20
D (Germany)	21.2	3.3	1	0	.88	.12
GR (Greece)	7.7	20.8	0	1	.27	.73
E (Spain)	10.4	9.9	0	1	.15	.85
F (France)	19.5	5.2	1	0	.88	.12
IRL (Ireland)	12.5	13.1	0	1	.16	.84
I (Italy)	14.8	7.7	1	0	.42	.58
L (Luxemburg)	28.8	3.2	1	0	.69	.31
NL (Netherlands)	18.1	4.0	1	0	.87	.13
A (Austria)	20.7	6.9	1	0	.80	.20
P (Portugal)	7.5	11.8	0	1	.17	.83
FIN (Finland)	16.2	8.6	1	0	.47	.53
S (Sweden)	18.9	3.4	1	0	.90	.10
UK (U.Kingdom)	14.7	2.1	1	0	.64	.36

Table 2: Agriculture data set about 15 European countries, with crisp memberships (obtained by `pam`) and fuzzy memberships (obtained by `fanny`).

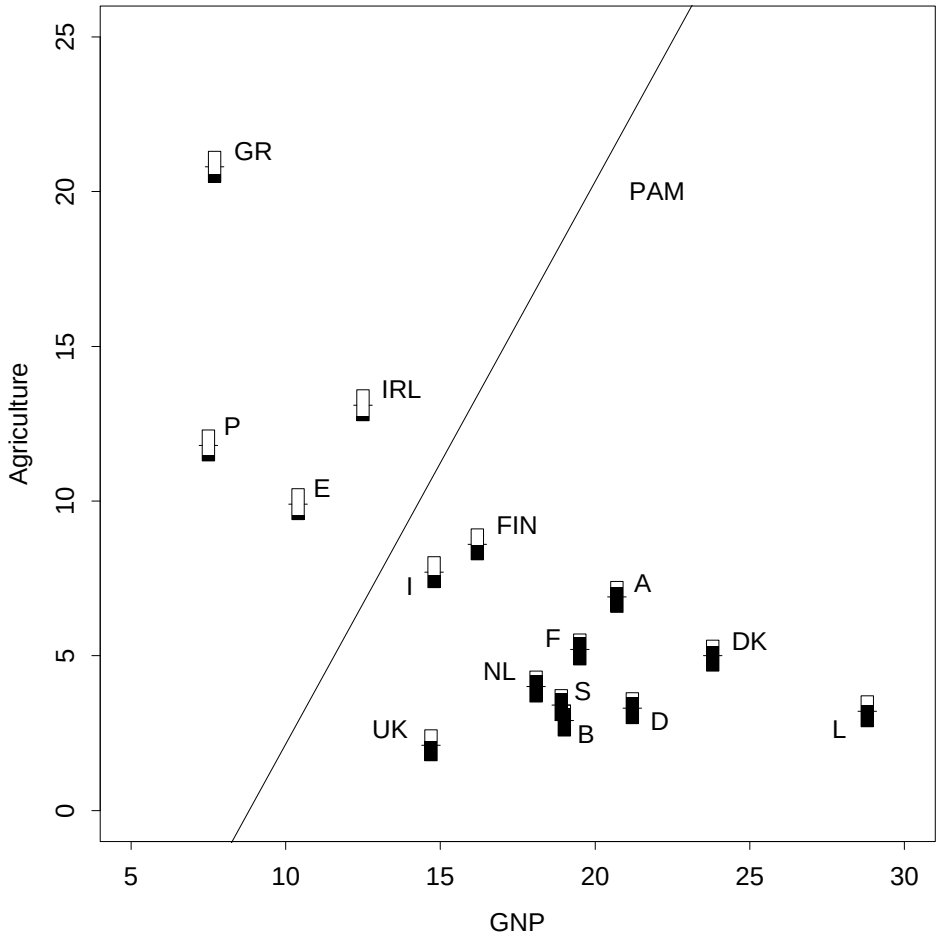


Figure 10: Agriculture data set about 15 European countries. The partitionings found by `pam` and `fanny` are given.

Algorithm:

In the method `fanny`, the memberships u_{iv} are defined through minimization of the objective function

$$\sum_{v=1}^{k} \frac{\sum_{i,j=1}^{n} u_{iv}^2 u_{jv}^2 d(i,j)}{2\sum_{j=1}^{n} u_{jv}^2}.$$

In this expression, the $d(i,j)$ are known and the u_{iv} are unknown. The minimization is carried out numerically.

To have an idea of "how fuzzy" the resulting clustering is, *Dunn's partition coefficient* is computed:

$$F_k \;=\; \sum_{i=1}^{n}\sum_{v=1}^{k} \frac{u_{iv}^2}{n}$$

which always lies in $[\frac{1}{k}, 1]$.

