What is Classification About

The purpose of classification is to group an arbitrary number, n, of observations into homogeneous classes.

There are two main types of classification, namely

- ✓ supervised classification;
- ✓ unsupervised classification, aka clustering.

Classified tree canopy layer in the Virginia Urban Tree Canopy Mapper – http://www.utcmapper.frec.vt.edu

Both types will be studied in this course.

Supervised Classification

Context

- ✓ The number, k, of groups is fixed and known.
- ✓ Group membership is known for all observations in a training sample.

The objective is to determine to which group each future observation will belong on the basis of explanatory variables.

Classical examples

- ✓ Assess whether a bank transaction is fraudulent or not
- ✓ Recognize hand-written numbers
- ✓ Identify the type of cancer a patient has

Unsupervised Classification

Context

- ✓ The number of groups is unknown.
- ✓ There is no certainty about group membership for any observation.

The objective is to classify data in homogeneous groups using covariates.

Examples

- ✓ In biology: animal taxonomy;
- ✓ In psychology: types of personality in a group;
- ✓ In text mining: detecting spam or classifying emails;
- ✓ In insurance: clientele segmentation to reflect risk types.

Main Types of Techniques

List of most common clustering techniques

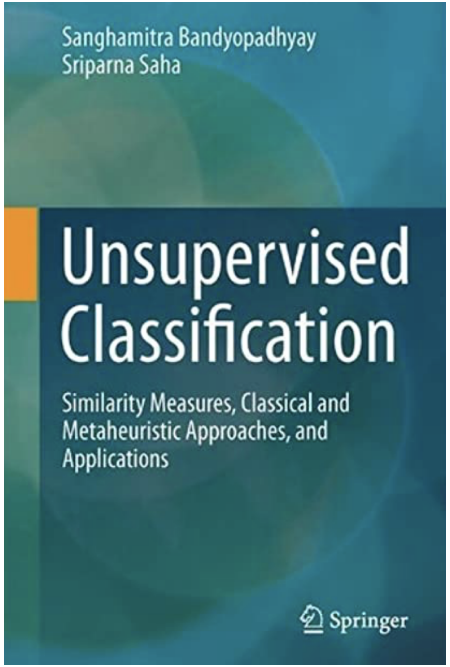
- ✓ hierarchical clustering;
- ✓ non-hierarchical clustering, e.g., the "k-means" algorithm;
- ✓ density-based clustering;
- ✓ clustering based on stochastic models, e.g., normal mixtures.

Our Objective

We will first consider the method of k-means.

Next, we will describe ascending algorithms for hierarchical clustering such as

- ✓ single and complete linkage;
- ✓ average and median method;
- ✓ centroid and Ward's method.



For more details, refer to the book by Bandyopadhyay and Saha (2013).

Critical First Step

Before observations can be clustered into homogeneous groups, one must define what it means for observations to be similar or not.

Therefore, one must quantify the similarity between two observations through a notion of distance.

This is often the most difficult step of the process. Nevertheless, it is an essential first step in a clustering algorithm.

When the observations consist of vectors in $\mathbb{R}^p$, then the Euclidean distance may be a reasonable choice.

Questions

What can be done when the observations consist of dichotomous variables (yes/no, M/F, present/absent, etc.)?

What about categorical variables, images, texts or — worse still — a mixture of all this?

Example: n individuals whose gender, age, income, and education level are provided, along with a one-paragraph job description.

Several measures have been developed with specific applications in mind, based on years of trial and error.

Some of the most classical options will be reviewed; they cover a broad range of situations.

Distance Measure

A distance measure d is a function that satisfies the following properties for all $i, j, k \in \{1, \ldots, n\}$:

1. Non-negativity: $d(i, j) \geq 0$;
2. Identity of indiscernibles: $d(i, j) = 0 \quad \Leftrightarrow \quad i = j$;
   impossible to see or clearly distinguish
3. Symmetry: $d(i, j) = d(j, i)$;
4. Triangle inequality: $d(i, k) \leq d(i, j) + d(j, k)$.

Classical Choices of Distance

A standard choice is the $\mathcal{L}_q$ distance between any vectors $x_i, x_j \in \mathbb{R}^p$, viz.

$$\|x_i - x_j\|_q = \left(\sum_{k=1}^{p} |x_{ik} - x_{jk}|^q \right)^{1/q},$$

where $x_i = (x_{i1}, \ldots, x_{ip})$ and $x_j = (x_{j1}, \ldots, x_{jp})$.

In particular, the Euclidean distance corresponds to the case $q = 2$, viz.

$$\|x_i - x_j\|_2 = \sqrt{\sum_{k=1}^{p} |x_{ik} - x_{jk}|^2}.$$

Note, however, that the $\mathcal{L}_q$ distance is not scale-invariant.

Numerical Example

To show that the $\mathcal{L}_2$ distance is not scale-invariant, consider the following data set:

Object	Weight (g)	Size (cm)
1	10	7
2	20	2
3	30	10

One finds

$$d_{12} \approx 11.2, \quad d_{13} \approx 20.2, \quad d_{23} \approx 12.8.$$

If the size of the objects is given in mm instead, then

$$d_{12} \approx 51.0, \quad d_{13} \approx 36.1, \quad d_{23} \approx 80.6.$$

So is the first object closer to the 2nd or to the 3rd object?

Standardization

To avoid this problem, it is often recommended to work with the distance between the standardized variables, viz.

$$d^2(x_i, x_j) = \sum_{k=1}^{p} \left(\frac{x_{ik} - \mu_k}{s_k} - \frac{x_{jk} - \mu_k}{s_k} \right)^2 = \sum_{k=1}^{p} \left(\frac{x_{ik} - x_{jk}}{s_k} \right)^2,$$

where

$$\mu_k = \text{mean of variable } k,$$

and

$$s_k = \text{standard deviation for variable } k.$$

Similarity Indices

A similarity index s is a function which satisfies the following properties for all $i, j, k \in \{1, \ldots, n\}$:

1. Non-negativity: $s(i,j) \geq 0$.
2. Symmetry: $s(i,j) = s(j,i)$.
3. Upper bound: $s(i,i) = 1 \geq s(i,j)$.

A distance measure can be transformed into a similarity index by setting

$$s(i,j) = \frac{1}{1 + d(i,j)}.$$

The reverse is not necessarily true, as the triangle inequality may fail.

A dissimilarity index is induced by $d^*(i,j) = 1 - s(i,j)$.

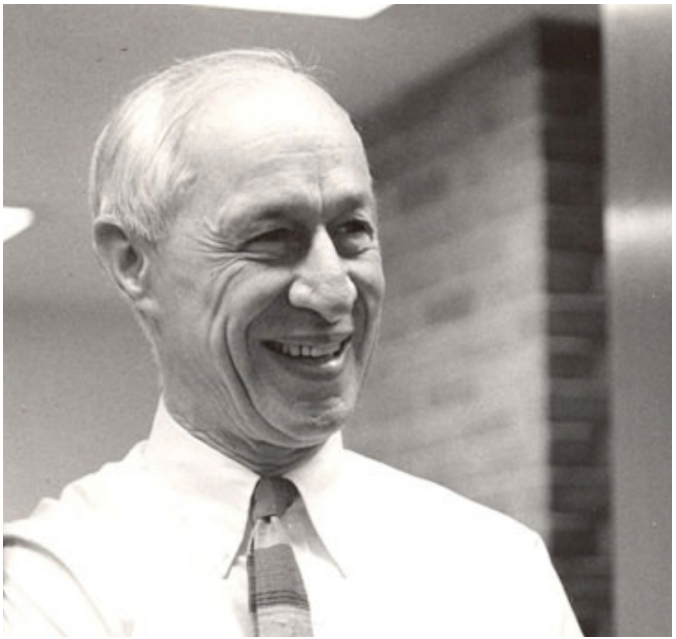
Main Types of Variables (1–2)

Numerical/Real-valued Variables: A variable whose numerical value measures something quantifiable and such that differences between values reflect differences between the objects, e.g., age, weight, income, etc.

Ordinal Variables: Variables that do not provide a precise quantification of a phenomenon but whose categories are naturally ordered, e.g., low, medium, high income, or scores on a 5-point or 7-point Likert scale.

Nominal Symmetric Variables: Qualitative variables (that are neither numerical nor nominal) whose categories are equally informative, e.g., gender (M/F), course section, etc.

Rensis Likert (1903–1981)



Rensis Likert was an American social psychologist mainly known for developing the Likert scale, an approach to creating a psychometrically sound scale based on responses to multiple questions or "items."

The scale has become a time-honored way to measure people's thoughts and feelings from opinion surveys to personality tests.

Main Types of Variables (2–2)

Nominal Asymmetric Variables: Qualitative variables whose possible categories do not contain the same level of information, e.g., because one of the categories is very frequent while others aren't.

Example: A variable that indicates whether someone is color-blind or not; two people share a common feature is they are both color-blind and not otherwise.

Other example: A financial transaction is fraudulent or not in an analysis where a very small proportion of transactions are.

Dichotomous Nominal Variables (1–3)

For vectors of symmetric dichotomous variables, one can use the matching coefficient ("proportion of agreements") between the vector components.

One answer may be coded as 0; the other may be coded as 1.

Assuming that p dichotomous variables are measured on two subjects i and j, one counts the number of variables for which the value is the same, viz.

$$m = \sum_{k=1}^{p} \mathbf{1}(x_{ik} = x_{jk}).$$

Similarity is then defined by $s(i, j) = m/p$.

Dichotomous Nominal Variables (2–3)

Consider, e.g., a questionnaire comprising 10 yes/no questions.

Suppose that 1 denotes "yes" and 0 means "no."

Individual	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
i	1	0	0	0	0	1	0	0	0	0
j	1	0	0	0	0	0	1	0	0	0

Here,

$$s(i,j) = 8/10 = 0.8,$$

because respondents i and j gave the same answer in 8 cases out of 10.

Dichotomous Nominal Variables (3–3)

For vectors of asymmetric dichotomous variables, assign 1 to the most important (or less frequent) outcome and 0 to the other.

One can then use Jaccard's index, defined by the number of variables for which i and j both take value 1, divided by the number of variables for which at least one of i or j is not equal to 0, viz.

$$J(i,j) = \frac{\sum_{k=1}^{p} x_{ik} x_{jk}}{\sum_{k=1}^{p} \{1 - (1 - x_{ik})(1 - x_{jk})\}}.$$

In the above example, one has $J(i,j) = 1/3 = 0.33$ because the two respondents gave the same answer for only one of the three questions where someone replied "1" (i.e., Q1, Q6, and Q7).

Paul Jaccard (1868–1944)

Paul Jaccard was a professor of botany and plant physiology at the ETH Zürich. He developed the Jaccard index of similarity (he called it "coefficient de communauté") and published it in 1901.

He also introduced the use of the species-to-genus ratio (he called it generic coefficient) in biogeography.

Polychotomous Nominal Variables

A variable with $M > 2$ values can be coded using $M - 1$ dichotomous variables.

For example, suppose that the possible answers are "Yes," "No," and "I don't know." One could then code the answers as follows:

	x_{i1}	x_{i2}
Yes	1	0
No	0	1
I don't know	0	0

The similarity between i and j can then be computed using the same methods as before.

Several Nominal Variables (1–2)

Consider q dichotomous or polychotomous variables of the same nature. Similarities are computed separately for each of them and then averaged.

For example, suppose that i and j have answered two questions for which they had three choices:

Individual	Q1a	Q1b	Q2a	Q2b
i	0	1	0	1
j	0	1	0	0

Similarity for each question, assuming that all categories are equally important: $s_{Q1}(i,j) = 2/2$, $s_{Q2}(i,j) = 1/2$.

Average of the two similarities: $s(i,j) = \{s_{Q1}(i,j) + s_{Q2}(i,j)\}/2 = 3/4$.

Several Nominal Variables (1–2)

Now assume that the variables corresponding to Question 1 are symmetric and that those corresponding to Question 2 are not.

Individual	Q1a	Q1b	Q2a	Q2b
i	0	1	0	1
j	0	1	0	0

Similarity $s_{Q1}(i,j) = 2/2$ remains unchanged. For Question 2, one takes $s_{Q2}(i,j) = 0/1$ because Q2a does not contribute to the calculation.

One then gets

$$s(i,j) = (2/2 + 0/1)/2 = 1/2.$$

Ordinal Variables

A numerical score is typically assigned to each category.

This score is treated as a numerical variable.

There is no set rule for assigning the scores, except that they should be positive and reflect the order of the categories.

For example, one could assign scores 1, 2, 3 to "low," "medium," and "high" income, respectively.

However, one could also decide to use 15,000, 50,000, 150,000.

In the absence of clear mathematical guidelines, the data analyst should use his/her judgment.

Gower Similarity

Given weights w_k assigned to variable k, set

$$G(i,j) = \sum_{k=1}^{p} w_k \gamma_k(i,j) s_k(i,j) \Big/ \sum_{k=1}^{p} w_k \gamma_k(i,j),$$

where $\gamma_k(i,j)$ and $s_k(i,j)$ are defined as follows:

- ✓ if k is quantitative or ordinal:

$$\gamma_k(i,j) = 1 \quad \text{and} \quad s_k(i,j) = 1 - |x_{ik} - x_{jk}|/r_k;$$

- ✓ if k is nominal symmetric: $\gamma_k(i,j) = 1$ and $s_k(i,j) = \mathbf{1}(x_{ik} = x_{jk})$;
- ✓ if k is nominal asymmetric:

$$\gamma_k(i,j) = \{1 - (1 - x_{ik})(1 - x_{jk})\} \text{ and } s_k(i,j) = \mathbf{1}(x_{ik} = x_{jk}).$$

Remarks

For numerical/ordinal variables, the value r_k in the similarity is the range of the variable k.

It is strongly recommended to standardize the numerical/ordinal variables at the outset.

The weights w_k make it possible to quantify the importance of each variable in the similarity measure.

Example (1–3)

The answers provided by subjects i and j have been recoded in such a way that variable 1 is numerical, variable 2 is ordinal, variables 3 and 4 are dichotomous are associated to a symmetric variable, while variable 5 is dichotomous and asymmetric.

The values for variables 1 and 2 have been standardized and we assume that they take values between -2.5 and 2.5 (so the range is 5).

