Lecture 13

# Introduction to Cluster Analysis

MCB 416A/516A
Statistical Bioinformatics and Genomic Analysis

Prof. Lingling An
Univ of Arizona

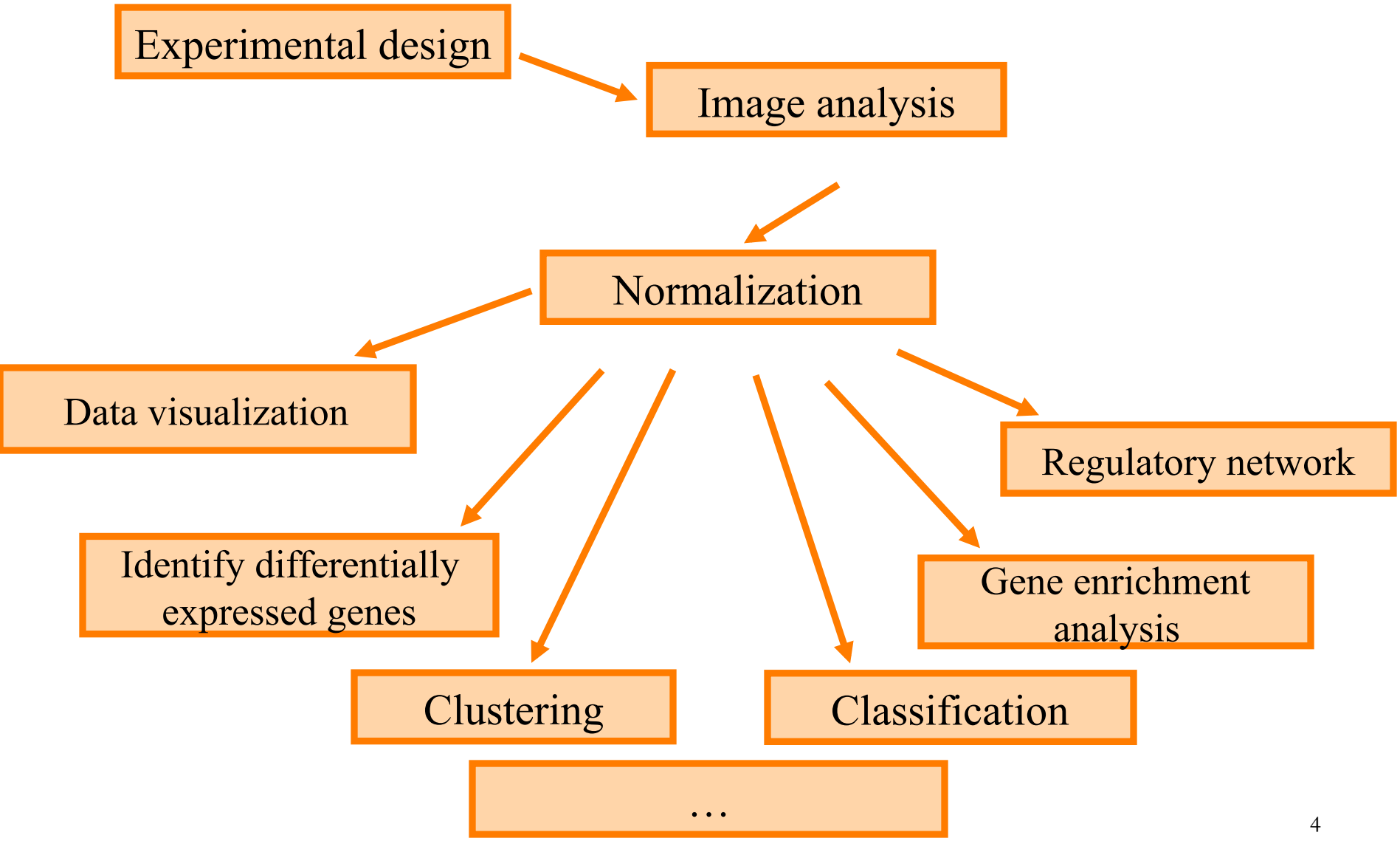
# Outline

- What is cluster analysis?
- Why cluster analysis for gene expression?
- Three main issues in cluster analysis
  - Distance measurements
  - Cluster methods
  - Determine number of clusters

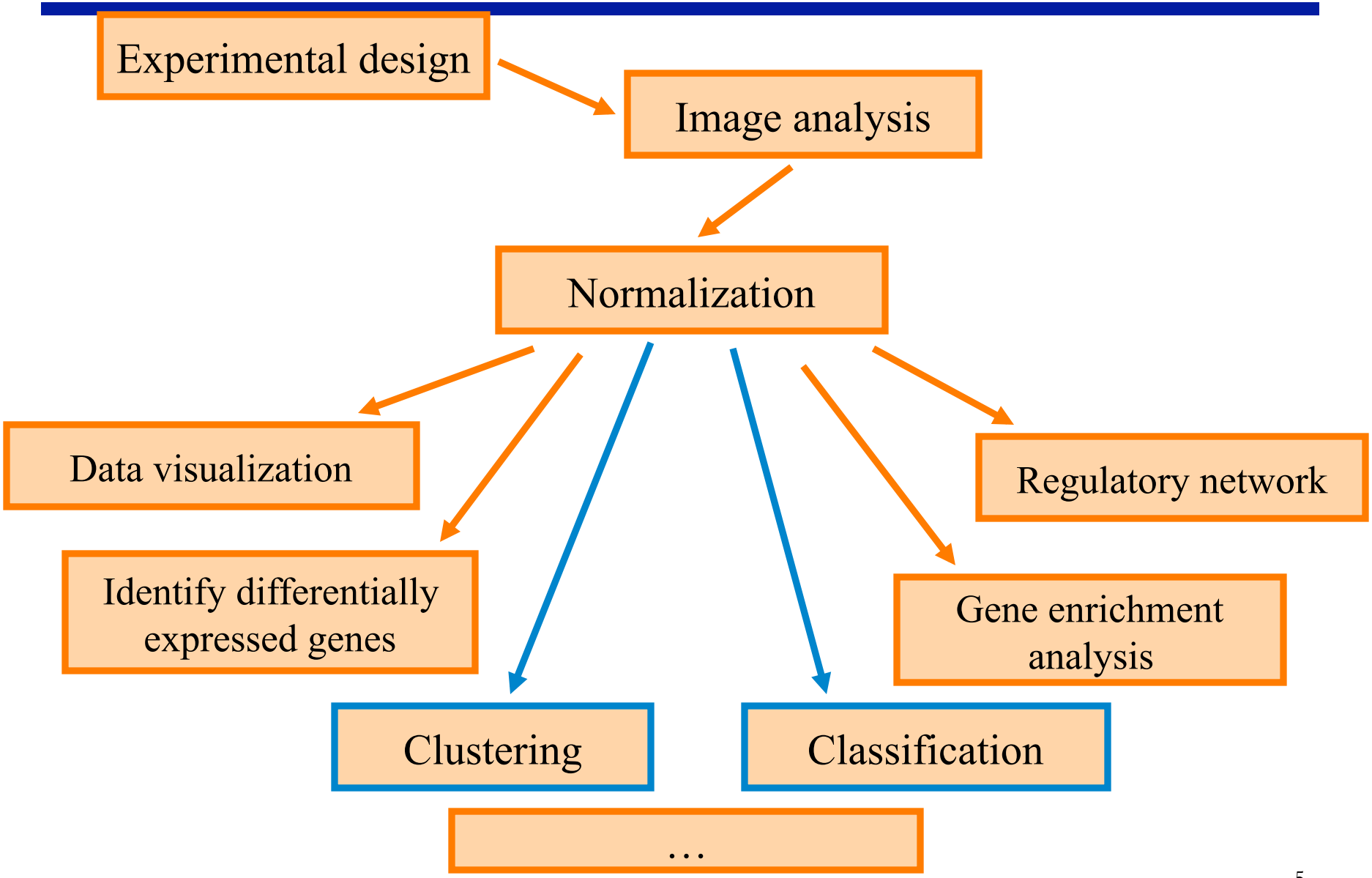
# Cluster Analysis

- Cluster analysis
  - First used by Tryon (1939)
  - Collecting "similar" objects in the same group or cluster
- Applications
  - Data mining
  - Pattern recognition
  - Spatial data analysis
  - Image processing
  - Market research
  - Biology – **gene expression data**
  - etc

# Statistical Issues in Microarray Analysis



# Statistical Issues in Microarray Analysis

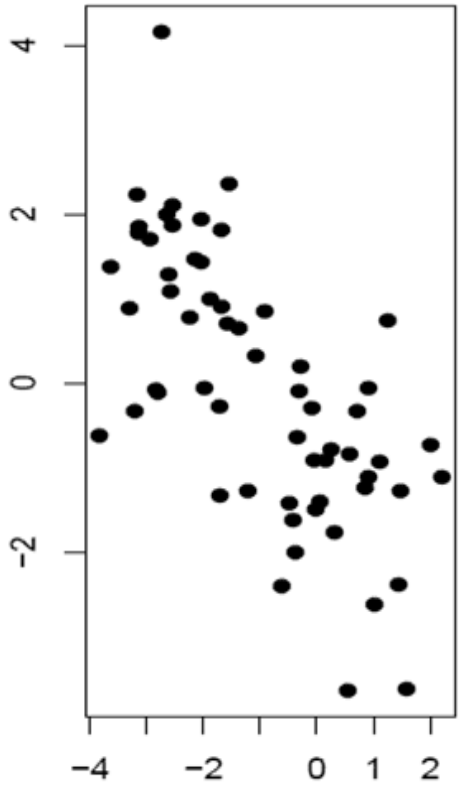


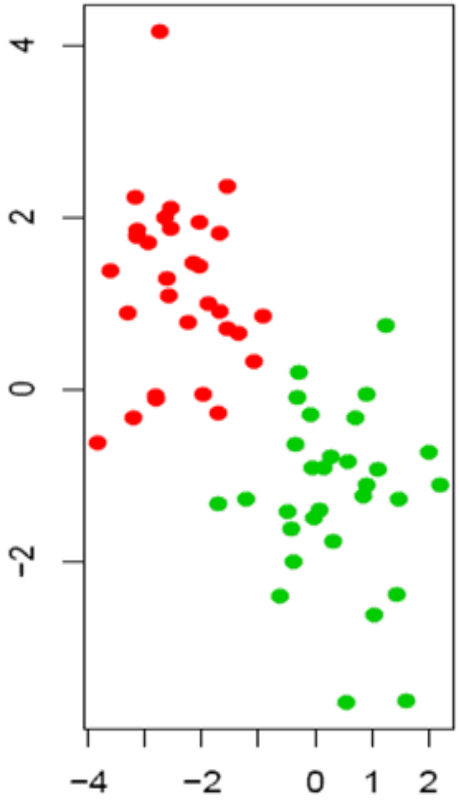
# Cluster analysis vs. Classification

- In **cluster analysis**, there are no predefined groups or labels for the observations, while **classification (i.e., discriminant ) analysis** is based on the existence of such groups or labels.
- Alternative terminology
  - Computer science: **unsupervised** and **supervised learning.**
  - Microarray literature: **class discovery** and **class prediction.**

# Aim of clustering: Group objects according to their similarity

**Cluster:** a set of objects that are similar to each other and separated from the other objects.

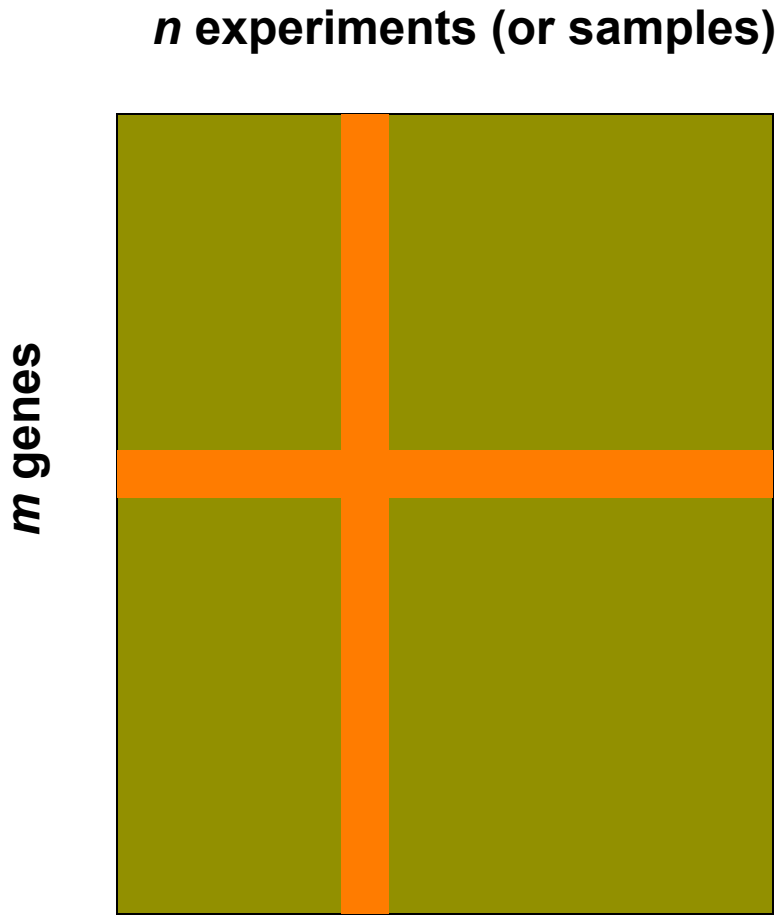



**Example:** green/ red data points were generated from two different normal distributions

# Microarray data structure

- Data matrix:
  - Genes and experiments/ samples are given as the row and column vectors.
  - *m* genes at *n* experiments/samples
- Can cluster genes and/ or experiments



# Why cluster genes?

A gene is a *n*-dimensional vector of profiles

- functionally related genes have similar profiles
- Identify groups of possibly co-regulated genes by their co-expression patterns
- Identify typical temporal or spatial gene expression patterns (e.g. cell cycle data)

# Why cluster experiments?

An individual is an *m*-dimensional vector of genes

- experiments with similar characteristics have similar gene expression profiles
- Identify groups of possibly correlated experiments by similar gene expression patterns

  — e.g., identify new subtype of leukemia from the gene expression patters from a group of leukemia patients

# Why cluster both genes and experiments?

- group of genes are co-regulated under a subset of experiments
  - —— Identify gene biomarkers related to a specific type (or subtype) of a disease

# Cluster analysis

Generally, cluster analysis is based on three ingredients:

- **Distance measure**: Quantification of (dis-) similarity of objects.
- **Cluster algorithm:** A procedure to group objects.

  —— Aim: small within-cluster distances, large between-cluster distances.
- **Method determining number of clusters**

# Some distance measures

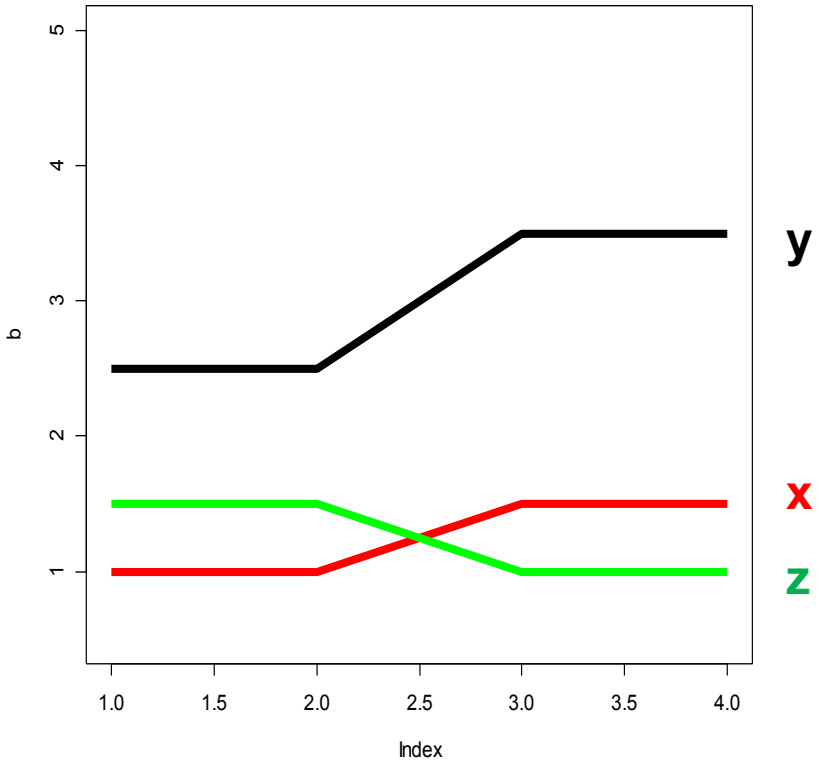
Given vectors $\boldsymbol{x} = (x_1, \ldots, x_n)$, $\boldsymbol{y} = (y_1, \ldots, y_n)$

- Euclidean distance: $$d_E(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$
- Manhattan distance: $$d_M(x,y) = \sum_{i=1}^{n}|x_i - y_i|.$$
- Correlation distance: $$d_C(x,y) = 1 - \frac{\sum_{i=1}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}(x_i - \bar{x})^2 \sum_{i=1}(y_i - \bar{y})^2}}.$$

# Which distance measure to use?

- The choice of distance measure should be based on the application area. What sort of similarities would you like to detect?
- Correlation distance $d_c$ measures trends/relative differences:

$d_c(x, y) = d_c(ax+b, y)$ if $a > 0$.




