Cluster Analysis

Introduction to Statistical Learning
Bachelor in *Global Governance*
University of Rome - Tor Vergata

Marco Stefanucci
Department of Economics and Finance
University of Rome - Tor Vergata
marco.stefanucci@uniroma2.it

Classification

- Usual and natural operation in the organization of knowledge
- Given a set of individuals / objects / elements we want to identify subsets that have somehow common characteristics
- Obtain classification criteria to locate aggregations, within the heterogeneity of a collective / population, capable of
  1. simplifying the perception of this collective
  2. interpreting the typical elements of differentiation

What does common characteristic mean?

In the simplest case, variables of classification:

- disease diagnosis: (healthy / diseased)
- degree of customer satisfaction: (zero / average / total)

proximity to a particular place of reference:

- real: topological / geographical classification
- ideal: by mapping individuals through scores based on internal / external variables

- Guessing an already defined classification
  - if already available, it tries to highlight the statistical link with other external variables, assess their limits and potential and perhaps isolate those most related to explicit classification variable
  - if not available yet, supervised classification!

- Explore, within a collective phenomenon on which numerous variables are detectable for each unit, the different possibilities of subdividing into subsets showing some degree of homogeneity: unsupervised classification!
  - understand a complex phenomenon and operate in a targeted manner in the subgroups
  - isolate some particularly interesting subgroups as represent a typology on which you want to deepen the research

Cluster analysis procedure

1. Selection of the variables of interest
2. Raw data manipulation
3. Selection and implementation of one of the clustering methodologies
4. Output analysis:
   - trade off in the choice of the number of groups (Occam's razor)
   - groups interpretation

Preliminary notions

- Partition of a set
- Dissimilarity and distance
- Deviance decomposition

Partition of a set

The partition of a set A is the collection of subset

$$\pi = \{I_1, I_2, \ldots, I_k\}$$

such that each subset has no elements in common with any of the others

$$I_g \cap I_r = \emptyset \;\; \forall g \neq r$$

and the union of all subsets I_g reconstitutes the given set A.

$$A = \bigcup_{g=1}^{k} I_g.$$

Dissimilarity and Euclidean distance

- Distance typically used between two units on which p measurements have been taken: the Euclidean distance

$$d_{ij} = d(x_i, x_j) = \sqrt{\sum_{r=1}^{p} (x_{ir} - x_{jr})^2}$$

- Dissimlilarity index:

$$d_{ij}^2 = d^2(x_i, x_j) = \sum_{r=1}^{p} (x_{ir} - x_{jr})^2 = ||x_{ir} - x_{jr}||^2$$

Dissimilarity and Euclidean distance

- Distance $d : D \times D \in \mathbb{R}$
  - $d(x_i, x_j) = 0 \Leftrightarrow x_i = x_j$
  - $d(x_i, x_j) = d(x_j, x_i)$ simmetry
  - $d(x_i, x_j) \leq d(x_i, x_k) + d(x_k, x_j)$ triangle inequality
- Dissimilarity $d : D \times D \in \mathbb{R}$
  - $d(x_i, x_j) \geq 0 \ \forall \ x_i, x_j$
  - $d(x_i, x_i) = 0 \ \forall x_i$
  - $d(x_i, x_j) = d(x_j, x_i)$ simmetry

Total deviance decomposition

$$\mathbf{T} = \mathbf{B} + \mathbf{W}$$

where:

- **T**→Total deviance
- **B**→ Between deviance
- **W**→ Within deviance

Suppose we observe one variable n times, the n units partitioned in G groups of size n_g, such that $\sum_{g=1}^{G} n_g = n$

$$\sum_{j=1}^{n}(x_j - \bar{x})^2 = \sum_{g=1}^{G}\sum_{i=1}^{n_g}(x_{ig} - \bar{x})^2 = \sum_{g=1}^{G} n_g(\bar{x}_g - \bar{x})^2 + \sum_{g=1}^{G}\sum_{i=1}^{n_g}(x_{ig} - \bar{x}_g)$$

Clustering algorithms

- Hierarchical
  - Aggregation Algorithms
  - Splitting Algorithms
- Not hierarchical
  - groupings around means (k-means)
  - groupings around representative units (k-medoids)

Not hierarchical algorithm

You get a single partition with number of components (groups) specified a priori.

Starting from an initial subdivision into a fixed number of clusters, we proceed sequentially obtaining at each step a new partition that improves the current partition according to a chosen criterion.