Individual	Q1	Q2	Q3	Q4	Q5
i	1	2	0	1	0
j	-1	1	0	0	1

Example (2–3)

Suppose that we want Question 1 to be three times as important as the others in the similarity measure. Then

$$w_1 = 3, \quad w_2 = w_3 = w_4 = w_5 = 1$$

and

$$\gamma_1(i,j) = \gamma_2(i,j) = \gamma_3(i,j) = \gamma_4(i,j) = 1$$

with

$$\gamma_5(i,j) = 1 - (1 - x_{i5})(1 - x_{j5}) = 1 - (1-1)(1-0) = 1,$$

so

$$s_1(i,j) = 1 - |1 - (-1)|/5 = 3/5, \quad s_2(i,j) = 1 - |2-1|/5 = 4/5,$$

and

$$s_3(i,j) = 1/1, \quad s_4(i,j) = 0/1, \quad s_5(i,j) = 0/1.$$

Example (3–3)

Therefore,

$$G(i,j) = \sum_{k=1}^{5} w_k \gamma_k(i,j) s_k(i,j) \Big/ \sum_{k=1}^{5} w_k \gamma_k(i,j)$$

$$= \left(3 \times 1 \times \frac{3}{5} + 1 \times 1 \times \frac{4}{5} + 1 \times 1 \times \frac{1}{1} + 1 \times 1 \times \frac{0}{1} + 1 \times 1 \times \frac{0}{1}\right)$$

$$\div (3 \times 1 + 1 \times 1 + 1 \times 1 + 1 \times 1 + 1 \times 1)$$

$$= \frac{18/5}{7} = \frac{18}{35} \approx 0.514.$$

Text Classification (1–2)

Text classification refers to a way of partitioning n written documents, e.g., webpages, emails, complaints, legal documents, etc.

Matrix of documents by terms:

- ✓ n rows $=$ n documents;
- ✓ p columns $=$ p words or expressions present in all documents.

Words that contain no information, called "stopwords," must be eliminated, e.g., articles, prepositions, conjunctions, pronouns, etc.

Text Classification (2–2)

Only the stem of the words is kept ("stemming"), e.g., "meteo" for "meteorology" and "meteorological."

At the intersection of row i and column k, one has the number of times (the frequency) with which the kth word appeared in the ith document.

As the number p of columns is high, the matrix becomes sparse, i.e., most of its entries are equal to 0.

Other possible codings: remplace the non-zero frequencies by 1, tf-idf, etc.

Similarity Measures

If coding of the presence/absence type is used along with Jaccard's index, then practically all pairs of documents will have a very low or zero similarity.

This is useful to detect plagiarism.

Cos Similarity

$$s_{\cos}(i,j) = \frac{\sum_{k=1}^{p} w_k \, x_{ik} x_{jk}}{\sqrt{\sum_{k=1}^{p} w_k x_{ik}^2 \; \sum_{k=1}^{p} w_k \, x_{jk}^2}}.$$

What We Learned Before Can Serve Again!

Dimension reduction techniques can simplify the calculation of the similarity/distance between observations.

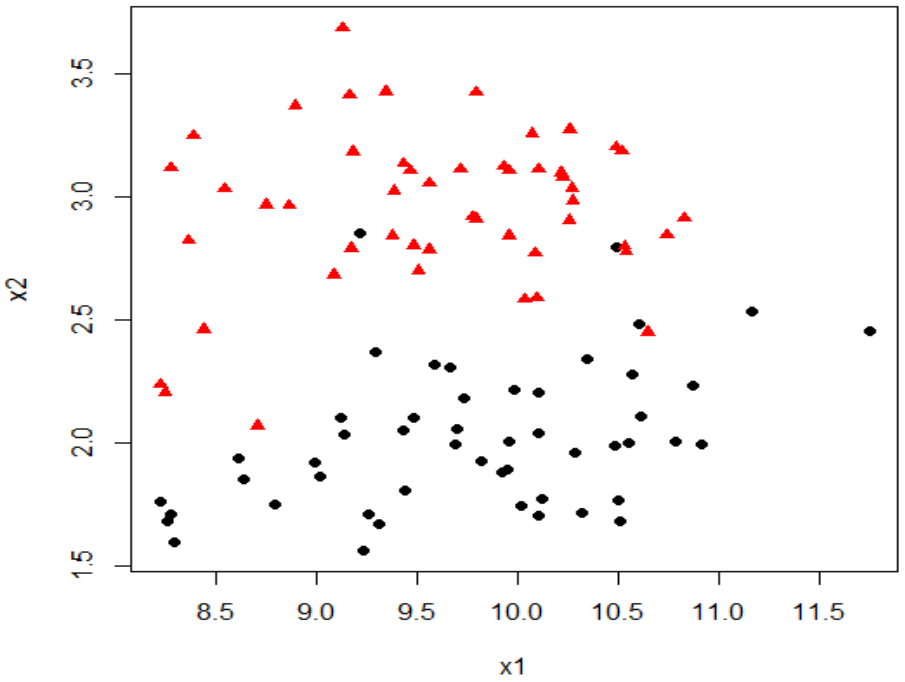
For example, if there are p numerical/ordinal variables then a PCA can be used to compute the score of each observation in $k \ll p$ principal axes before computing the Euclidean distance.

As another example, a questionnaire with Q multiple choice (i.e., with Q categorical variables) could be processed by MCA and the score of each observation in the first k first principal axes could be used to compute the Euclidean distance.

Such strategies are generally fruitful but it may happen that groups that are nearly separated in dimension p are not so easily distinguished in dimension $k < p$.

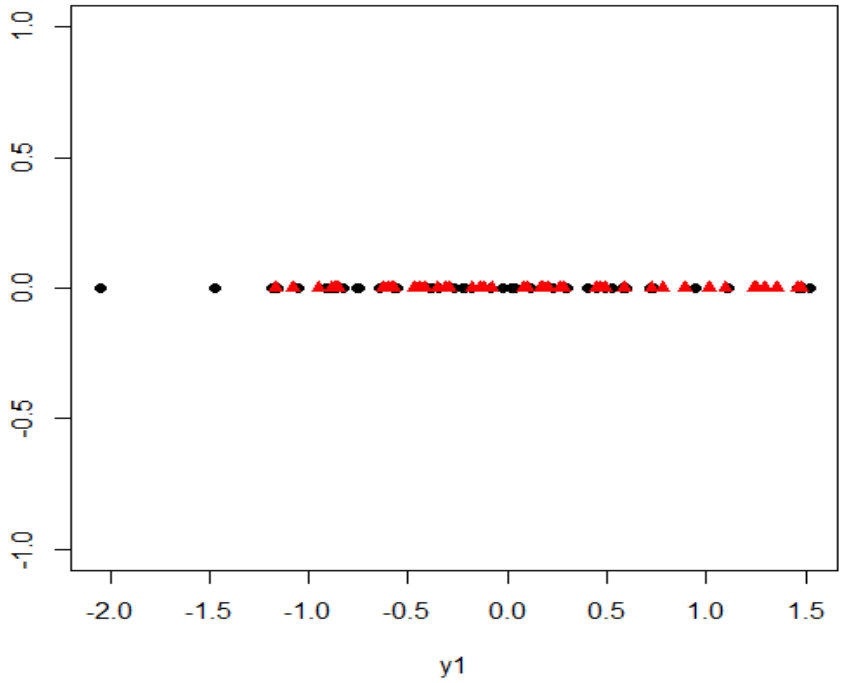
From $p = 2$ to $k = 1$ (1–2)

50 pairs of observations (x_1, x_2), divided into two groups.



From $p = 2$ to $k = 1$ (2–2)

Same 50 pairs of observations ranked as per the first principal component.



Mathematical Formulation of the Objective Function

The objective is to assign n observations to K groups (classes, categories) in such a way that

- ✓ observations within the same group are as similar as possible;
- ✓ observations from different groups are as different as possible

To this end, a classifier C will be sought which meets these two conditions in some sense.

The map C maps every observation i into a group $C(i)$, viz.

$$\begin{aligned} C : \{1, \ldots, n\} &\rightarrow \{1, \ldots, K\} \\ i &\mapsto C(i). \end{aligned}$$

Objective Function

It can be shown that the group assignment C which minimizes the objective function below meets the two requirements:

$$W(C) = \sum_{k=1}^{K} \sum_{i:C(i)=k} \sum_{j:C(j)=k} d^*(x_i, x_j),$$

where $d^*(x_i, x_j)$ is the dissimilarity between observations i and j.

However, one cannot find C by trying out all possible assignments of n objects into K classes because there are too many, viz.

$$\frac{1}{K!} \sum_{k=1}^{K} (-1)^{K-k} \binom{K}{k} k^n.$$

A Glimpse at the Size of the Problem

Sample Size (n)	Number of Groups (K)	Number of Assignments
10	2	511
10	3	9 330
10	4	34 105
20	2	524 287
20	3	580 606 446
20	4	45 232 115 901
100	2	6.338253e+29
100	3	8.589625e+46
100	4	6.695575e+58
1000	2	5.357543e+300
1000	3	"Inf" according to R
1000	4	"Inf" according to R

Greedy Algorithms

As all options can't be tried, one must resort to greedy algorithms, e.g.,

- ✓ k-means algorithm;
- ✓ hierarchical clustering algorithms, e.g.,
  - ✓ descending algorithms;
  - ✓ ascending algorithms.

Such algorithms find an allocation rule C that minimizes $W(C)$ on a restricted space.

They do not guarantee that the global minimum has been reached.

k-means Algorithm

Set-up

- ✓ We have p ordinal/quantitative variables at our disposal (which are usually standardized).
- ✓ We wish to partition n observations into a predetermined number K of groups.

Advantage

The algorithm is fairly simple and at each step, the value of $W(C)$ is guaranteed to decrease.

k-means Algorithm

1. Choose the number K of classes or groups.
2. Divide the n observations randomly into K groups.
3. Compute the mean vector of each group $k \in \{1, \ldots, K\}$, viz.

$$\mu_k = \frac{1}{N_k} \sum_{i:C(i)=k} x_i,$$

   where N_k is the number of observations in Group k.
4. Compute the distance from the observations to the K mean vectors.
5. Assign each n observation to the group with the closest mean.
6. Repeat Steps 3 to 5 until convergence.

Example (1–10)

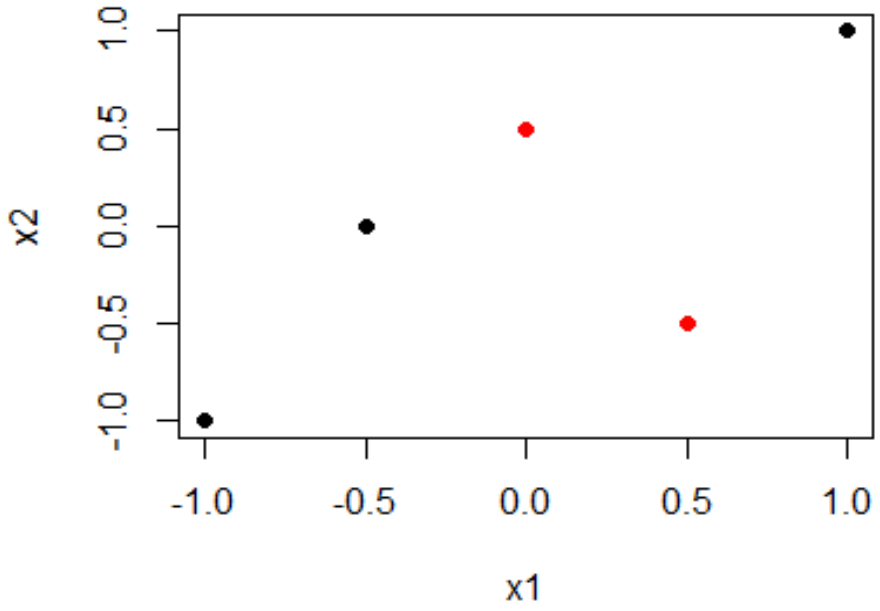
This small artificial example is intended to explain in detail how the algorithm works.

Suppose that it is desired to partition 5 observations into $K = 2$ groups.

i	1	2	3	4	5
x_{i1}	-1	-0.5	0	0.5	1
x_{i2}	-1	0	0.5	-0.5	1

Observations 1, 2 and 5 are initially assigned to Group 1 at random; the remaining observations, 3 and 4, form Group 2.

Example (2–10)



Example (3–10)

The group means are computed as follows:

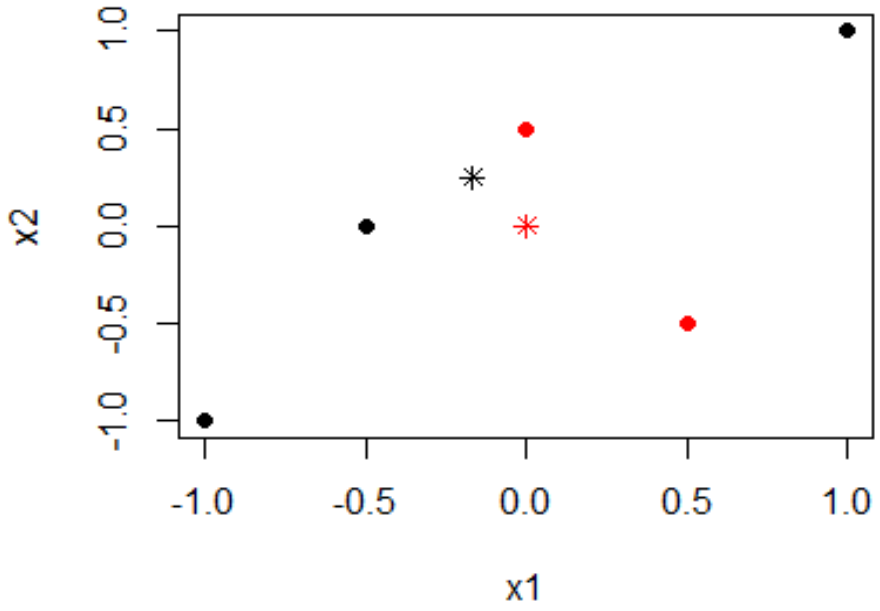
$$\mu_1 = \frac{1}{3}\left\{\begin{pmatrix}-1\\-1\end{pmatrix} + \begin{pmatrix}-0.5\\0\end{pmatrix} + \begin{pmatrix}1\\1\end{pmatrix}\right\} = \begin{pmatrix}-1/6\\0\end{pmatrix},$$
$$\mu_2 = \frac{1}{2}\left\{\begin{pmatrix}0\\0.5\end{pmatrix} + \begin{pmatrix}0.5\\-0.5\end{pmatrix}\right\} = \begin{pmatrix}1/4\\0\end{pmatrix}.$$

