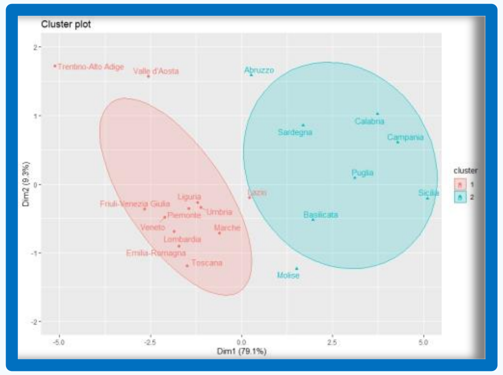
Clustering and Classification

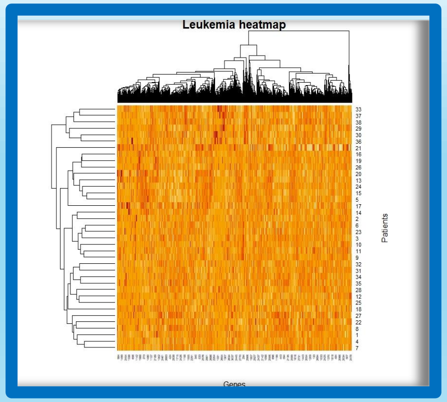
Dr. Annamaria Porreca

University of «G.D'Annunzio»,

Chieti-Pescara, Italy

annamaria.porreca@unich.it





Outline

1. Classification in real life
2. Supervised vs unsupervised classification
3. Clustering
4. Dissimilarity measures
5. Hierarchical clustering
   - Agglomerative hierarchical clustering
   - Divisive hierarchical clustering
6. Selecting the number of groups
7. Pre-processing operations for clustering
8. Non-hierarchical clustering
   - K-Means
9. Fuzzy set theory
10. Fuzzy clustering

Classification in real life

We as humans have an innate tendency to **classify everything.**

Given a set of individuals, we have an innate need in our mind to classify all units into **groups.**

It doesn't matter if there are differences between the individuals within a group, but we put them in the same group, if these **differences are negligible** (in our mind).

We are children of **Aristotelian logic**, that is, **bivalent logic**, in which **everything is black or white**. Thus, **<u>individual MUST be in one group or not.</u>**

Our brains are programmed to classify since we were born!

POISONOUS FOOD

GOOD FOOD

Our brains are programmed to classify since we were born!

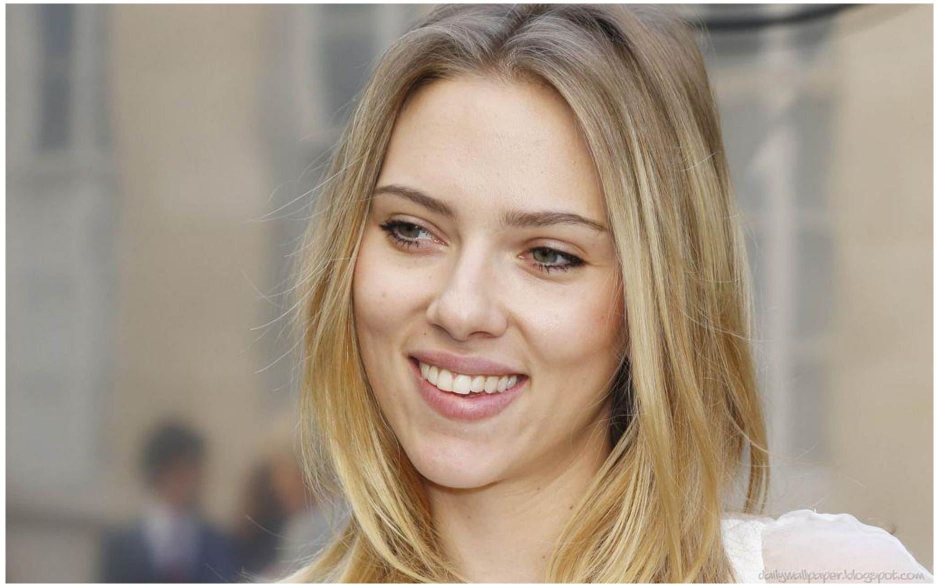
ANIMAL TO AVOID

ANIMAL TO HUNT

…and we continue to classify in our everyday life!

BEAUTIFUL WOMAN

VERY UGLY WOMAN



Classification

So, **classification is not a statistical thing but it is an absolutely natural stuff.**

Thus, which is the role of statistics?

What statistics does, is to **help us to classify, giving us analytical tools.**

When we talk about **supervised and unsupervised classification**, we talk about these tools.

Some examples of classification problems

Email Spam: to predict whether an email is a spam and should be delivered to the Junk folder.

Handwritten Digit Recognition

Image segmentation

Speech Recognition

DNA Expression Microarray: to identify disease or tissue types based on the gene expression levels.

DNA Sequence Classification: to find DNA segments which are non-coding, i.e. they do not have any biological function.

Supervised classification

Sometimes we know the number of groups because they are an objective fact and we have the availability of a training set with predictors and the group variable.

In these cases, we talk about “Supervised classification”.

SICK PERSON

HEALTHY PERSON

TRAINING SET

	GROUP	Predictor X1	Predictor X2
Little man	SICK	…	…
Arnold	HEALTHY	…	…
….	HEALTHY	…	…
….	SICK	…	…

Supervised classification

Some of the most used approaches are:

- Logistic Regression
- K-Nearest Neighbors
- Linear Discriminant Analysis
- Bayesian approach
- Classification Trees
- Bagging and Random Forest

Unsupervised classification (Clustering)

Sometimes we do NOT know a-priori the number of groups and we have NOT a training set with the group variable.

In these cases, we talk about “Un-supervised classification” or “Clustering”.

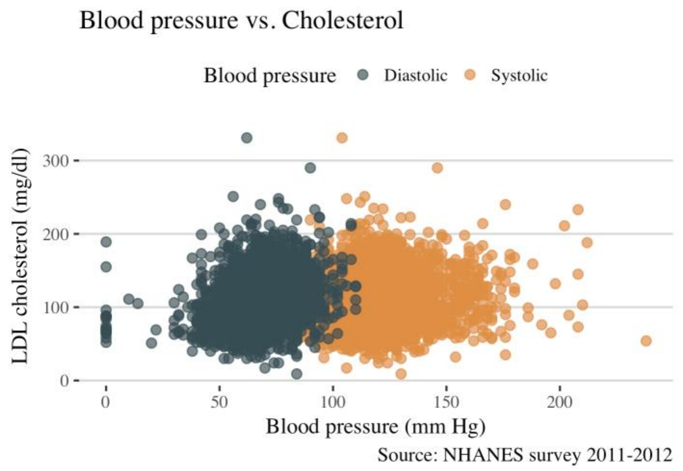
	GROUP	Predictor X1	Predictor X2
Ettore	?	…	…
Simone	?	…	…
Angelo	?	…	…
Giuseppe	?	…	…

- We do not know the number of groups
- We do not have the data about the group
- Often, we do not know what the group is about
- We can only have some a-priori ideas

We can only have some a-priori ideas to fix the number of groups: how?

1) A-priori knowledge of a phenomenon.

2) Visualisation is helpful to see any structure that might be present: we plot to see graphically if the data are grouped



Note: more difficult when there are many variables.

Figure from: https://datascienceplus.com/ggplot2-features-for-visualizing-the-nhanes-data/

Unsupervised classification (Clustering)

In summary, clustering is a methodology for **creating groups of statistical units that are homogeneous within them and the most diverse among them.**

The key concept of clustering is that of the "**dissimilarity"** between statistical units because we want similar units to be in the same group and dissimilar ones to different groups.

We reason in terms of **dissimilarity matrices between the statistical units**, and to calculate it, we use the observed values for all the variables of these statistical units.

The **quality of a clustering** result depends on the algorithm, the distance function, and the application.

Types of dissimilarity

The choice of a **dissimilarity metric** should be made based on **theoretical concerns** from the domain of study. That is, a distance metric needs to define **similarity in a way that is sensible for the field** of study.

The definitions of a **dissimilarity functions** are usually **very different** for boolean, categorical,and numerical variables.

It is hard to define "similar enough" or "good enough", the answer is typically **highly subjective**.

Euclidean Vs. Non-Euclidean

Two major classes of distance measure:

1. **Euclidean distance**: A Euclidean space has some number of real-valued dimensions and “dense” points. There is a notion of “average” of two points.

A Euclidean distance is based on the locations of points in such a space.

2. **Non-Euclidean**: it is based on properties of points, but not their “location” in a space; es. Jaccard distance, Cosine distance, etc.

Common types of data

- Binary variables.
- Nominal (Categorical) variables.
- Numerical variables,
- Mixed types of variables.

Symmetric versus Asymmetric Nominal Variables (a brief clarification)

A binary variable contains two possible outcomes: 1 (positive/present) or 0 (negative/absent).

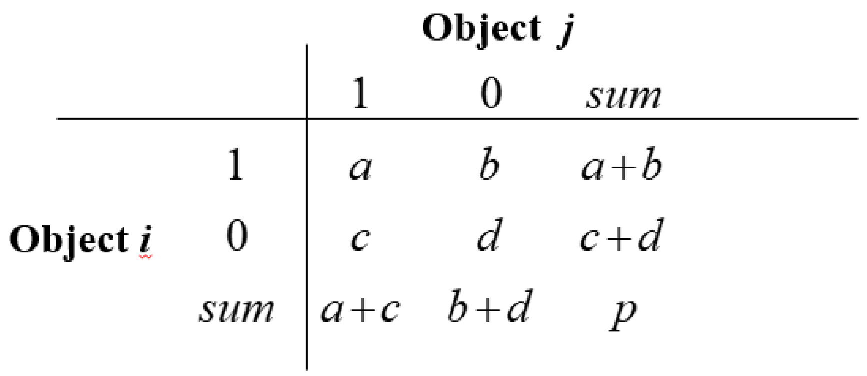
If there is no preference for which outcome should be coded as 0 and which as 1, the binary variable is called symmetric. For example, the binary variable "is evergreen?" for a plant has the possible states "loses leaves in winter" and "does not lose leaves in winter." Both are equally valuable and carry the same weight when a proximity measure is computed. Commonly used measures that accept symmetric binary variables include the Simple Matching, Hamann, Roger and Tanimoto, Sokal and Sneath 1, and Sokal and Sneath 3 coefficients.