**$x$ = (1, 1, 1.5, 1.5)**
**$y$ = (2.5, 2.5, 3.5, 3.5) = 2$x$ + 0.5**
**$z$ = (1.5, 1.5, 1, 1)**

**$d_c(x, y) = 0$, $d_c(x, z) = 2$.**
**$d_E(x, z) = 1$, $d_E(x, y) \sim 3.54$.**

# Which distance measure to use?

- Euclidean and Manhattan distance both measure absolute differences between vectors. Manhattan distance is more robust against outliers.
- May apply **standardization** to the observations:

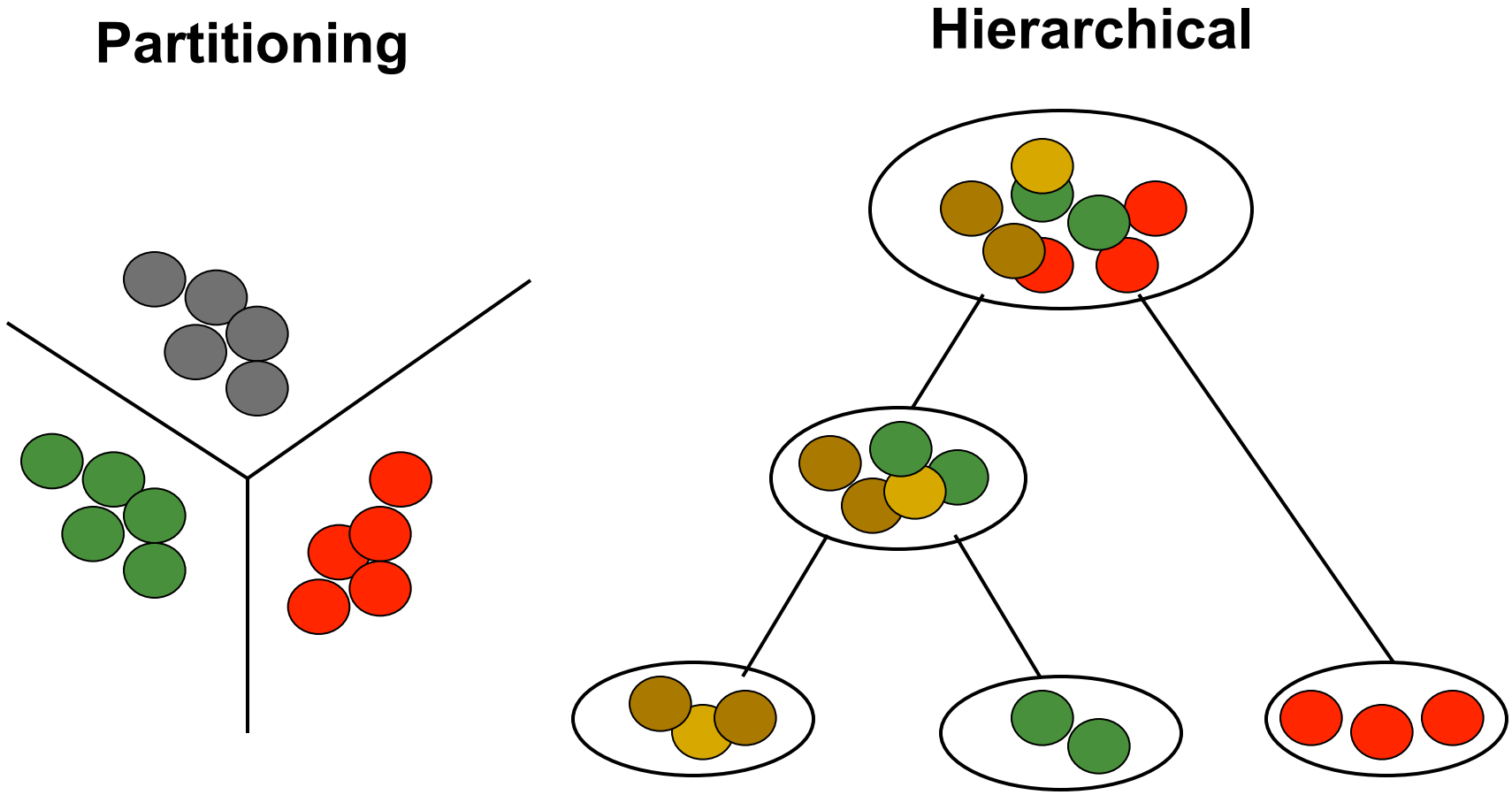
  Subtract mean and divide by standard deviation:

$$x \mapsto \frac{x - \bar{x}}{\hat{\sigma}_x}$$

  —— After standardization, Euclidean and correlation distance are equivalent:

$$d_E(x_1, x_2)^2 = 2n d_C(x_1, x_2).$$

# Two basic types of cluster methods



# (1) Partitioning methods

Partition the data into a pre-specified number *k* of mutually exclusive and exhaustive groups.

Iteratively reallocate the observations to clusters until some criterion is met, e.g. minimize within cluster sums of squares.

*Examples*:

— *k*-means, self-organizing maps (SOM), *PAM (i.e. K-medoids)*, etc.

# (1) Partitioning methods -- *K*-means

1. Define k = number of clusters
2. Randomly initialize a seed vector for each cluster
3. Go through all genes, and assign each gene to the cluster which it is most similar to
4. Recalculate all seed vectors as means of patterns of each cluster
5. Repeat 3&4 until <stop condition>

# (1) K-means clustering: stop conditions

- Until the change in seed vectors is < <constant>
- Until there is no change on the cluster assignment for all genes.

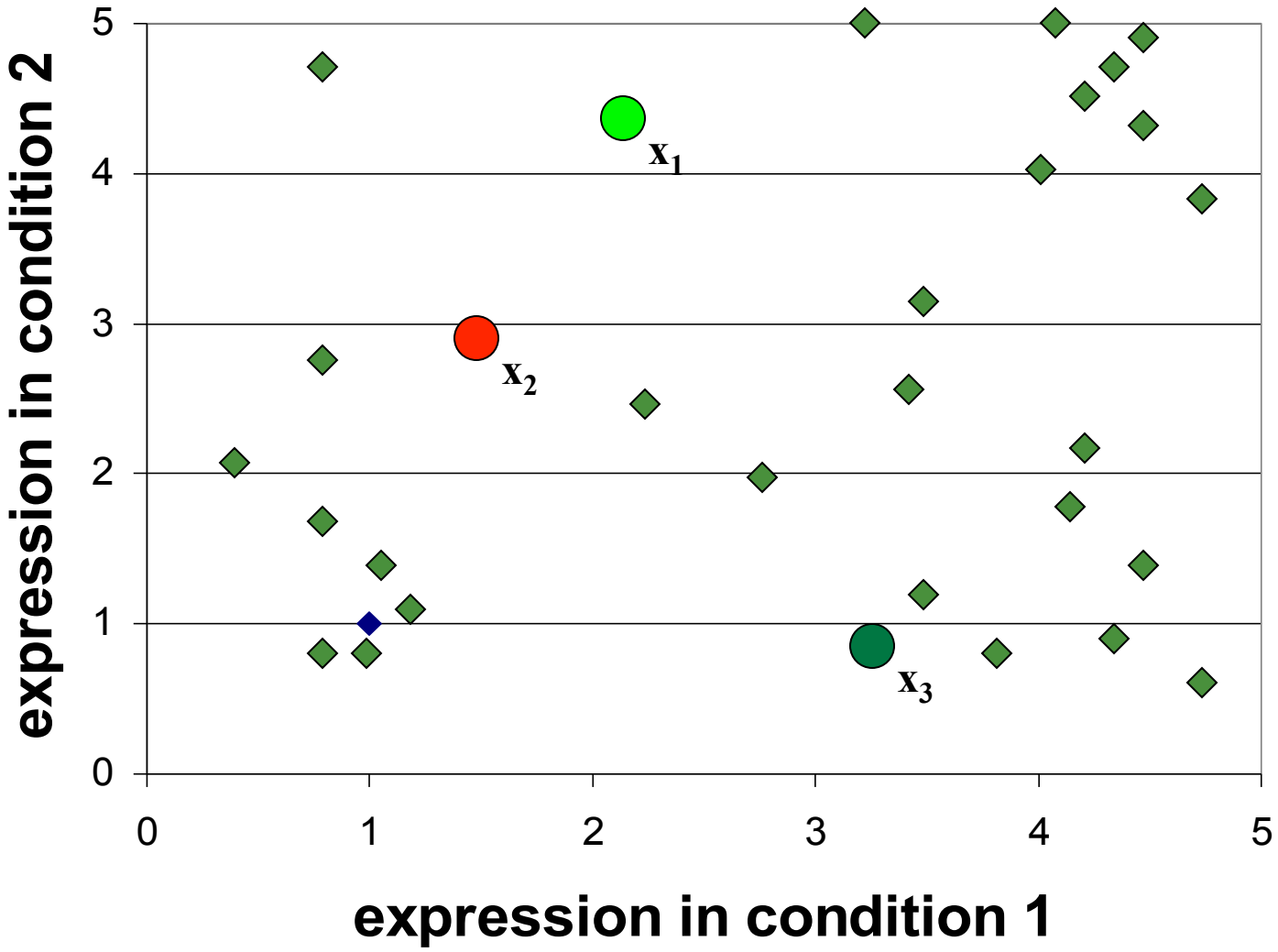
- *K*-means clustering is based on Euclidean distance.
- **Partitioning around medoids** (PAM) generalizes the idea and can be used with any distance measure *d.*
- The cluster centers/prototypes are required to be observations. (for 2-dimensional data, medoid =median)

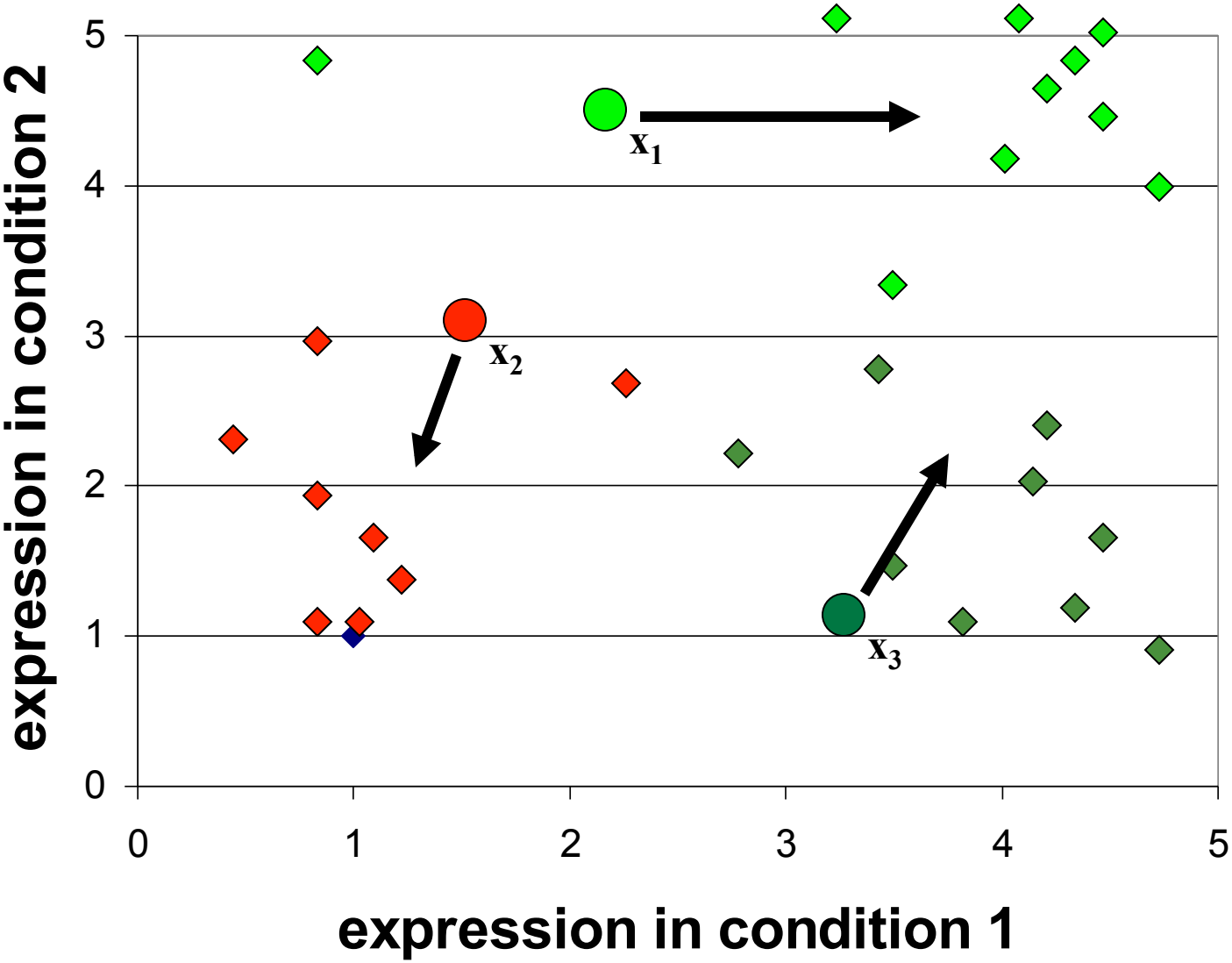
  Should be 1-dimensional
- Try to minimize the sum of distances of the objects to their cluster centers,

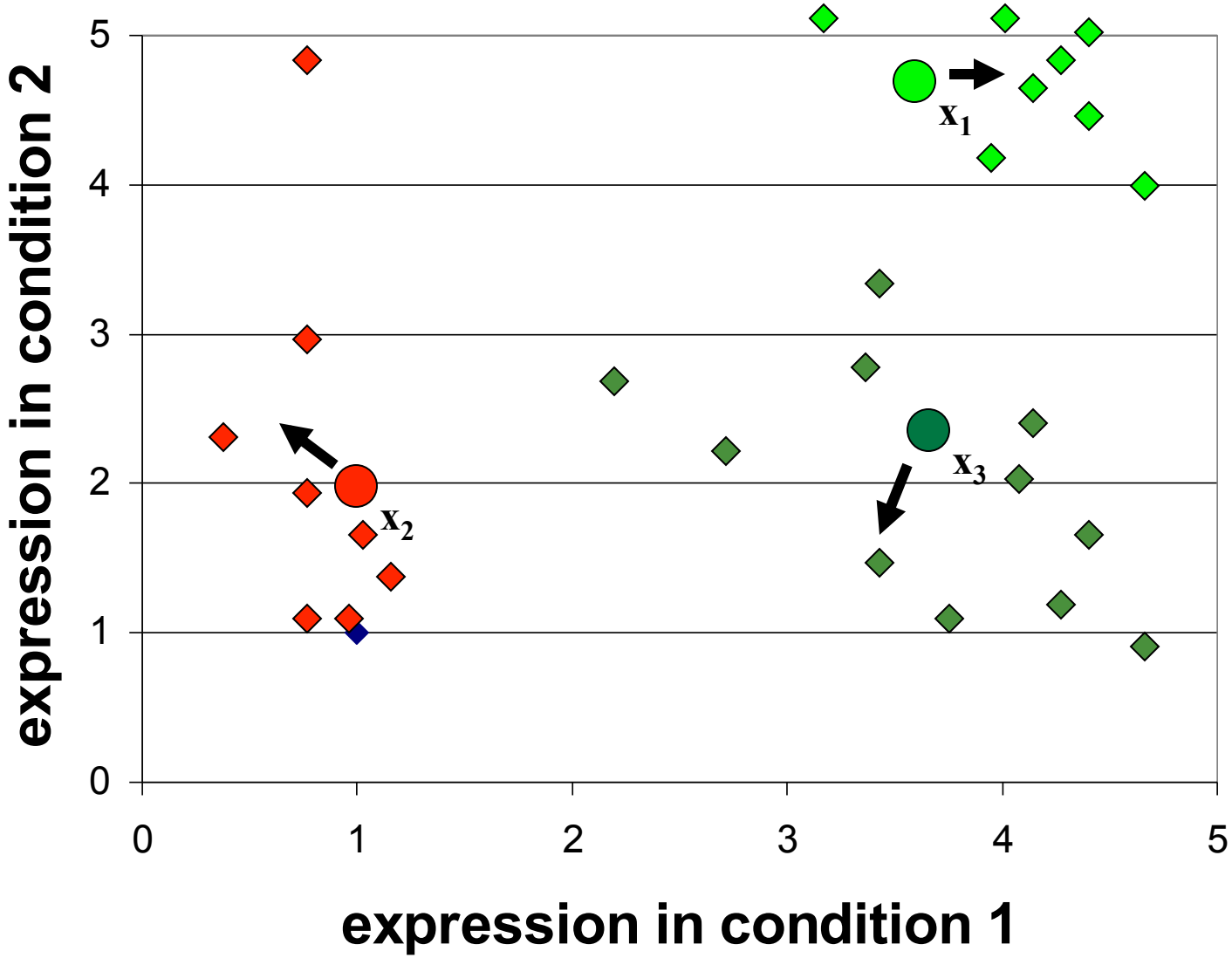
$$\sum_{i=1}^{n} d(x_i, m_{j(i)}), \qquad m_j = x_{i_j}, j = 1, \ldots, K.$$