The distances between the observations and the group means are given by

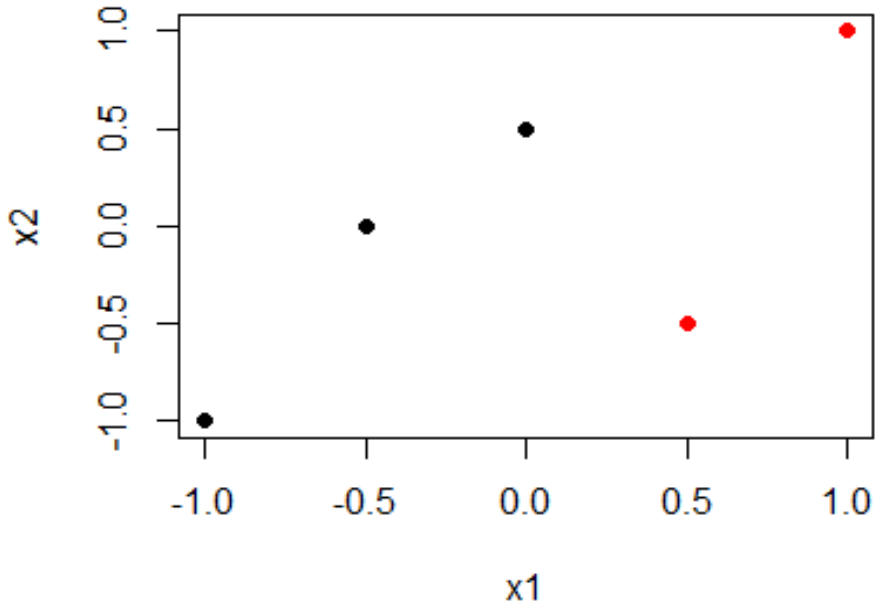
i	1	2	3	4	5
$d^2(i, \mu_1)$	**1.69**	**0.11**	**0.28**	0.69	2.36
$d^2(i, \mu_2)$	2.56	0.56	0.31	**0.31**	**1.57**

The group to which each observation belongs is highlighted in bold.

Example (4–10)



Example (5–10)



Example (6–10)

Step 2:

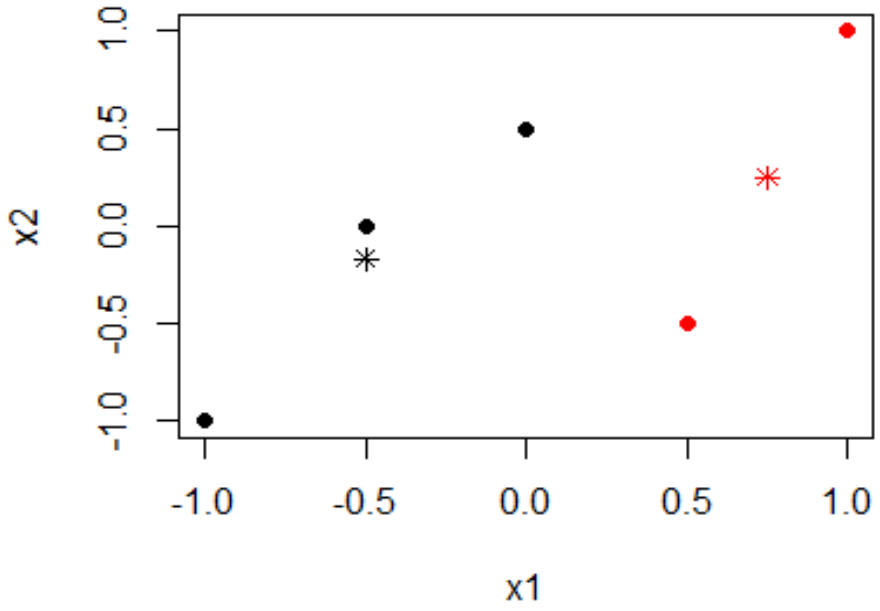
$$\mu_1 = \frac{1}{3}\left\{\begin{pmatrix}-1\\-1\end{pmatrix} + \begin{pmatrix}-0.5\\0\end{pmatrix} + \begin{pmatrix}0\\0.5\end{pmatrix}\right\} = \begin{pmatrix}-1/2\\-1/6\end{pmatrix},$$

$$\mu_2 = \frac{1}{2}\left\{\begin{pmatrix}0.5\\-0.5\end{pmatrix} + \begin{pmatrix}1\\1\end{pmatrix}\right\} = \begin{pmatrix}3/4\\1/4\end{pmatrix}.$$

i	1	2	3	4	5
$d^2(i, \mu_1)$	**0.94**	**0.03**	0.69	1.11	3.61
$d^2(i, \mu_2)$	4.63	1.63	**0.63**	**0.63**	**0.63**

Observation no. 3 is now in Group 2.

Example (7–10)



Example (8–10)

Step 3:

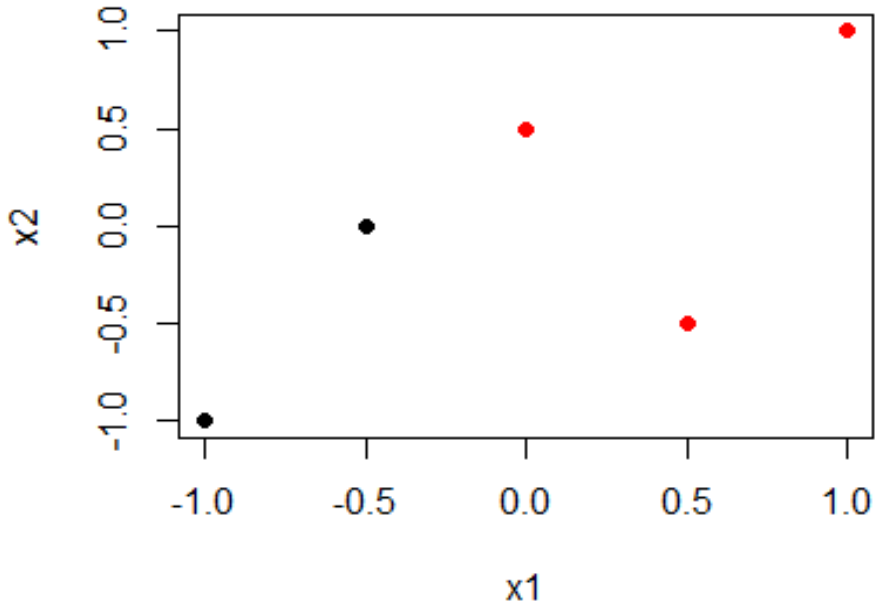
$$\mu_1 = \frac{1}{2}\left\{\begin{pmatrix}-1\\-1\end{pmatrix} + \begin{pmatrix}-0.5\\0\end{pmatrix}\right\} = \begin{pmatrix}-3/4\\-1/2\end{pmatrix},$$
$$\mu_2 = \frac{1}{3}\left\{\begin{pmatrix}0\\0.5\end{pmatrix} + \begin{pmatrix}0.5\\-0.5\end{pmatrix} + \begin{pmatrix}1\\1\end{pmatrix}\right\} = \begin{pmatrix}1/2\\1/3\end{pmatrix}.$$

i	1	2	3	4	5
$d^2(i, \mu_1)$	**0.31**	**0.31**	1.56	1.56	5.31
$d^2(i, \mu_2)$	4.03	1.11	**0.28**	**0.69**	**0.69**

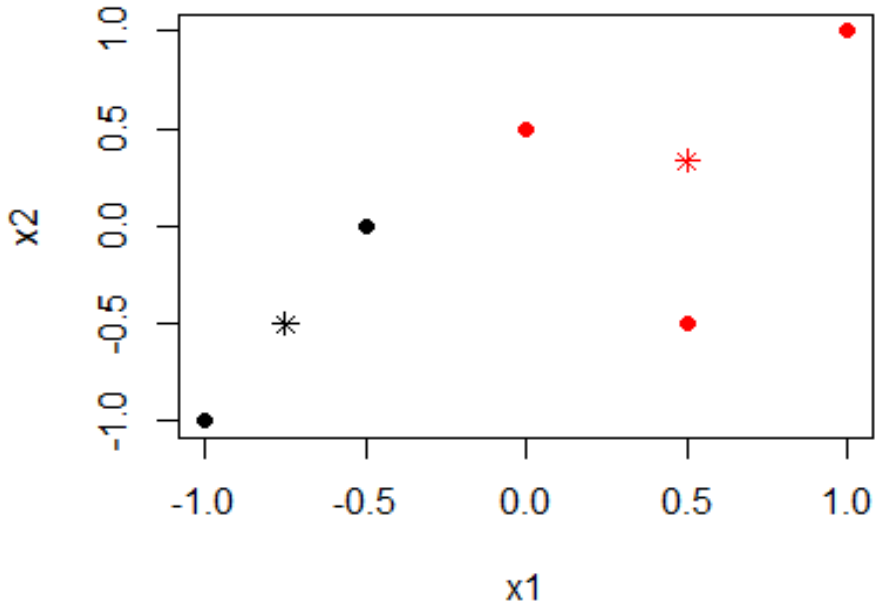
As group assignment does not change, the algorithm halts.

Suggestion: Try calculating $W(C)$ after each step.

Example (9–10)



Example (10–10)



Practical Example of k-means Algorithm (1–11)

As an illustration of the k-means algorithm, consider crime rates per 100,000 people in seven categories for each of the 50 US states in 1977.

```
Alabama    14.2 25.2  96.8 278.3 1135.5 1881.9 280.7
Alaska     10.8 51.6  96.8 284.0 1331.7 3369.8 753.3
Arizona     9.5 34.2 138.2 312.3 2346.1 4467.4 439.5
Arkansas    8.8 27.6  83.2 203.4  972.6 1862.1 183.4
```

The variables are

- ✓ Murder
- ✓ Rape
- ✓ Robbery
- ✓ Assault
- ✓ Burglary
- ✓ Larceny
- ✓ Auto Theft

Practical Example of k-means Algorithm (2–11)

First we read in the data:

```
crime <- read.table("crime.txt", header=T, sep="\t")
```

Next, we identify column 1 as containing names; it is not a variable.

```
states <- crime[,1]
crime <- crime[,-1]
crime.st <- scale(crime)
dimnames(crime.st) <- list(states, names(crime))
```

Practical Example of k-means Algorithm (3–11)

Suppose we want to constitute 5 groups. The command `kmeans` in R can be used to carry out the work:

```
crime.km5 <- kmeans(crime.st, centers=5)
```

#class

```
[1]  9 10  9 12 10
```

The following command allows us to determine how many states are in each group:

```
crime.km5$size
```

Practical Example of k-means Algorithm (4–11)

The composition of each group can be determined as follows:

```
states[crime.km5$cluster==1]

[1] "Iowa"      "Kentucky"  "Nebraska"  "NewHamp"   "NDakota"
[6] "Penn"      "SDakota"   "WVirginia" "Wisconsin"

states[crime.km5$cluster==2]

 [1] "Alaska"   "Arizona"  "Califor"  "Colorado" "Florida"
 [6] "Maryland" "Michigan" "Nevada"   "NewYork"  "Oregon"
```

Practical Example of *k*-means Algorithm (5–11)

The composition of each group can be determined as follows:

```
states[crime.km5$cluster==3]

[1] "Connec"     "Delaware"   "Hawaii"     "Illinois"
[5] "Mass"       "NewJersey"  "Ohio"       "RhodeIsl"
[9] "Washington"

states[crime.km5$cluster==4]

 [1] "Alabama"   "Arkansas"  "Georgia"   "Louisiana" "Missi"
 [6] "Missouri"  "NewMexico" "NCarolina" "Oklahoma"  "SCarolina"
[11] "Tennessee" "Texas"

states[crime.km5$cluster==5]

 [1] "Idaho"     "Indiana"   "Kansas"    "Maine"     "Minnesota"
 [6] "Montana"   "Utah"      "Vermont"   "Virginia"  "Wyoming"
```

Practical Example of k-means Algorithm (6–11)

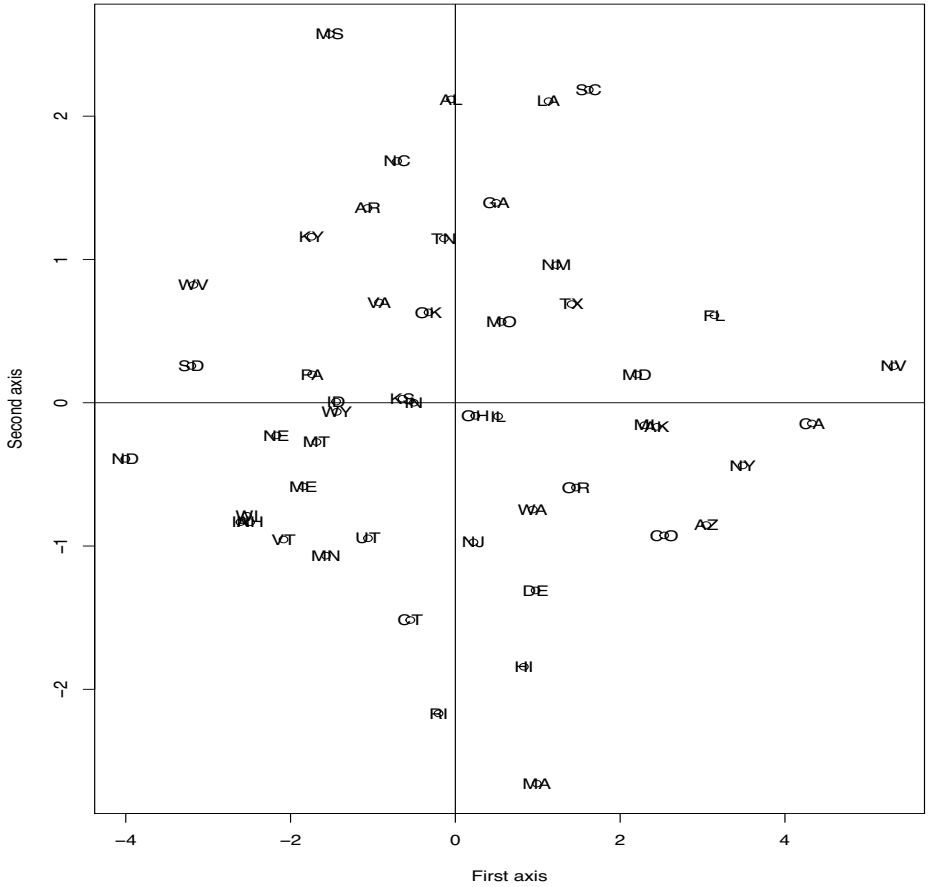
To visualize the groups, one can use the biplot from principal component analysis.

```
crime.acp <- princomp(~Murder + Rape + Robbery + Assault + Burglary
   + Larceny + Auto_Theft, cor=T, data = crime)
```