K-means algorithm

1. Initialization: we choose the desired number of groups (K), and set as starting points K centers or means (K p-dimensional vectors): $(\mathbf{m}_1, \ldots, \mathbf{m}_K)$
2. Allocate the i-th unit for $i = 1, \ldots, n$ into the group j such that the distance between the unit x_i and the center $\mathbf{m}_k$ is minimized;
3. Calculate the new centers $(\mathbf{m}_1, \ldots, \mathbf{m}_K)$ basing on the new allocations to the groups (clusters).

Repeat the steps 2 and 3 until convergence, i.e. when the allocation of the observations does not vary between two iterations.

K-means algorithm

This algorithm only works starting with a data matrix in which all variables are quantitative (measurements). Let n_j be the number of units that are assigned in step 2 to the j-th group G_j

$$\mathbf{m}_j = (m_{j1}, \ldots, m_{jr}, \ldots, m_{jp})$$

$$m_{jr} = \frac{1}{n_j} \sum_{i \in G_j} x_{ir}$$

The objective to be achieved is formalized in the following problem of optimization: identify K groups for which the internal variability is minimal and (automatically) the between variability is maximal.

Generalization of the K-means method

To allow using only one dissimilarity matrix among units, the identification of centers through averages (means) can be modified with corresponding actual centers being the coordinates of central units (medoids) which minimize the distance between the unit candidate to play the role of center and the other units of the group.

There modification of step 3 becomes much more expensive from the computational point of view.

The modification of step 2 is immediate and does not involve additional difficulties. Such modifications can lead to algorithms more robust to the presence of anomalous distances between single pairs of units.

PAM (Partitioning Around Medoids) algorithm

1. Initialization:
   - calculation of the dissimilarity matrix
   - choice of K points as initial candidate medoids
2. (build phase) assign each unit to the nearest medoid to form the K matching clusters
3. (swap phase) for each cluster, check whether any unit in the cluster is able to decrease the overall dissimilarity of the same cluster and eventually proceed with the replacement of the medoid with that unit, decreasing the dissimilarity of the cluster more.

   If for at least one cluster the medoid is replaced, go back to step 2.

Hierarchical algorithms

Do not specify the number of groups, which ranges from the extreme minimum (1 group only) to the extreme maximum (many groups how many units are considered). This range of possibilities is typically represented by the dendrogram.

- Aggregative algorithms: start from the consideration of the maximum number of groups n obtainable considering each group formed by a single unit; proceed sequentially to obtain a smaller number of groups ($n \to n-1 \to n-2 \dots$).
- Splitting algorithms: start from the consideration of all the units belonging to only one group and proceed by subdivision according to suitable criteria of optimality.

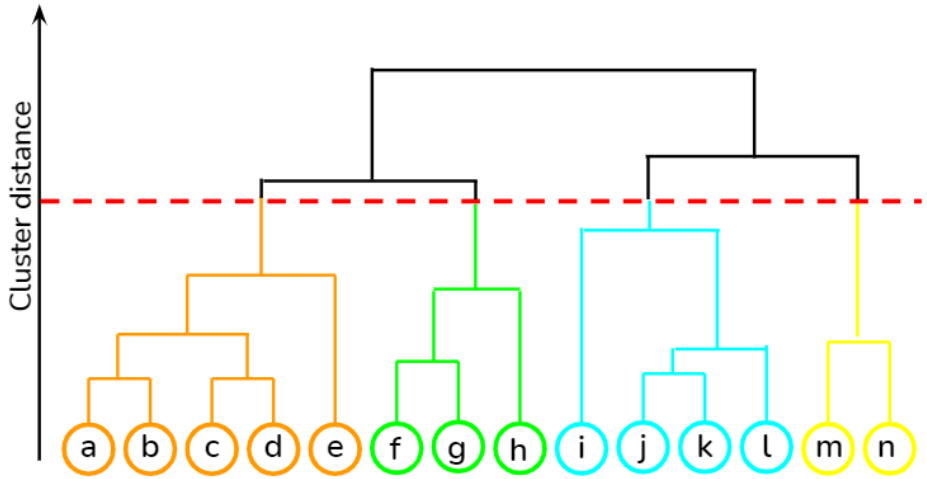
Dendrogram

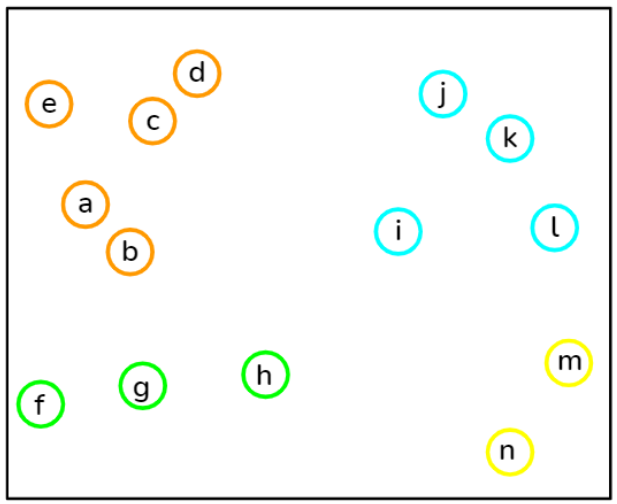
It is a visualization of the subsequent aggregation process (aggregation methods) of the n minimal subgroups (each consisting of a single unit) in groupings composed of the aggregation of pairs of subgroups according to a pre-established criterion; or the subsequent subdivision process(splitting methods) starting from the set of all units.