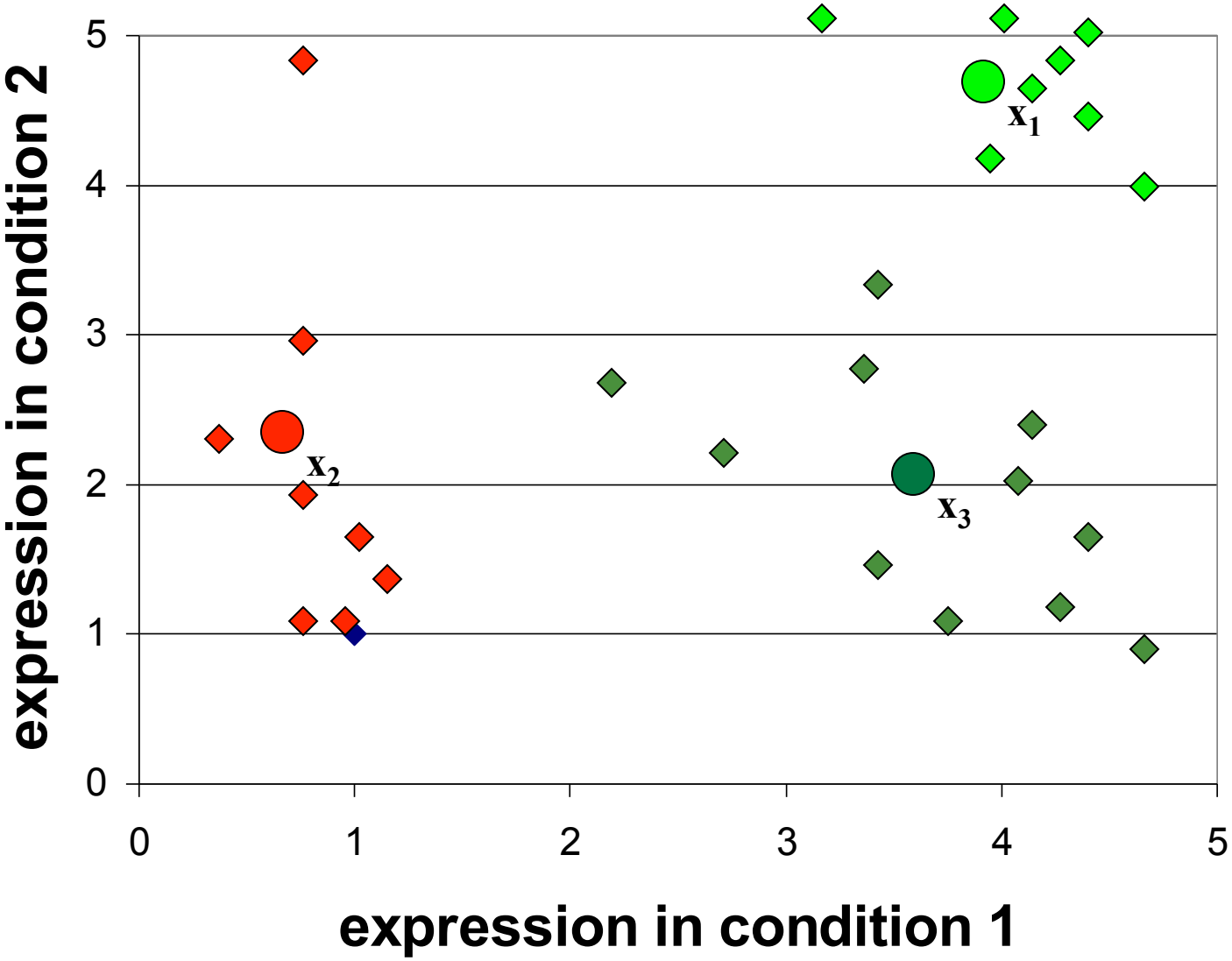
  using an iterative procedure analogous to the one in *K*-means clustering.

- Example: K-mean clustering


expression in condition 2
expression in condition 1
0
1
2
3
4
5
x1
x2
x3


expression in condition 2
expression in condition 1
0
1
2
3
4
5
x1
x2
x3


expression in condition 2
expression in condition 1
0
1
2
3
4
5
x1
x2
x3

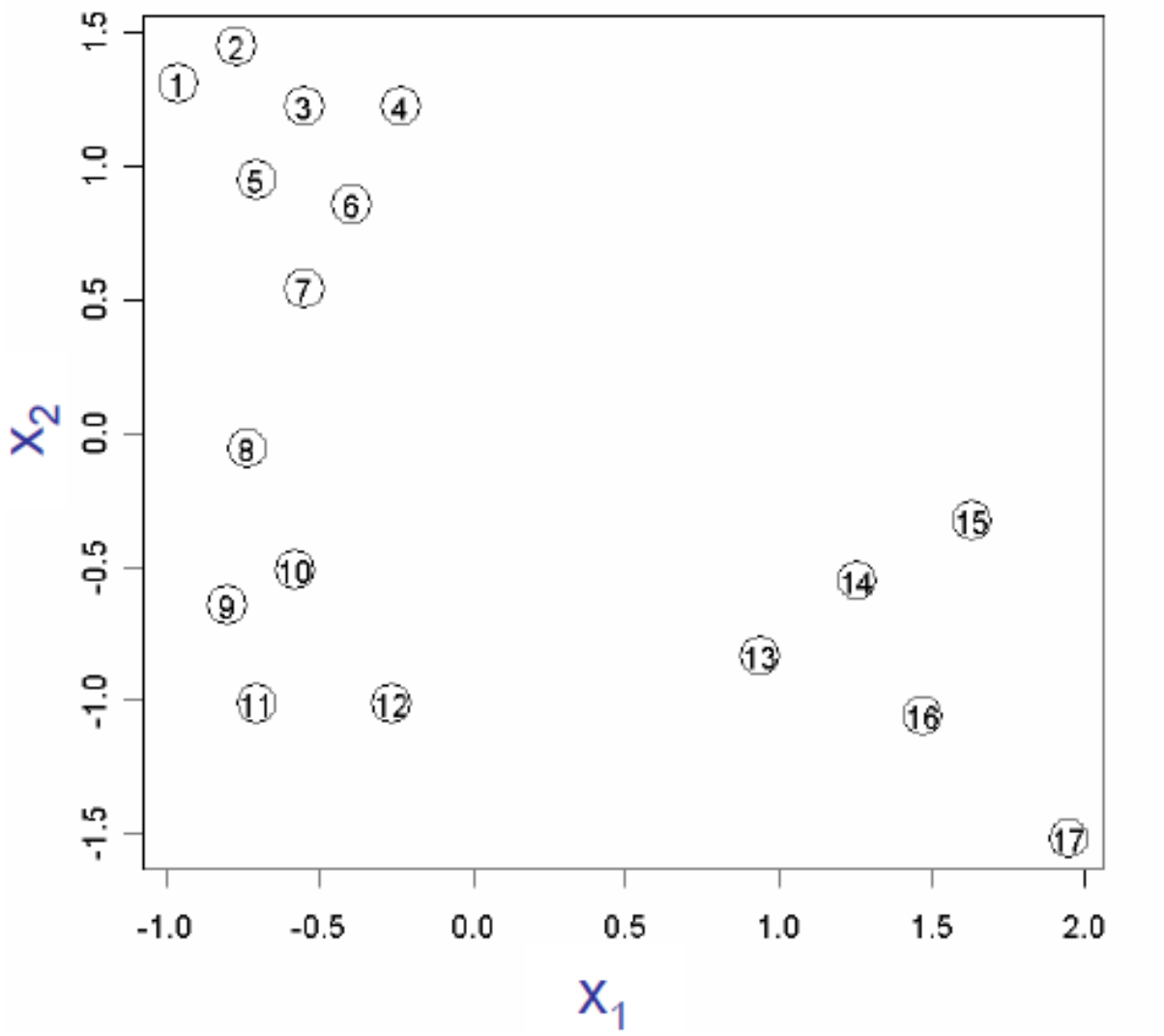

expression in condition 2
expression in condition 1
0
1
2
3
4
5
x1
x2
x3

# (2) Hierarchical cluster

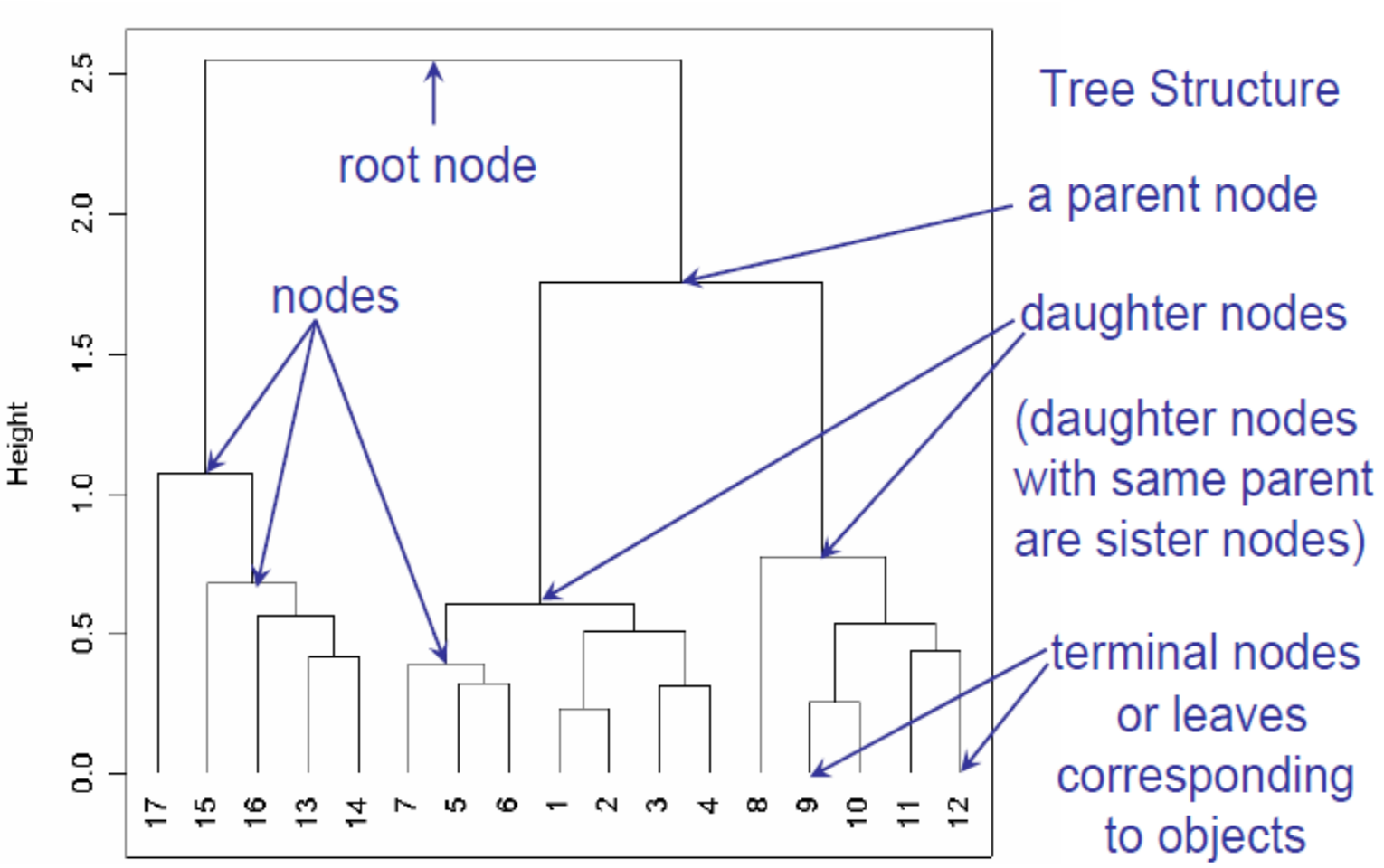
- Hierarchical clustering methods build a nested sequence of clusters that can be displayed using a *dendrogram.*

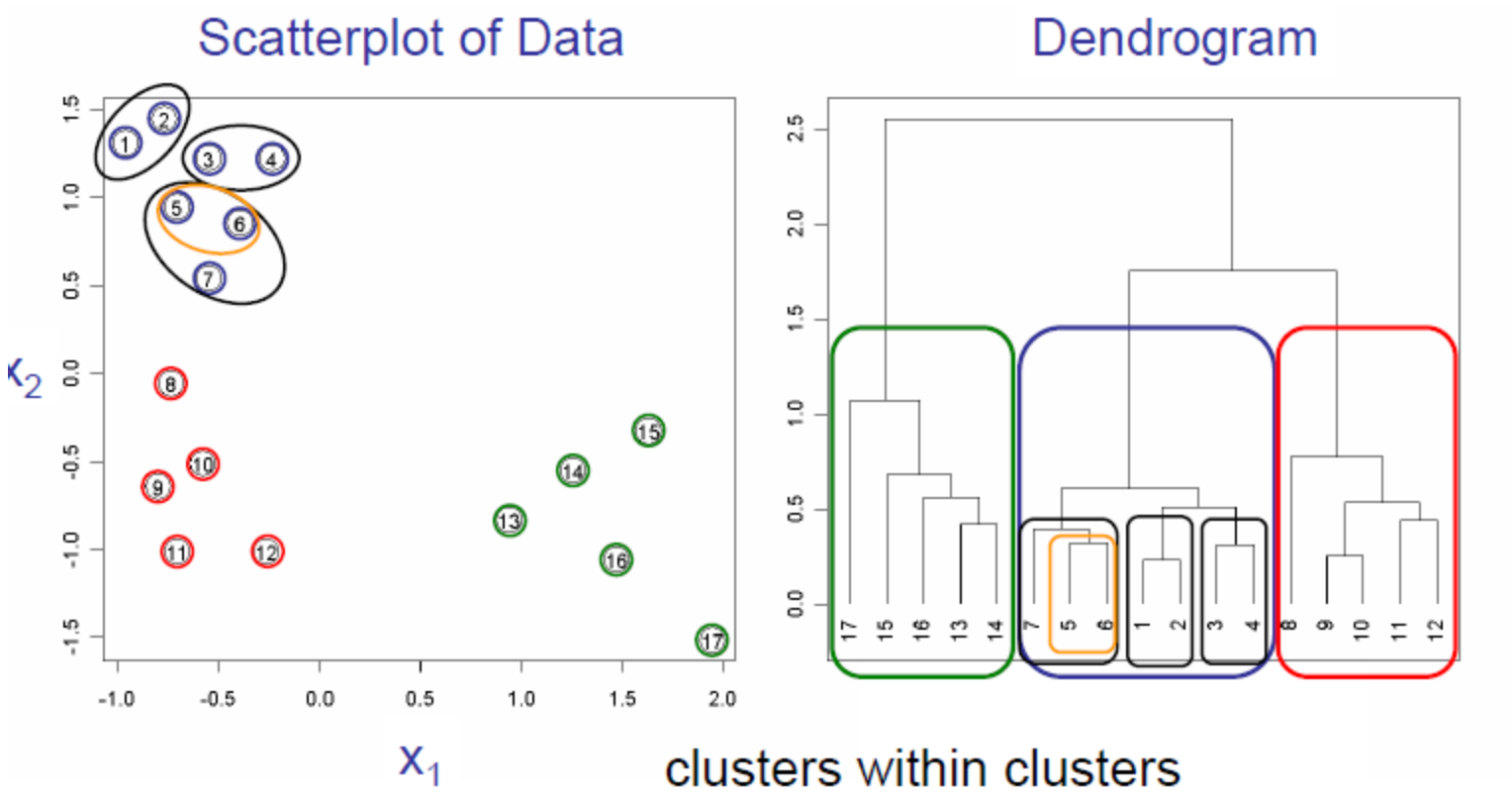
- We will begin with some simple illustrations and then move on to a more general discussion.