This coefficient attains its extreme values in the following situations:

1. entirely fuzzy clustering:
   all $u_{iv} = \frac{1}{k} \Rightarrow F_k = nk\frac{1}{nk^2} = \frac{1}{k}$;
2. crisp clustering:
   all $u_{iv} = 0$ or $1 \Rightarrow F_k = \frac{n}{n} = 1$.

The normalized version of this coefficient is

$$F_k{}' = \frac{F_k - \frac{1}{k}}{1 - \frac{1}{k}} = \frac{kF_k - 1}{k - 1}$$

which always lies in $[0, 1]$.
In the agriculture example:

$$F_k = 0.68 \qquad \text{and} \qquad F_k{}' = 0.37.$$

Graphical representation:

Nearest Crisp Clustering = assign each object i to the cluster v in which it has the highest membership u_{iv}.

This crisp clustering can then be represented by a silhouette plot or a clusplot (as in Section 3). In the agriculture example, the crisp clustering closest to that of `fanny` happens to be different of that of `pam`, hence yielding another silhouette plot (Figure 11).

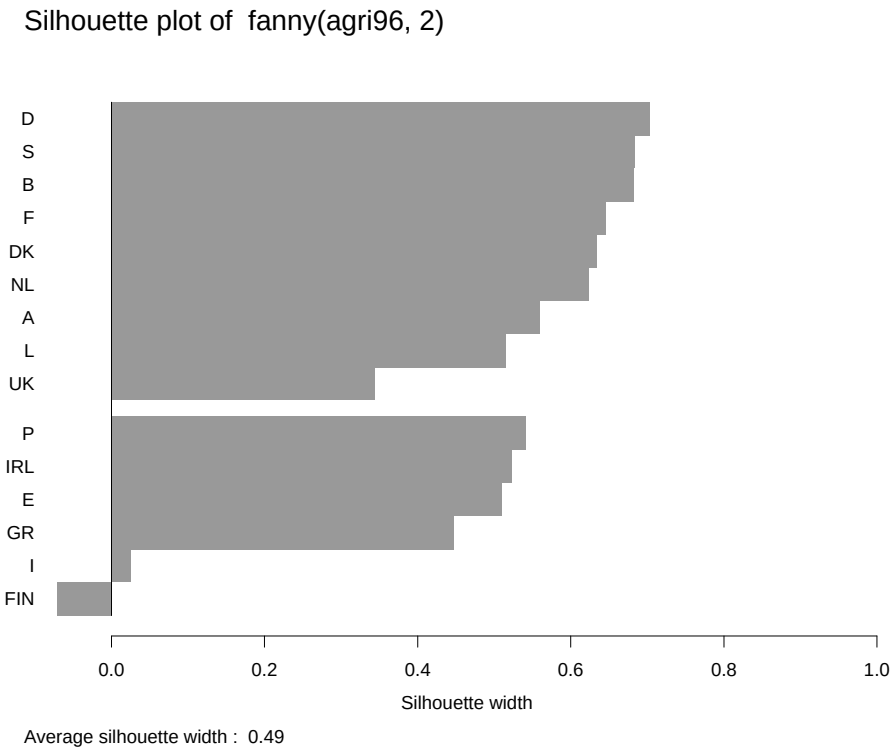


Figure 11: Silhouette plot of the `fanny` clustering of Figure 10.

6 Agglomerative Nesting (agnes)

Idea:

`agnes` is an agglomerative hierarchical clustering method. Hence it yields a sequence of clusterings. In the first clustering each of the n objects forms its own separate cluster. In subsequent steps clusters are merged, until (after $n-1$ steps) only one large cluster remains.

Algorithm:

Step 0:

consider each object as a separate cluster.

Step i: $(i = 1, \ldots, n-1)$

1. Merge the two clusters of step $i-1$ with the smallest between-cluster dissimilarity.
2. Compute the dissimilarity between the new cluster and all remaining clusters.

The between-cluster dissimilarity can be defined in various ways, e.g.:

- *Group Average Method*

$$d(R, Q) = \frac{1}{|R||Q|} \sum_{i \in R, j \in Q} d(i, j);$$

- *Nearest Neighbor Method = Single Linkage Method*

$$d(R, Q) = \min_{i \in R, j \in Q} d(i, j);$$

- *Furthest Neighbor Method = Complete Linkage Method*

$$d(R,Q) = \max_{i \in R, j \in Q} d(i,j).$$

Example: Group average method.

	a	b	c	d	e
a	0				
b	2	0			
c	6	5	0		
d	10	9	4	0	
e	9	8	5	3	0

$\Rightarrow$

	{a,b}	c	d	e
{a,b}	0			
c	5.5	0		
d	9.5	4	0	
e	8.5	5	3	0

$\Rightarrow$

	{a,b}	c	{d,e}
{a,b}	0		
c	5.5	0	
{d,e}	9	4.5	0

$\Rightarrow$

	{a,b}	{c,d,e}
{a,b}	0	
{c,d,e}	7.83	0

Graphical representation:

Agglomerative Tree: a tree in which the leaves represent objects. The vertical coordinate of the junction of two branches is the dissimilarity between the corresponding clusters.