A biplot is then produced as follows. The result is given on the next slide.

```
plot(crime.acp$scores[,1], crime.acp$scores[,2],
     xlab = "First axis", ylab = "Second axis")
abline(h = 0,v = 0)
text(crime.acp$scores[,1], crime.acp$scores[,2],
     labels = states, cex = 1)
```

Practical Example of k-means Algorithm (7–11)



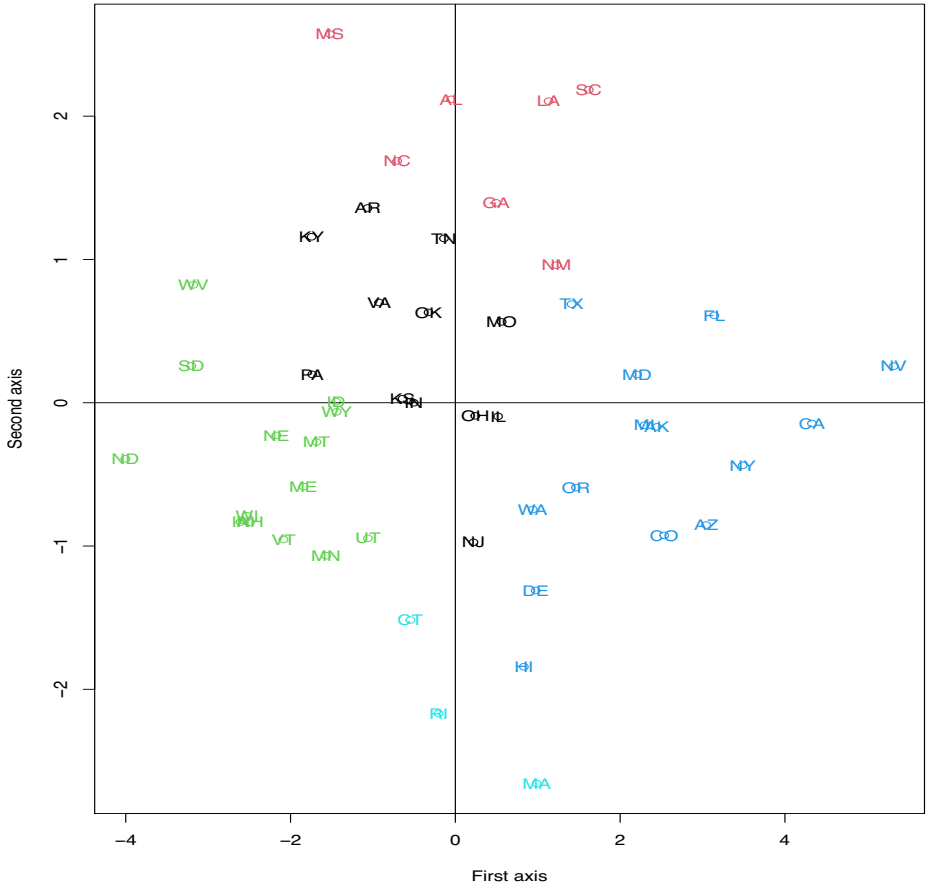
Practical Example of k-means Algorithm (8–11)

To visualize the groups, one can use different colors as follows.

```
plot(crime.acp$scores[,1], crime.acp$scores[,2],
     xlab = "First axis", ylab = "Second axis",
     col = crime.km5$cluster)
abline(h = 0, v = 0)
text(crime.acp$scores[,1], crime.acp$scores[,2],
     labels = states, cex = 1, col = crime.km5$cluster)
```

The resulting biplot is given on the next slide.

Practical Example of k-means Algorithm (9–11)



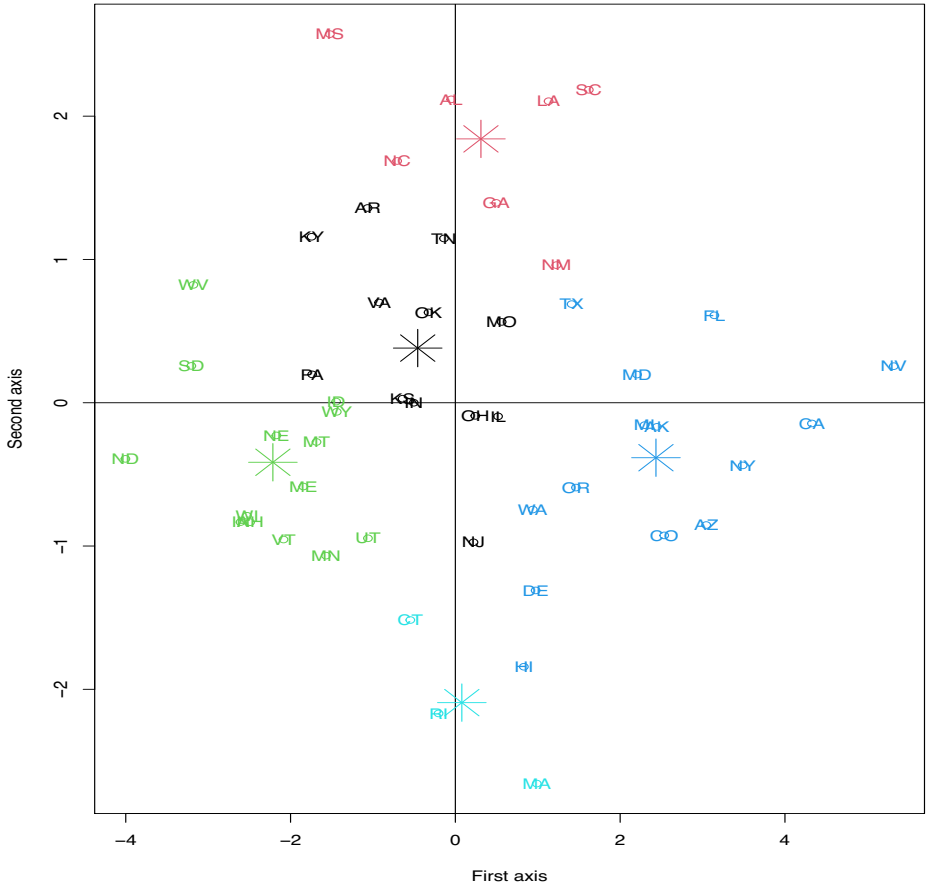
Practical Example of k-means Algorithm (10–11)

Group centers can be added to the biplot as follows.

```
means <- colMeans(crime.st)
means <- matrix(means, ncol = 7, row = 5, byrow = T)
sds <- sqrt(diag(var(crime.st)))
sds <- matrix(sds,ncol=7,nrow=5,byrow=T)
centers.km5 <- (crime.km5$centers - means)/sds
centers.axe1 <- centers.km5%*%loadings(crime.acp)[,1]
centers.axe2 <- centers.km5%*%loadings(crime.acp)[,2]
points(centers.axe1, centers.axe2, col = 1:5, pch = 8, cex = 4)
```

The resulting biplot is given on the next slide.

Practical Example of k-means Algorithm (11–11)



Hierarchical Clustering

Because it is based on the Euclidean distance, the k-means algorithm cannot be used when

- ✓ the observations do not consist of numerical or ordinal variables;
- ✓ it is not clear what value of K should be used.

As an alternative, one can call on hierarchical clustering algorithms.

Such algorithms lead to partitions of the data that are nested in one another. The algorithm can be ascending or descending.

In both cases, one gets n hierarchical partitions comprising 1 to n groups.

Descending Algorithms

Descending algorithms proceed as follows:

At Step 1, all the observations are in the same group of n observations.

At each successive step, the least homogeneous group is divided into two parts.

At the end, i.e., after n steps, each observation constitutes its own group, i.e., there are n groups consisting of a single observation each.

Remarks

Such an algorithm does not result in a single partition but in n partitions:

- ✓ one partition with a single group;
- ✓ one partition with two groups, etc.
- ✓ one partition with n groups.

Criteria can then be used to choose one of the n partitions proposed by the algorithm.

Descending algorithms require a lot of computing time because once the group to be split is identified, one must determine how it should be split.

For this reason, descending algorithms are not used as often as ascending algorithms.

Ascending Algorithms

Ascending algorithms proceed in exactly the opposite way:

At Step 1, each observation forms its own group, i.e., there are at first n groups each containing a single observation. $\binom{50}{2} = 1225$

At each step, the two groups which are most alike are merged. $\binom{49}{2}$ ...

After n steps, one has a single group consisting of n observations.

In addition, hierarchical clustering algorithms depend on the way in which

- ✓ distances or similarities between two observations are measured;
- ✓ distances or similarities between two groups are measured.

Distance and Similarity Between Two Groups

In order to implement a hierarchical clustering algorithm, one must define

$$d(R, S) = \text{the distance between any two groups } R \text{ and } S \text{ of observations.}$$

For example, given three points 1, 2, 3 and a notion of distance between them, how does one measure the distance between two sets, viz.

$$\{1, 2\} \quad \longleftrightarrow \quad \{3\}.$$

There are several ways in which such a distance could be computed.

Some of them are listed next. This is followed by an illustration.

Single Linkage (1–2)

Distance between two groups:

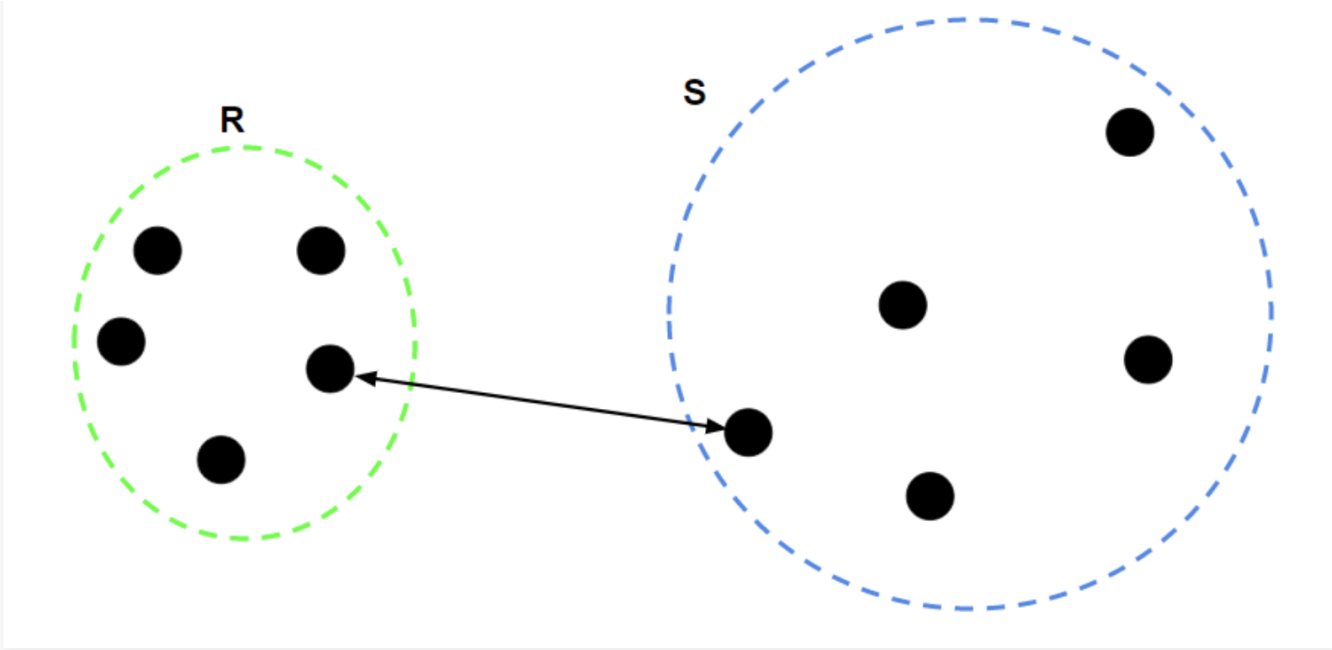
$$d(R, S) = \min \{d_{ij} : i \in R, j \in S\}.$$

If similarity indices are used instead, then

$$s(R, S) = \max \{s_{ij} : i \in R, j \in S\}.$$

Therefore, by definition, the distance/similarity between two groups of observations is given by the distance/similarity between the two group representatives that are closed or most similar.

Single Linkage (2–2)



Complete Linkage (1–2)

Distance between two groups:

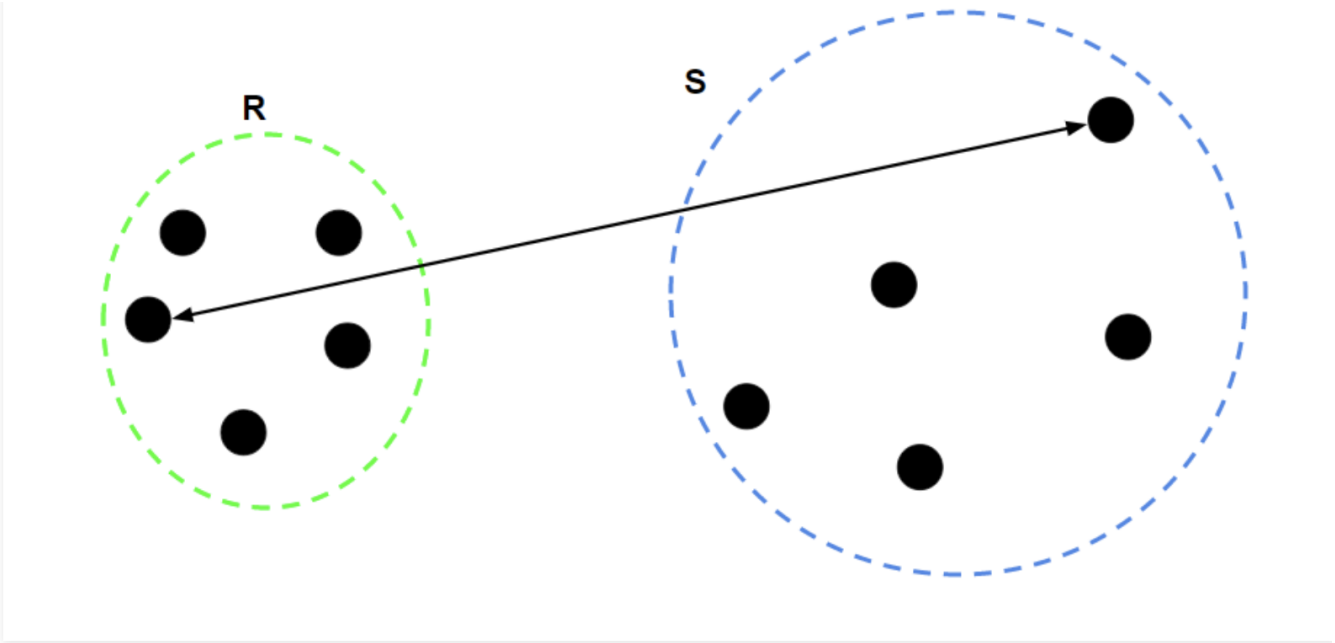
$$d(R, S) = \max \{d_{ij} : i \in R, j \in S\}.$$

If similarity indices are used instead, then

$$s(R, S) = \min \{s_{ij} : i \in R, j \in S\}.$$

Therefore, by definition, the distance/similarity between two groups of observations is given by the distance/similarity between the two group representatives that are furthest apart or most dissimilar.