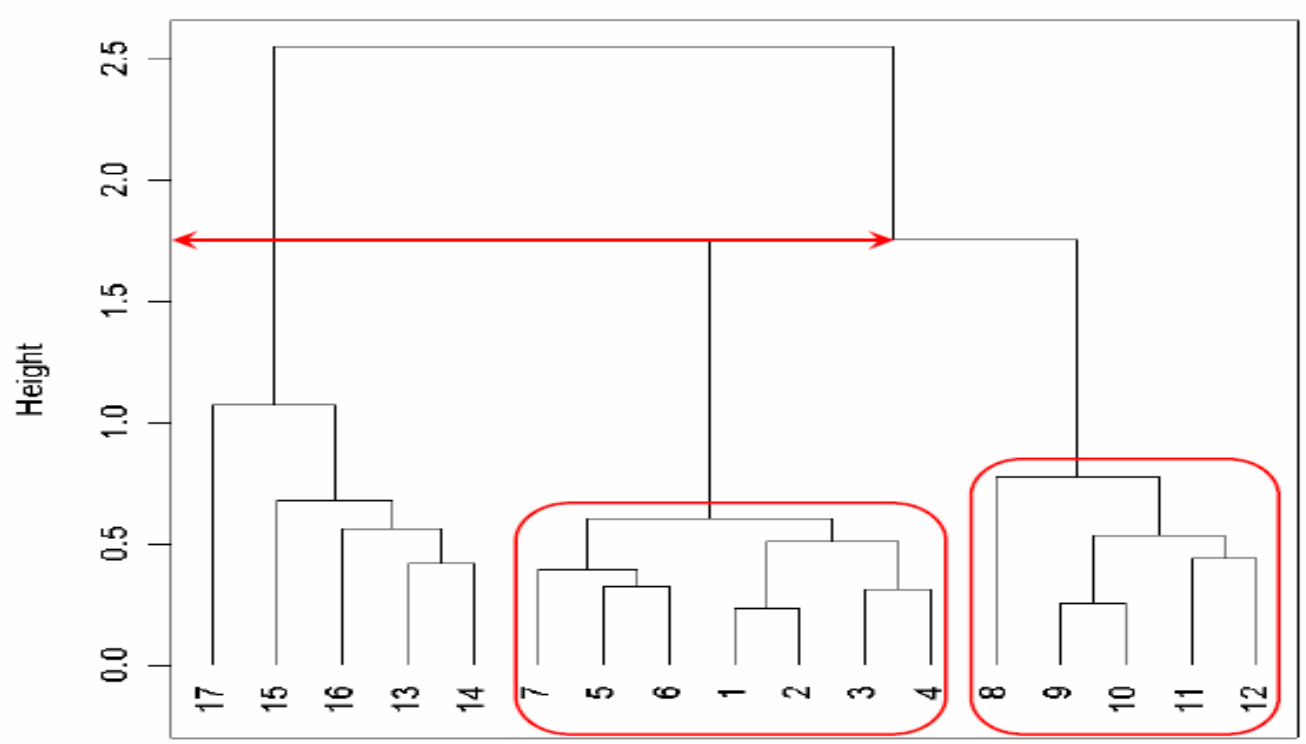
# Simple example dataset



# Basic concept: dendrogram

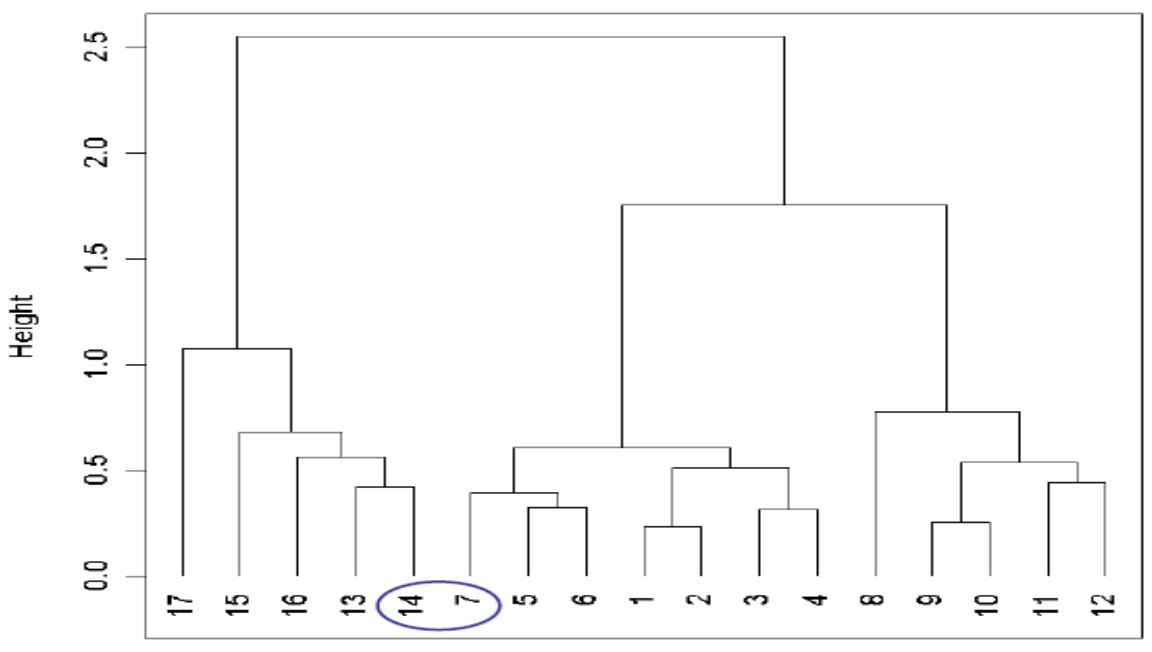



Scatterplot of Data
Dendrogram
$x_2$
$x_1$
clusters within clusters
within clusters...



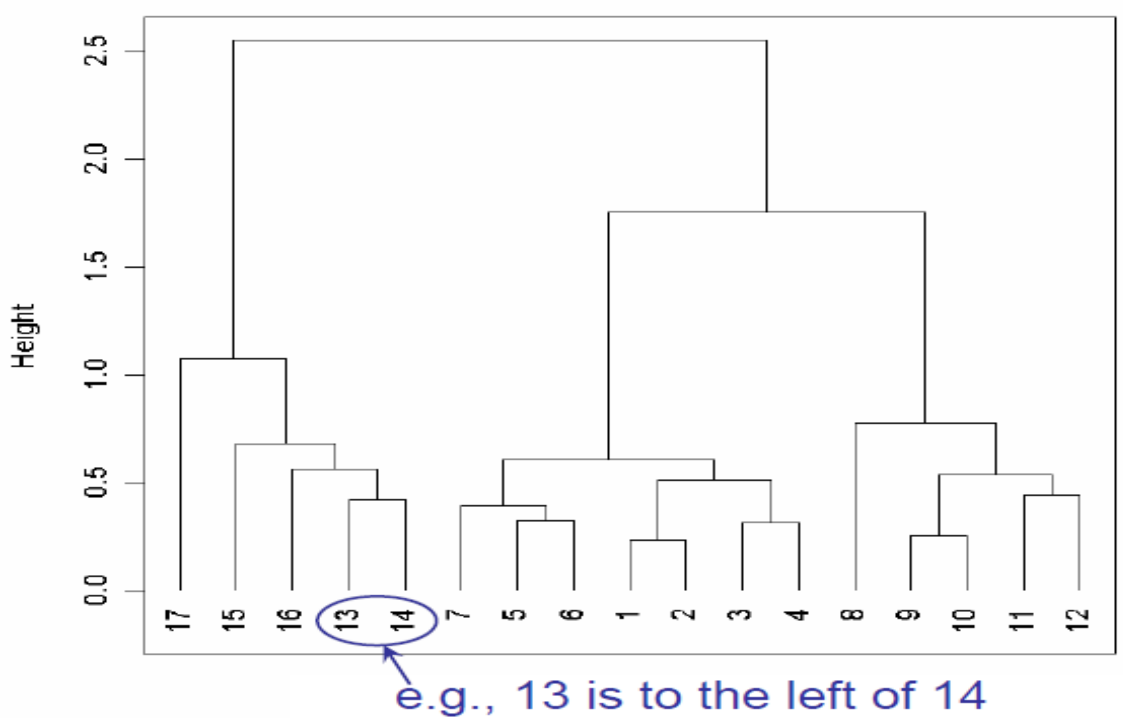

- The height of a node represents the dissimilarity between the two clusters merged together at the node.
- These two clusters have a dissimilarity of about 1.75.

# The appearance of a dendrogram is not unique

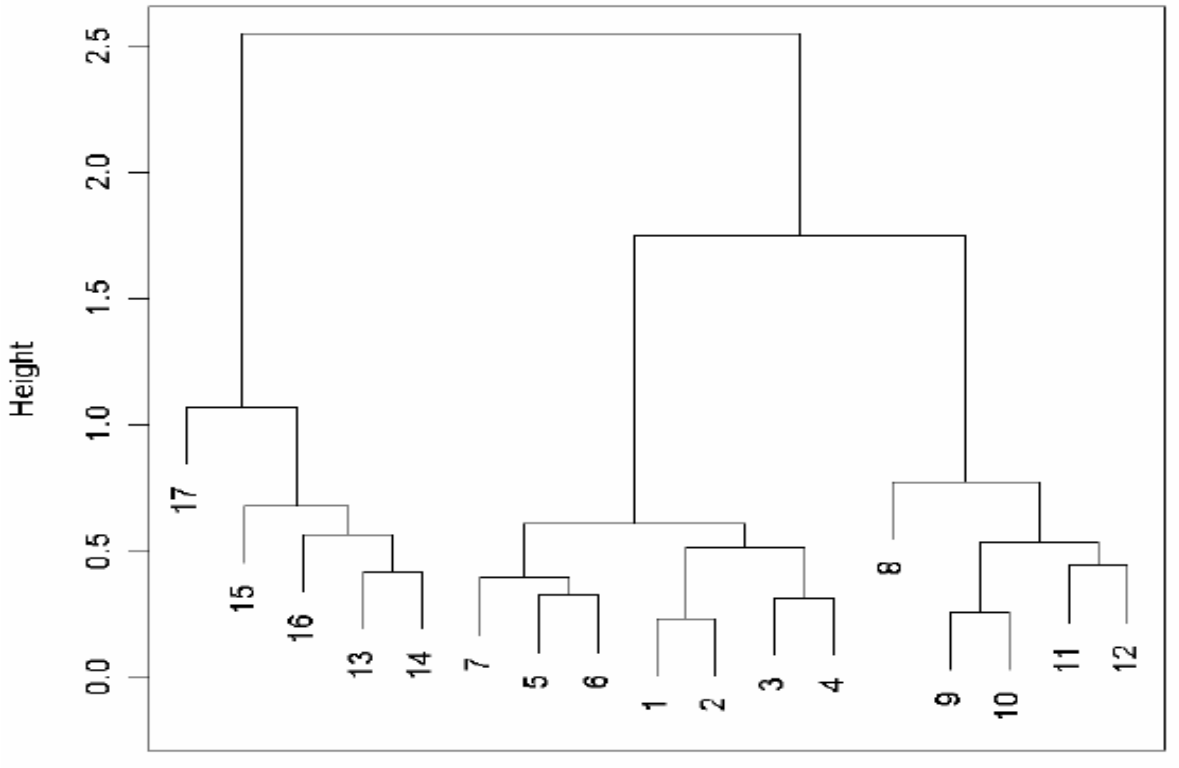



- Any two sister nodes could trade places without changing the meaning of the dendrogram.
- Thus 14 next to 7 does not imply that these objects are similar.

# The appearance of a dendrogram is not unique.

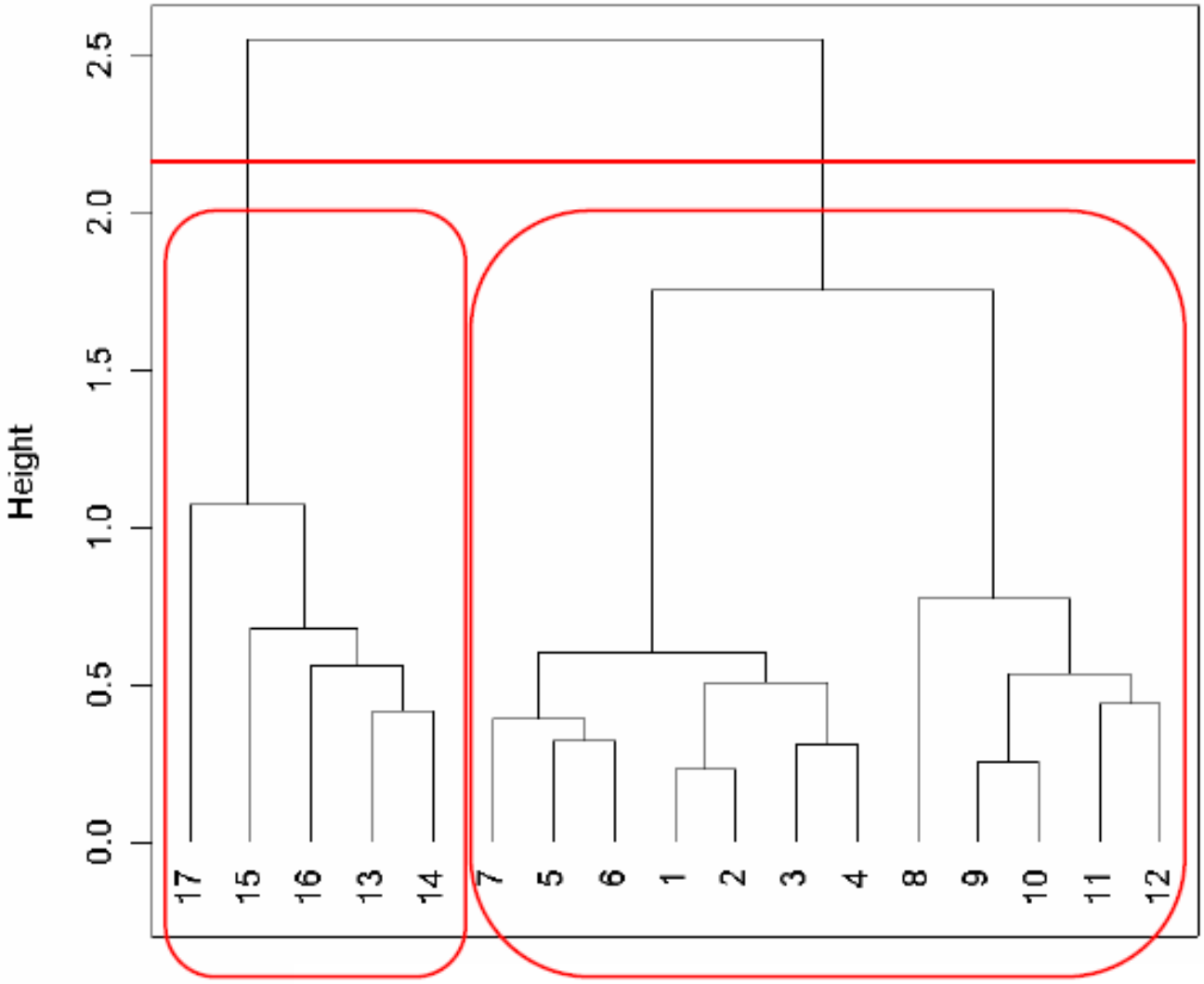



- By convention, R dendrograms how the lower sister node on the left.
- Ties are broken by observation number.