Agglomerative Banner: the banner shows the successive mergers from left to right. (Imagine the ragged flag parts at the left, and the flagstaff at the right.) The objects are listed vertically. The merger of two clusters is represented by a horizontal bar which commences at the between-cluster dissimilarity.

The quality of an agglomerative clustering of the data can be measured by the *agglomerative coefficient* (Rousseeuw 1986). It is defined as the average width (the percentage filled) of the banner.

Note: the AC tends to increase with the number of objects.

Example. Agriculture data.

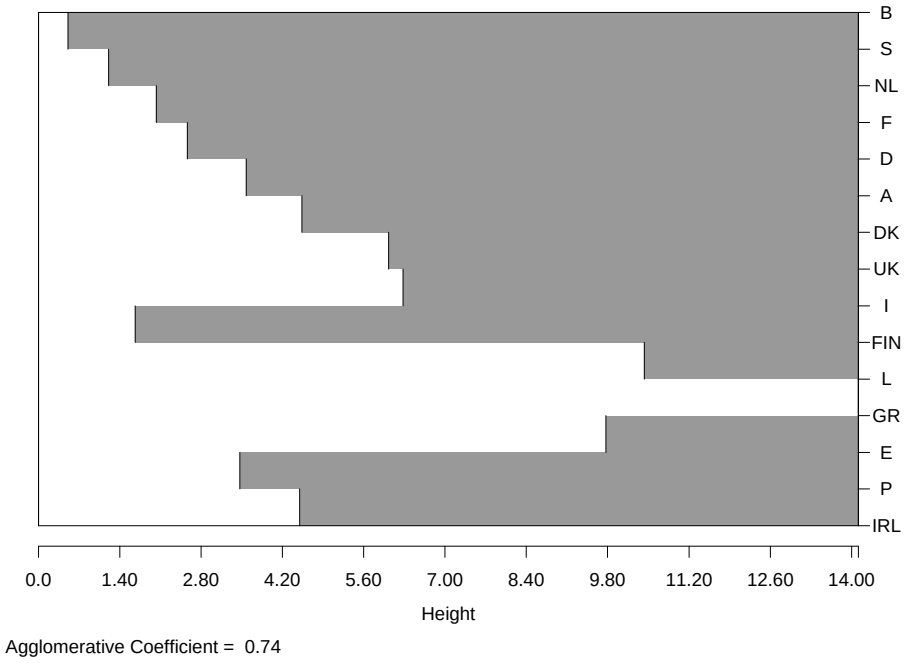
In the agriculture data set `agnes` ends up with the same clustering as `pam` and `fanny` found. The agglomerative coefficient is rather high (0.74) which indicates a good clustering structure. Both the banner (Figure 12a) and the clustering tree (Figure 12b) have been plotted, to facilitate comparisons between them.

Example. Congress data.

The dissimilarities give the number of times that 15 congressmen in New Jersey voted differently on 19 environmental bills. The banner, obtained by `agnes` is given in Figure 13.

```
R1     0
R2     8    0
D3    15   17    0
D4    15   12    9    0
R5    10   13   16   14    0
R6     9   13   12   12    8    0
R7     7   12   15   13    9    7    0
D8    15   16    5   10   13   12   17    0
D9    16   17    5    8   14   11   16    4    0
D10   14   15    6    8   12   10   15    5    3    0
D11   15   16    5    8   12    9   14    5    2    1    0
R12   16   17    4    6   12   10   15    3    1    2    1    0
R13    7   13   11   15   10    6   10   12   13   11   12   12    0
D14   11   12   10   10   11    6   11    7    7    4    5    6    9    0
D15   13   16    7    7   11   10   13    6    5    6    5    4   13    9    0
```

(a)



(b)

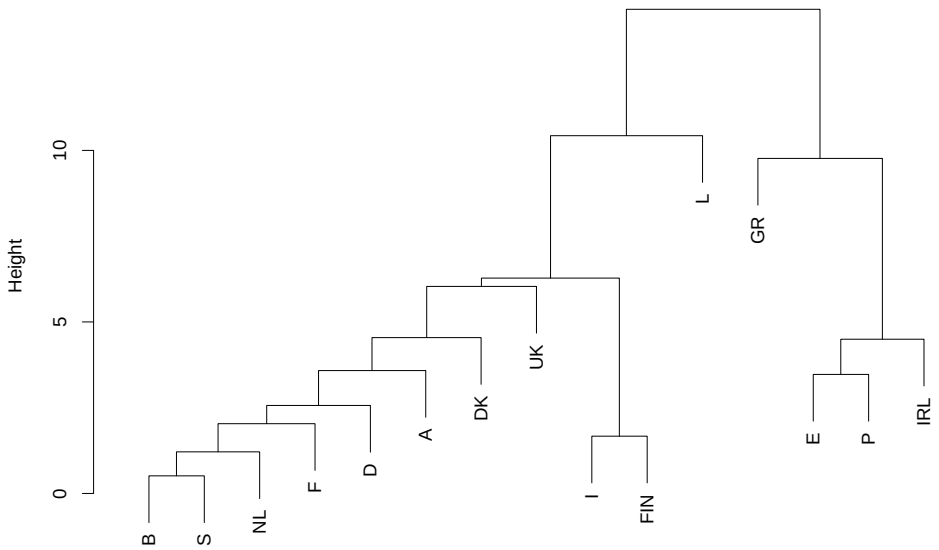


Figure 12: Result of applying `agnes` to the agriculture data, represented in the form of (a) a banner; and (b) a clustering tree.