What you must be able to see in a dendrogram:

1. how to identify the subdivision into K groups;
2. which units belong to a group;
3. indications on the selection of an adequate number of groups to well represent the (possible) grouping structure;

Dendrogram





Aggregative methods

The general operating principle is as follows:

1. It is initialized considering $K = n$ distinct groups each consisting of one element starts from a matrix of distances/dissimilarities between the K groups

2. aggregate the two groups that are less distant between all the $K(K-1) = 2$ possible pairs and the new distance matrix relating to $(K-1)$ groups is recalculated

3. If $K = 1$ we stop otherwise we decrease the index K by 1 and go back to step 2.

- The matrix of distances is necessary for the elaboration. It is provided directly as input or it is calculated as a function of raw data.
- The calculation of the new matrix of distances / dissimilarity when the number of groups diminishes by one presupposes the notion of distance between sets of units (between two groups). For example the distance between group I and the group J will be denoted by $d(I;J)$.
- When the two groups are made up of a single unit, the definition is natural. In the non-trivial case of groups consisting of several units, different definitions can be used. Among these we point out the most frequently used:

$$d_{SL} = \min_{i \in I, j \in J} d(i,j)$$

$$d_{CL} = \max_{i \in I, j \in J} d(i,j)$$

$$d_{AL} = \text{mean } d(i,j)$$

Note that the operations in question can be calculated starting from the last matrix of distances obtained. If group I is obtained by aggregation of groups G_1 and G_2 from the last distance matrix obtained, I can identify the quantities a $d(G1; J)$ and $d(G2; J)$ from which I deduce:

- Single-Linkage or Nearest neighbor:

$$d_{SL} = \min_{i=1,2} d(G_i, J)$$

- Complete-Linkage:

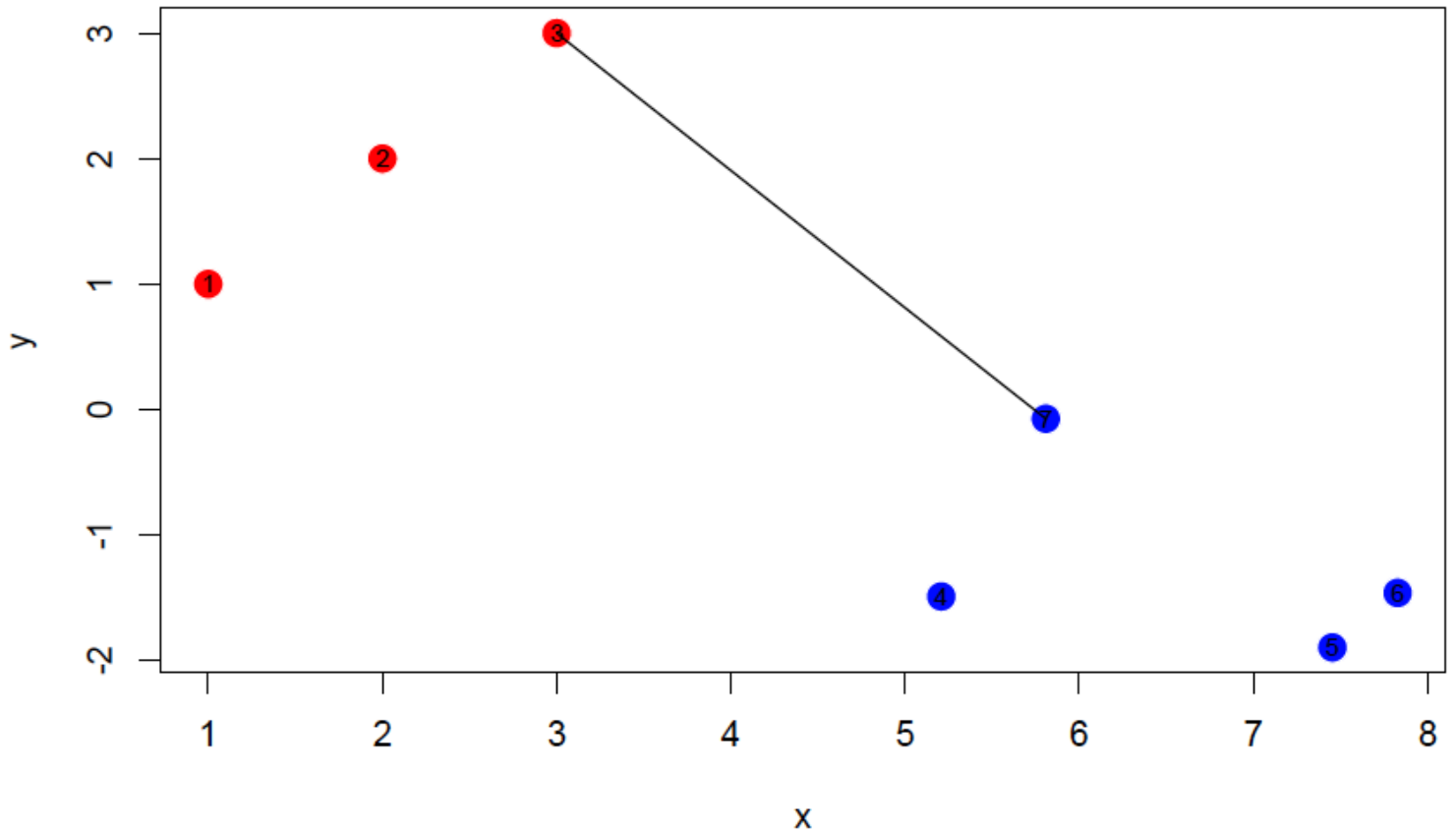
$$d_{CL} = \max_{i=1,2} d(G_i, J)$$

- Average-Linkage or Mean distance:

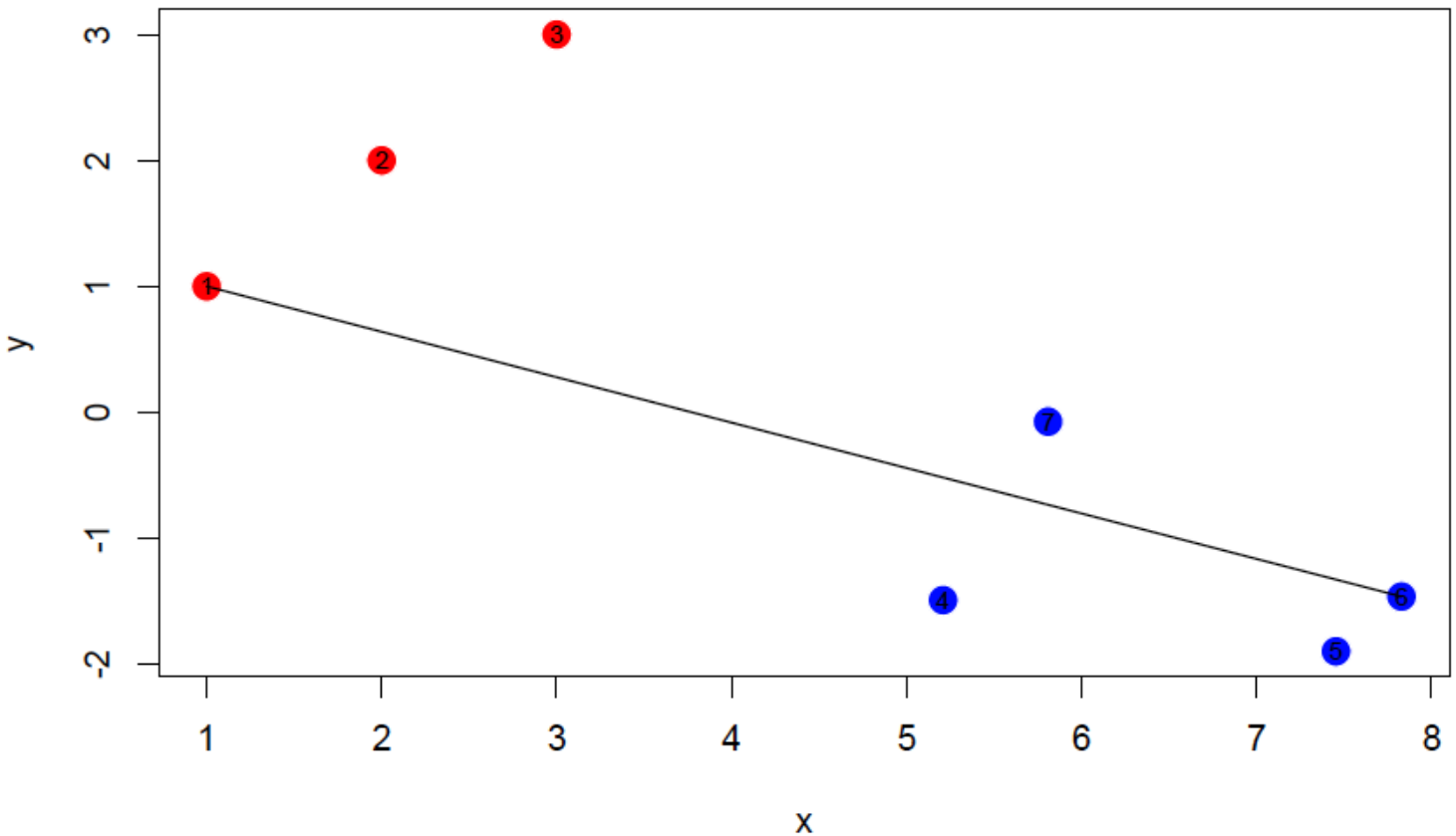
$$d_{AL} = \frac{n_{G_1}}{n_{G_1} + n_{G_2}} d(G_1, J) + \frac{n_{G_2}}{n_{G_1} + n_{G_2}} d(G_1, J)$$