If the outcomes of a binary variable are not equally important, the binary variable is called asymmetric. An example of such a variable is the presence or absence of a relatively rare attribute, such as "is color-blind" for a human being. While you say that two people who are color-blind have something in common, you cannot say that people who are not color-blind have something in common. The most important outcome is usually coded as 1 (present) and the other is coded as 0 (absent). The agreement of two 1's (a present-present match or a positive match) is more significant than the agreement of two 0's (an absent-absent match or a negative match). Usually, the negative match is treated as irrelevant. Commonly used measures that accept asymmetric binary variables include Jaccard, Dice, Russell and Rao, Binary Lance and Williams nonmetric, and Kulcynski coefficients.

Taken from: https://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#statug_distance_sect003.htm

Dissimilarity in the case of Binary variables

If we are interested in the distance between i and j, we do a contingency table for binary data:



		Object j		
		1	0	sum
	1	a	b	$a+b$
Object i	0	c	d	$c+d$
	sum	$a+c$	$b+d$	p

Simple matching coefficient (invariant, if the binary variable is symmetric): $$d(i,j) = \frac{b+c}{a+b+c+d}$$

Jaccard coefficient (noninvariant if the binary variable is asymmetric): $$d(i,j) = \frac{b+c}{a+b+c}$$

Dissimilarity in the case of Nominal (categorical) variables

Method 1: **Simple matching**

$$d(i,j) = \frac{p-m}{p}$$

Where m is the # of matches, and p is the total # of variables.

Method 2: Use **a large number of binary variables**

Creating a new binary variable for each of the M nominal states.

Dissimilarity in the case of numerical variables

Distances are normally used to measure the similarity or dissimilarity between two data objects:

The most important are:

- **Minkowski distance**: $$d(i,j) = \sqrt[q]{(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \ldots + |x_{ip} - x_{jp}|^q)}$$

  where $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two p-dimensional data objects, and q is a positive integer.

- **Manhattan distance**: $$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \ldots + |x_{ip} - x_{jp}|$$

  When **q = 1**, Minkowshi distance is same as Manhattan distance.

- **Euclidean distance**: $$d(i,j) = \sqrt{(|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \ldots + |x_{ip} - x_{jp}|^2)}$$

  When **q = 2**, Minkowshi distance is same as Euclidean distance.

Note: We can also use weighted distance, parametric Pearson product moment correlation, or **other dissimilarity measures.**

The meaning of the Euclidean distance

For example, take two cities, say, Baltimore and Washington D.C., and put them on a map. If you imagine that the center of each city has an X and a Y coordinate (say, longitude and latitude), and you want to map the distance between the centers of the two cities, then you can draw a straight diagonal line between the two cities.

In the two-dimensional plane, you take the distance in the x coordinates, square it, take the difference in the y coordinates, square that, and then add the two squares together and take the square root of the whole thing.

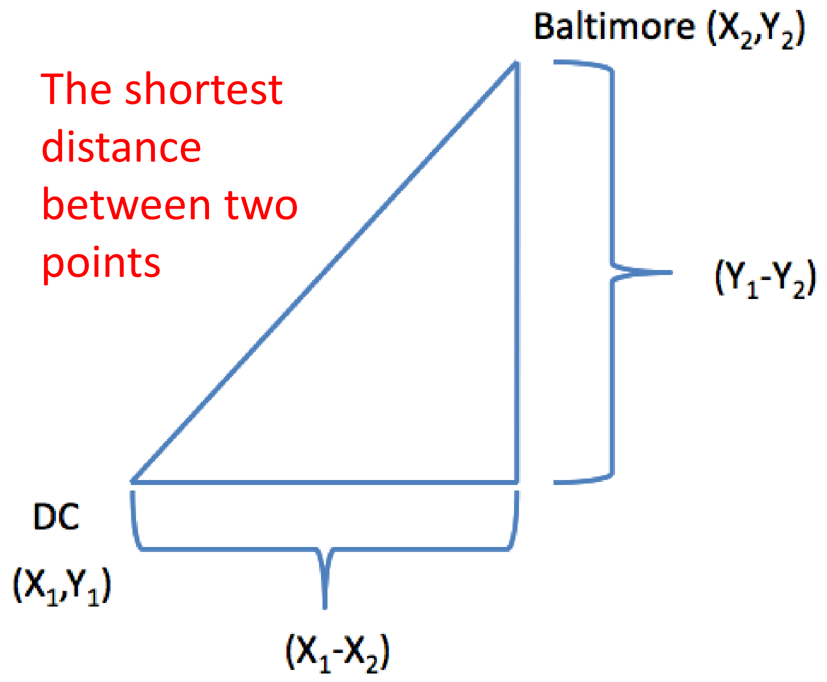


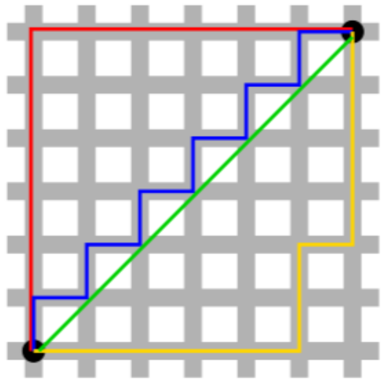
Image from: https://bookdown.org/rdpeng/exdata/hierarchical-clustering.html

The meaning of the Manhattan distance

It gets its name from the idea that you can look at points as being on a grid or lattice, not unlike the grid making up the streets of Manhattan in New York City.

In a city, if you want to go from point A to point B, you usually cannot take the direct route there because there will be buildings in the way. So instead, you have to follow the streets, or the grid layout, of the city to navigate around.

The red, blue, and yellow lines show various way of getting between the two black circles using the grid layout, while the green line shows the Euclidean distance. The Manhattan distance between the points is simply the sum of the right-left moves plus the sum of all the up-down moves on the grid.



Quoting from the paper, "On the Surprising Behavior of Distance Metrics in High Dimensional Space", by Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Kiem. " for a given problem with a fixed (high) value of the dimensionality d, it may be preferable to use lower values of p. This means that the L1 distance metric (Manhattan Distance metric) is the most preferable for high dimensional applications.

Image from: https://bookdown.org/rdpeng/exdata/hierarchical-clustering.html

Other distance functions

Other distance functions that can be used in Bioinformatics are:

- Standardized Euclidean Distance.
- Chebychev Distance.
- Cosine Similarity.
- Correlation Distance.
- Mahalanobis distance.

Standardized Euclidean Distance

The basic idea is that not all the predictors are necessarily the same.

The SED between two n-dimensional vectors x=(x_1,x_2,…,x_n) and y=(y_1,y_2,…,y_n) is:

$$d_{SE}(x,y) = \sqrt{\frac{1}{s_1^2}(x_1 - y_1)^2 + \ldots + \frac{1}{s_n^2}(x_n - y_n)^2} = \sqrt{\sum_{i=1}^{n} \frac{1}{s_i^2}(x_i - y_i)^2}$$

Where s^2_i is the sample variance over the i-th dimension in the input space.

Thus, we are considering the effect of the variability of each predictor (e.g. expression of the gene) weighting each dimension by a quantity inversely proportional to the amount of variability along that dimension. It eliminates the influence of different range of variation.

Chebychev Distance

CD picks the largest difference between any two corresponding coordinates.

Is to be used when the goal is to reflect any big difference between any corresponding coordinates.

The CD between two n-dimensional vectors $x=(x_1,x_2,\dots,x_n)$ and $y=(y_1,y_2,\dots,y_n)$ is:

$$d_{\max}(x,y) = \max_i |x_i - y_i|$$

CD is <u>very sensitive to outliers.</u>

Pearson Correlation Distance

The PCD tells us how far the point are from the regression line.

The PCD between two *n*-dimensional vectors x=(x_1,x_2,…,x_n) and y=(y_1,y_2,…,y_n) is:

$$d_R(x,y) = 1 - r_{xy}$$

where $r_{x,y}$ is the Pearson Correlation Coefficient. Thus, the PCD varies between 0 and 2.

It looks for similar variation as opposed to similar numerical values.

Example: a gene g1 that has an expression of g1=(1,2,3,4,5) in the 5 experiments.

A gene g2 that has an expression of g2=(100,200,300,400,500) in the 5 experiments.

g1 is in the same cluster of g2 even if they have very different values.

Mahalanobis distance

MD distance between two *n*-dimensional vectors $x=(x_1, x_2, \dots, x_n)$ and $y=(y_1, y_2, \dots, y_n)$ is:

$$d_{Ml}(x,y) = \sqrt{(x-y)^T S^{-1} (x-y)}$$

where S is a *n* x *m* positive definite matrix which distorts the space as desidered.

Usually, this matrix is the covariance matrix of the data set.

It differs from the Euclidean distance in that it takes into account correlations within the dataset.

If the space warping matrix S is taken to be the identity matrix (S=I), the MD reduces to the classical Euclidean distance.

Dissimilarity in the case of mixed types of variables

A database may contain all the types of variables.

One may use a **<u>weighted formula</u>** to combine their effects.

<u>Out of the scope</u> of this talk.