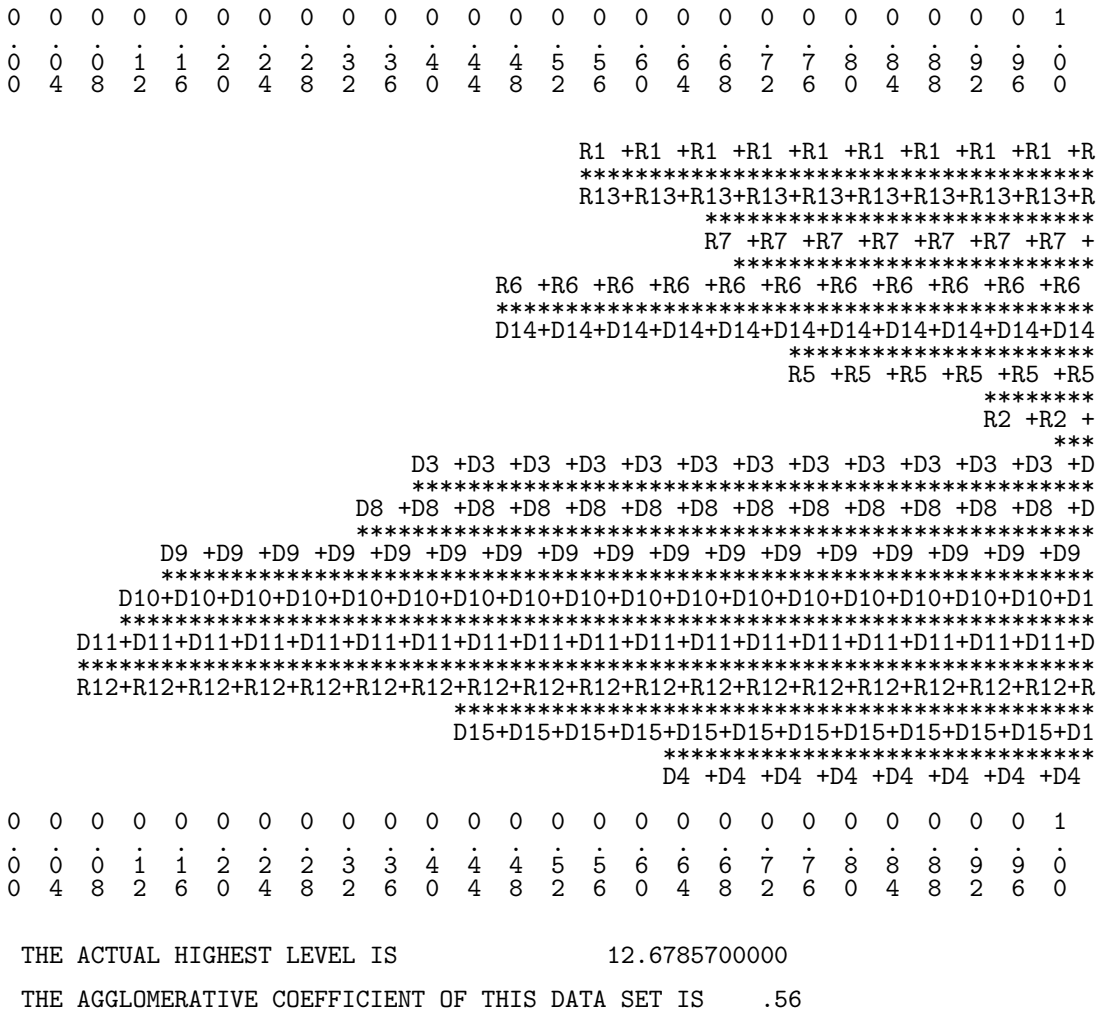


Figure 13: Agglomerative banner of the congress data.

7 Divisive Analysis (`diana`)

Idea:

`diana` is a divisive hierarchical method. The initial clustering consists of one large cluster containing all n objects. In each subsequent step, the largest available cluster is split into two smaller clusters, until finally all clusters contain but a single object.

The algorithm consists of $n-1$ successive splits. In each step, we select the cluster C with the largest diameter, where

$$\text{diam}(C) := \max_{i,j \in C} d(i,j).$$

Assuming $\text{diam}(C) > 0$ we then split up C into two clusters A and B.