where n_{G_i} is the number of units composing the group G_i.

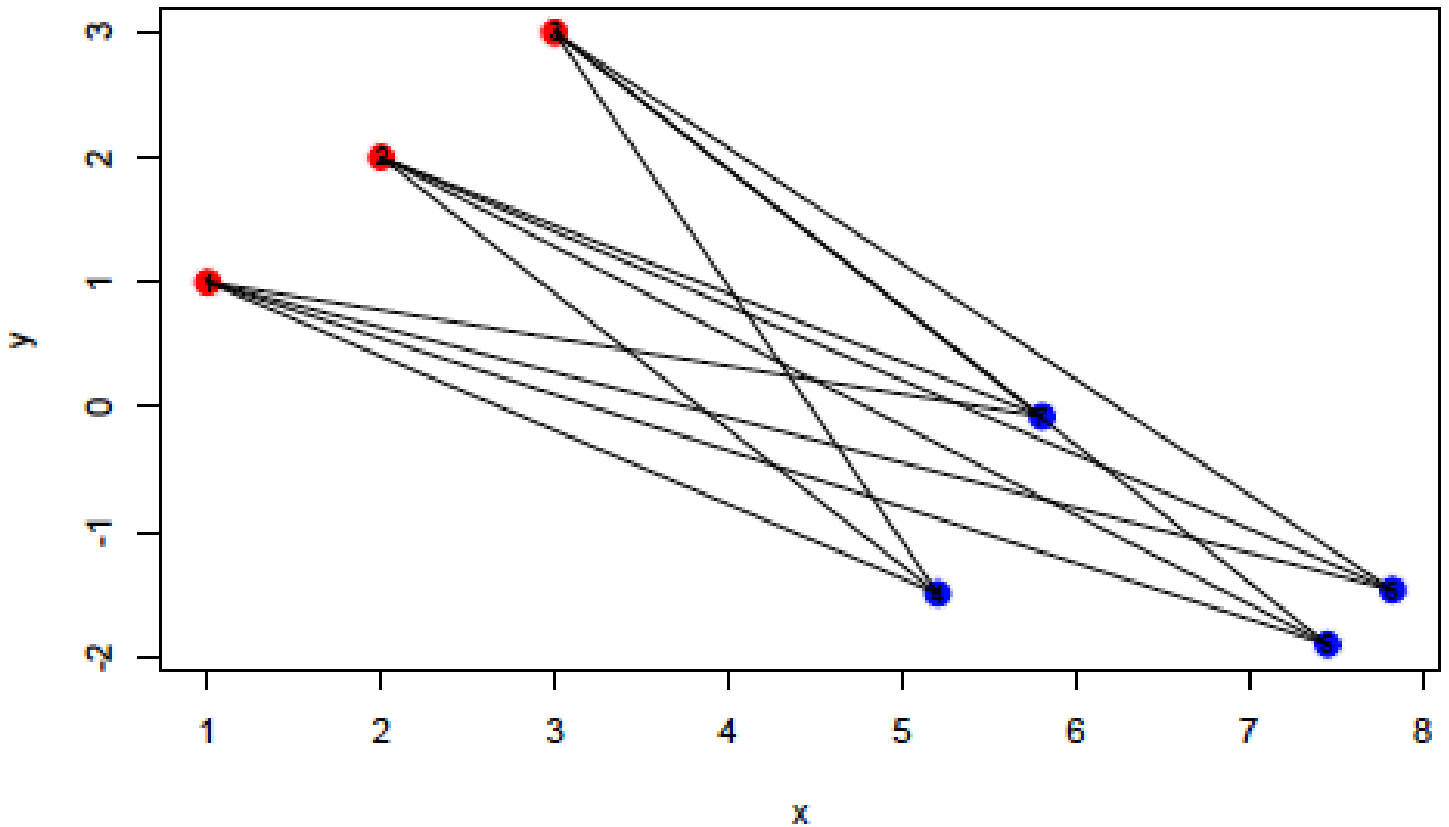
Single-Linkage or nearest neighbor



Complete-Linkage



Average-Linkage or Mean distance



How to choose the number of groups

- Graphic inspection of the dendrogram
- Some tests based on formal criteria: Calinski and Harabasz
- Silhouette
- Gap statistics

Silhouette

The silhouette is an index which takes values in the interval $[-1, 1]$.

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

where $a(i)$ is the average dissimilarity in the cluster considered, $b(i)$ is the average dissimilarity w.r.t. any other clusters. If:

- $s \approx 1 \rightarrow$ the unit i is assigned to the right cluster
- $s \approx 0 \rightarrow$ the unit i should not be assigned to either the compared clusters
- $s \approx -1 \rightarrow$ the unit i is assigned to the wrong cluster

Choice of the number of clusters

The value of $s(i)$ depends on the partition π and so on the chosen number of clusters. We should adopt a notation of the type $s_\pi(i)$. We use as criteria for the choice of the number of groups the one based on the mean of the silhouette, therefore the number K which maximizes