Axioms of a distance measure

1. $d(i,j) \geq 0$	non-negativity
2. $d(i,i) = 0$	identity
3. $d(i,j) = d(j,i)$	symmetry
4. $d(i,j) \leq d(i,k) + d(k,j)$	subadditivity or triangle inequality, i.e. the sum of two sides of a triangle is at least as big as the third side

Data Structures

Data matrix

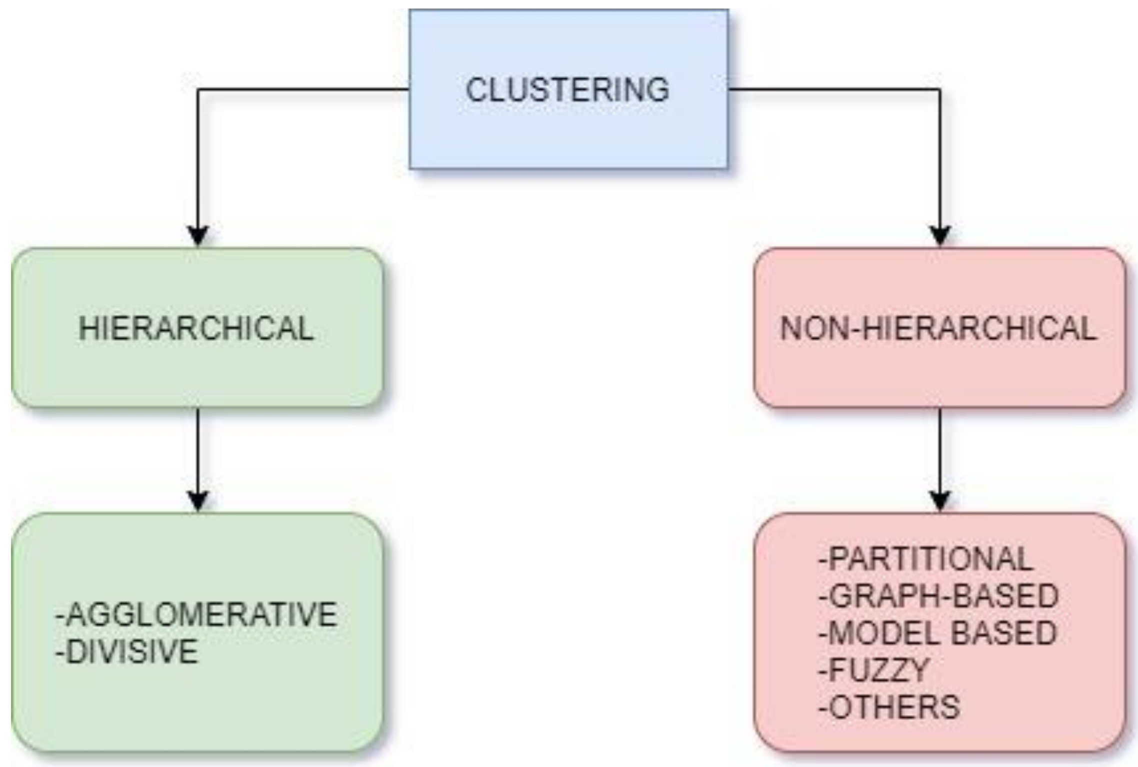
$$\begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix}$$

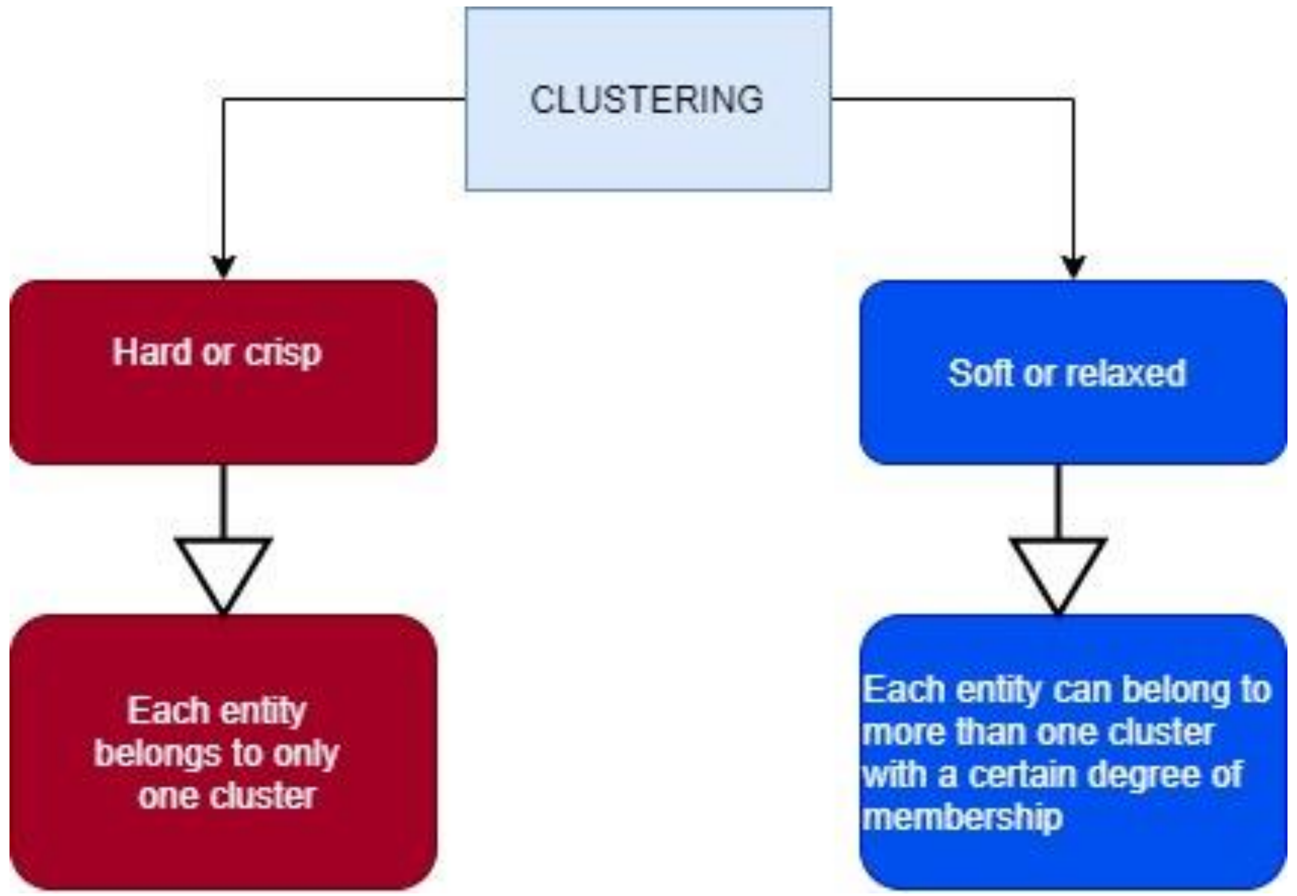
Dissimilarity matrix

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

Types of Clustering Algorithms

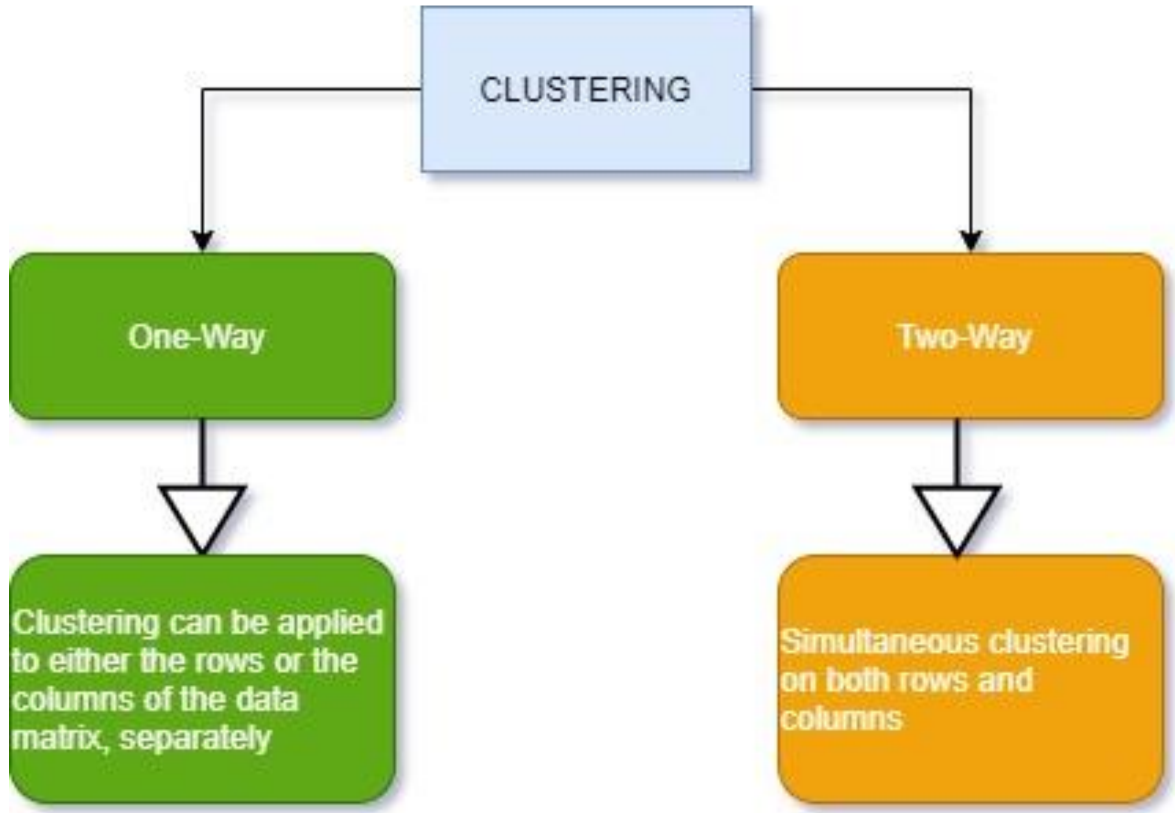
A more complete summary scheme of the taxonomy of clustering algorithms can be given by the following figure:




CLUSTERING
Hard or crisp
Each entity belongs to only one cluster
Soft or relaxed
Each entity can belong to more than one cluster with a certain degree of membership

Types of Clustering Algorithms

Types of Clustering Algorithms


CLUSTERING
One-Way
Two-Way
Clustering can be applied to either the rows or the columns of the data matrix, separately
Simultaneous clustering on both rows and columns

Hierarchical clustering

In hierarchical clustering there is a **precise order** in all the steps that have taken place, so that all the work of the algorithm can be reconstructed “hierarchically”, in the sense that it **identifies levels.**

In this type of clustering, a statistical unit, **once entered into a group, remains in that group** forever (agglomerative).

Unlike non-hierarchical clustering where statistical units can also change the group at each step.

There are **two types** of Hierarchical clustering:

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

Agglomerative hierarchical clustering

The Agglomerative Hierarchical Clustering is the most common type of hierarchical clustering used to group objects in clusters based on their similarity. It's also known as **AGNES** (Agglomerative Nesting).

It's a "**bottom-up**" approach: The algorithm **starts by treating each object as a singleton cluster**.

Next, **clusters are successively merged until all clusters have been merged into one big cluster** containing all objects.

The result is a tree-based representation of the objects, named ***dendrogram*** (https://www.kdnuggets.com/2019/09/hierarchical-clustering.html).

Steps:

1. Make each data point a single-point cluster → forms N clusters
2. Take the two closest data points and make them one cluster → forms N-1 clusters
3. Take the two closest clusters and make them one cluster → Forms N-2 clusters.
4. Repeat step-3 until you are left with only one cluster.