Below we describe in pseudocode how such a split is performed.

1. $A := C$ and $B := \varnothing$.

2. Move one object from A to B: ($B =$ "splinter group")
   for each object $i \in A$ we calculate $a(i)$, the average dissimilarity to all other objects of A, as in (3.2). The object m of A for which $a(m)$ is the largest, is moved to B:

$$A := A \setminus \{m\},\ B := \{m\}.$$

3. Move other objects from A to B:
   if $|A| = 1$, stop.
   Otherwise, calculate for all $i \in A$, $a(i)$, and the average dissimilarity of

i to all objects of B, denoted as $d(i, B)$ in (3.3). Select the object $h \in A$ for which $a(h) - d(h, B) = \max_{i \in A}(a(i) - d(i, B))$.

If $a(h) - d(h, B) > 0 \Rightarrow$ move h from A to B, and go to 3.

If $a(h) - d(h, B) \leq 0 \Rightarrow$ the process stops. Keep A and B as they are now.

Example :

	a	b	c	d	e
a	0				
b	2	0			
c	6	5	0		
d	10	9	4	0	
e	9	8	5	3	0

Compute for each object the average dissimilarity to all other objects:

$a(a) = 6.75, \qquad a(b) = 6, \qquad a(c) = 5, \qquad a(d) = 6.5, \qquad a(e) = 6.25.$

Maximal for $a \Rightarrow$ splinter group $= \{a\}$.

Compute differences for objects outside the splinter group:

$a(b) - d(b, \{a\}) = 7.33 - 2 = 5.33, \qquad a(c) - d(c, \{a\}) = 4.67 - 6 = -1.33,$

$a(d) - d(d, \{a\}) = 5.33 - 10 = -4.67, \qquad a(e) - d(e, \{a\}) = 5.33 - 9 = -3.67.$

Maximal for $b \Rightarrow$ splinter group $= \{a, b\}$.

Compute differences for objects outside the splinter group:

$a(c) - d(c, \{a, b\}) = -1, \quad a(d) - d(d, \{a, b\}) = -6, \quad a(e) - d(e, \{a, b\}) = -4.5.$

All negative $\Rightarrow$ clusters $\{a, b\}$ and $\{c, d, e\}$.

Now split cluster $\{c, d, e\}$:

$a(c) = 4.5, \qquad a(d) = 3.5, \qquad a(e) = 4.$

Maximal for $c \Rightarrow$ splinter group $= \{c\}$.

$a(d) - d(d, \{c\}) = -1, \qquad a(e) - d(e, \{c\}) = -2.$

All negative $\Rightarrow$ clusters $\{c\}$ and $\{d, e\}$.

Graphical representation:

Divisive Tree: here the stem represents the entire data set. The vertical coordinate where a branch splits in two equals the diameter of that cluster before splitting.

Divisive Banner: the banner shows the successive splits from left to right. (Imagine the flagstaff at the left, and the ragged endings at the right.) The objects are stitched together by horizontal bars, which end at the diameter of the cluster being split.

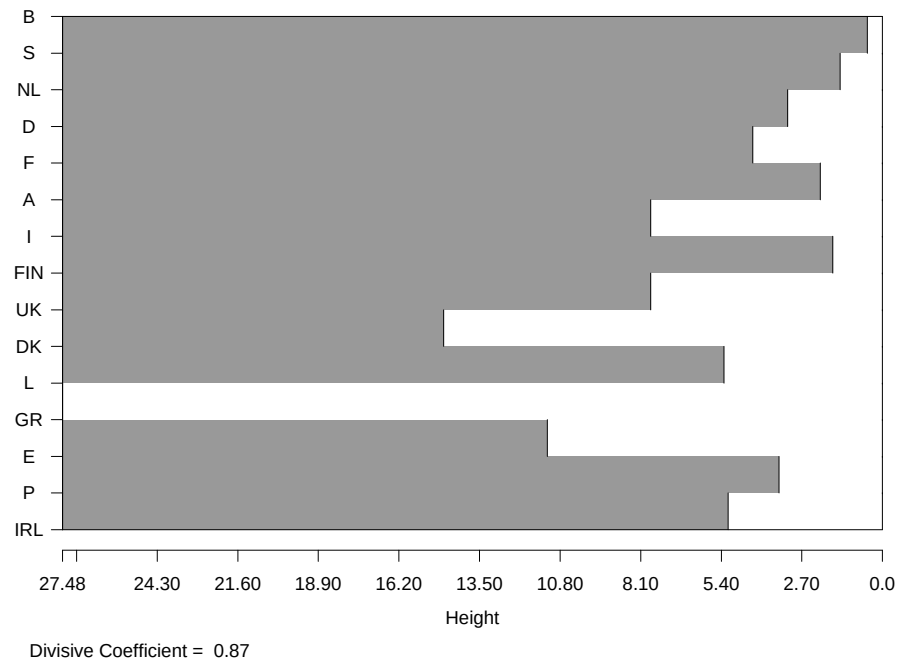
The *divisive coefficient* (Rousseeuw 1986), measures the clustering structure of the data set. It is defined as the average width (or the percentage filled) of the banner.