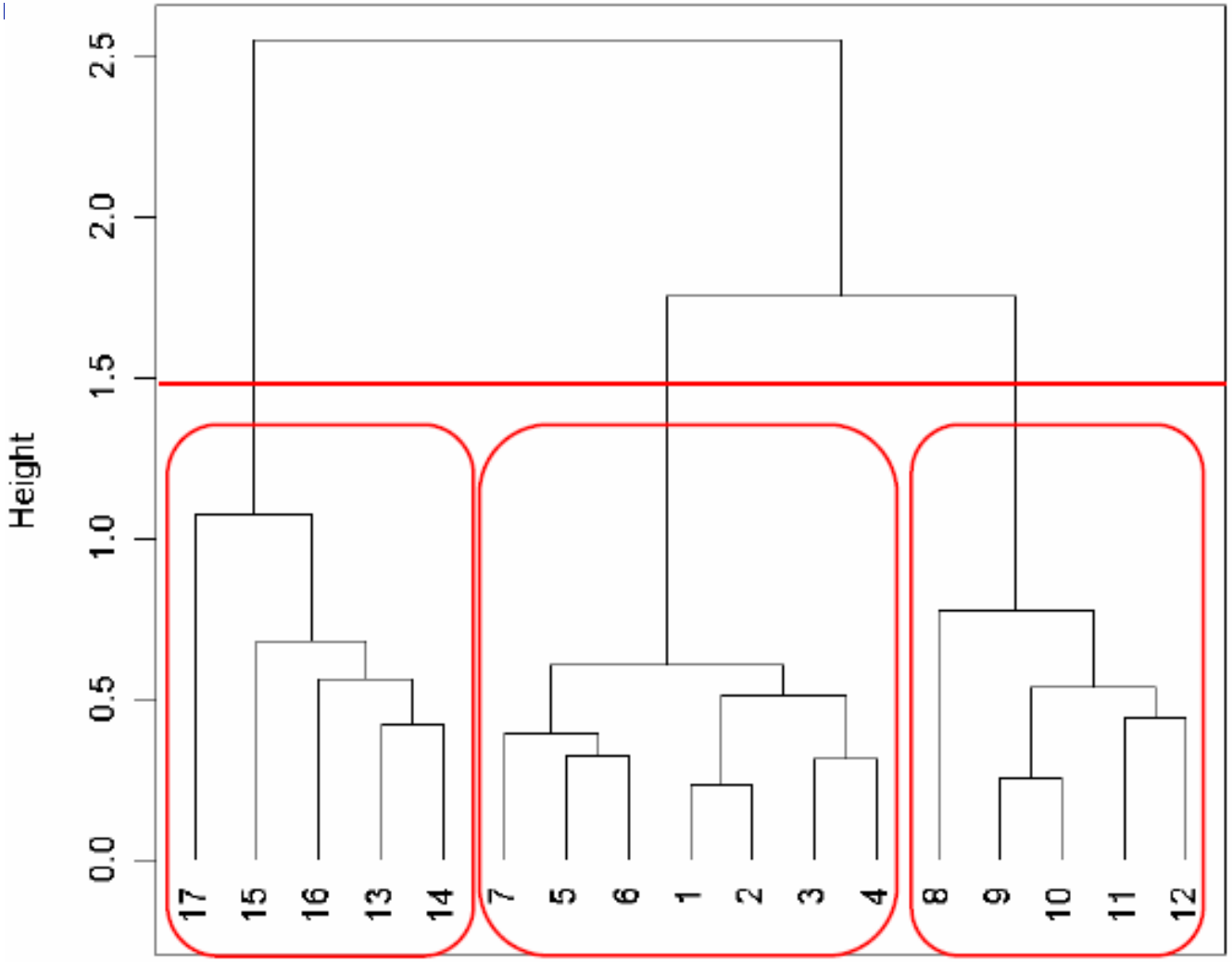

- The lengths of the branches leading to **terminal** nodes have no particular meaning in R dendrograms.

# Cutting the tree at a given height will correspond to a partitioning of the data into k clusters.



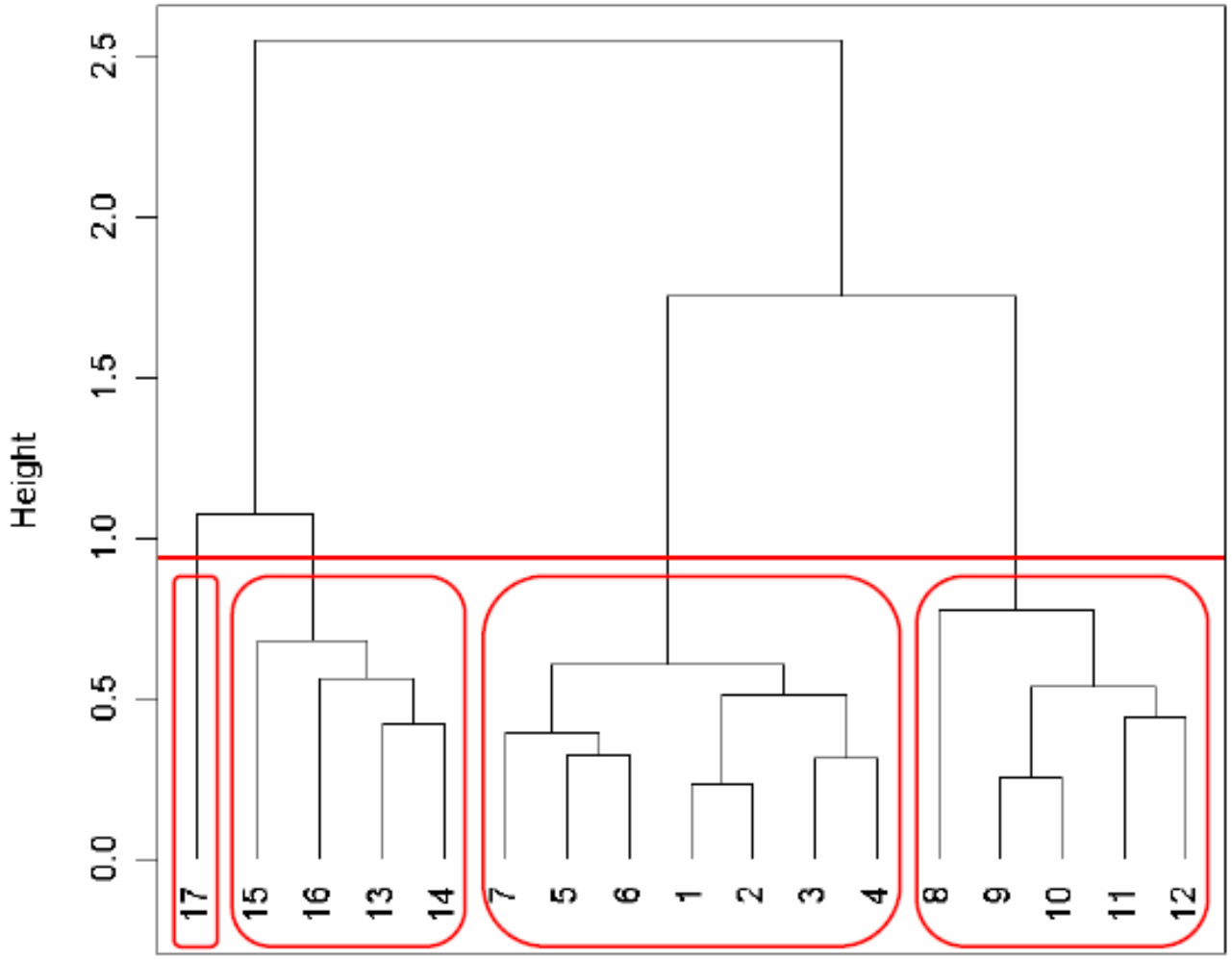

k=2 Clusters

# Cutting the tree at a given height will correspond to a partitioning of the data into k clusters.



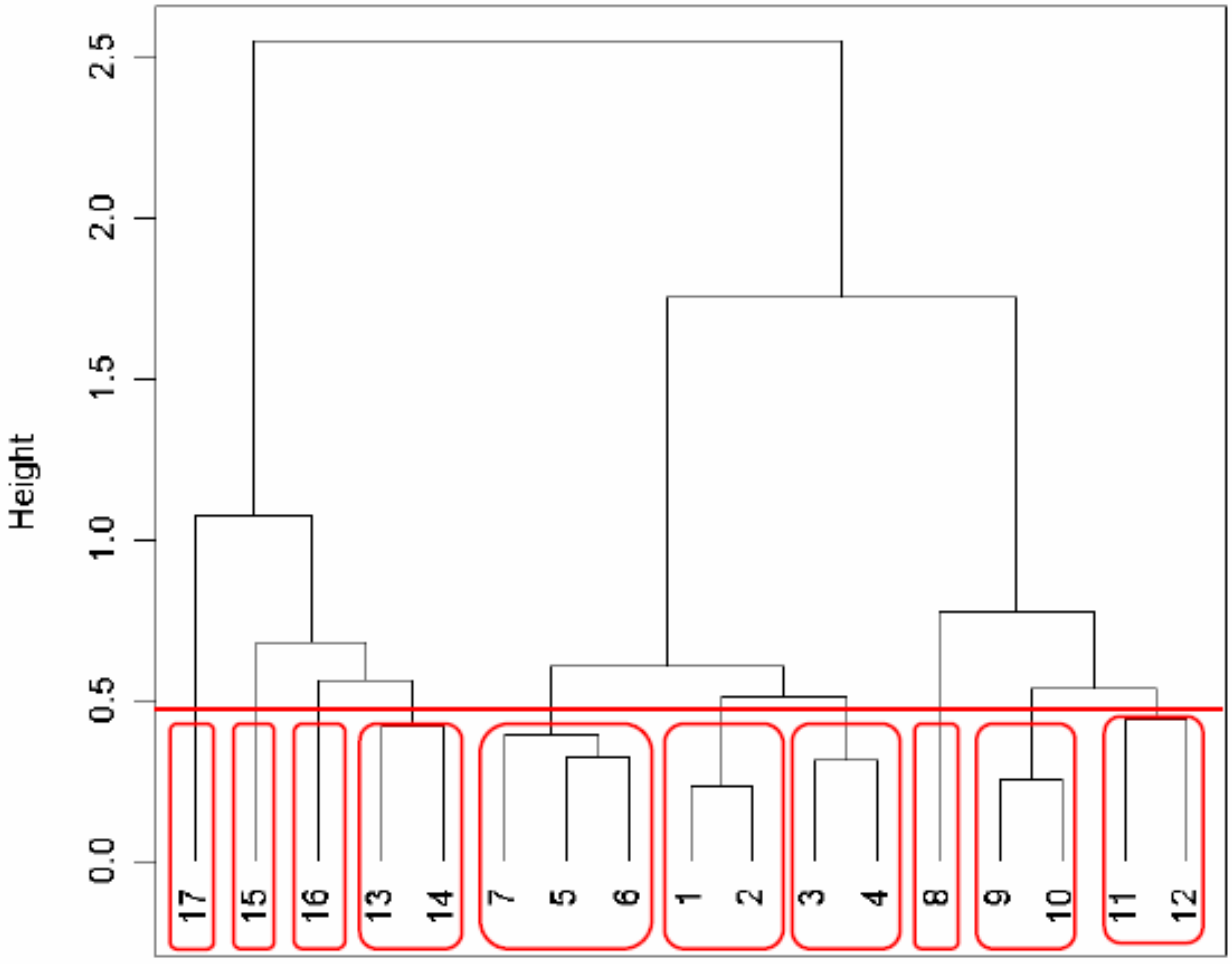

k=3 Clusters

# Cutting the tree at a given height will correspond to a partitioning of the data into k clusters.




k=4 Clusters

# Cutting the tree at a given height will correspond to a partitioning of the data into k clusters.




k=10 Clusters

# (2) Hierarchical methods

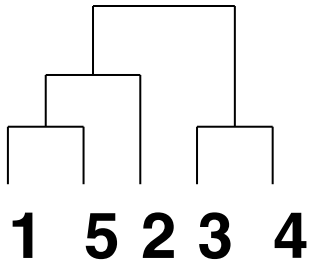
The tree can be built in two distinct ways

- bottom-up: agglomerative clustering;
- top-down: divisive clustering.

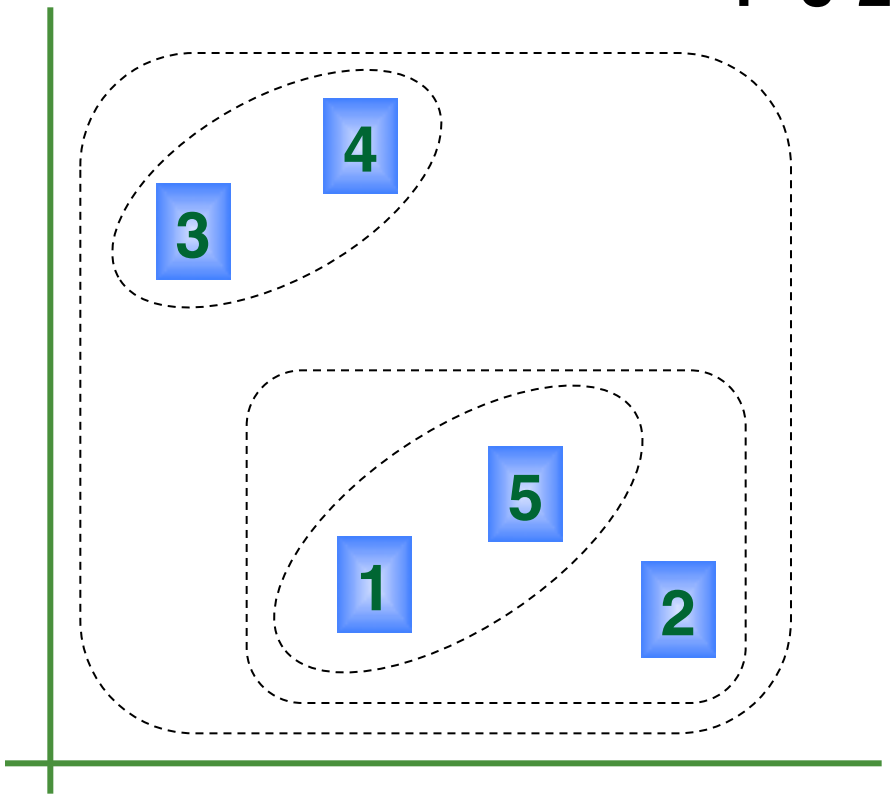
# (2) Hierarchical clustering - Agglomerative algorithm

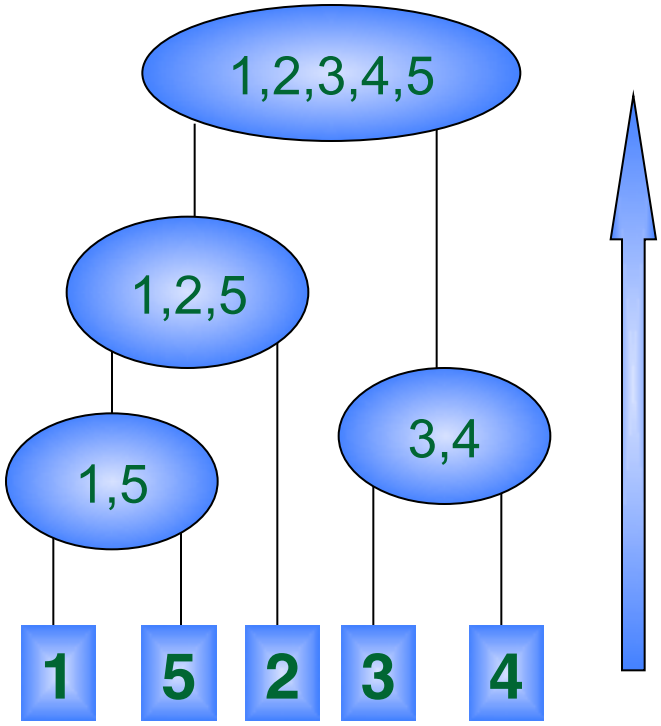
- **Bottom-up** algorithm
- Start with the objects as clusters.
- In each iteration, merge the two clusters with the minimal distance from each other - until you are left with a single cluster comprising all objects.

## Illustration of points In two dimensional space




## Agglomerative

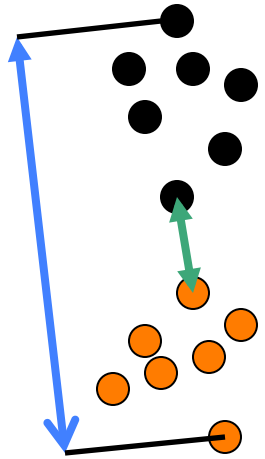


# (2) Hierarchical clustering --distances between clusters

Calculation of the distance between two clusters is based on the pairwise distances between members of the clusters.