The choice of a Linkage Method

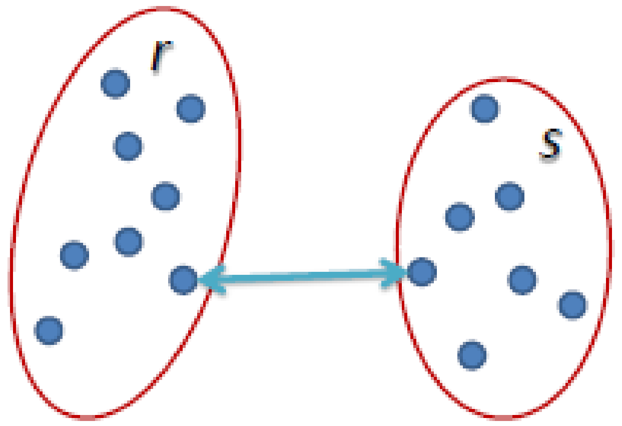
There are several ways to **measure the distance between clusters** in order to decide the rules for clustering, and they are often called Linkage Methods (https://www.saedsayad.com/clustering_hierarchical.htm).

Some of the common linkage methods are:

1. Single-linkage (or nearestneighbor): it favors elongated classes.
2. Complete-linkage: it favors compact classes.
3. Average-linkage: it is more robust to outliers.

Single-linkage

The distance between two clusters is defined as the shortest distance between two points in each cluster. This linkage may be used to detect high values in your dataset which may be outliers as they will be merged at the end.

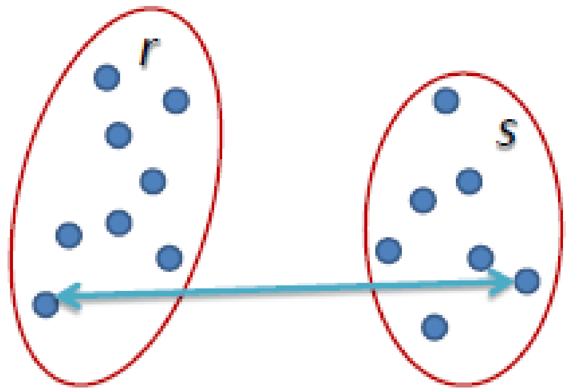


$$L(r,s) = \min(D(x_{ri}, x_{sj}))$$

Image from: https://www.saedsayad.com/clustering_hierarchical.htm

Complete-linkage

The distance between two clusters is defined as the longest distance between two points in each cluster.

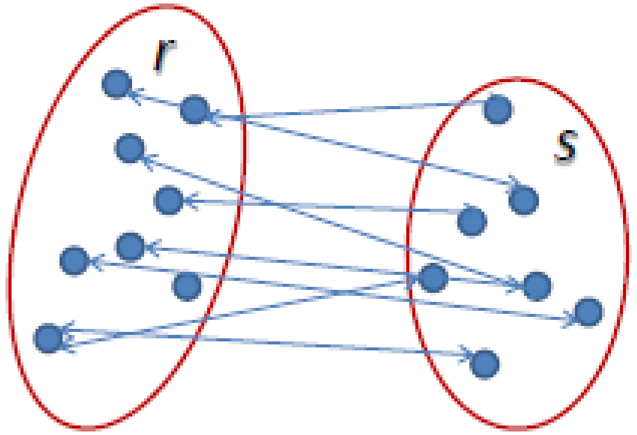


$$L(r,s) = \max(D(x_{ri}, x_{sj}))$$

Image from: https://www.saedsayad.com/clustering_hierarchical.htm

Average-linkage

The distance between two clusters is defined as the average distance between each point in one cluster to every point in the other cluster.



$$L(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$

Image from: https://www.saedsayad.com/clustering_hierarchical.htm

Dendrogram

A Dendrogram is a type of tree diagram showing hierarchical relationships between different sets of data.

It contains the memory of hierarchical clustering algorithm, so just by looking at the Dendrogram we can tell how the cluster is formed.

Distance between data points represents dissimilarities.

Height of the blocks represents the distance between clusters.

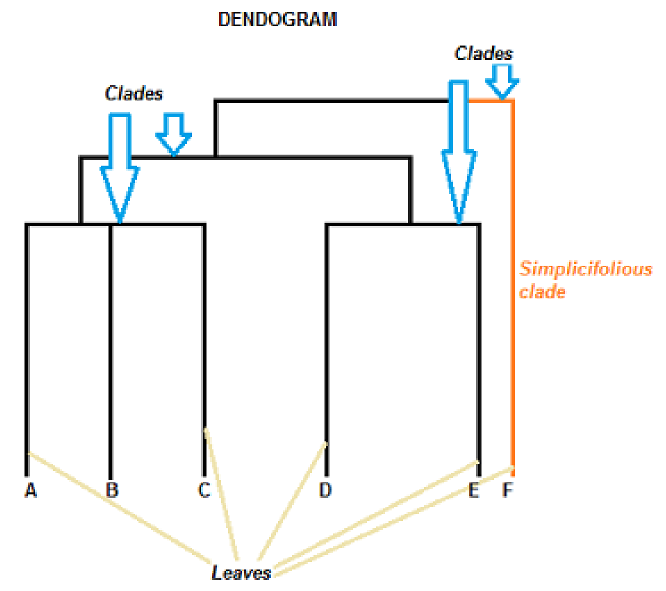
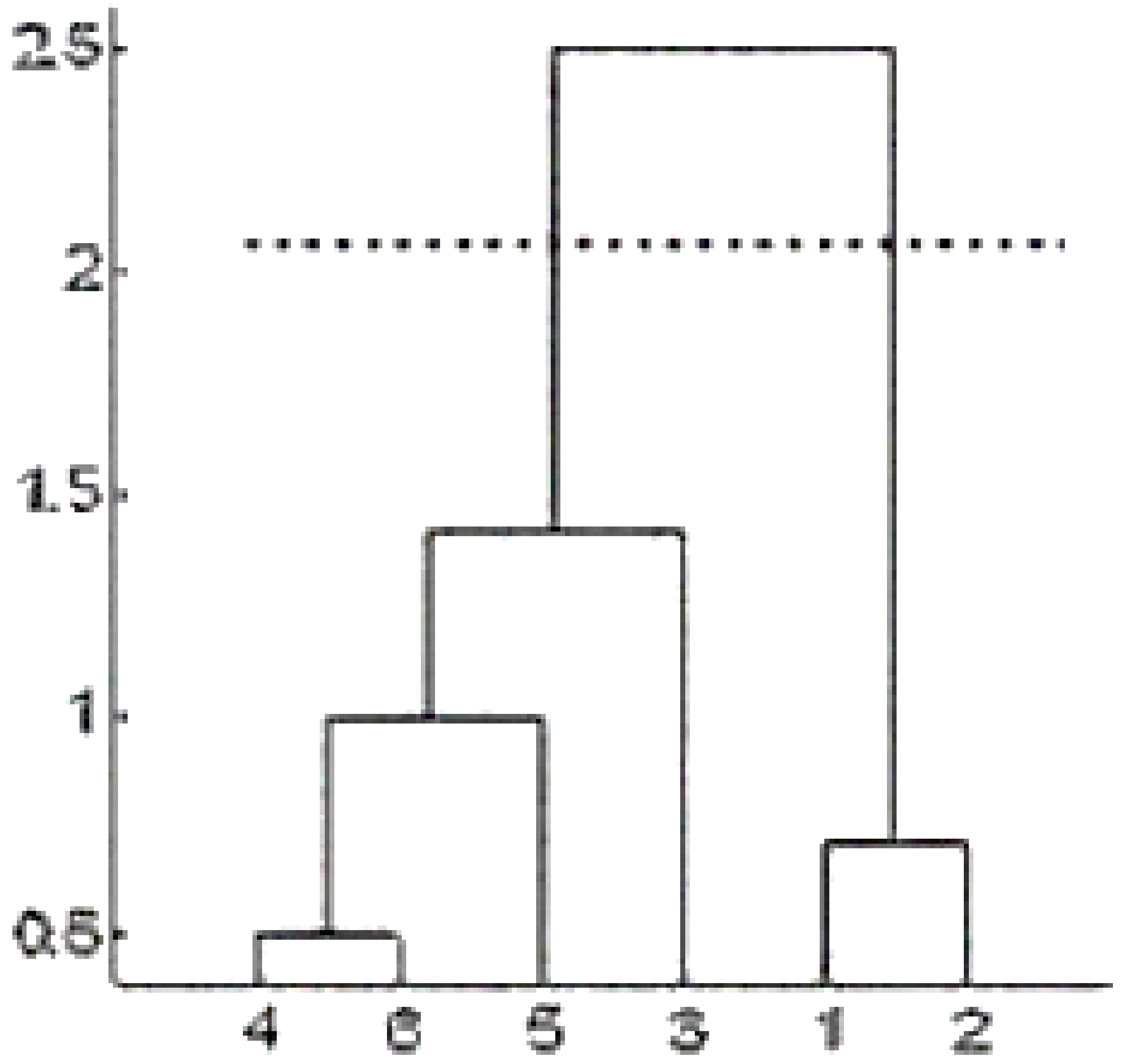


Image from: https://www.statisticshowto.datasciencecentral.com/hierarchical-clustering/

Choosing the optimal number of clusters

We cut the dendrogram tree with a horizontal line at a height where the line can traverse the maximum distance up and down without intersecting the merging point.



Divisive hierarchical clustering

The **divisive hierarchical clustering**, also known as *DIANA* (*DIvisive ANAlysis*) is the inverse of agglomerative clustering (https://www.datanovia.com/en/lessons/divisive-hierarchical-clustering/).

It's a ***"top-down"*** approach: It starts by including all objects in a single large cluster. At each step of iteration, the most heterogeneous cluster is divided into two. The process is iterated until all objects are in their own cluster.

There is evidence that divisive algorithms produce more **accurate hierarchies** than agglomerative algorithms in some circumstances but is **conceptually more complex**.

It is good at identifying large clusters while agglomerative clustering is good at identifying small clusters.