Note: like the AC of `agnes`, also the DC grows with the number of objects. Therefore, neither the AC or the DC can be used to compare data sets of very different sizes.

Example. Agriculture data.

`diana` finds essentially the same clustering structure as `agnes`, with a high divisive coefficient (0.87). The divisive banner in Figure 14a is almost a mirror image of the agglomerative banner in Figure 12a.

(a)



(b)

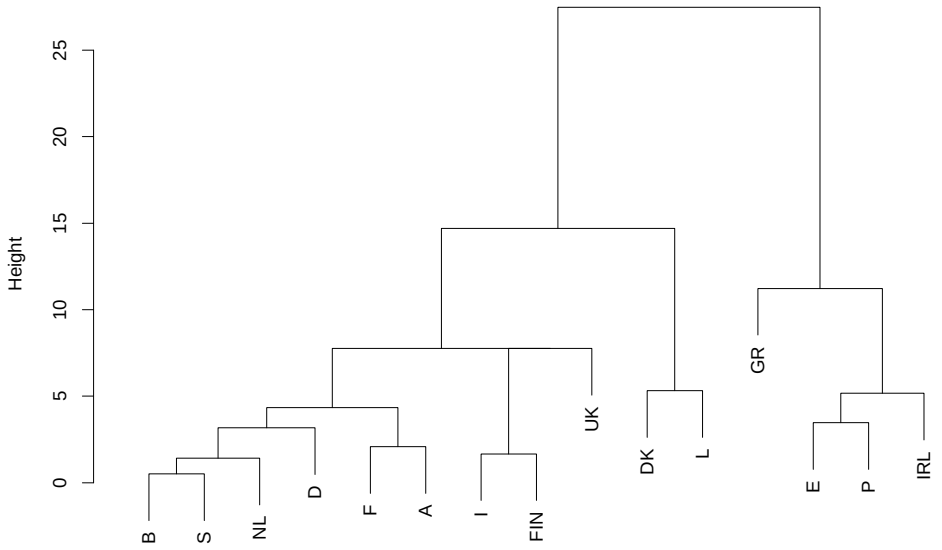


Figure 14: Result of applying `diana` to the agriculture data, represented in the form of (a) a banner; and (b) a clustering tree.

8 Monothetic Analysis (`mona`)

Idea:

The function `mona` is a monothetic divisive hierarchical method, and only accepts binary data as input.

Monothetic Divisive Methods: methods using a single variable for each split.
Example: `mona`.

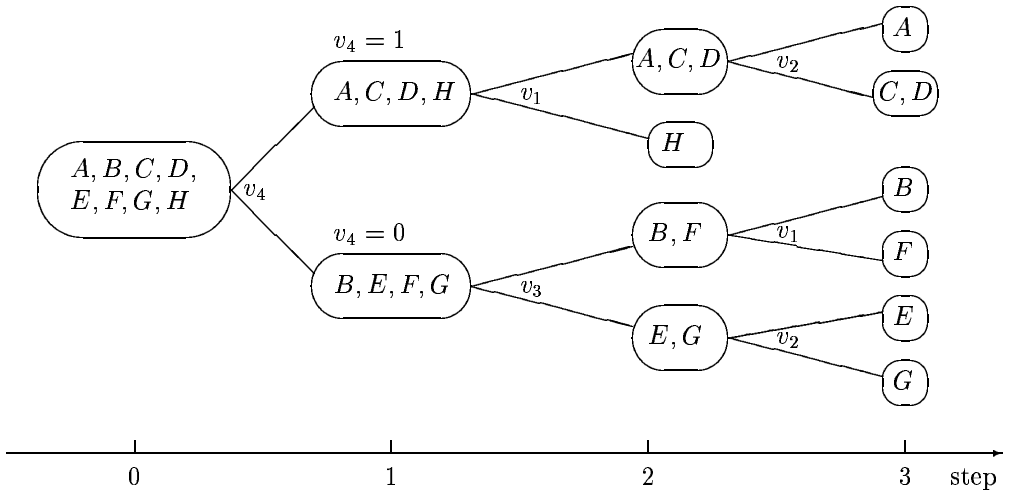
Polythetic Divisive Methods: methods using all variables simultaneously for each split.
Example: `diana`.

Example: 4 variables, 8 objects.
Figure 15a.

Graphical representation:

The clustering hierarchy constructed by `mona` can be represented by means of a divisive banner (Figure 15b). The length of a bar is now given by the number of divisive steps needed to make that split. Next to the bar, the variable is listed which was responsible for the split. A bar continuing to the right margin (here, to the imaginary "step 4") indicates a cluster that cannot be split.