Complete Linkage (2–2)



Average Linkage (1–2)

Distance between two groups:

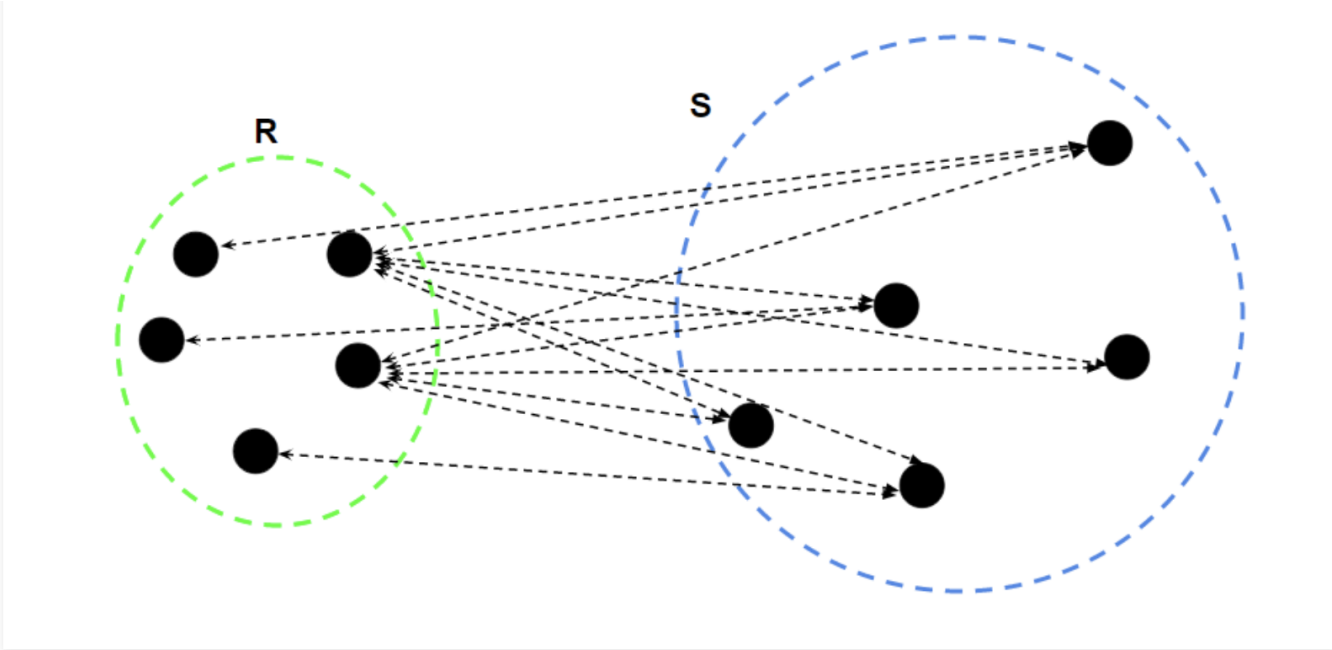
$$d(R,S) = \frac{1}{n_R n_S} \sum_{i \in R} \sum_{j \in S} d(x_i, x_j),$$

where n_R is the number of observations in group R and n_S is the number of observations in group S.

To compute $d(R,S)$, one must determine the $n_R \times n_S$ distances [similarities] between all pairs of points, one from each group.

Once this is done, one takes the average of the distances [similarities] as a measure of the distance [similarity] between the two groups.

Average Linkage (2–2)



Centroid Method (1–2)

Distance between two groups:

$$d(R, S) = d(\bar{x}_R, \bar{x}_S),$$

where

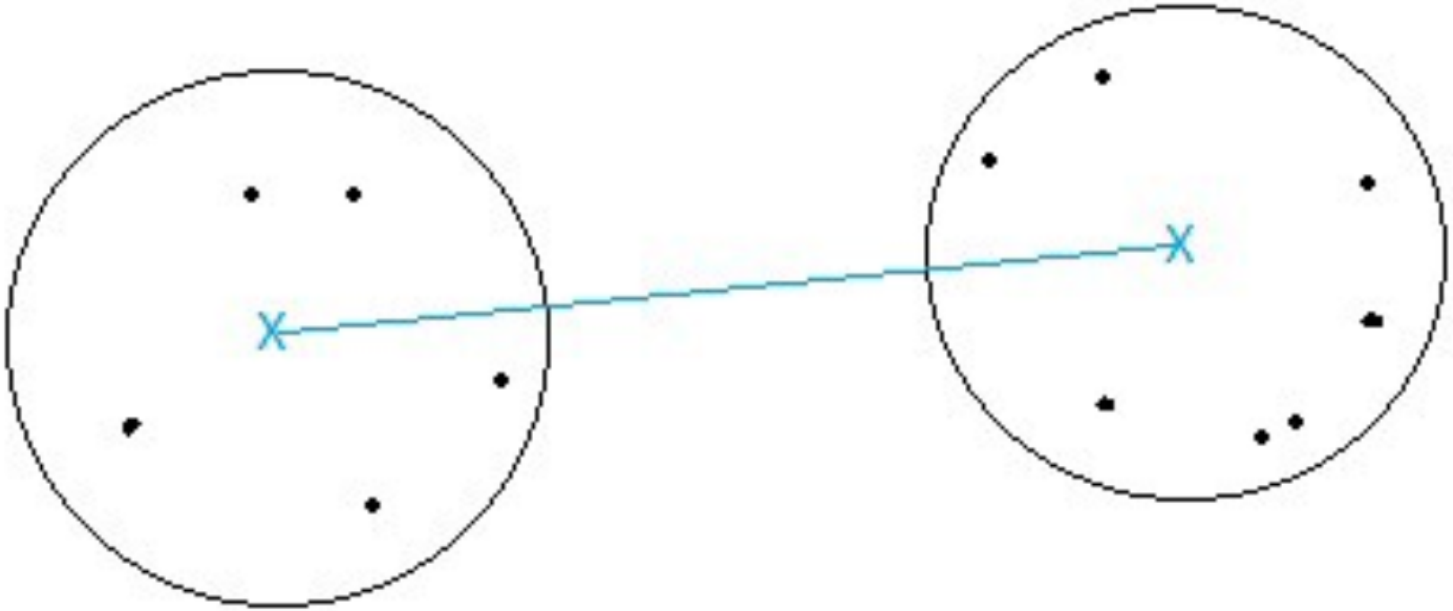
$$\bar{x}_R = \frac{1}{n_R} \sum_{i \in R} x_i, \quad \bar{x}_S = \frac{1}{n_S} \sum_{j \in S} x_j.$$

The mean $\bar{x}_{R \cup S}$ of the group resulting from the merger of groups R and S can be computed as follows:

$$\bar{x}_{R \cup S} = \frac{n_R \bar{x}_R + n_S \bar{x}_S}{n_R + n_S}.$$

weighted average

Centroid Method (2–2)



Median Method

At any given step, we always have at our disposal the distance between the already existing groups.

We merge the two groups which are closest [most similar], say R and S, to form the group $R \cup S$.

We then update the distance matrix by defining the distance between the new group $R \cup S$ and any other group T by setting

$$d(R \cup S, T) = \{d(R, T) + d(S, T)\}/2 - d(R, S)/4.$$

Ward's Method (1–2)

Ward's method is a variant of the centroid method which takes into account the size of the groups.

It is designed to be optimal if the n vectors $x_1, \ldots, x_n$ are multivariate Normal with K different means but the same variance-covariance matrix.

This method is based on the sums of squares

$$SS_R = \sum_{i \in R} (x_i - \bar{x}_R)^\top (x_i - \bar{x}_R),$$
$$SS_S = \sum_{j \in S} (x_j - \bar{x}_S)^\top (x_j - \bar{x}_S),$$
$$SS_{R \cup S} = \sum_{k \in R \cup S} (x_k - \bar{x}_{R \cup S})^\top (x_k - \bar{x}_{R \cup S}),$$

where $\bar{x}_R$, $\bar{x}_S$ and $\bar{x}_{R \cup S}$ are computed as in the centroid method.

Ward's Method (2–2)

Ward's method consists of merging the classes R and S for which

$$I_{R\cup S} = SS_{R\cup S} - SS_R - SS_S = \frac{n_R n_S}{n_R + n_S} (\bar{x}_R - \bar{x}_S)^\top (\bar{x}_R - \bar{x}_S)$$
$$= \frac{d^2(\bar{x}_R, \bar{x}_S)}{1/n_R + 1/n_S}$$

is minimal.

This method was developed by Joe H. Ward Jr., who was a research scientist for the US Air Force early in his career. He later collaborated with Earl Jennings at The University of Texas at Austin to author *Introduction to Linear Models* in 1973.

An elementary school bears Ward's name in San Antonio, Texas, to honor his many years of volunteer work with the district.

Joe H. Ward Jr. (1927–2011)

Flexible Method (1–3)

It can be observed that for several of the most commonly used methods, one has

$$d(T, R \cup S) = \alpha_R \, d(T, R) + \alpha_S \, d(T, S) + \beta \, d(R, S) + \gamma \, |d(T, R) - d(T, S)|.$$

The appropriate values of α_R, α_S, β and γ are given in the table below.

Flexible Method (2–3)

Method	α_R	α_S	β	γ
Single	$1/2$	$1/2$	0	$-1/2$
Complete	$1/2$	$1/2$	0	$1/2$
Median	$1/2$	$1/2$	$-1/4$	0
Average	$\frac{n_R}{n_R+n_S}$	$\frac{n_S}{n_R+n_S}$	0	0
Centroid	$\frac{n_R}{n_R+n_S}$	$\frac{n_S}{n_R+n_S}$	$-\frac{n_R n_S}{n_R+n_S}$	0
Ward	$\frac{n_R+n_T}{n_R+n_S+n_T}$	$\frac{n_S+n_T}{n_R+n_S+n_T}$	$-\frac{n_T}{n_R+n_S+n_T}$	0

Flexible Method (3–3)

With the flexible method, one imposes arbitrarily the following constraints:

$$\alpha_R + \alpha_S + \beta = 1, \quad \alpha_R = \alpha_S, \quad \gamma = 0.$$

Accordingly,

$$\alpha_R = \alpha_S = (1 - \beta)/2$$

and it only remains to choose β.

It is often suggested to set

$$\beta = -0.25$$

except when there are reasons to suspect the presence of outliers. In the latter case, the recommendation is usually to take

$$\beta = -0.5.$$

An Overarching Principle (1–3)

In the previous segment, various ways of computing the distance or dissimilarity between two groups of observations were presented.

It can be observed that for several of these methods, the measure of distance between a set T and the union of two other sets R and S can be expressed in the form

$$d(T, R \cup S) = \alpha_R \, d(T, R) + \alpha_S \, d(T, S) + \beta \, d(R, S) + \gamma \, |d(T, R) - d(T, S)|,$$

for appropriate values of α_R, α_S, β and γ.

The correspondence is given in the table below for the various methods reviewing in the previous segment.

An Overarching Principle (2–3)

Method	α_R	α_S	β	γ
Single	$1/2$	$1/2$	0	$-1/2$
Complete	$1/2$	$1/2$	0	$1/2$
Median	$1/2$	$1/2$	$-1/4$	0
Average	$\frac{n_R}{n_R+n_S}$	$\frac{n_S}{n_R+n_S}$	0	0
Centroid	$\frac{n_R}{n_R+n_S}$	$\frac{n_S}{n_R+n_S}$	$-\frac{n_R n_S}{n_R+n_S}$	0
Ward	$\frac{n_R+n_T}{n_R+n_S+n_T}$	$\frac{n_S+n_T}{n_R+n_S+n_T}$	$-\frac{n_T}{n_R+n_S+n_T}$	0

An Overarching Principle (3–3)

With the flexible method, one imposes arbitrarily the following constraints:

$$\alpha_R + \alpha_S + \beta = 1, \quad \alpha_R = \alpha_S, \quad \gamma = 0.$$

Accordingly,

$$\alpha_R = \alpha_S = (1 - \beta)/2$$

and it only remains to choose β.

It is often suggested to set

$$\beta = -0.25$$

except when there are reasons to suspect the presence of outliers. In the latter case, the recommendation is usually to take

$$\beta = -0.5.$$

Illustration With the Crime Data

One must first construct a distance or a dissimilarity matrix, viz.