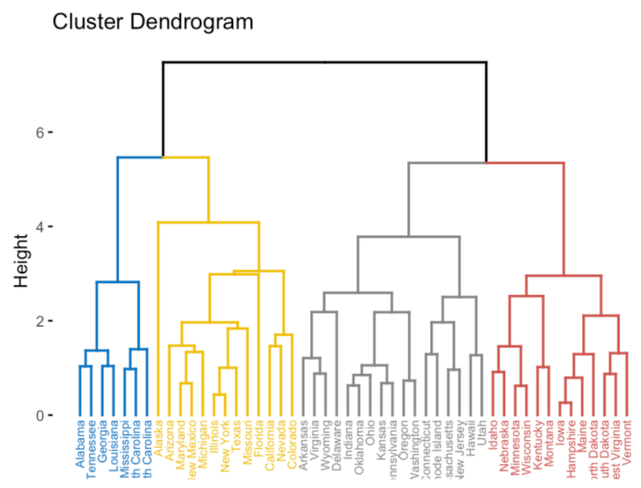


Image from: http://www.sthda.com/english/articles/28-hierarchical-clustering-

Pre-processing operations for clustering

There are two things we must do before starting to cluster datasets with many different variables:

1) **Scaling:** each observation's feature values are represented as coordinates in n-dimensional space (n is the number of features) and then the distances between these coordinates are calculated. If these coordinates are not normalized, then it may lead to false results.

Alternatives:

Min-max normalization

`standardize <- function(x){(x-min(x))/(max(x)-min(x))}` or the function `normalize()`

Standardisation

x(s)=x(i)−mean(x)/sd(x) or the function `scale()`

```
library(BBmisc)

?normalize

?scale

normalize(data)
```

Pre-processing operations for clustering

2) Missing Value imputation: if there are missing values, we have two alternatives:

-Missing values **imputation:**

?impute

-**Delete** observations with missing values:

?na.omit # look at the slides of the Statistical Learning course

any(is.na(data)) #check if any

Selecting the number of groups

We can often still apply a priori knowledge for potential groupings.

A more common case is that k is unknown.

Although hierarchical clustering provides a fully connected dendrogram representing the cluster relationships, we may still need to choose the preferred number of clusters to extract.

A commonly used rule of thumb is $k=\sqrt{n/2}$, where n is the number of observations to cluster. However, this rule can result in very large values of k for larger data sets.

There are multiple statistical methods but many of these measures suffer from the curse of dimensionality as they require multiple iterations and clustering large data sets is not efficient.

Possible Methods for selecting the # of groups

The most common techniques are:

- The “*Elbow*” method.
- The “*Silhouette*” method.

1-The Elbow method

The **total within-cluster sum of squares (WSS)** measures the compactness of the clustering and we want it to be as small as possible.

Thus, we can use the following approach to define the optimal # of clusters:

1. Compute clustering for different values of k. For instance, by varying k from 1–10 clusters.
2. For each k, calculate WSS.
3. Plot the curve of WSS according to the number of clusters k.
4. The location of a bend (i.e., elbow) in the plot is generally considered as an indicator of the appropriate number of clusters.

From: https://bradleyboehmke.github.io/HOML/kmeans.html#determine-k

2-The Average Silhouette method

The average silhouette determines how well each object lies within its cluster.

It computes the average silhouette of observations for different values of k.

The optimal number of clusters k is the one that maximizes the average silhouette over a range of possible values for k.

The silhouette for the unit i-th is computed as follows:

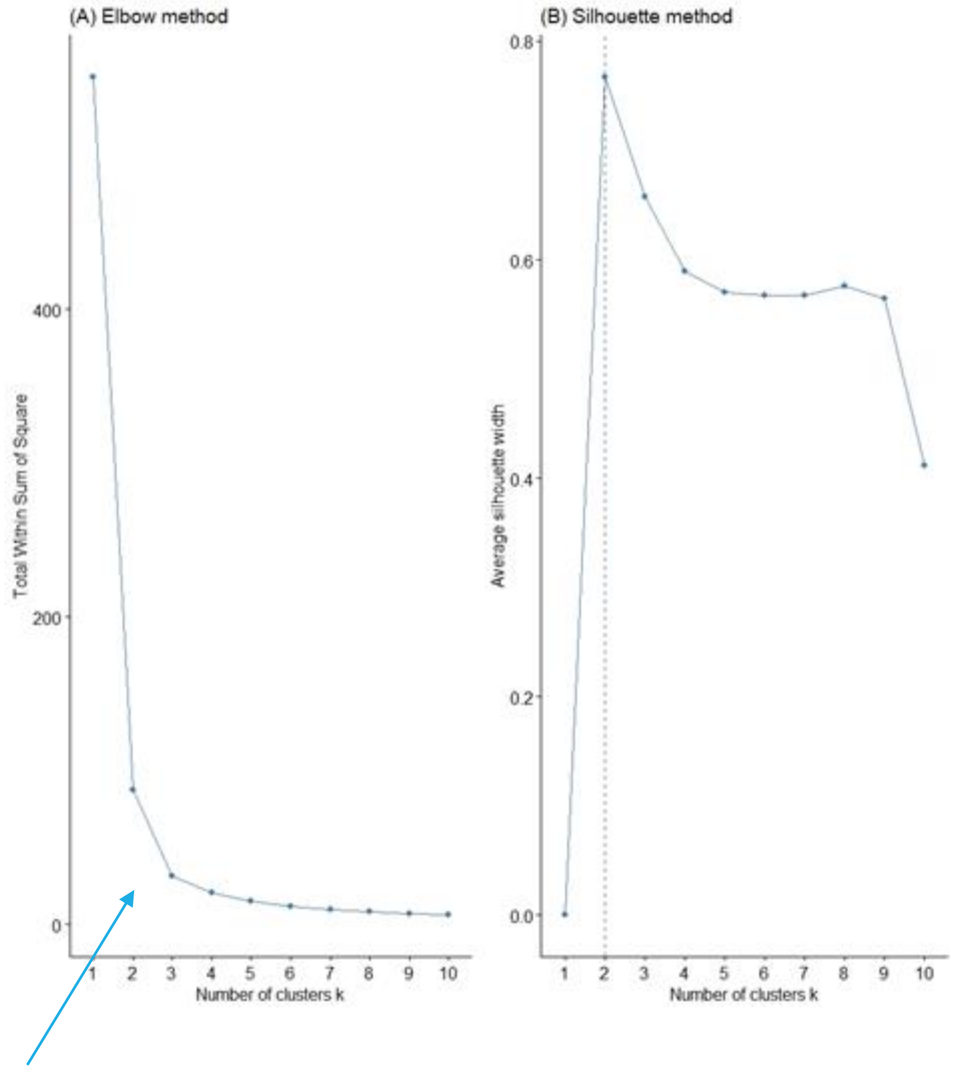
$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

Where:

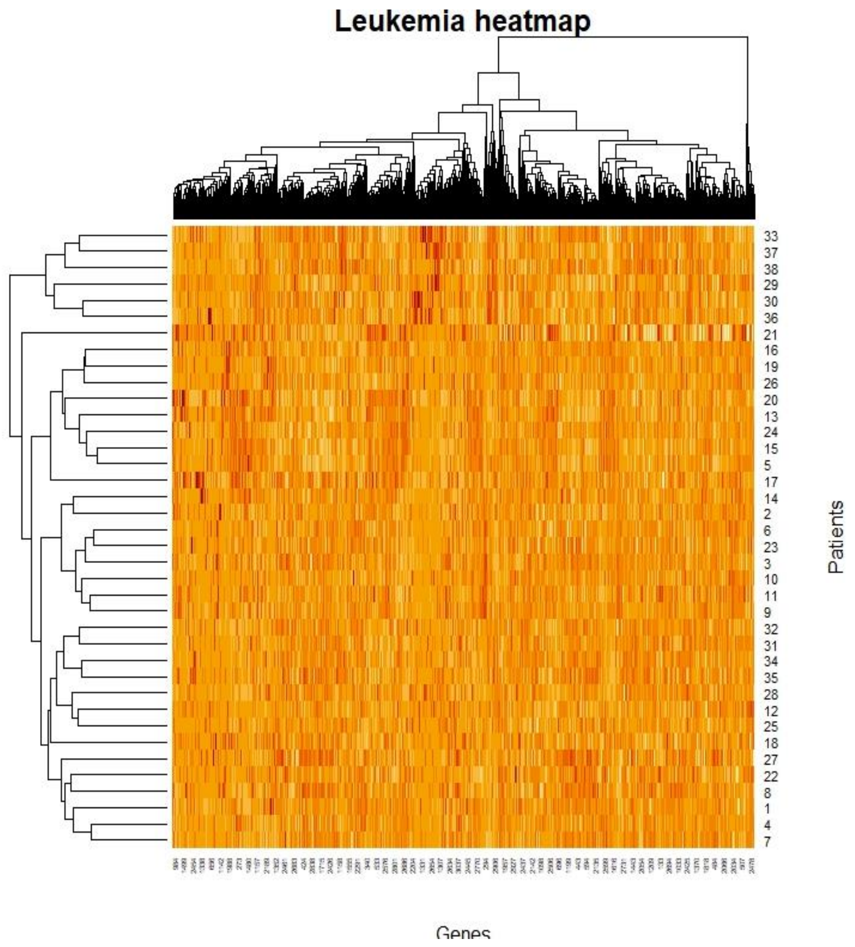
-a(i) is the mean distance (dissimilarity) of the i-th unit with respect to all the units belonging to the cluster; thus, we want, on average, low a(i);

-b(i) is the minimum distance (dissimilarity) between the i-th unit and the other non-cluster units; thus, we want, on average, high b(i).