$$\bar{S}_K = \frac{1}{n} \sum_{i=1}^{n} s_K(i).$$

Diagnostics

- Internal validation: uses data only and is based on the quantitative measures of distance used and / or on the definition of reference structures for the presence / absence of groupings.
  - Average silhouette $(\bar{S})$
    - $\bar{S} \in (0.7, 1.00]$ the partition obtained is extremely reliable
    - $\bar{S} \in (0.5, 0.7)$ the partition obtained is reliable
    - $\bar{S} \in (0.25, 0.5)$ the partition obtained is not very reliable
    - $\bar{S} \in [-1, 0.25)$ the partition obtained is not reliable
  - Between deviance ratio on total deviance
- External validation: uses a priori information or structures of existing classification.
  - Rand Index
  - Adjusted Rand Index

The Iris dataset

```
> library(datasets)
> data(iris)
> summary(iris)
```

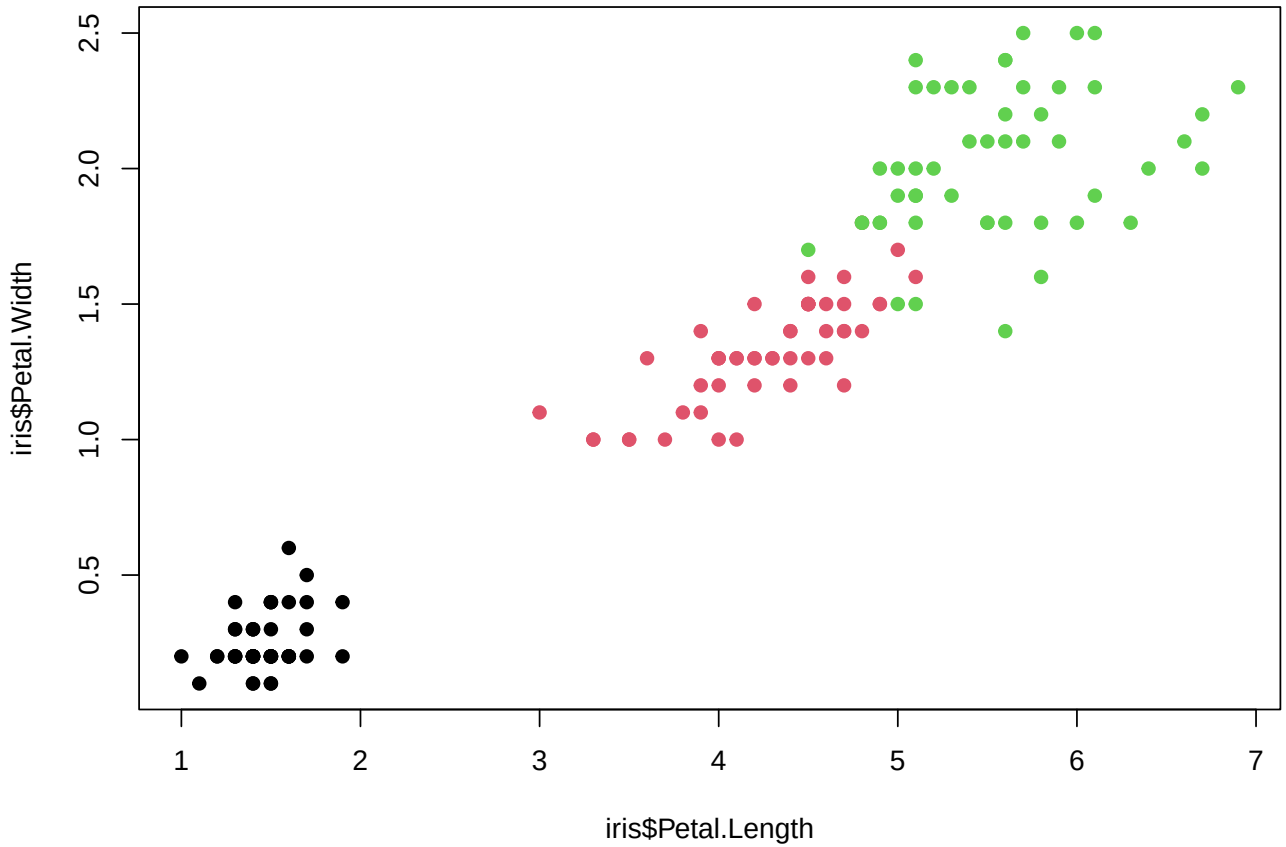
```
Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