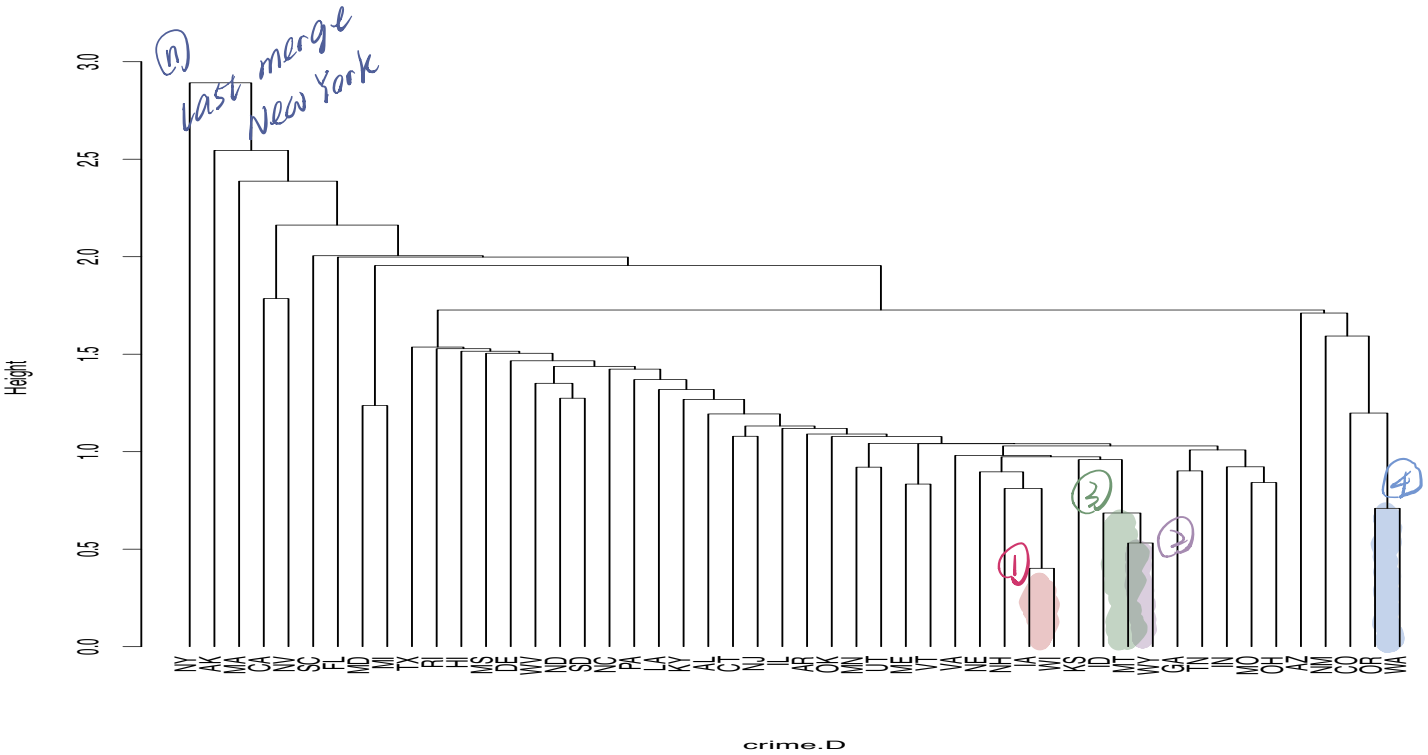
```
crime.D <- dist(crime.st, method="euclidean")
```

The command for distances is `dist()` and the command for dissimilarities is `daisy()`.

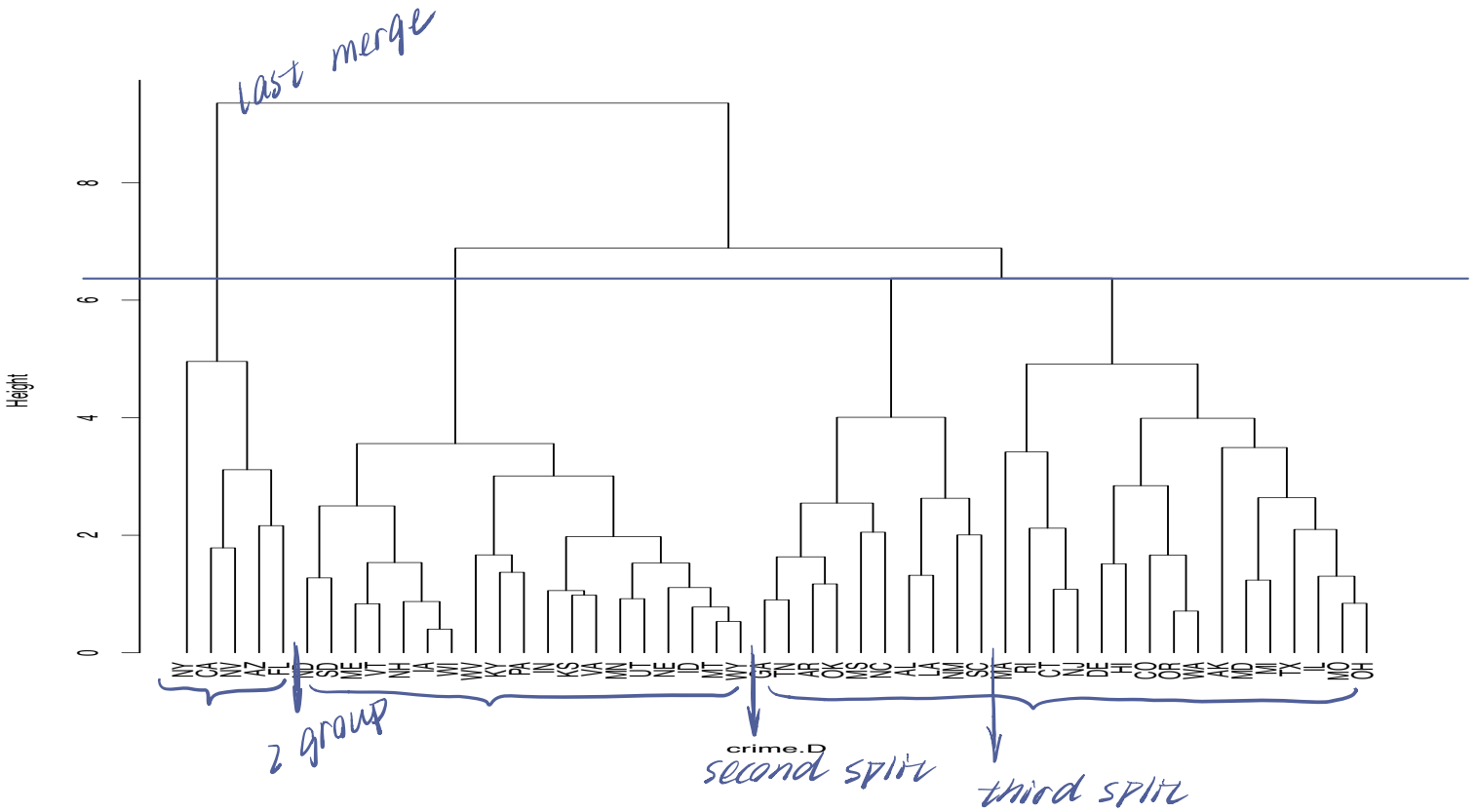
One can then apply any method of one's choice, e.g.,

```
crime.single <- hclust(crime.D, method="single")
plot(crime.single, labels = states, hang = -1,
     main = "Single linkage\nEuclidean distance",
     sub="Function hclust")
```

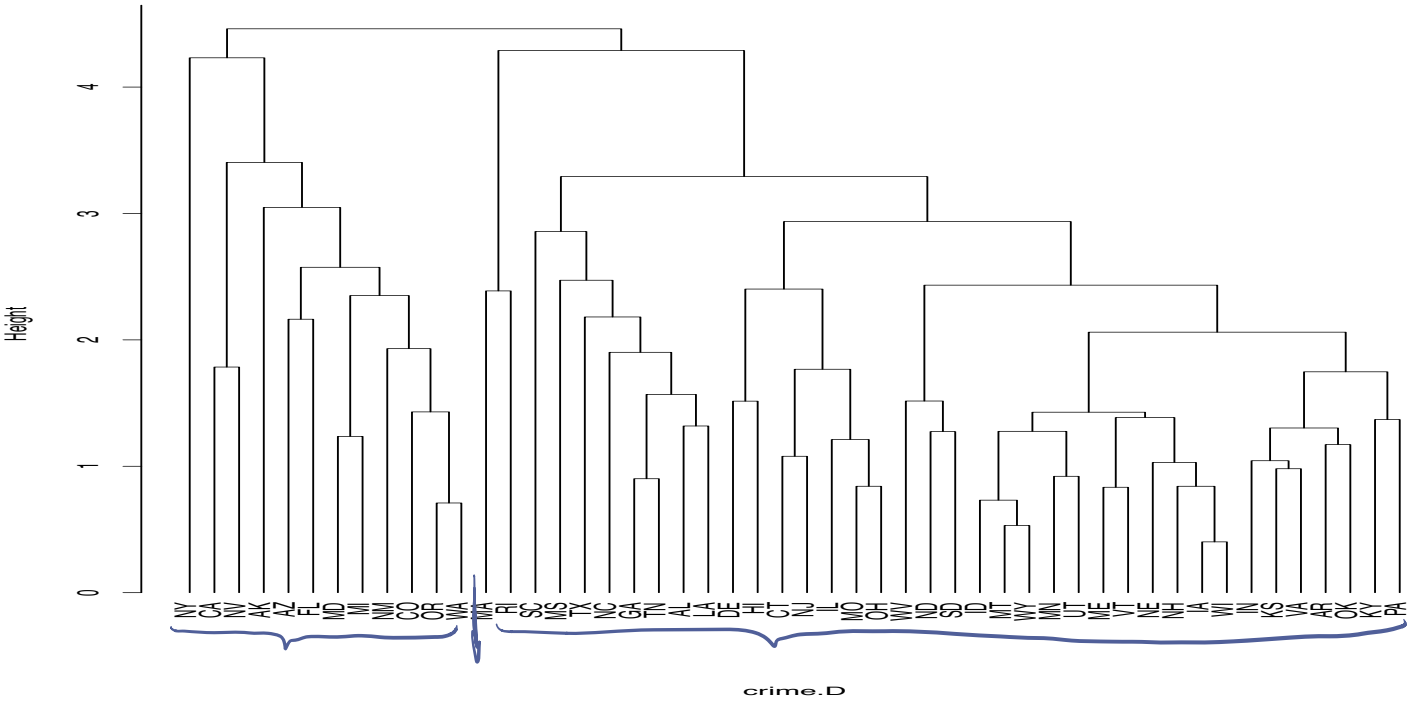
Single Linkage



Complete Linkage



Average Linkage



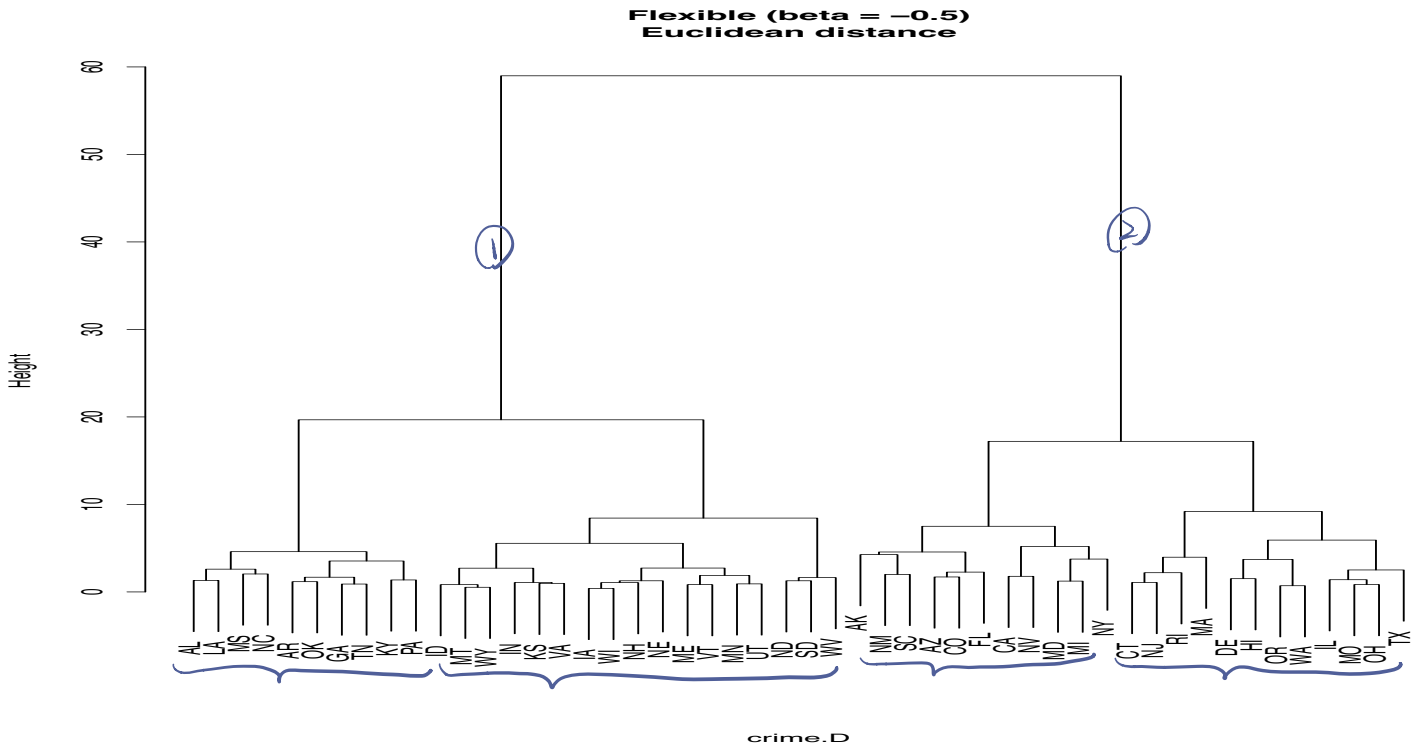
Flexible Method

For the flexible method, one can proceed as follows:

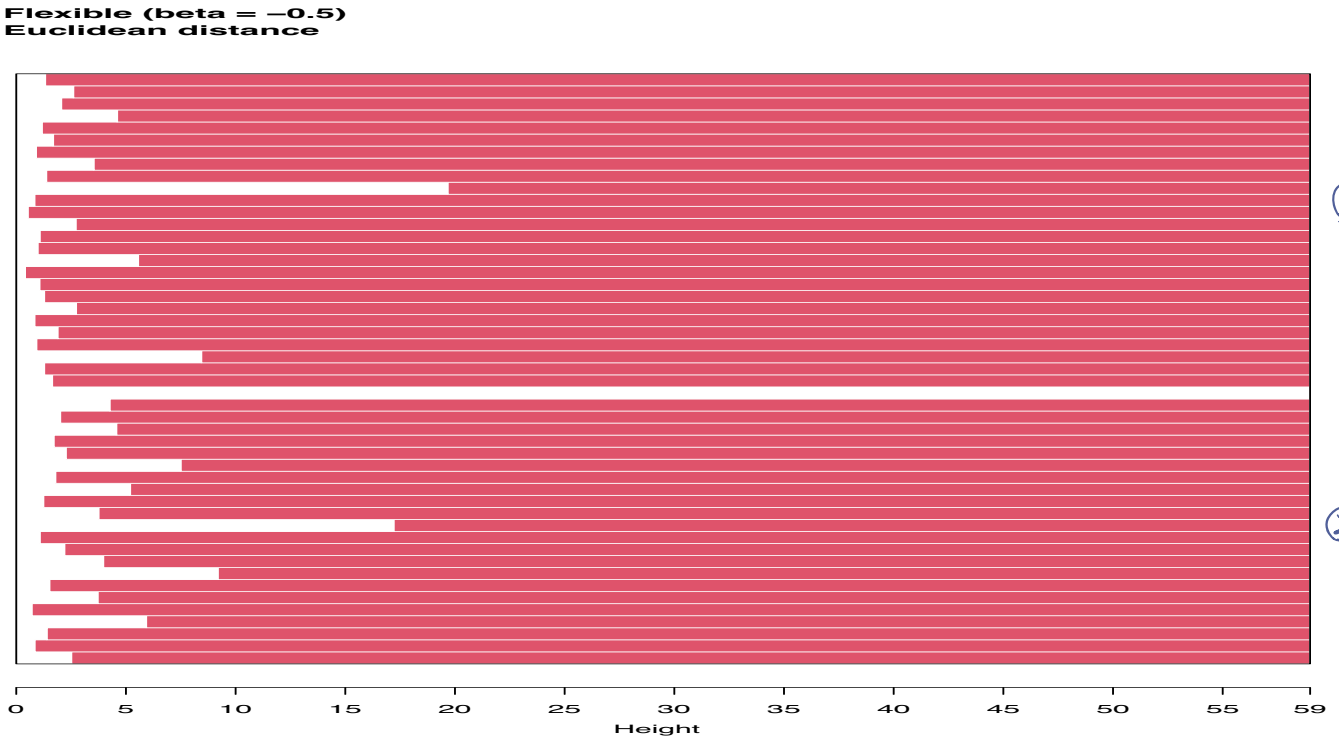
```
library(cluster)
crime.flex <- agnes(crime.D,method="flexible",par.method = 0.75)
plot(crime.flex,labels=states,
     main="Flexible (beta = -0.5)\nEuclidean distance",sub="")
```

This actually produces three plots, which appear in the subsequent slides.

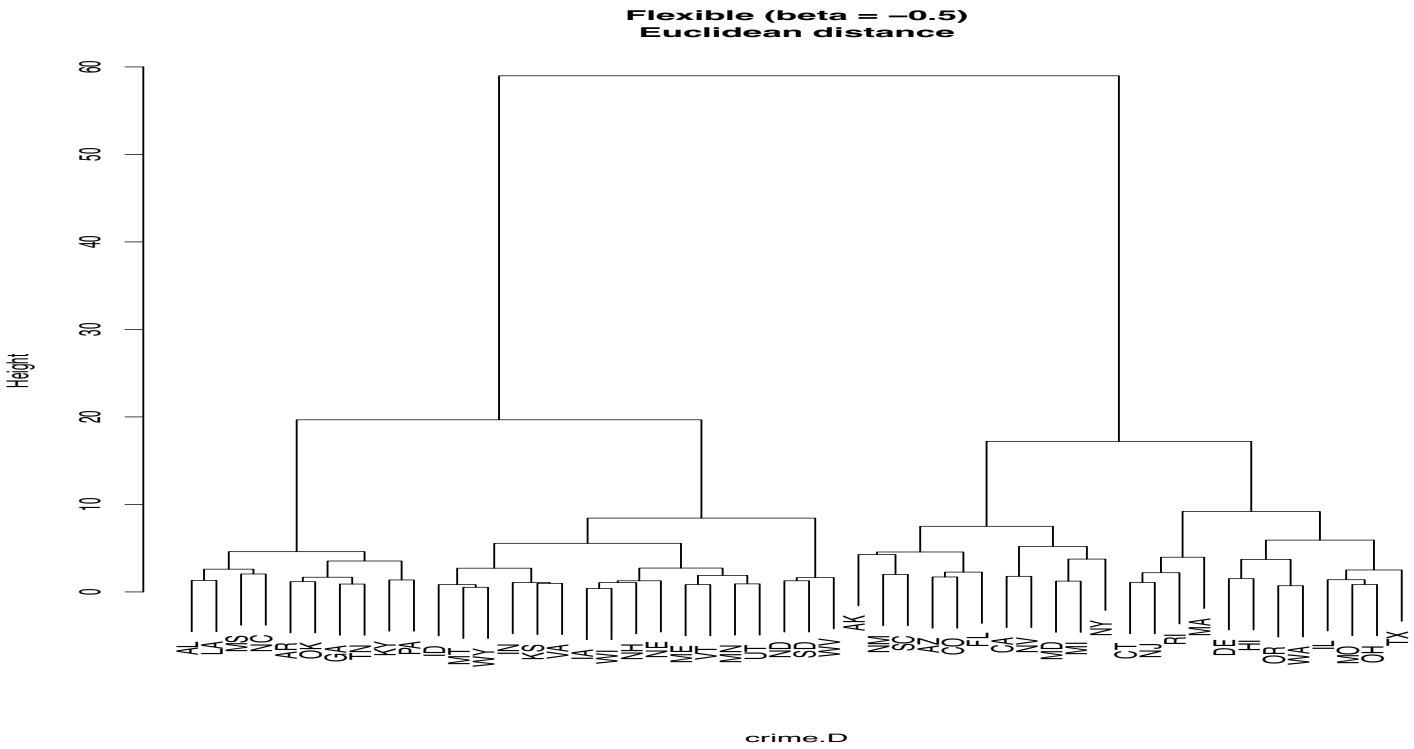
Flexible Method, $\beta = -0.5$ (1–3)



Flexible Method, $\beta = -0.5$ (2–3)



Flexible Method, $\beta = -0.5$ (3–3)



How to Choose a Partition

The above algorithms each provide a sequence of n partitions having 1 to n groups.

Which partition should be chosen, and how?

This is a difficult question, to which there is no unique answer.

One should start by asking the following questions:

- ✓ Is one of the n partitions particularly easy to interpret?
- ✓ Does one of the n partitions have a practical use?

If there is no clear answer to these questions, then criteria can be invoked to make a choice.

Cubic Clustering Criterion (1–2)

The cubic clustering criterion (CCC) can be used to estimate the number of clusters using Ward's minimum variance method, k-means, or other methods based on minimizing the within-cluster sum of squares.

The value of CCC (defined later) is plotted as a function of the number of groups, where $K = 1$ to $K = n/10$ are considered.

- ✓ If CCC > 2, the clustering is deemed excellent.
- ✓ If $0 <$ CCC < 2, the clustering is deemed average.
- ✓ If CCC < 0, the clustering is deemed poor.

Cubic Clustering Criterion (2–2)

To select the number of groups, one considers those that are associated to large increases of CCC between two successive numbers of classes.

An increase is considered large if it is larger than 2 or 3.

Warning:

- ✓ The CCC criterion should not be used with the single linkage method, or when the groups are suspected to be stretched out or irregular in shape.
- ✓ The CCC criterion does not work well either when some of the groups have fewer than 10 observations.

Pseudo-F Statistic

The decision is based on a statistic having an F distribution when the data are multivariate normal with equal variances across all groups.

This statistic can still be informative, even when the data are far from Gaussian.

Desirable are numbers of groups for which the pseudo-F statistic is large.

Note, however, that the pseudo-F statistic should not be used together with the single linkage method.

Pseudo-t^2 Statistic

The decision is based on a statistic having a t^2 distribution when the data are multivariate normal with equal variances across all groups.

A graph of the value of the pseudo-t^2 statistic is drawn as a function of the number of classes.

The graph is scanned from right to left, and one looks for values of the statistic which are much higher than the previous one.

Suppose that a large increase occurs between k and $k-1$ groups. Then one chooses k as the appropriate number of groups for the data at hand.

Again, the pseudo-t^2 statistic should not be used together with the single linkage method.

Computation of the Criteria

To compute the criteria CCC, Pseudo-t2, Pseudo-F, one must provide the data and the distance matrix. One must also specify the clustering method and the selected criterion.

```
library(NbClust)
crime.ave.ccc <- NbClust(data = crime, diss = crime.D, distance = NULL,
                 method = "average", index = "ccc")
crime.ave.ccc
plot(2:15, crime.ave.ccc$All.index, xlab = "K", ylab = "ccc",
     type = "l", main = "Average method")
crime.ave.t2 <- NbClust(data = crime, diss = crime.D, distance = NULL,
                         method = "average", index = "pseudot2")
crime.ave.t2
plot(2:15,crime.ave.t2$All.index, xlab = "K", ylab = "pseudo t2",
     type = "l", main = "Average method")
```

You set `distance = NULL` if you supply the matrix D yourself.

Crime.ave.ccc

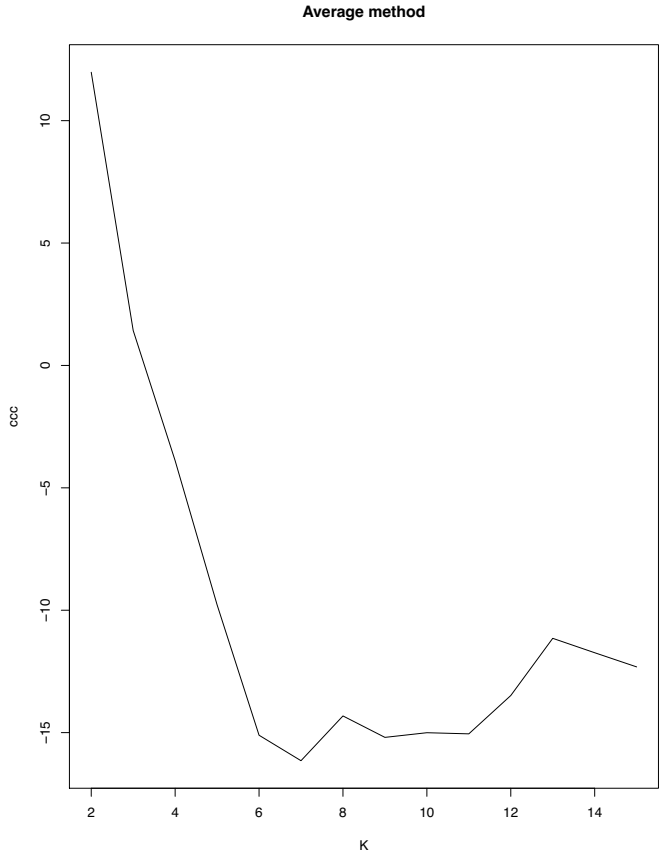
```
$All.index
       2        3        4        5        6        7        8        9        1
 11.9779   1.4285  -3.8919  -9.7794 -15.1066 -16.1474 -14.3186 -15.1945 -15.003
      11       12       13       14       15
-15.0500 -13.4846 -11.1483 -11.7363 -12.3153

$Best.nc
Number_clusters     Value_Index
         2.0000         11.9779

$Best.partition
AL AK AZ AR CA CO CT DE FL GA HI ID IL IN IA KS KY LA ME MD MA MI MN MS MO
 1  2  2  1  2  2  1  1  2  1  1  1  1  1  1  1  1  1  2  1  2  1  1  1
MT NE NV NH NJ NM NY NC ND OH OK OR PA RI SC SD TN TX UT VT VA WA WV WI WY
1  1  2  1  1  2  2  1  1  1  1  2  1  1  1  1  1  1  1  1  2  1  1  1
```

objective

Plot For Criterion CCC



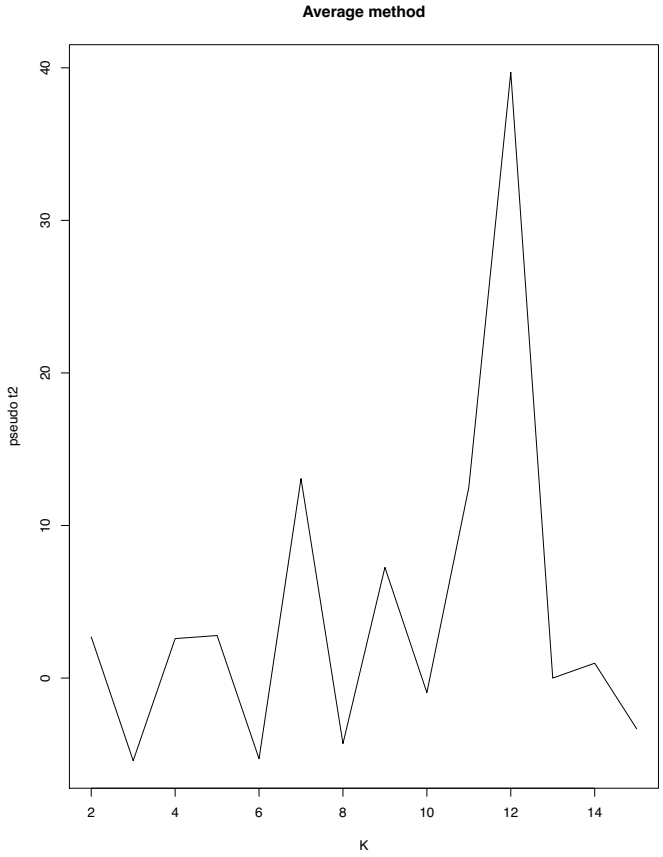
crime.ave.t2

```
$All.index
      2       3       4       5       6       7
 2.6987 -5.4171  2.5870  2.7884 -5.2872 13.0721
      8       9      10      11      12      13
-4.2958  7.2550 -0.9550 12.5044 39.7108  0.0000
     14      15
 0.9726 -3.3242

$All.CriticalValues
      2       3       4       5       6       7
19.5437 12.4845 12.2211 19.0431 11.7706 16.9684
      8       9      10      11      12      13
11.6263 11.6263 11.6036 15.0457 11.6036  0.0000
     14      15
11.8459 11.8459

$Best.nc
Number_clusters     Value_Index
         2.0000          2.6987
```

Plot For Criterion Pseudo-t^2



crime.ave.t2

```
$Best.nc
Number_clusters     Value_Index
         2.0000          2.6987

$Best.partition
AL AK AZ AR CA CO CT DE FL GA HI ID IL IN IA KS KY LA
 1  2  2  1  2  2  1  1  2  1  1  1  1  1  1  1  1  1
ME MD MA MI MN MS MO MT NE NV NH NJ NM NY NC ND OH OK
 1  2  1  2  1  1  1  1  1  2  1  1  2  2  1  1  1  1
OR PA RI SC SD TN TX UT VT VA WA WV WI WY
 2  1  1  1  1  1  1  1  1  1  2  1  1  1
```

Additional Code (1–3)

To see the group numbers after clustering, one can use the following code, say for $k = 2$ and $k = 5$ groups.