Methods for selecting the # of groups in R



We tend to see that error decreases steadily as our K increases. However, we reach a point where the error stops decreasing.



```
?heatmap
heatmap(leukemia$X,scale = "column",
xlab = "Genes", ylab =  "Patients",
       main = "Leukemia heatmap")
```

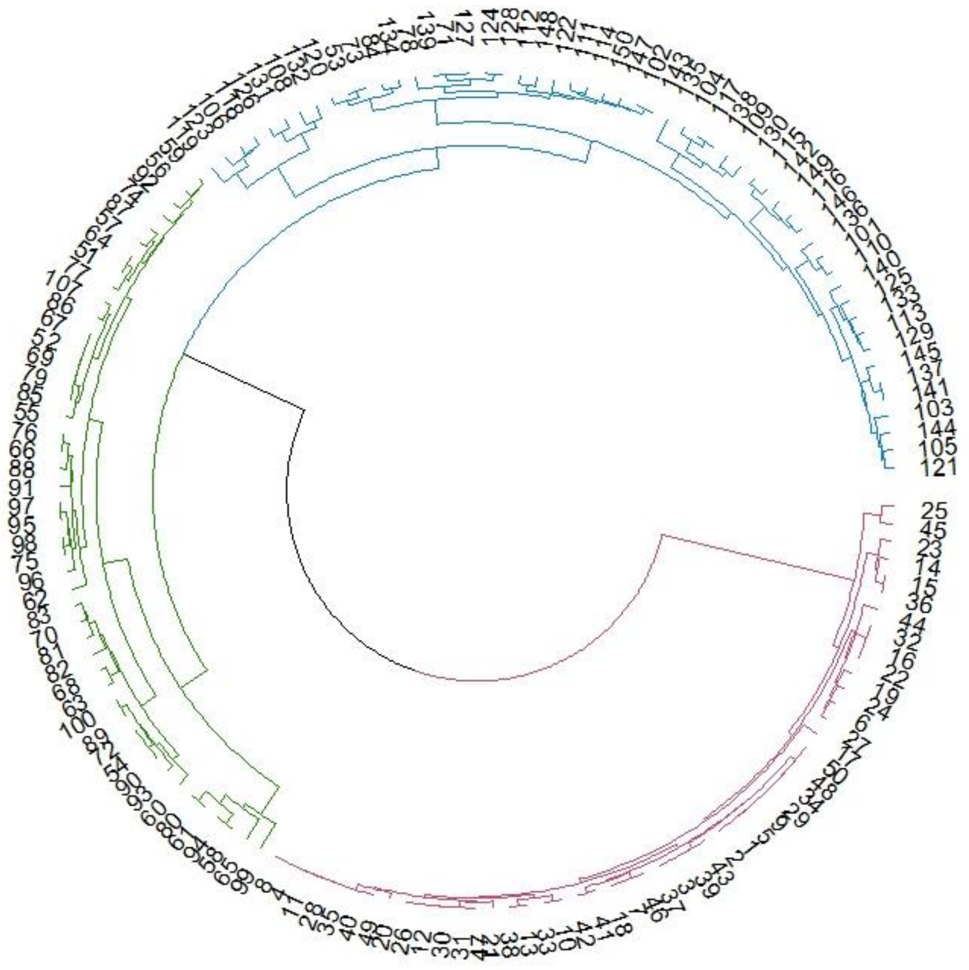
The heatmap() function

The heatmap() function is a helpful approach to visualize matrix data.

The idea is to sorts the rows and columns according to the clustering determined by a call to hclust().

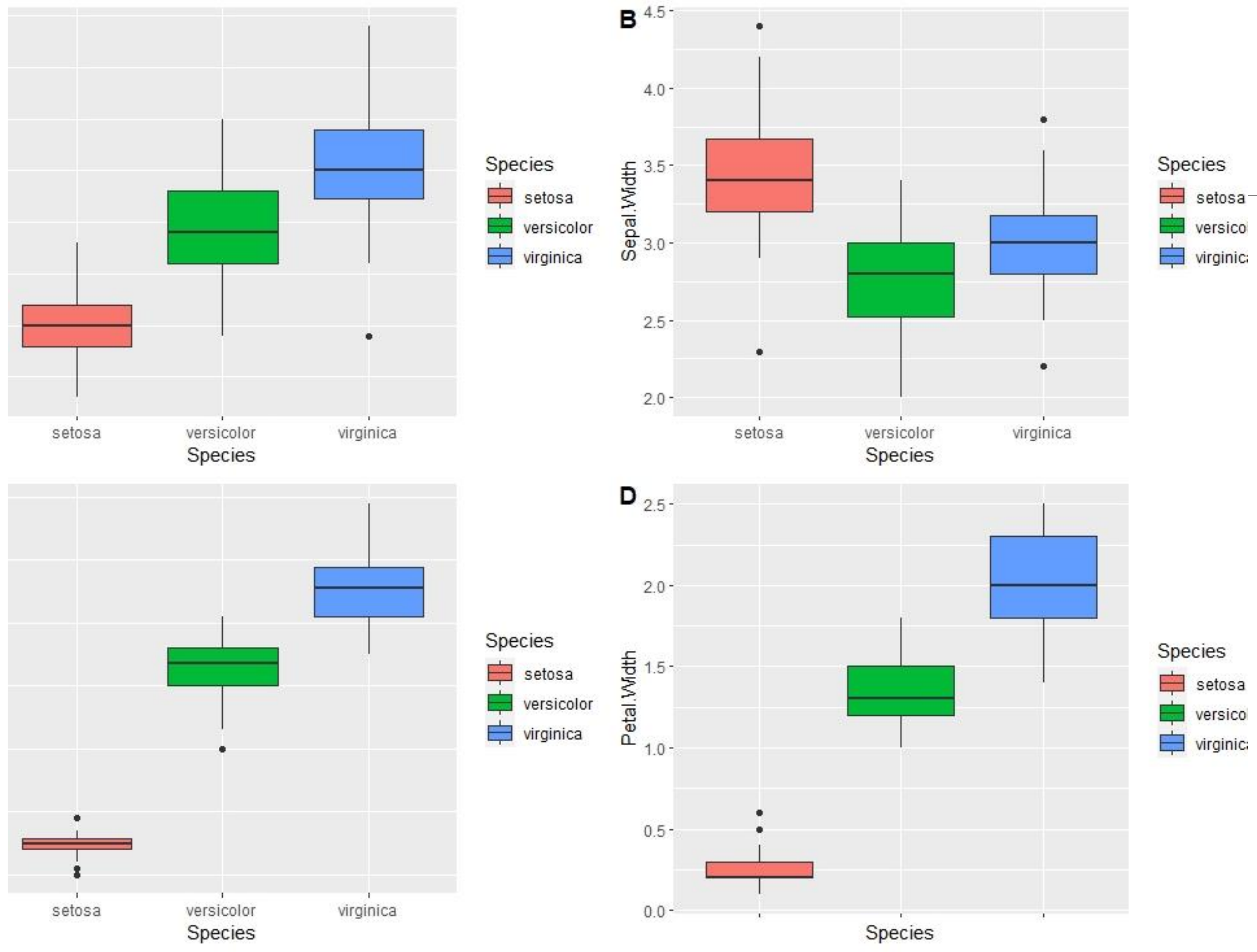
It first treats the rows of a matrix as observations and calls hclust() on them, then it treats the columns of a matrix as observations and calls hclust() on those values.

We get a dendrogram associated with both the rows and columns of a matrix, which can help you to recognize patterns in the data.



Visualization

```
library(dendextend)

library(circlize)

?as.dendrogram

dd=dist(data[, 3:4], method = "euclidean")

clusters = hclust(dd,method = "ave")

dend <- as.dendrogram(clusters)

dend <- color_branches(dend, k=3) #, groupLabels=iris_species)

circlize_dendrogram(dend)
```



Help for interpreting

```
p1 <- ggplot(iris, aes(x=Species, y=Sepal.Length,
fill=Species)) +
  geom_boxplot()

p2 <- ggplot(iris, aes(x=Species, y=Sepal.Width,
fill=Species)) +
  geom_boxplot()

p3 <- ggplot(iris, aes(x=Species, y=Petal.Length,
fill=Species)) +
  geom_boxplot()

p4 <- ggplot(iris, aes(x=Species, y=Petal.Width,
fill=Species)) +
  geom_boxplot()

install.packages("ggpubr")
library(ggpubr)

ggarrange(p1,p2,p3,p4 + rremove("x.text"),
        labels = c("A", "B", "C","D"),
        ncol = 2, nrow = 2)
```

Advantages and disadvantages of hierarchical clustering

Does not require the number of clusters to be known in advance.

No need of input parameters (besides the choice of the (dis)similarity).

Provides a complete hierarchy of clusters.

Nice visualizations.

No explicit clusters: a "flat" partition can be derived afterwards (e.g. via a cut through the dendrogram or termination condition in the construction).

No automatic discovering of "optimal clusters".

Non-hierarchical clustering

In the case of non-hierarchical methods, also called **Partitional clustering,** the algorithms aim to divide the n units into a predefined number of clusters, providing as a final result a unique partition in k groups based on the optimization of a specific objective function.

The procedure is iterative and starts indicating c initial centers around which to aggregate the units.

Differently from hierarchical clustering, at each step, units can be assigned to a different cluster if the initial allocation is inappropriate.

Therefore, we look for clusters that minimize or maximize an objective function, e.g. the sum of squares of distances to cluster center.

Non-hierarchical clustering

There are many partitional clustering algorithms available. The most used are:

K-means (MacQueen, 1967)

K-medoids or PAM method (Partition Around Medoids) (Kaufman & Rousseeuw, 1987)

Clustering Large Applications or CLARA method (Kaufmann and Rousseeuw in 1990)

Fuzzy C-means (Bezdek, 1981)

K-means

K-means is the most famous non-hierarchical clustering algorithm. It consists in an **iterative procedure** that can be summarized in the following phases:

1. Choice of the number of groups K and distance metric to be used.
2. Fix the coordinates of the K initial centroids. They are often chosen randomly but could be also existing points.
3. Compute the distance of each statistical unit from each centroid.
4. Allocate each statistical unit to the cluster whose centroid is closest. The centroid is the point that minimizes the sum of dissimilarities from the mean or the sum of the square errors from the mean, e.g. the mean of the coordinates of the points belonging to a group.
5. Because new groups have been created, new centroids must be computed.
6. Because the centroids have changed, and there (probably) aren't those fixed at step 2, we need to compute again the distance of all statistical units from the new centroids.
7. Repeat iteratively from step 3 to step 5 until the units do not change group.

K-means

In summary, K-means clustering aims to assign K clusters in such a way that maximises the separation of those clusters while minimising intra-cluster distances relative to the cluster's mean or centroid.

The algorithm typically defaults to Euclidean distances, however, alternate criteria, such as different distance or dissimilarity measures, can be implemented.