```
> head(iris)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

The Iris dataset

```
> plot(iris$Petal.Length, iris$Petal.Width, col = iris$Species, pch = 19)
```

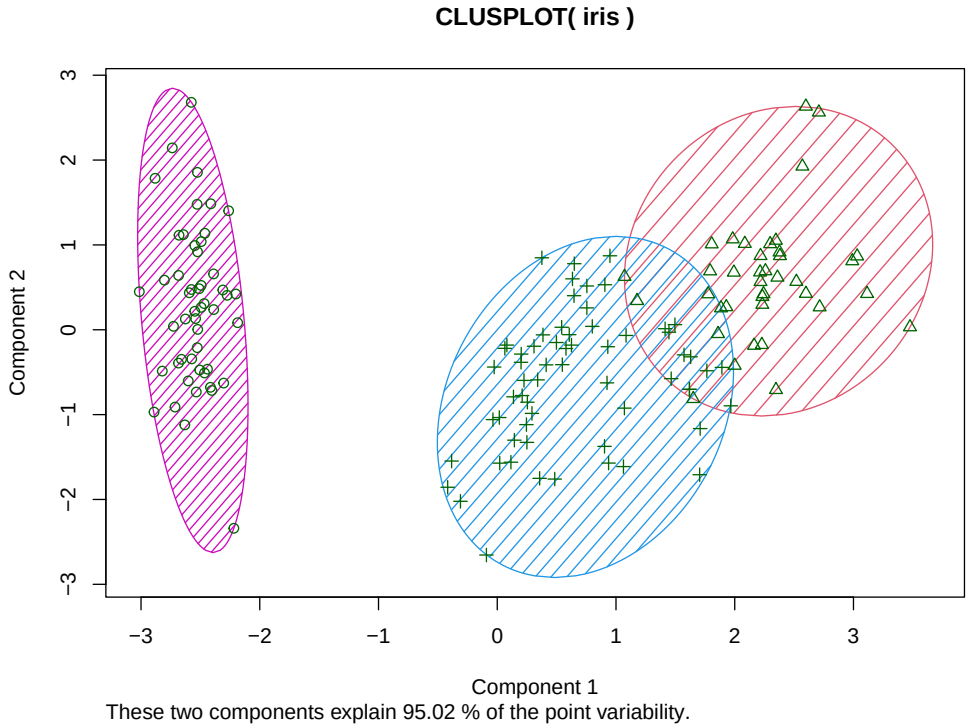


The Iris dataset - K-means

```
> irisKmeans <- kmeans(iris[,1:4], center=3, nstart=20)
> irisKmeans
K-means clustering with 3 clusters of sizes 50, 38, 62

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.006000    3.428000     1.462000    0.246000
2     6.850000    3.073684     5.742105    2.071053
3     5.901613    2.748387     4.393548    1.433871

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [39] 1 1 1 1 1 1 1 1 1 1 1 1 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [77] 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 2 2 2 2 3 2 2 2 2 2 2 3
[115] 3 2 2 2 2 3 2 3 2 3 2 2 3 3 2 2 2 2 2 3 2 2 2 2 3 2 2 2 3 2 2 2 3 2 2 3

Within cluster sum of squares by cluster:
[1] 15.15100 23.87947 39.82097
 (between_SS / total_SS =  88.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

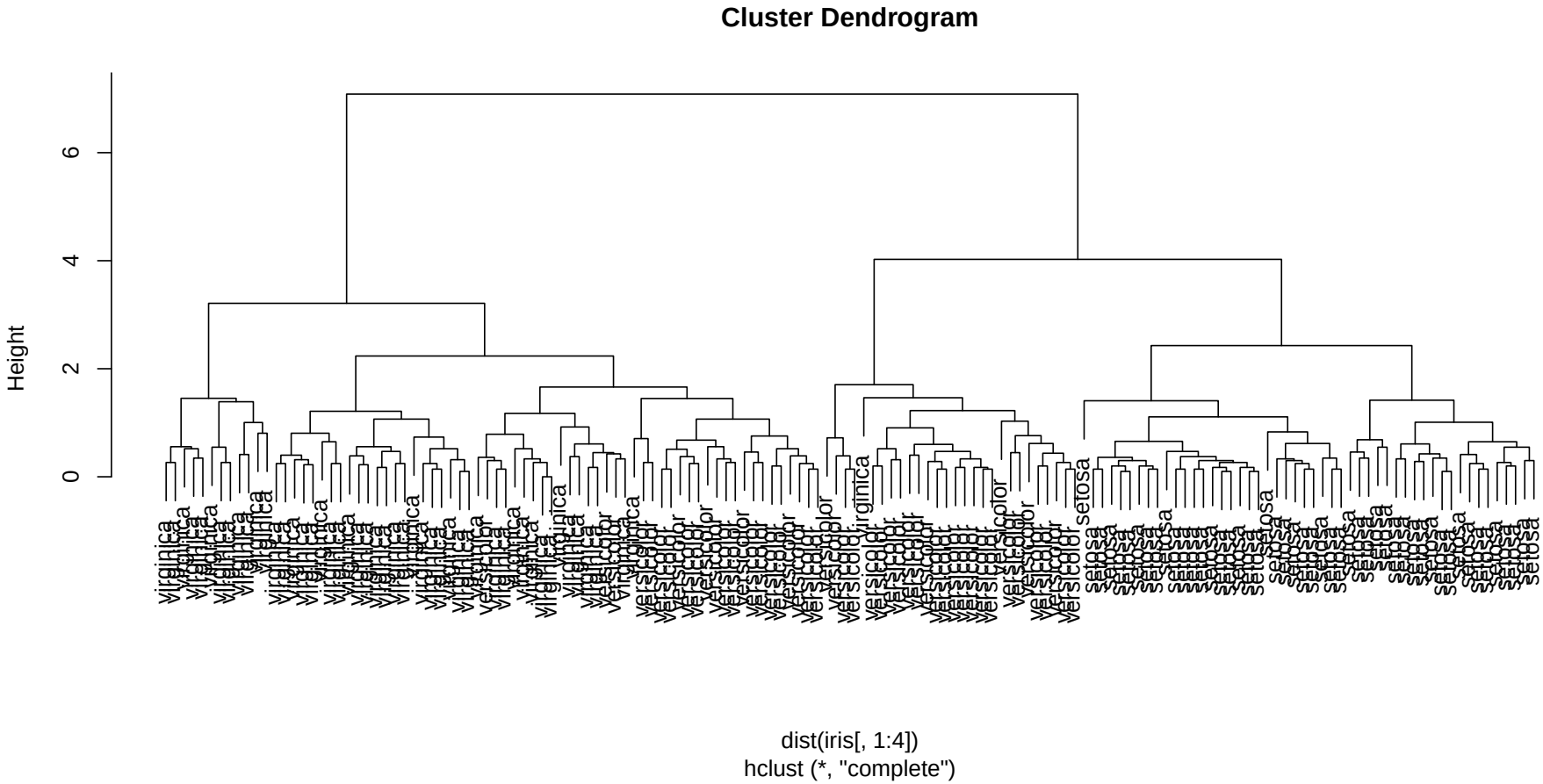
The Iris dataset - K-means

```
> table(irisKmeans$cluster, iris$Species)
   setosa versicolor virginica
1      50          0         0
2       0          2        36
3       0         48        14

> library(cluster)
> clusplot(iris, irisKmeans$cluster, color=T, shade=T, labels=0, lines=0)
```



The Iris dataset - Hierarchical clustering

```
> d <- dist(iris[,1:4], method = "euclidean")
> irisHclus <- hclust(d, method = "complete")
> plot(irisHclus, labels = iris$Species)
```



The Iris dataset - Hierarchical clustering

```
> groups <- cutree(irisHclus, k = 3)
> table(groups, iris$Species)
```

```
groups setosa versicolor virginica
     1     50          0         0
     2      0         23        49
     3      0         27         1
```