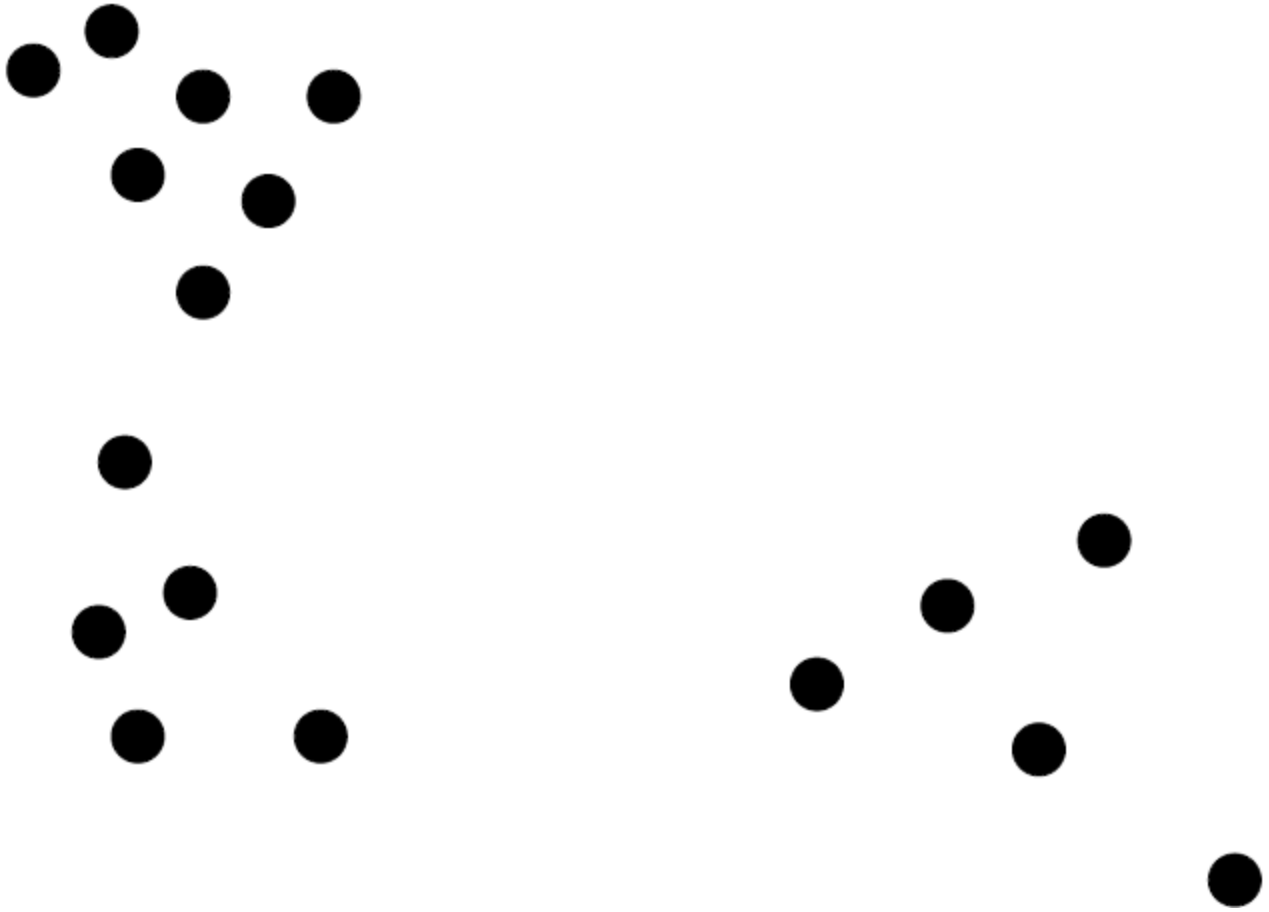
- **Complete linkage**: largest distance
- **Average linkage**: average distance
- **Single linkage**: smallest distance
- **Centroid linkage**: distance between centroids



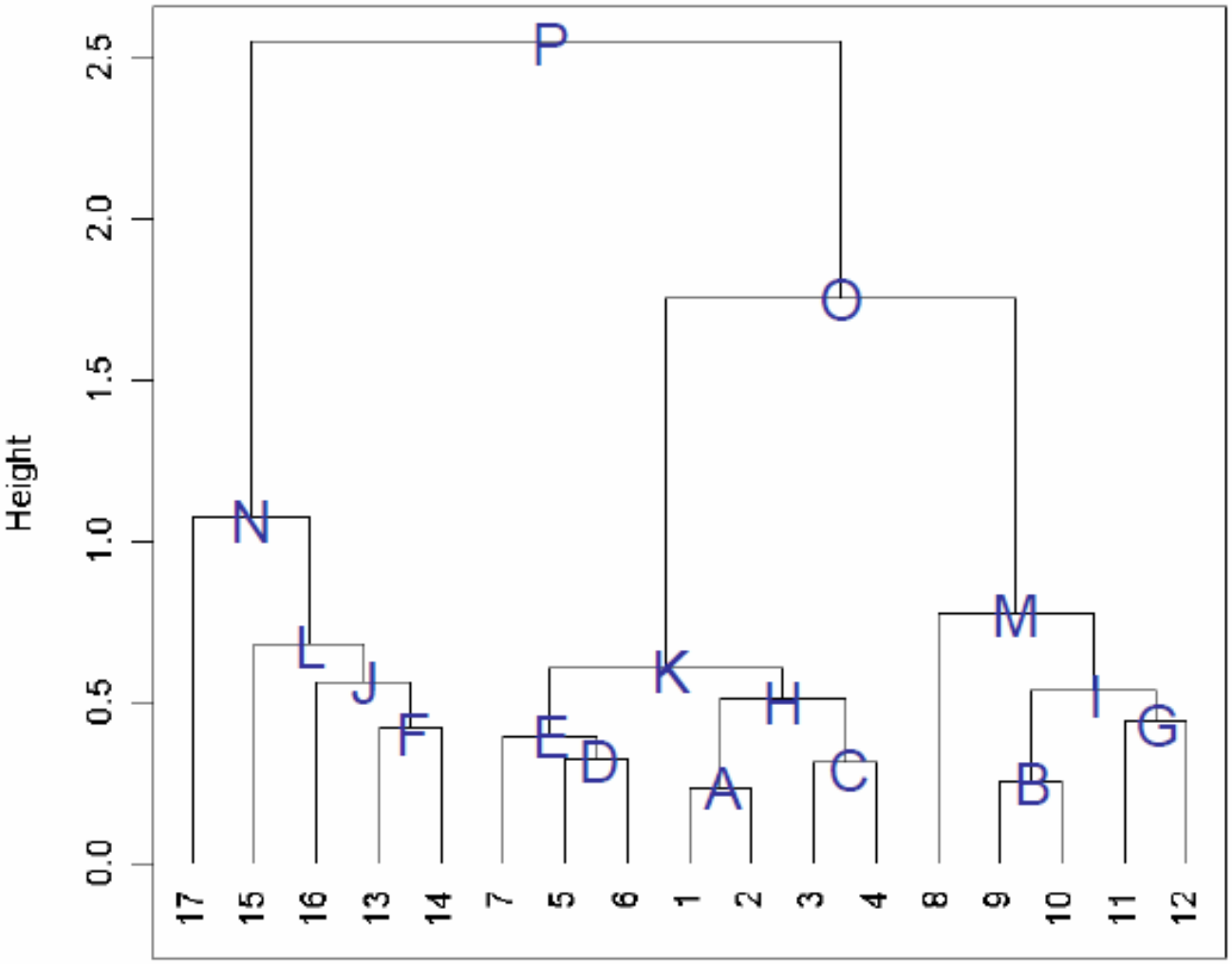
Complete linkage gives preference to compact/spherical clusters. Single linkage can produce long stretched clusters.

**Very often, use average linkage or centroid linkage**

# Simple Example Data Revisited



# Agglomerative Clustering Using Average Linkage for the Simple Example Data Set




A. 1-2

B. 9-10

C. 3-4

D. 5-6

E. 7-(5,6)

F. 13-14

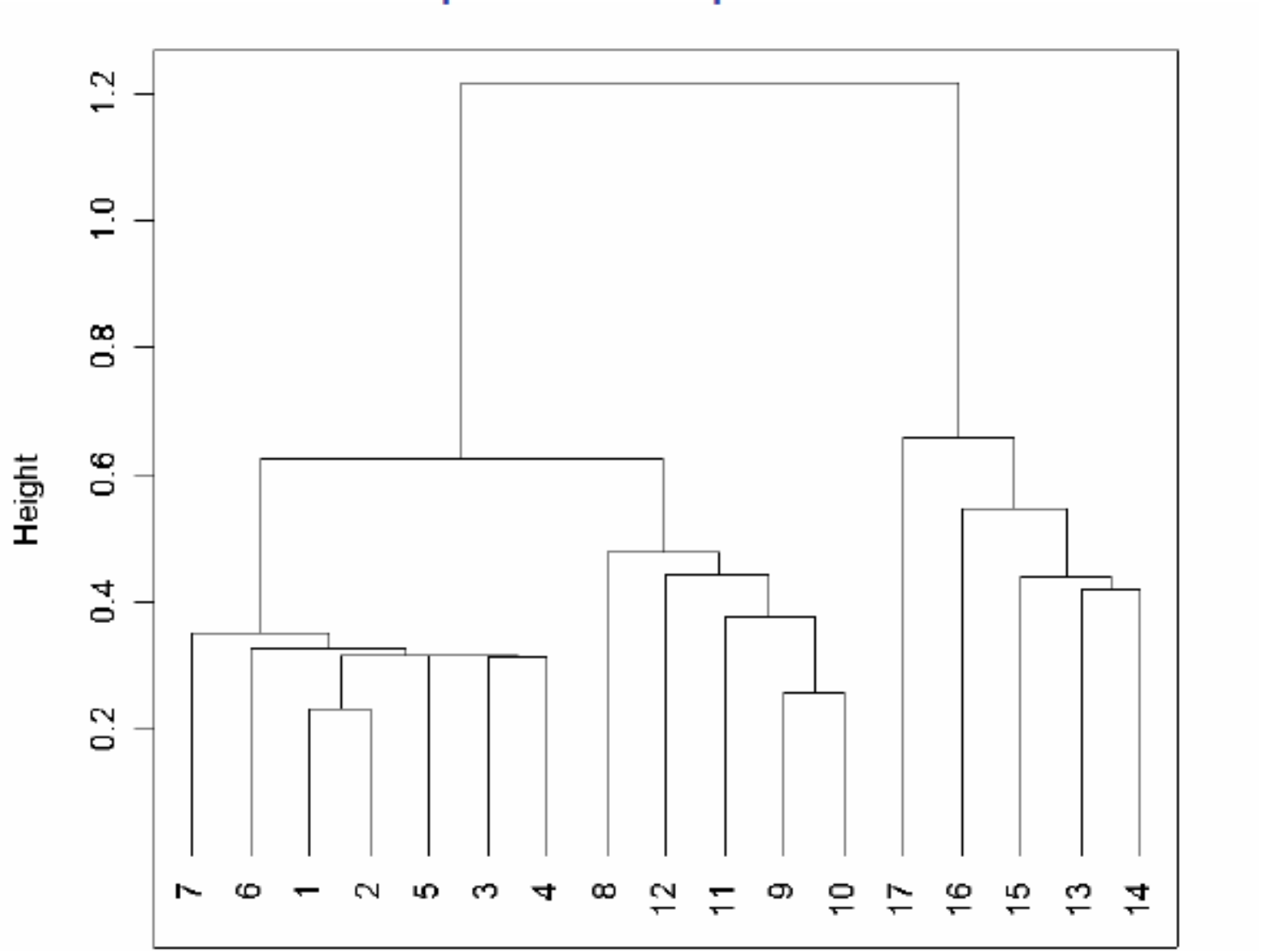
G. 11-12

H. (1,2)-(3,4)

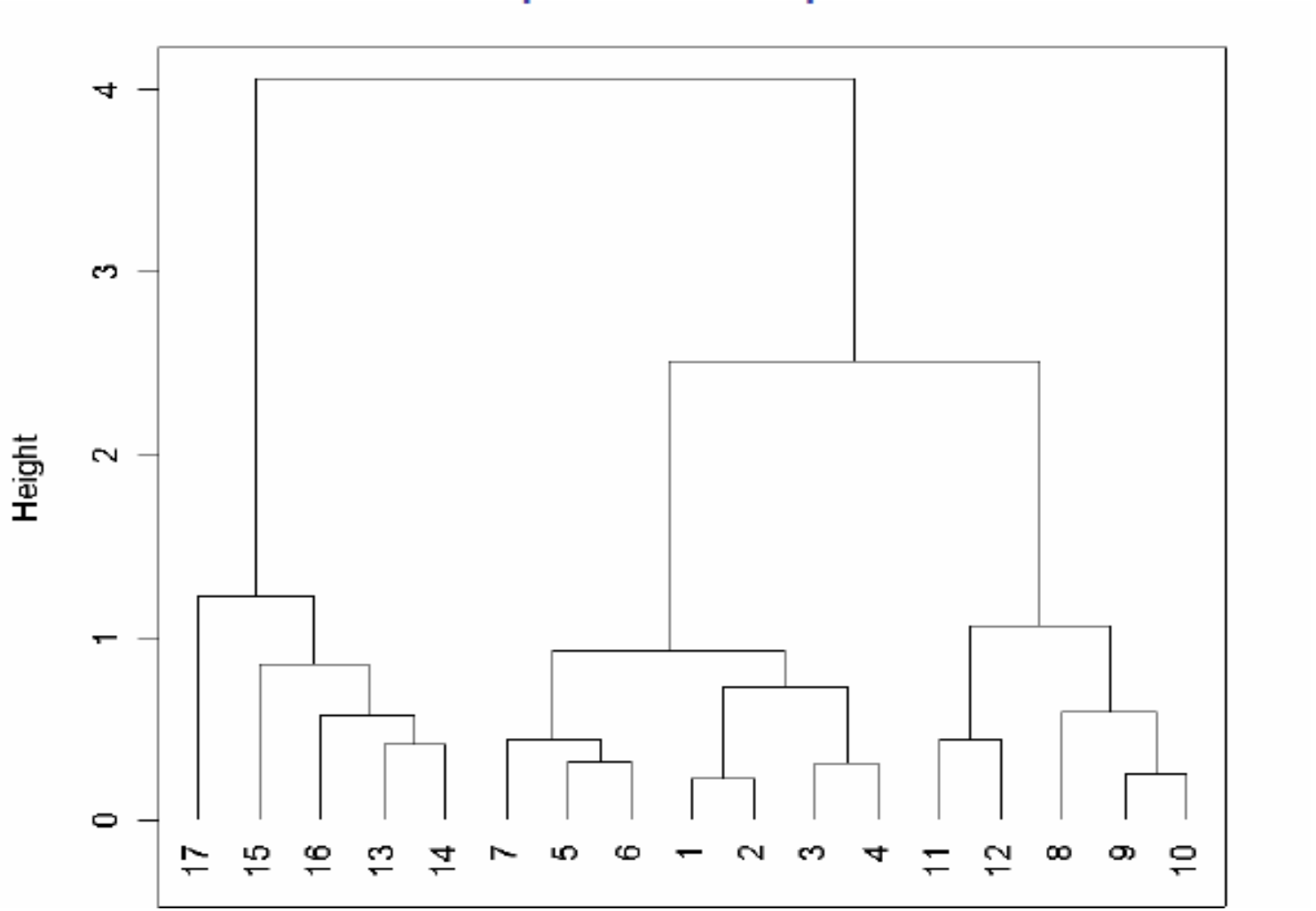
I. (9,10)-(11,12)

etc....

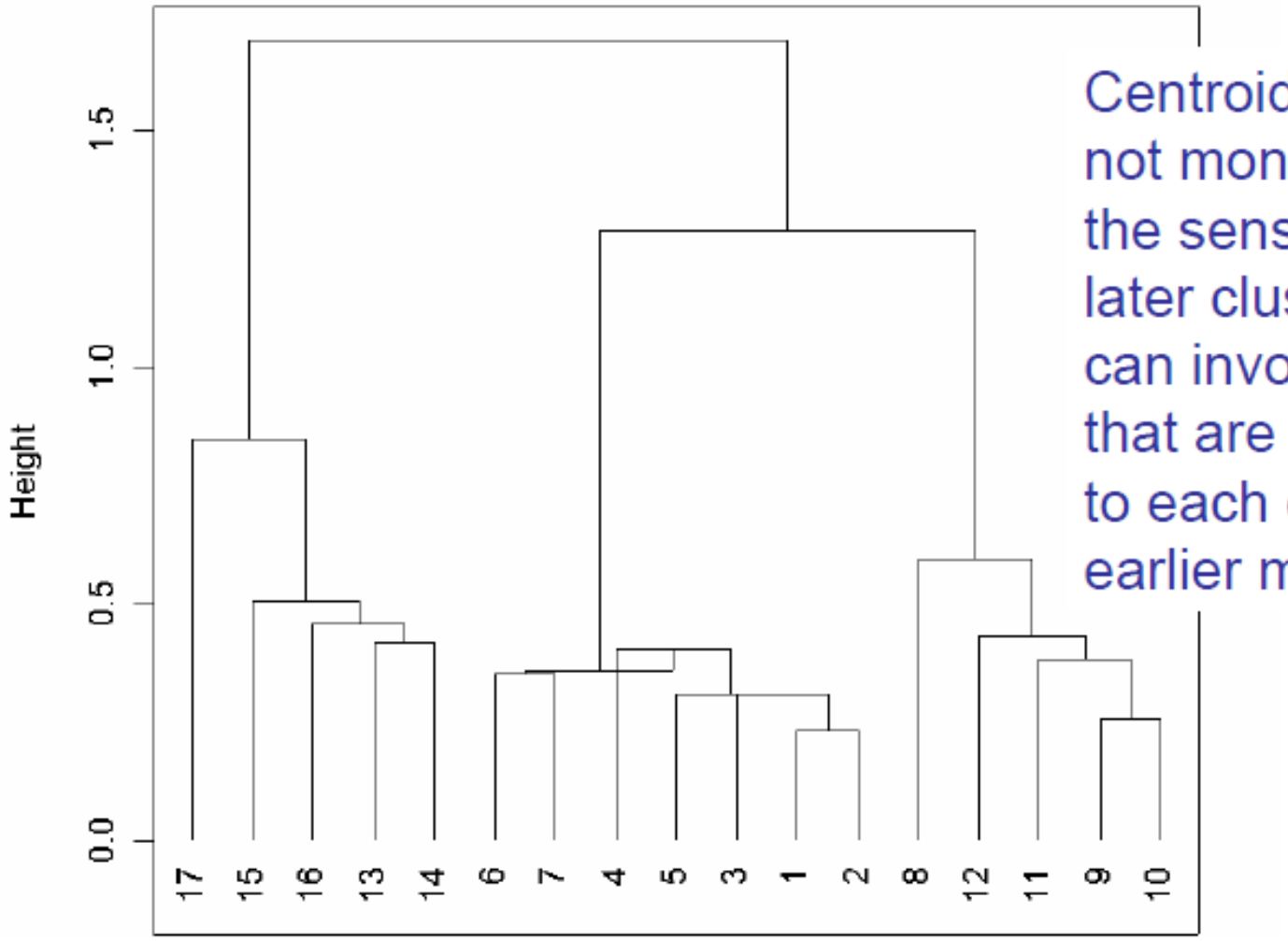
# Agglomerative Clustering Using Single Linkage for the Simple Example Data Set