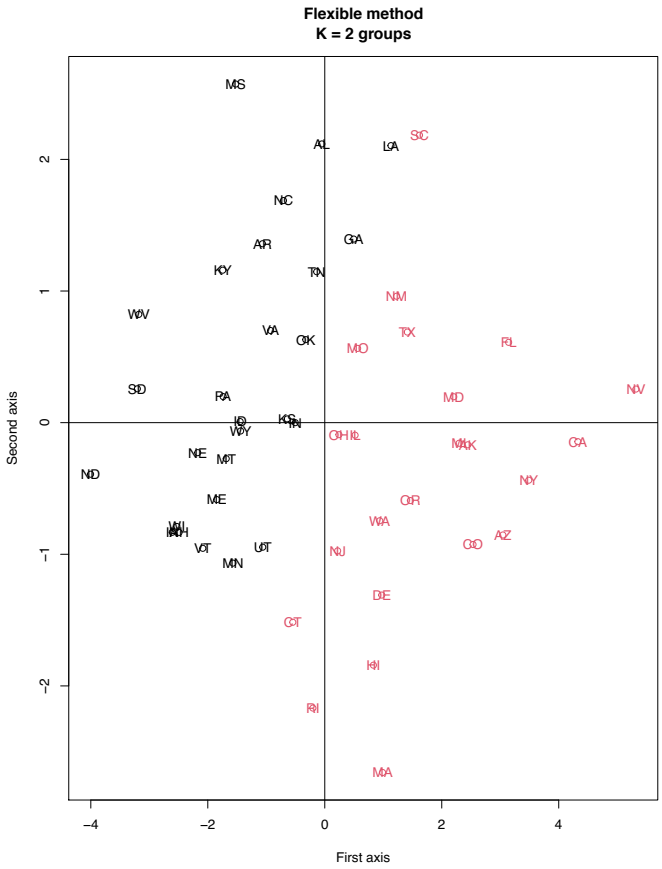
```
crime.25gr <- cutree(crime.flex,k=c(2,5))
crime.25gr
```

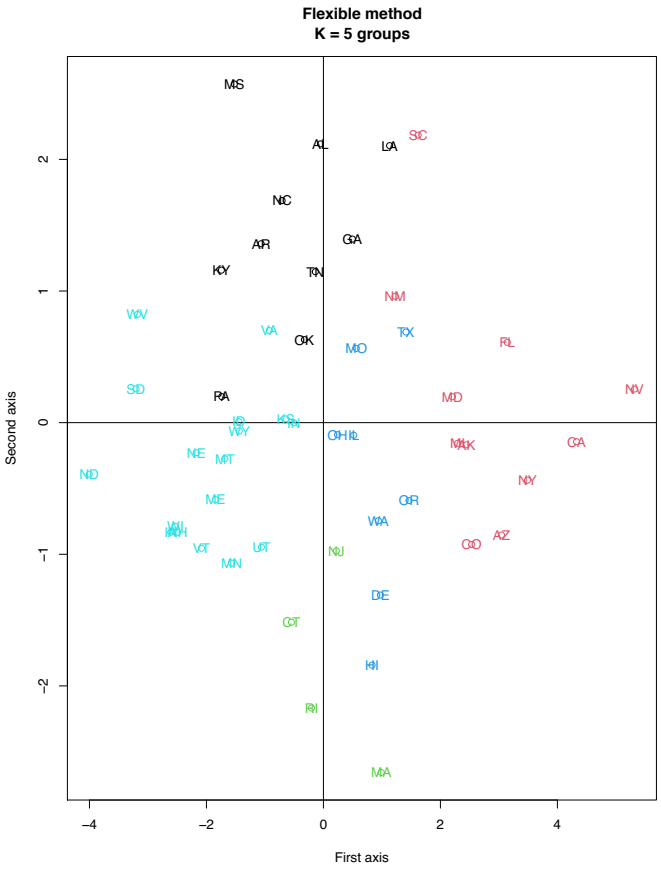
```
     2 5
 [1,] 1 1
 [2,] 2 2
 [3,] 2 2
 [4,] 1 1
 [5,] 2 2
 [6,] 2 2
 [7,] 2 3
 [8,] 2 4
 ... ...
```

Additional Code (2–3)

```
plot(crime.acp$scores[,1], crime.acp$scores[,2],
     xlab = "First axis", ylab = "Second axis", col = crime.25gr[,1],
     main = "Flexible method\nK = 2 groups")
abline(h = 0, v = 0)
text(crime.acp$scores[,1], crime.acp$scores[,2],
     labels = states, cex = 1, col = crime.25gr[,1])
plot(crime.acp$scores[,1],crime.acp$scores[,2],
     xlab = "First axis", ylab = "Second axis", col = crime.25gr[,2],
     main = "Flexible method\nK = 5 groups")
abline(h = 0 , v = 0)
text(crime.acp$scores[,1], crime.acp$scores[,2],
     labels = states, cex = 1, col = crime.25gr[,2])
```

Additional Code (3–3)





Review of Properties: Single Linkage (1–4)

Pros:

- performs well when the variables are of different types;
- has good theoretical properties under certain conditions;
- makes it possible to create groups with irregular shapes;
- is robust to outliers.

Against:

- tends to create a large group surrounded by small satellite groups;
- looses in efficiency if the underlying groups are regular in shape;
- is well behaved under conditions that are rarely met in practice.

Review of Properties: Complete Linkage (2–4)

Pros:

- performs well when the variables are of different types;
- tends to form groups of equal size.

Against:

- tends to form groups of equal size;
- is very sensitive to outliers;
- is rarely used in practice.

Review of Properties: Other Methods (3–4)

Average linkage:

- **Pros:** tends to form groups with *small* variance;
- **Against:** tends to form groups with *equal* variance.

Centroid:

- **Pros:** is robust to outliers;
- **Against:** is not very efficient in the absence of outliers.

Median:

- **Pros:** is even more robust to outliers;
- **Against:** is very inefficient in the absence of outliers.

Review of Properties: Ward's Methods (4–4)

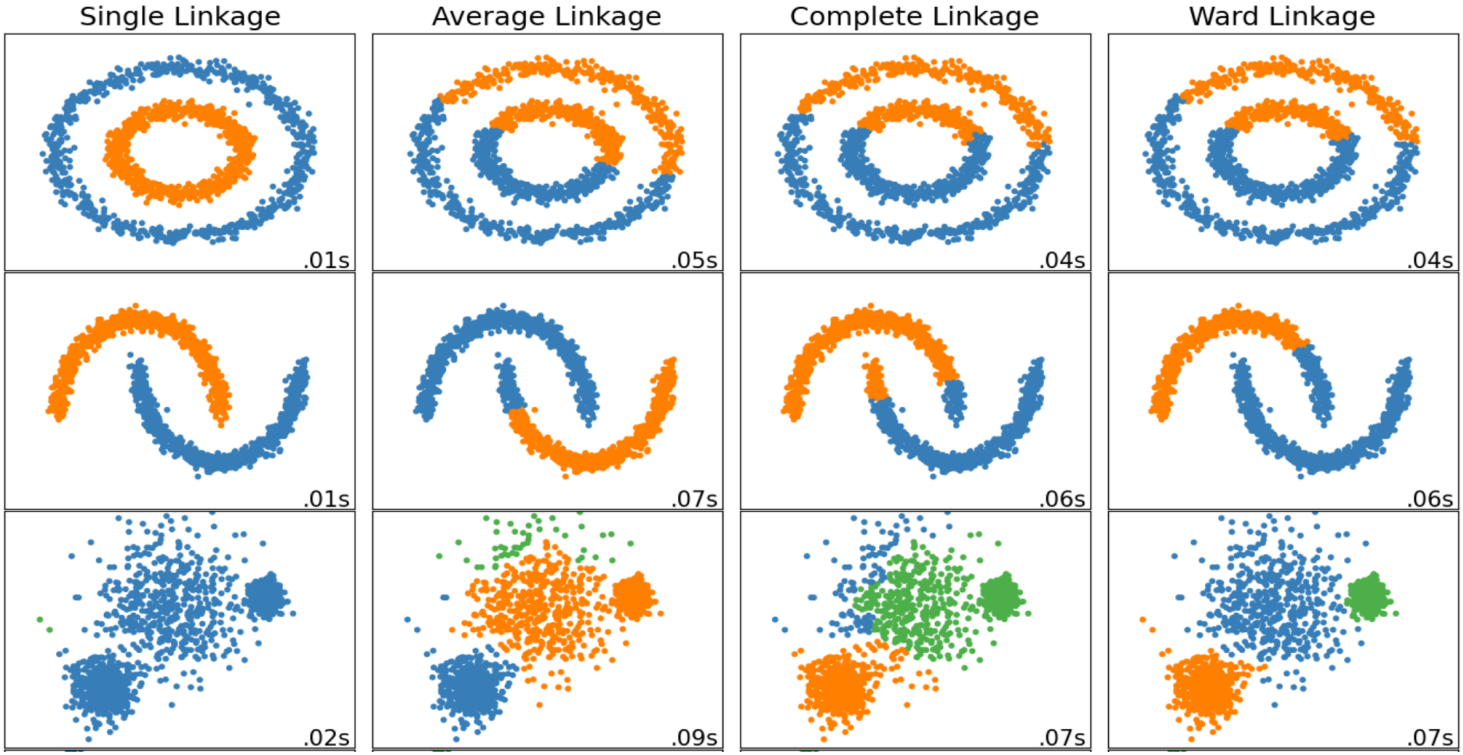
Pros:

- is optimal when the observations are multivariate normal with the same covariance matrix.

Against:

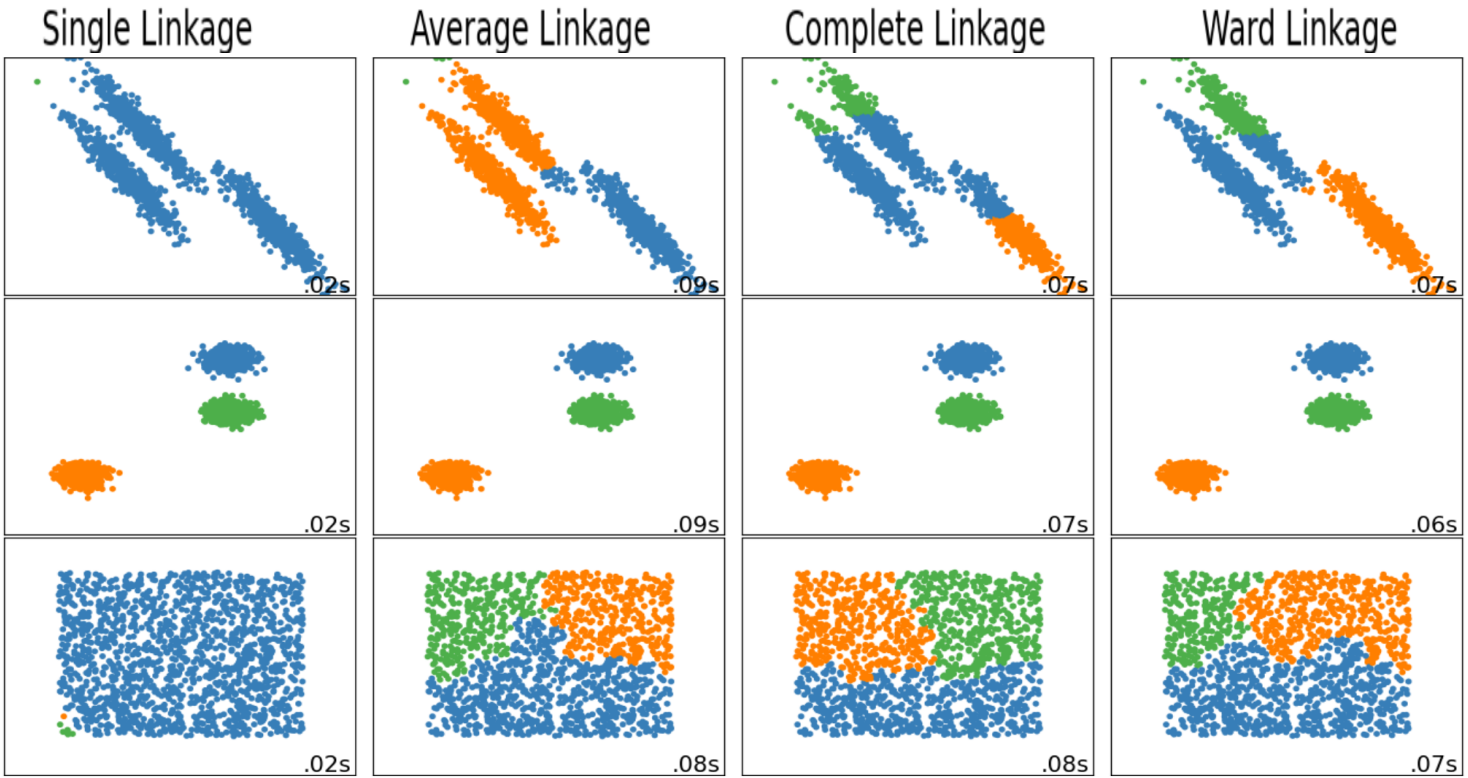
- tends to form small groups;
- tends to form groups of equal size;
- is sensitive to outliers.

Illustration (1–2)



Source: `https://scikit-learn.org/stable/auto_examples/cluster/plot_linkage_comparison.html`

Illustration (2–2)



Total running time of the script: (0 minutes 3.219 seconds)

Source: `https://scikit-learn.org/stable/auto_examples/cluster/plot_linkage_comparison.html`