(a)



(b)

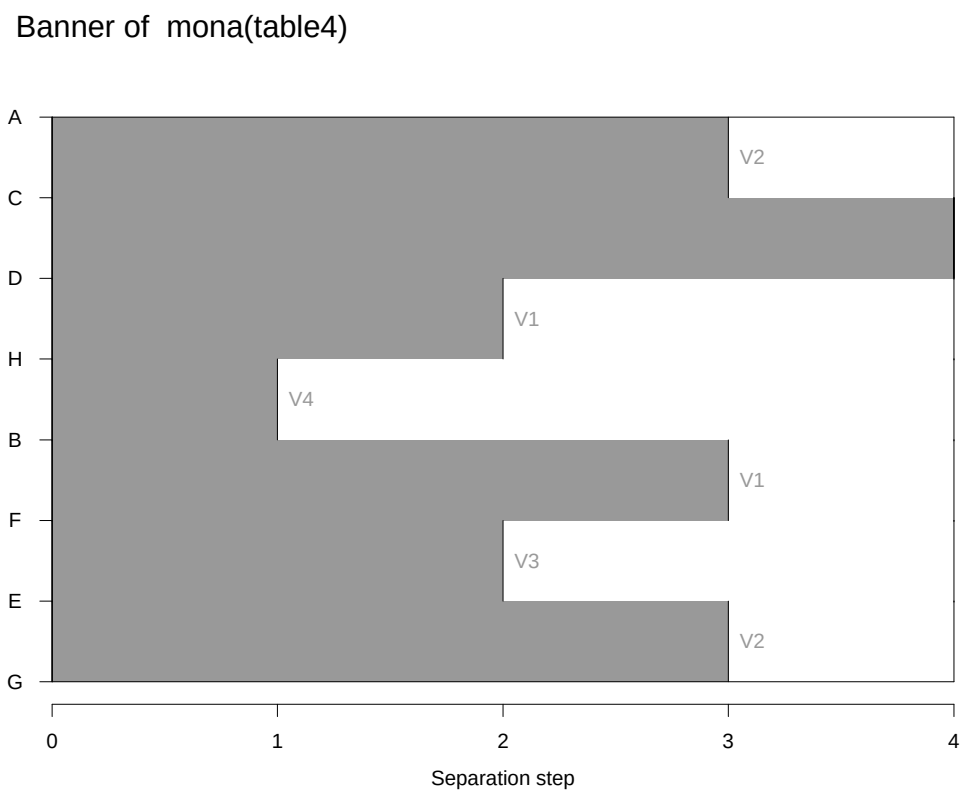


Figure 15: Clustering hierarchy obtained by `mona` for 8 objects with 4 binary variables: (a) depicted as a tree; (b) in the form of a banner.

Algorithm:

1. Replace all missing values in the binary data matrix (i.e., all values $\neq 0$ or 1) by estimated values:

   suppose that x_{if} is missing.

   Let g be any variable $\neq f$, and construct the contingency table.

$f \backslash g$	1	0
1	a_{fg}	b_{fg}
0	c_{fg}	d_{fg}

   The association between f and g is defined as

$$A_{fg} := |a_{fg}d_{fg} - b_{fg}c_{fg}|. \tag{8.1}$$

   The variable t for which

$$A_{ft} = \max_g A_{fg},$$

   is the most correlated with variable f.

   The missing values of f are then estimated by means of variable t in the following way:

   - put $x_{if} := x_{it}$ when $a_{ft}d_{ft} - b_{ft}c_{ft} > 0$,
   - put $x_{if} := 1 - x_{it}$ when $a_{ft}d_{ft} - b_{ft}c_{ft} < 0$.

2. Split each available cluster according to one variable:

   a cluster is divided into a cluster with all objects having value 1 for the chosen variable, and another cluster with all objects having value 0 for that variable.

   The variable used for splitting a cluster is the variable with largest total association to the other variables. The association between variables f and g is given by the expression A_{fg} (8.1), but now the contingency table only contains the objects of the cluster to be split. The total association of a variable f is then defined as:

$$A_f = \sum_{g \neq f} A_{fg}.$$

   The variable t which satisfies

$$A_t = \max_f A_f,$$

   is selected for splitting the cluster.

3. Repeat step 2 until each cluster consists of objects having identical values for all variables. Such clusters cannot be split any more. A final cluster is thus a singleton or an indivisible cluster.

Example: Animals data.