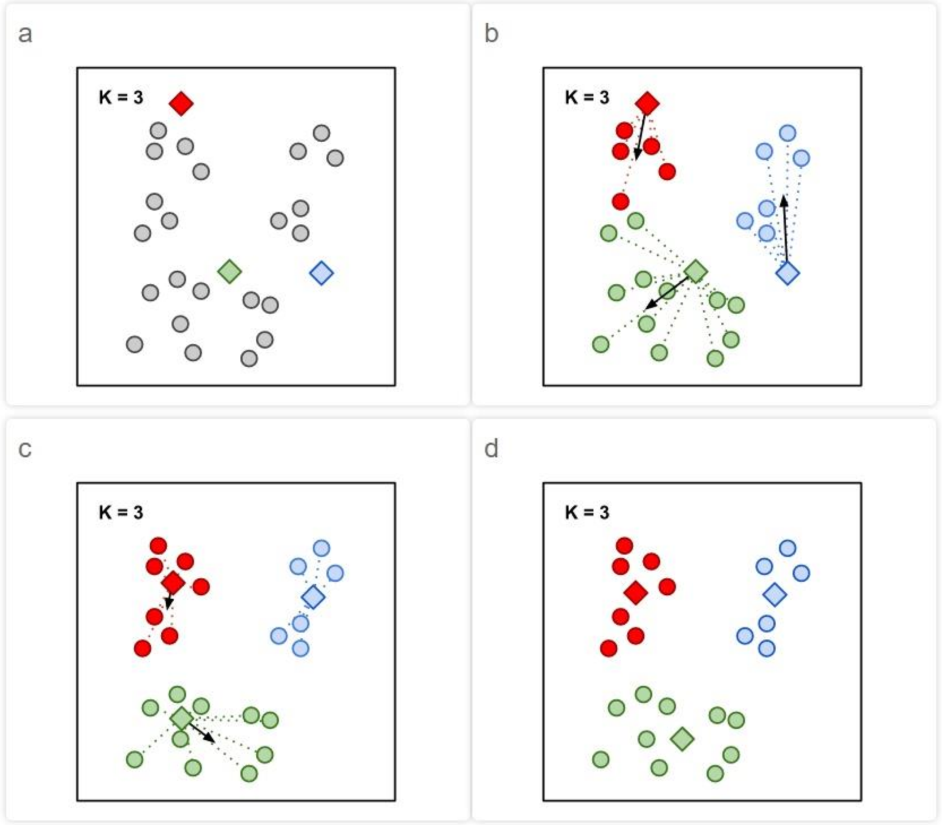
Most of the convergence happens in the first few iterations; however, generally, a maximum number of iterations is established in case the alghoritm does not converge.

Graphic example using three centroids

The **total within-cluster sum of square** measures the compactness (i.e *goodness*) of the clustering and we want it to be as small as possible.

$$tot.withinss = \sum_{k=1}^{k} W(C_k) = \sum_{k=1}^{k} \sum_{x_i \in C_k} (x_i - \mu_k)^2$$

where:
-x_i design a data point belonging to the cluster C_k
-μ_k is the mean value of the points assigned to the cluster C_k



From: https://mb3is.megx.net/gustame/dissimilarity-based-methods/cluster-analysis/non-hierarchical-cluster-analysis

Some comments on K-means

1. Solutions may vary with different distance measures.
2. It is problematic when an object is equidistant from two or more centroids.
3. Standardising variables may change the solution. Prior to standardisation, variables with larger ranges will contribute more to the distance of objects.
4. When using Euclidean distances, variables which are correlated will naturally influence object positioning in similar ways. Davidson (2002) recommends performing factor analysis on the variables and, if it is successful, using the extracted factors as input for *k*-means clustering.

https://mb3is.megx.net/gustame/dissimilarity-based-methods/cluster-analysis/non-hierarchical-cluster-analysis

Some comments on K-means

5. We need to specify the number of clusters in advance.
6. It works best on data which contains spherical clusters; clusters with other geometry may not be found.
7. It is sensitive to noisy data and outliers since a small number of such data can substantially influence the mean value.
8. It has problems when clusters are of differing sizes (biased toward the larger clusters)
9. It can become 'stuck' in local optima. Thus, repeating the clustering by adding noise to the data can help evaluate the robustness of the solution.

Fuzzy Clustering

The clusters produced by the k-means procedure are sometimes called **"hard" or "crisp"** clusters, since any feature vector x either is or is not a member of a particular cluster.

This contrasts with **"soft" or "fuzzy" clusters**, in which a feature vector x can have a degree of membership in each cluster.

Before to talk about fuzzy clustering, we should understand what is fuzzy logic and its motivation.

A brief overview on Fuzzy Logic

"One day I understood that science was not true. I do not remember the day but I remember the moment. The god of the twentieth century was no longer God. There is an error and seems common to all those who deal with science: they claim that everything is true or false; ***they do not always know with certainty which things are true and which are false, but they are sure that things are true or false*** *". Bart Kosko*

Bivalence sacrifices accuracy to simplicity; in fact, it requires a forced approximation of things: almost A must become A, almost non-A must become non-A.

According to **Zadeh's perspective**, we need to take into account the **vagueness (fuzzyness) of the real world.**

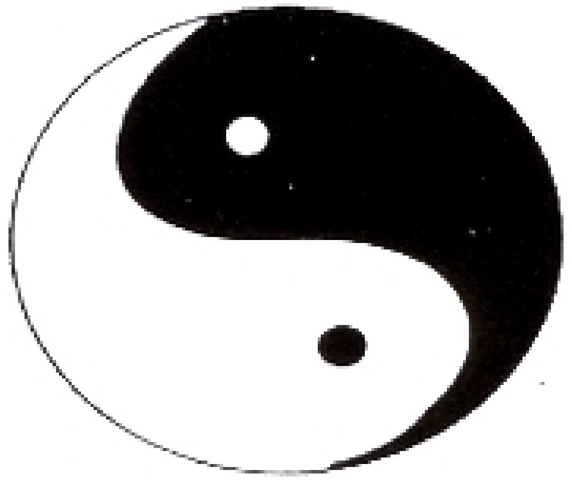
Fuzzy Thinking

“For any philosophy or religion there is always a devil to avoid or destroy..... The existence of contradiction A and not-A is the devil of Aristotelian logic “ (Bart Kosko)

"God does not play dice“ (Albert Einstein)

With this sentence he meant that dichotomy was far from the multi-purpose nature of the reality in which we live.

Fuzzy logic is much more useful where start contradictions, where A and non-A melt to a certain extent.



Zeno's Paradox

Zeno picked up a grain of sand from a pile and asked if the pile was still a heap.

Then he took another grain pile and asked again if this was or if it was turned into a non-heap.

The grain, being small by definition, could not bring change to the heap pile of his qualification, since the variation he brought the total was infinitesimal.

So Zeno, was never able to figure out **<u>what was the grain of sand that would change the heap into a non-heap</u>**; in fact, continuing to take other grains of sand, seemed to get a heap and a non-heap, A and not-A.

The limits of traditional logic

Traditional logic answers to two fundamental principles:
- The logical principle of the excluded middle;
- The logical principle of non-contradiction.

For the logical principle of the excluded middle can not exist a third truth value in addition to "true" and "false.“

For the logical principle of non-contradiction it is impossible that a think could be "true" and “false "at the same time.

The difficulty in understanding what is the critical point at which a "thing" becomes a "no thing", undermines the traditional logic which is based on black and white, all or nothing, true or false, according to Aristotelian logic.

1

Often in statistical surveys we use crisp sets relying on traditional logic, using the conventions that have little to do with reality. For example: if in a survey of customer satisfaction we get as a result that **adults** believe that a product is **good**; this is a meaningless result.

2

What does the word **adult** mean? Conventions establish that an adult is a person between 18 and 60 years.

3

But…who has decided this? How could you say that there is a difference between a person who in nearly 18 years, and one that has just made them.

4

What does the word **good** mean? Who decided that good must result in a value of 8 on a scale from one to 10? Someone may give a different meaning.

5

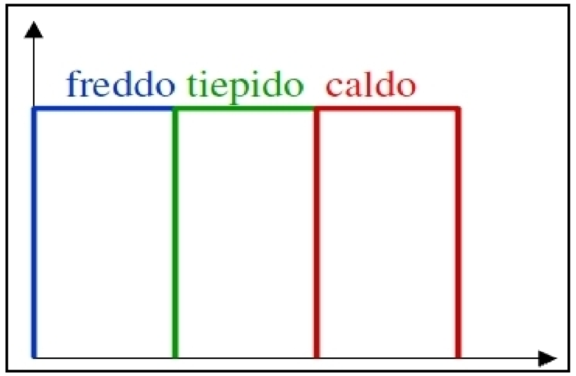
Fuzzy logic tries to solve this problem by eliminating sharp edges, blurring the boundaries, overcoming the paradoxes due to the fact of having to choose whether an element belongs to a group rather than to another.

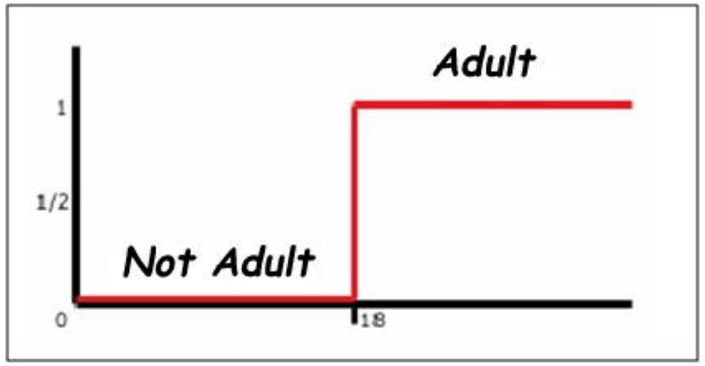
Conventions & Language

Classical Set

For a **traditional set**, which we define **crisp**, to distinguish it from a fuzzy set, the **membership function is Boolean**, it associates to each element x of the universe a value of “0" or “1" depending on whether or not x belongs to the set.

$$\mu_A(x) = \begin{cases} 1 \text{ se } x \in A \\ 0 \text{ se } x \notin A \end{cases}$$





Fuzzy Set

The fuzzy set theory extends the classical set theory, introducing the concept of membership function. The theory of fuzzy sets provides that an element can belong partially to a set, according to a membership function to real values in the interval [0,1].