# Agglomerative Clustering Using Complete Linkage for the Simple Example Data Set

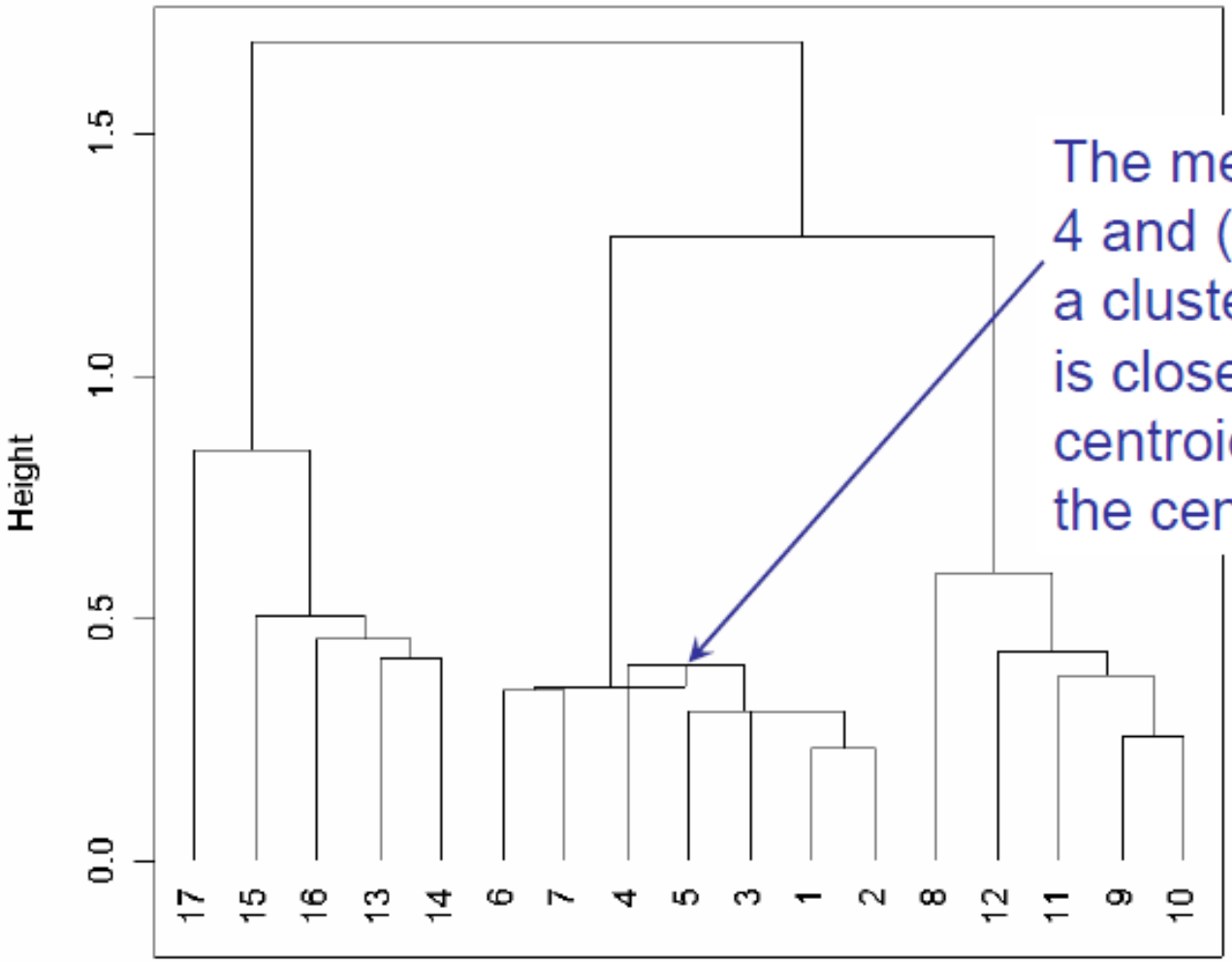


# Agglomerative Clustering Using Centroid Linkage for the Simple Example Data Set




Centroid linkage is not monotone in the sense that later cluster merges can involve clusters that are more similar to each other than earlier merges.

# Agglomerative Clustering Using Centroid Linkage for the Simple Example Data Set




The merge between 4 and (1,2,3,5) creates a cluster whose centroid is closer to the (6,7) centroid than 4 was to the centroid of (1,2,3,5).

# Which linkage (i.e., Between-Cluster Distance) is Best?

- Depends, of course, on what is meant by "best".
  - — Single linkage tends to produce "long stringy" clusters.
  - — Complete linkage produces compact spherical clusters.
  - — Average linkage is a compromise between single and complete linkage.
  - — Centroid linkage is not monotone.

# (2) Hierarchical clustering: Divisive algorithm

- Start with only one cluster.
- At each step, split clusters into two parts.
- Split to give greatest distance between two new clusters
- *Advantages*.
  - Obtain the main structure of the data, i.e. focus on upper levels of dendogram.
- *Disadvantages*.

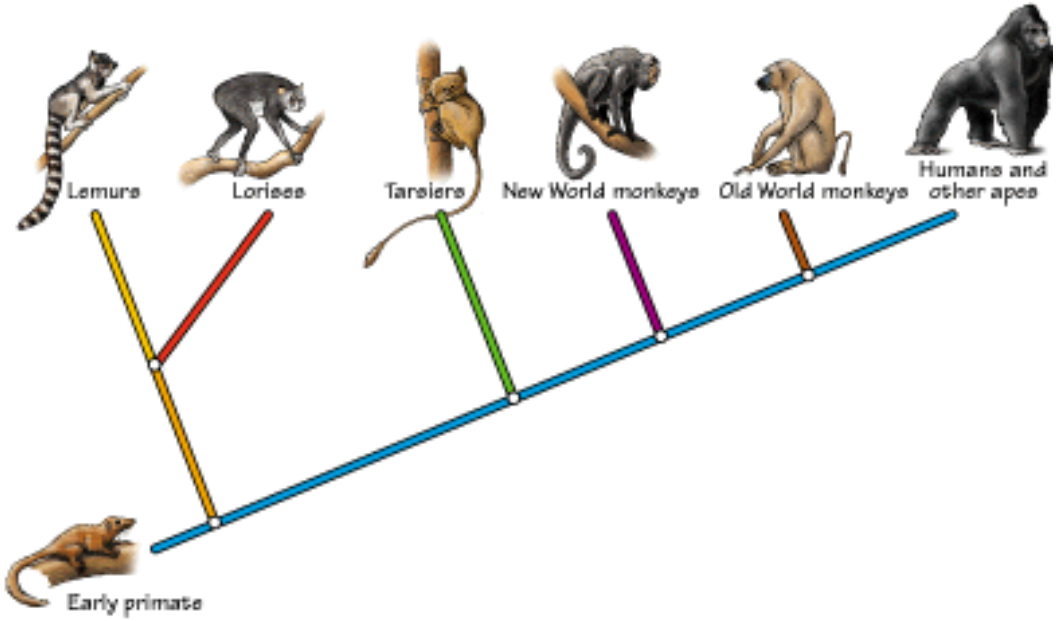
—— Computational difficulties when considering all possible divisions into two groups.

# Agglomerative vs. Divisive Clustering

- Divisive clustering has not been studied as extensively as agglomerative clustering.
- Divisive clustering may be preferred if only a small number of large clusters is desired.
- Agglomerative clustering may be preferred if a large number of small clusters is desired.

# (2) Hierarchical Clustering -- application

- Hierarchical Clustering is often used to reveal evolutionary history



# Partitioning or Hierarchical?

- Partitioning:
  - Advantages
    - Optimal for certain criteria.
    - Genes automatically assigned to clusters
  - Disadvantages
    - Need initial k;
    - Often require long computation time.
    - All genes are forced into a cluster.

# Partitioning or Hierarchical? - 2

- Hierarchical
  - Advantages
    - Faster computation.
    - Visual.
  - Disadvantages
    - Unrelated objects are eventually joined
    - Rigid, cannot correct later for erroneous decisions made earlier.
    - Tend to be sensitive to small changes in the data
    - Hard to define clusters.

# (3) How to determine $K$ (the number of clusters)?

- There is no easy answer.
- Many heuristic approaches try to compare the quality of clustering results for different values of *K.*
  - **Silhouette width**: Choose K that maximizes the average *silhouette width.*
    - Rousseeuw, P.J. (1987). *Journal of Computational and Applied Mathematics,* ***20, 53-65.***
    - Kaufman, L. and Rousseeuw, P.J. (1990). Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York.
  - **gap statistic**: Choose K according to the *gap statistic.*
    - Tibshirani, R., Walther, G., Hastie, T. (2001). *Journal of the Royal Statistics Society, Series B-Statistical Methodology,* ***63, 411-423.***

# (3) Estimating number of clusters using silhouette

- The silhouette width of an object is

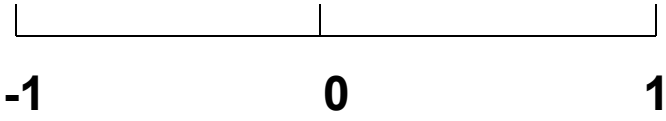
$$(B-W)/\max(B,W)$$

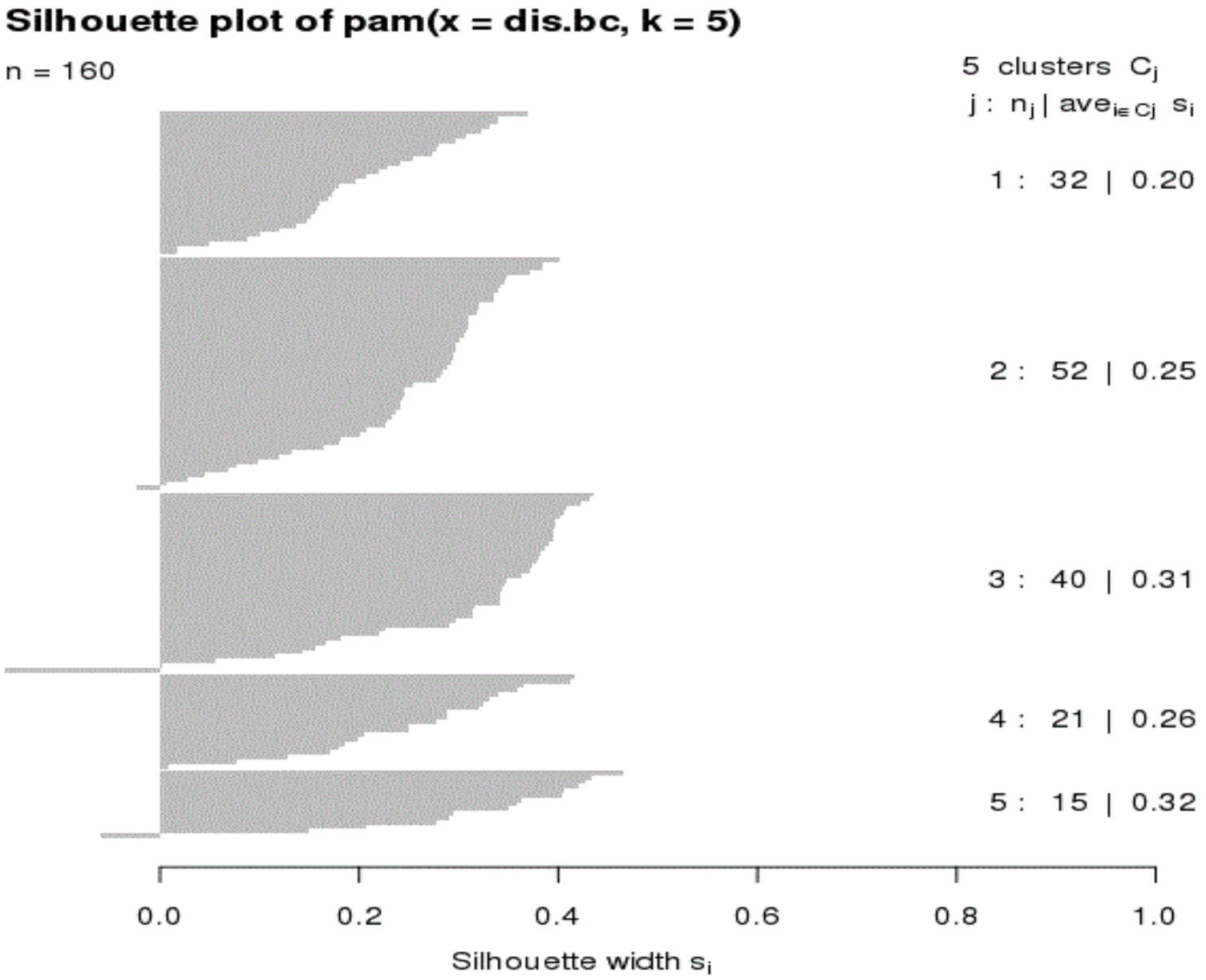
where W=average distance of the object to all other objects ***within*** its cluster and B=average distance of the object ***in its nearest*** neighboring cluster.

- The silhouette width will be between -1 and 1.

# Silhouette width

- Values near 1 indicate that an object is near the center of a tight cluster.
- Values near 0 indicate that an object is between clusters.
- Negative values indicate that an object may be in the wrong cluster.

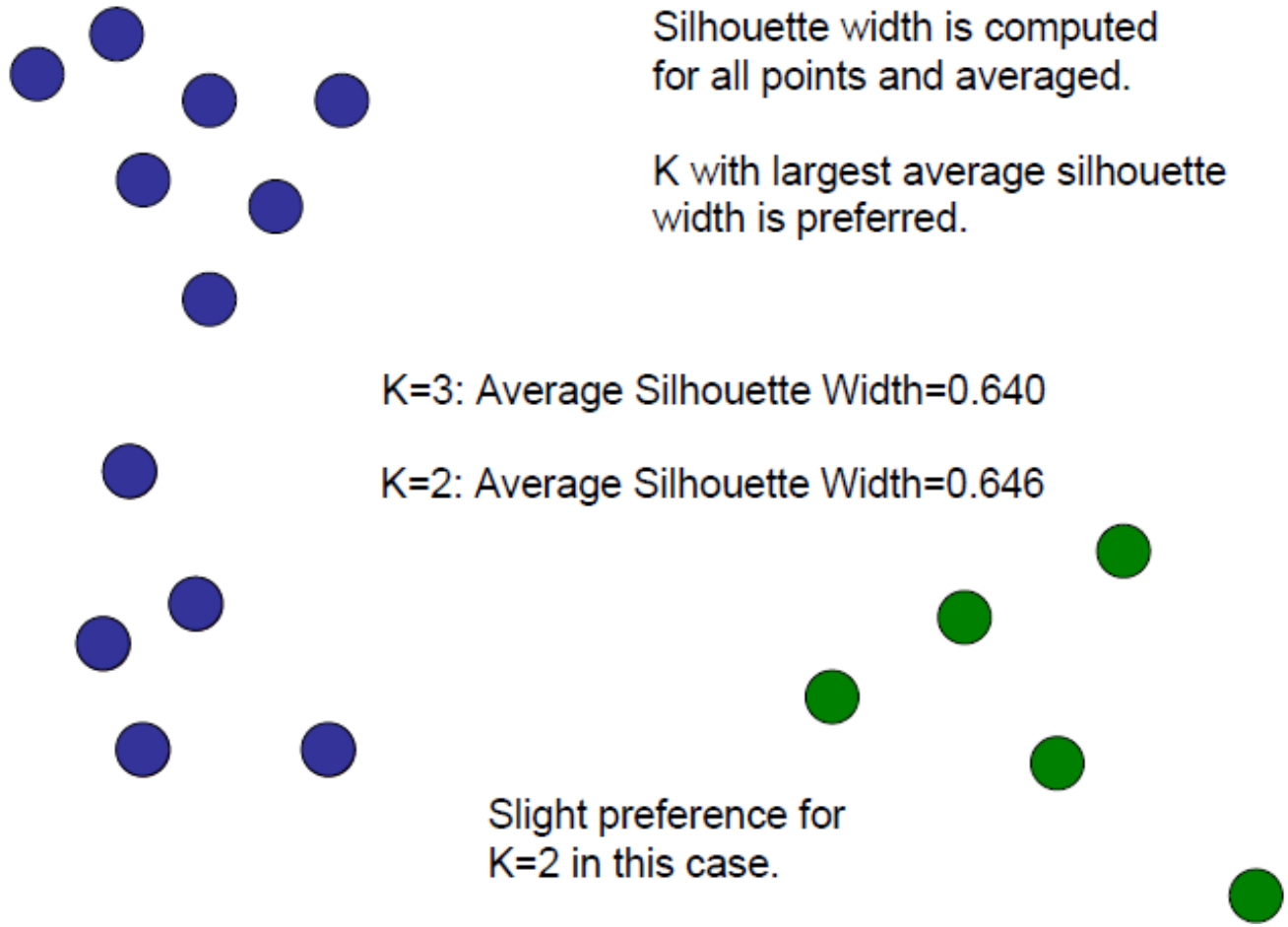



Silhouette plot of pam(x = dis.bc, k = 5)
n = 160
5 clusters $C_j$
j : $n_j$ | $ave_{i \in Cj}$ $s_i$
1 : 32 | 0.20
2 : 52 | 0.25
3 : 40 | 0.31
4 : 21 | 0.26
5 : 15 | 0.32
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette width $s_i$
Average silhouette width : 0.26

# Silhouette width

- The silhouette widths of clustered objects can be averaged.
- A clustering with a high average silhouette width is preferred.
- For a given method of clustering, we may wish to choose the value of K that maximizes the average silhouette width.