Original data

	war	fly	ver	end	gro	hai
ant	1	1	1	1	2	1
bee	1	2	1	1	2	2
cat	2	1	2	1	1	2
cpl	1	1	1	1	1	2
chi	2	1	2	2	2	2
cow	2	1	2	1	2	2
duc	2	2	2	1	2	1
eag	2	2	2	2	1	1
ele	2	1	2	2	2	1
fly	1	2	1	1	1	1
fro	1	1	2	2	NA	1
her	1	1	2	1	2	1
lio	2	1	2	NA	2	2
liz	1	1	2	1	1	1
lob	1	1	1	1	NA	1
man	2	1	2	2	2	2
rab	2	1	2	1	2	2
sal	1	1	2	1	NA	1
spi	1	1	1	NA	1	2
wha	2	1	2	2	2	1

Revised data

	war	fly	ver	end	gro	hai
ant	0	0	0	0	1	0
bee	0	1	0	0	1	1
cat	1	0	1	0	0	1
cpl	0	0	0	0	0	1
chi	1	0	1	1	1	1
cow	1	0	1	0	1	1
duc	1	1	1	0	1	0
eag	1	1	1	1	0	0
ele	1	0	1	1	1	0
fly	0	1	0	0	0	0
fro	0	0	1	1	0	0
her	0	0	1	0	1	0
lio	1	0	1	1	1	1
liz	0	0	1	0	0	0
lob	0	0	0	0	0	0
man	1	0	1	1	1	1
rab	1	0	1	0	1	1
sal	0	0	1	0	0	0
spi	0	0	0	0	0	1
wha	1	0	1	1	1	0

`war` = warm-blooded or cold-blooded animal,

`fly` = flying or non-flying animal,

`ver` = vertebrate or invertebrate,

`end` = endangered or not,

`gro` = live in groups,

`hai` = have hair or not.

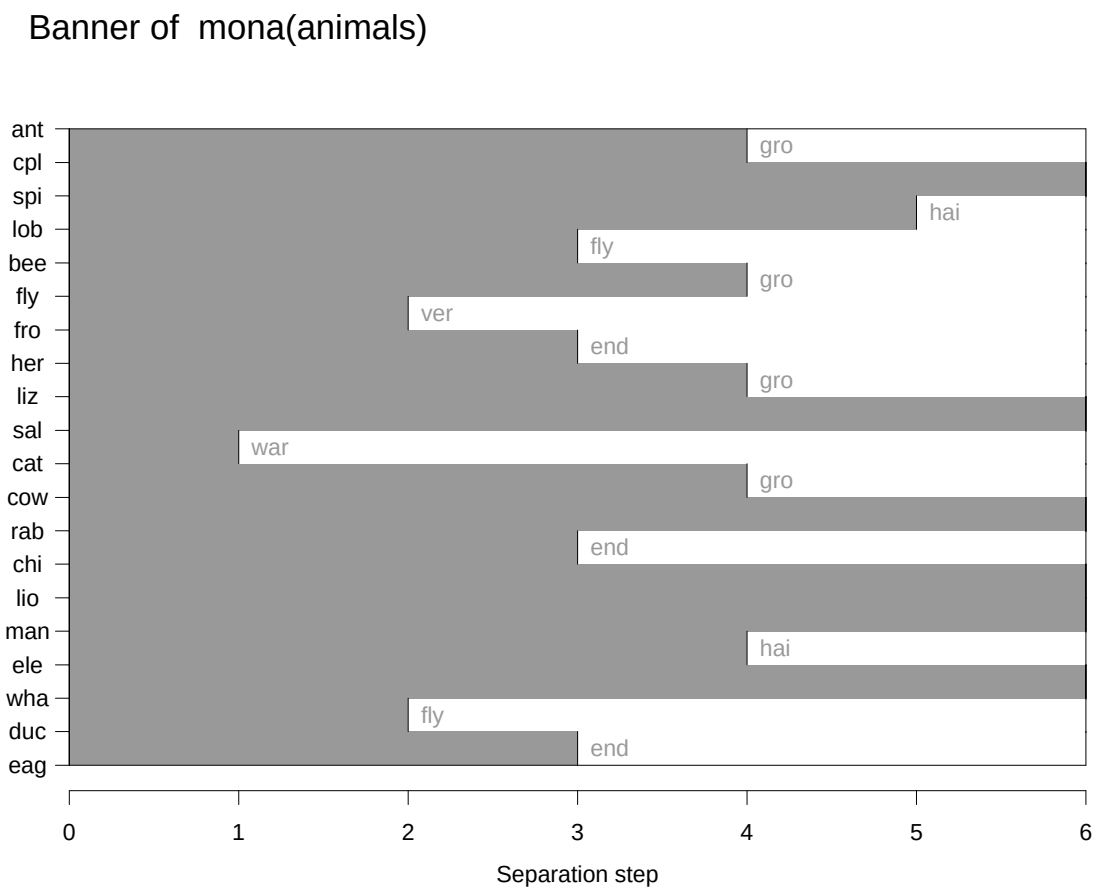


Figure 16: Animals data: divisive banner of the `mona` clustering. The clusters at the imaginary last step (here, "step 6") are unsplittable.

9 References

Book: Kaufman L. and Rousseeuw P. (1990),
"Finding Groups in Data, an Introduction to Cluster Analysis",
Wiley, New York.

Software: S-PLUS library `cluster` (1996).