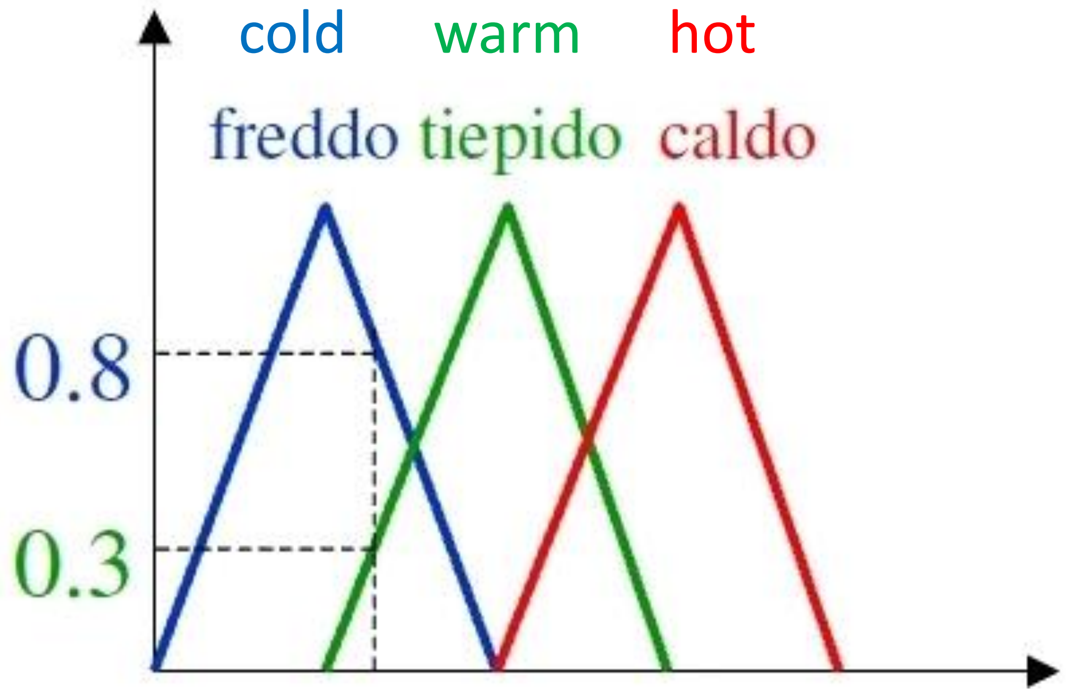
A fuzzy set A is defined by its characteristic function: $\mu_A: X \to [0,1]$ being X the universe of definition.

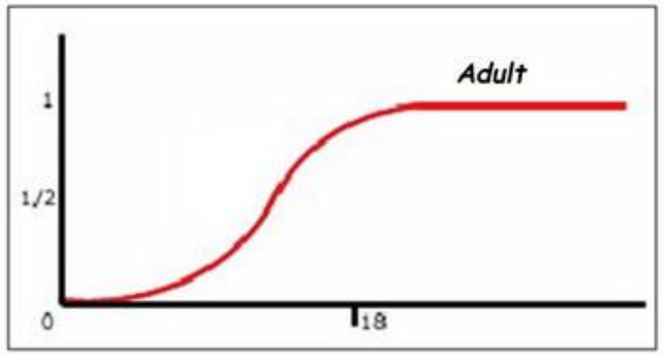
A fuzzy set can be defined as the set of ordered pairs formed by the elements of X and the corresponding value of the membership function :

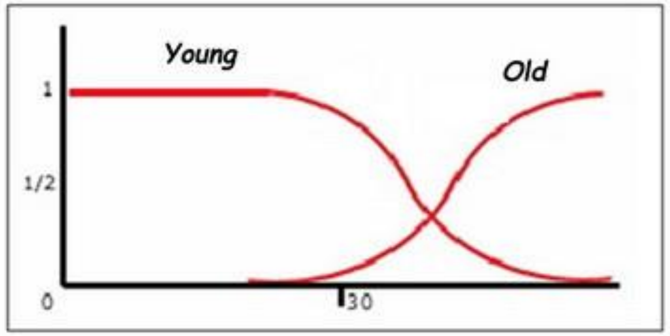
$$A = \{(x, \mu_A(x) \mid x \in X\}$$

The value μ_A (x) is called degree of membership of the element x to the fuzzy set A, that is, the subjective truth value associated with the proposition "x is an element of A", in a logic of continuity.

Normally in the literature under the name of fuzzy set means its characteristic function.







Fuzzy Set Examples

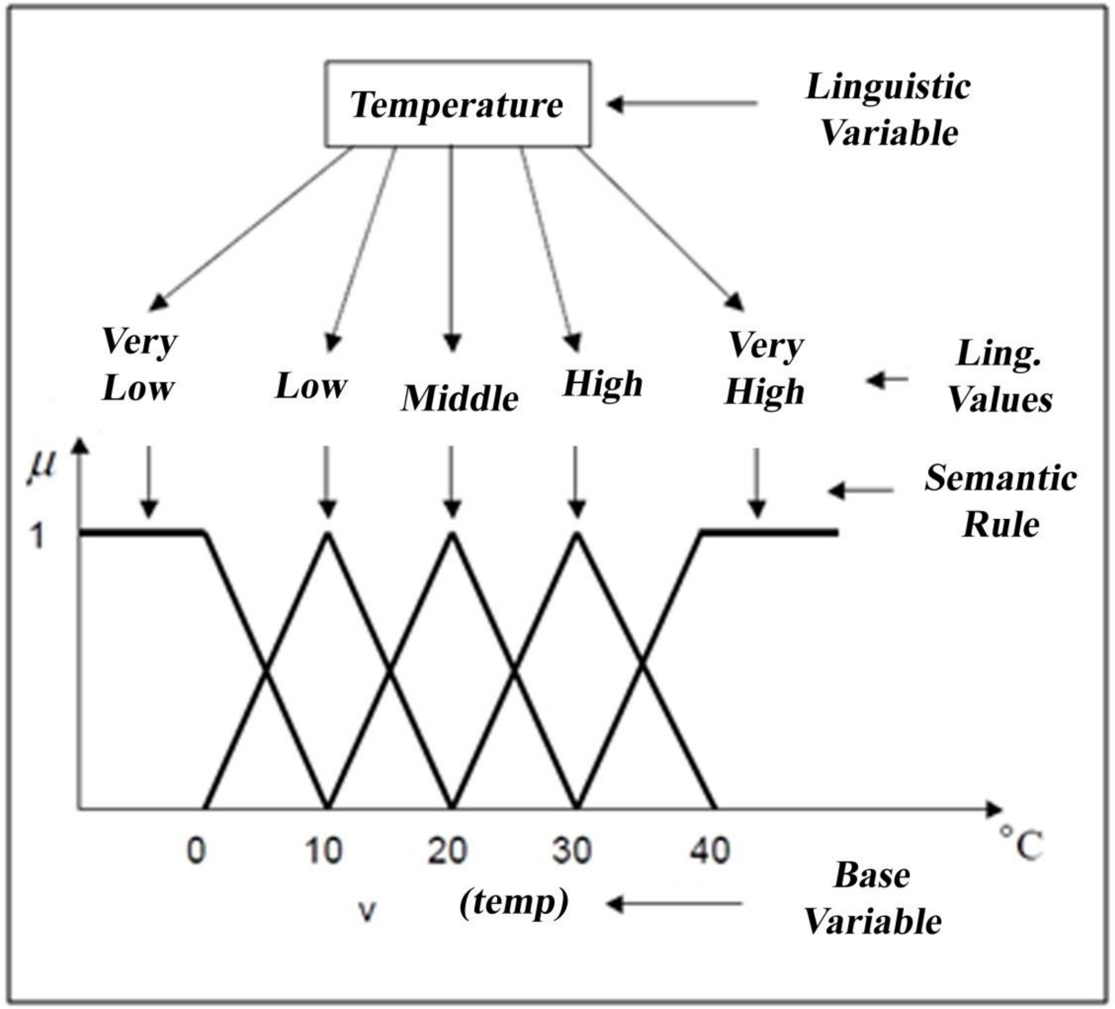
Uncertainty

From the statistical point of view, the uncertainty can derive from:

• imprecision related to **measurement** of phenomena;
• imprecision of **language**;
• from the total or partial ignorance of **values** or the theoretical underlying **assumptions**;
• from the **links** between the observed data and the universe of possible data.

Although the last of them fall within the usual probability theory, in the context of fuzzy logic, the proposed solutions tend to **overcome all the limitations that are typical of the classical approach**, such as those relating the need to introduce very restrictive assumptions about the nature and distribution of the data.

Arithmetic with Fuzzy Numbers and Properties of Fuzzy Sets are out of the scope of this presentation, hence for more detail see: Zadeh, L., 1975. The concept of a linguistic variable and its application to approximate reasoning. Information Scienze I.



Example of a Linguistic Variable

Let's come back to «soft» clustering

Once we understand what Fuzzy Logic is and what its motivations are, we can go back to clustering algorithms to try to understand what the basic idea of a soft algorithm is.

Fuzzy classification methods **do not pretend to give precise answers on how to aggregate the data**, but, on the contrary, they try to represent the imprecision inherent in the data.

At this point, however, a clarification must be made, when we talk about soft clustering, we must distinguish the case in which the data is fuzzy or the "belonging to the groups" is fuzzy.

The most famous and used methods of Soft clustering are those that treat the "belonging to the groups" as fuzzy, and in particular the most used method is the so-called "fuzzy k-means".

Fuzzy K-means

Fuzzy K-Means does not force a unit to belong to a unique group but **allows to associate it to n groups with a certain degree of membership in the interval [0,1].**

In this context, the objective function determines, for each solution, a **measure of the error in terms of efficiency or cost, based on the distance between the data and the representative elements of the clusters.**

Constraints

-The sum of the membership functions (u_{ik}) of the unit i satisfies the following constraint:

$$\sum_{k=1}^{c} u_{ik} = 1$$

Where k=1,2, … , c.

-The membership functions belong to the interval [0,1].

$$0 \leq u_{ik} \leq 1$$

The Objective Function

Let J_m be the **objective function** to minimize:

$$J_m(U, v) = \sum_{k=1}^{c} \sum_{i=1}^{n} (u_{ik})^{m} (d_{ik})^{2}$$

where:

- - $(d_{ik})^2 = |x_i - v_k|^2$ is a suitable norm on R^p (e.g. Euclidean norm);
- $v_k \in R^p$ is the k-th component of the vector of centroids;
- $x_i \in R^p$ is the i-th component of the vector of units;
- $U[u_{ik}]$ is the matrix n x c containing the membership functions.

Thus, <u>the variables with respect to which minimize are the cluster centers and the degrees of membership. The meaning of the objective function is that each centroid is the best representation of the units that make up the group</u>, as it minimizes the sum of the squares distaces $(d_{ik})^2$ **. The optimal partition defined as least variance partition is the one that minimizes J_m.**

The value of the m parameter to be chosen at the beginning of the procedure expresses <u>**the degree of fuzziness,**</u> that is, how the resulting partition will be blurred.

LET’S TRY IT ON R….