# Example: silhouette



Silhouette width is computed for all points and averaged.

K with largest average silhouette width is preferred.

K=3: Average Silhouette Width=0.640

K=2: Average Silhouette Width=0.646

Slight preference for K=2 in this case.

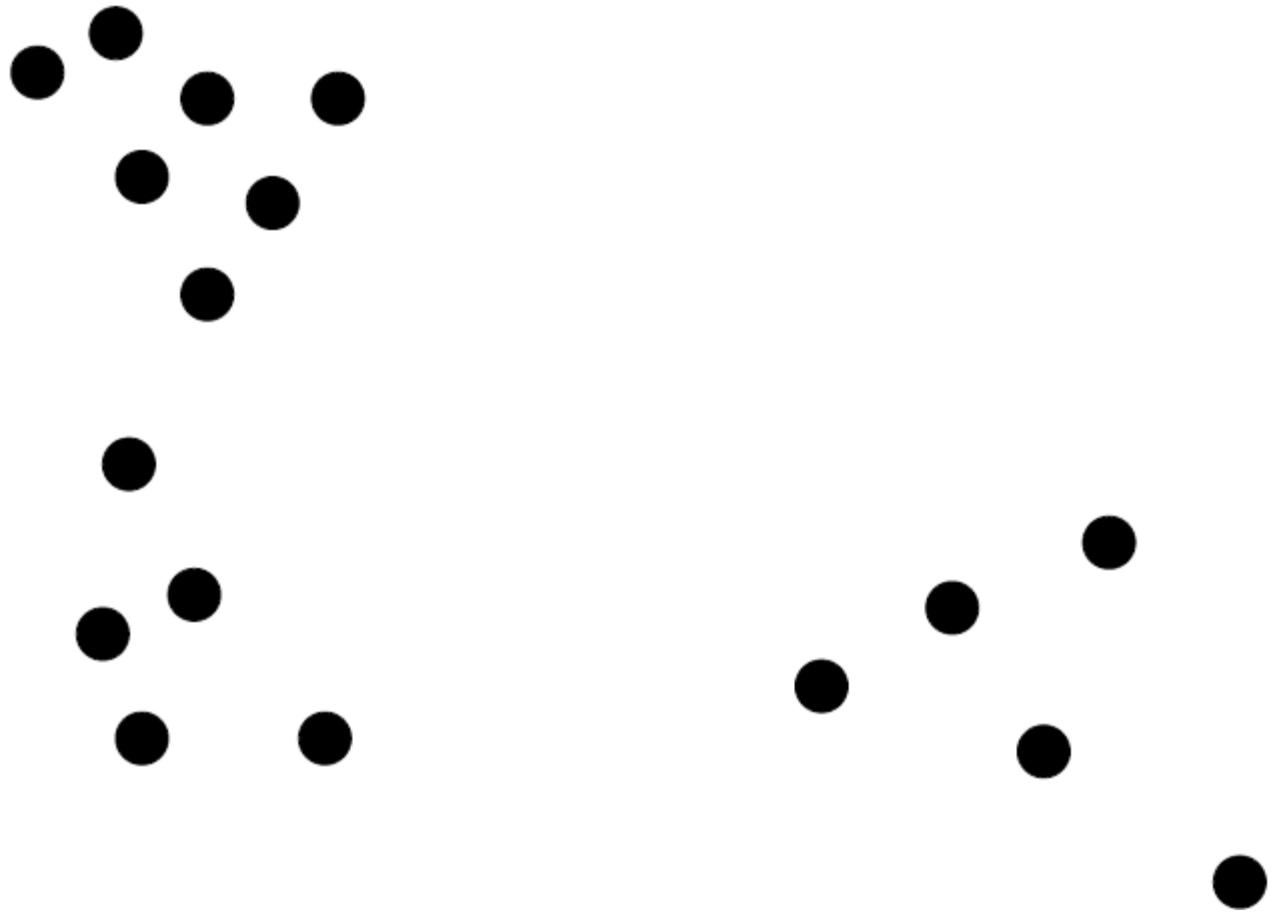
# Gap statistic

Find k at which within-cluster variation is min

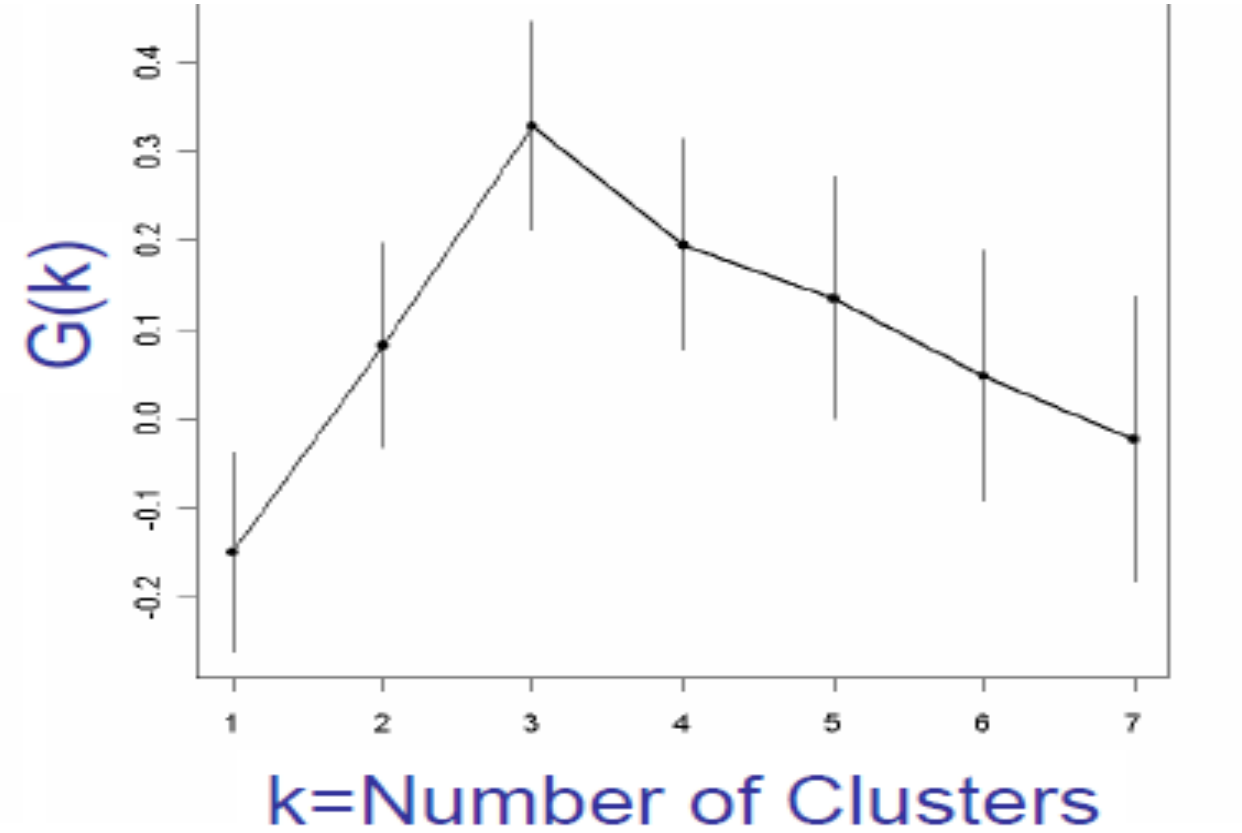
That is: $\hat{K} = \min \{ k : G(k) \geq G(k+1) - S_{k+1}\sqrt{1+1/N} \}$

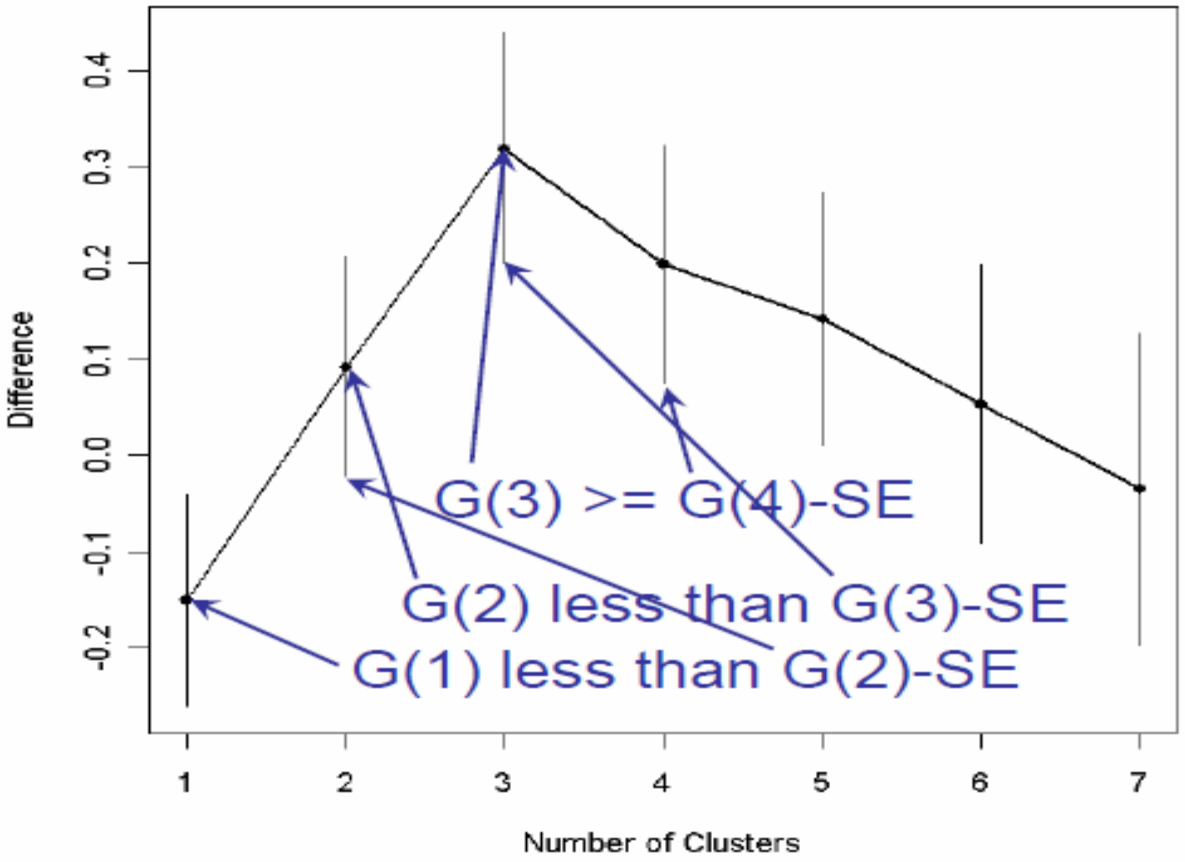
where $S_k\sqrt{1+1/N}$ denotes an approximate of standard error of gap statistic $G(k)$ and N is the number of randomly generated data sets with null distribution

# Gap statistic: simple example data revisited



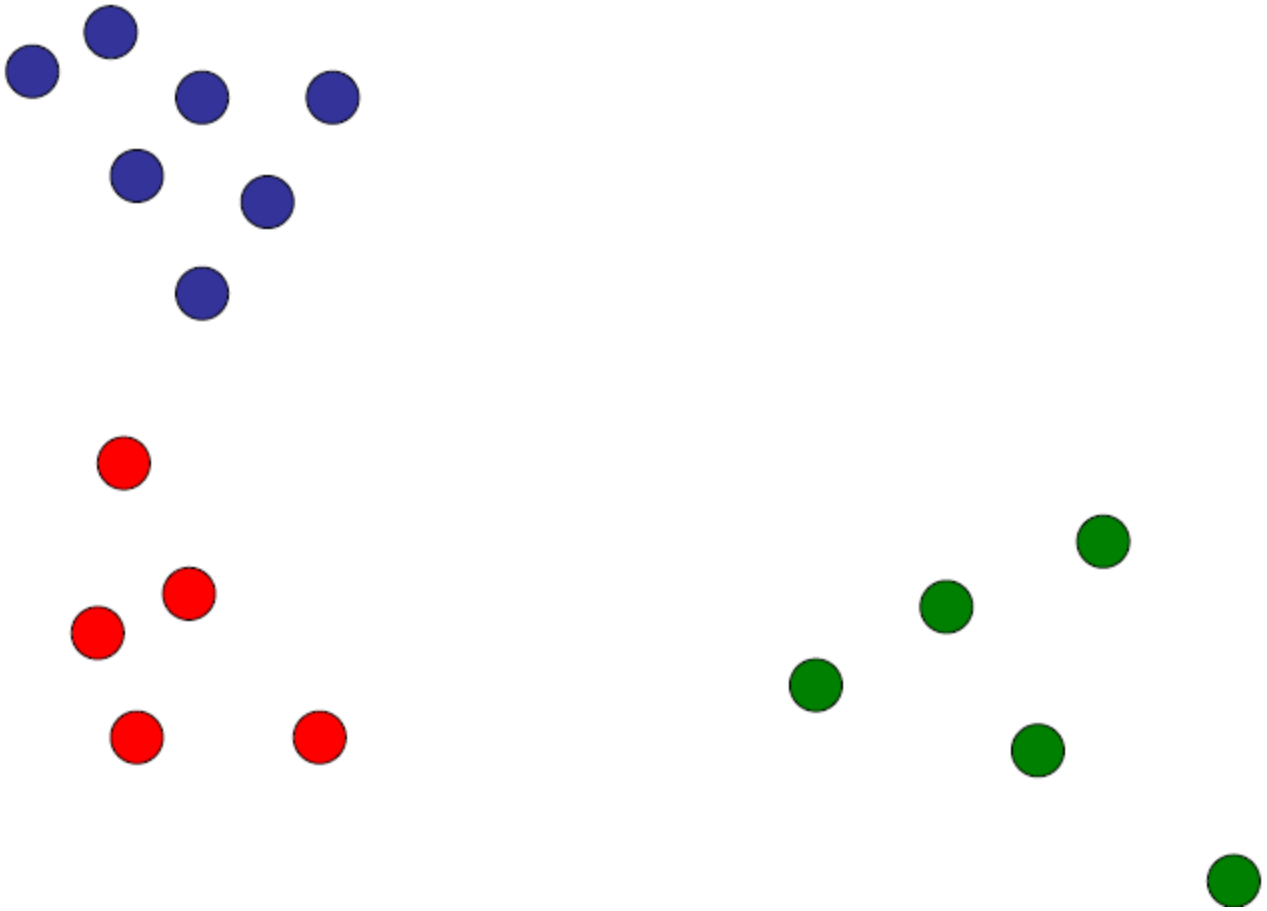
# N=1000






**Suggest: K= 3 clusters**

# Gap result:



# Other methods estimating number of clusters

- resampling and  non-resampling methods

The bottom line is that none work very well in complicated situation and, to a  large extent,  clustering lies outside a usual statistical framework.

It is always reassuring when you are able to characterize a newly discovered clusters using information that was not used for clustering.