FIB UPC UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

# Grado en Inteligencia Artificial

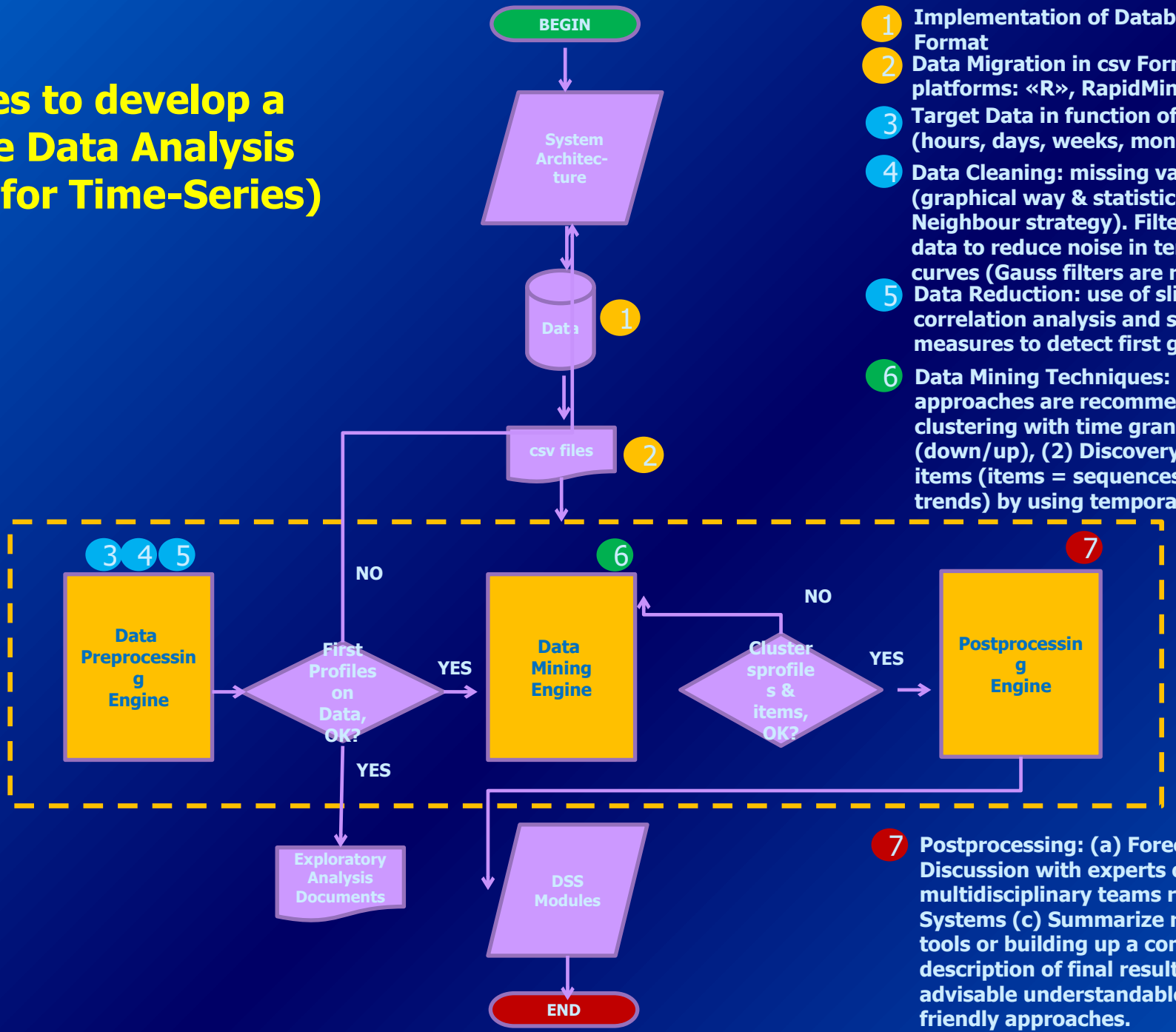
## Advanced Clustering – Time Series



**Prof. Dante Conti**

**Prof. Sergi Ramírez**

# Outlines to develop a suitable Data Analysis Platform for Time-Series)




Architecture of Data Analysis Engine

1. Implementation of Database Model & Data Format
2. Data Migration in csv Format to GNU platforms: «R», RapidMiner, mySQL, Weka
3. Target Data in function of time granularity (hours, days, weeks, months)
4. Data Cleaning: missing values, outliers (graphical way & statistical tests & Neighbour strategy). Filtering audited data to reduce noise in temporal series curves (Gauss filters are recommended)
5. Data Reduction: use of sliding windows, correlation analysis and similarity measures to detect first grouping signals.
6. Data Mining Techniques: two major approaches are recommended: (1) clustering with time granularity (down/up), (2) Discovery of frequents items (items = sequences, temporal trends) by using temporal data mining.
7. Postprocessing: (a) Forecasting tasks (b) Discussion with experts or multidisciplinary teams regarding Water Systems (c) Summarize results with visual tools or building up a conceptual description of final results. For (c) it is advisable understandable and user-friendly approaches.

# Time Series Mining

- ## Data Format

| DATE | X0 | X1 | X2 | X3 | X4 | ...X23 |
|---|---|---|---|---|---|---|
| 01/03/2011 | 16630,1 | 13392,17 | 12757,23 | 12584,18 | 12707,2 | ... |
| 02/03/2011 | 16310,55 | 13541,09 | 12576,87 | 12296,47 | 12348,89 | ... |
| 03/03/2011 | 16758,89 | 13829,13 | 12960,01 | 12717,28 | 12851,31 | ... |
| 04/03/2011 | 16378,45 | 13269,95 | 12614,3 | 12325,09 | 12401,48 | ... |
| 05/03/2011 | 16412,62 | 14245,78 | 13147,6 | 12781,1 | 12786,73 | ... |
| 06/03/2011 | 17462,05 | 14869,66 | 13534,39 | 13157,48 | 13002,9 | ... |
| 07/03/2011 | 15759,4 | 13438,4 | 12387,89 | 12084,77 | 12151,04 | ... |
| 30/04/2012 | ..... | ..... | ..... | ..... | ..... | ..... |

Data from 03/01/2011 till 04/30/2012 – 427 records in 14 months. X0, X1, ..., X23 represent hourly water flow (cubic meters per hour)., i.e., X0 is water flow from 00.00 hours to 00.59 hours and so on.

- ## Data Analytics Techniques

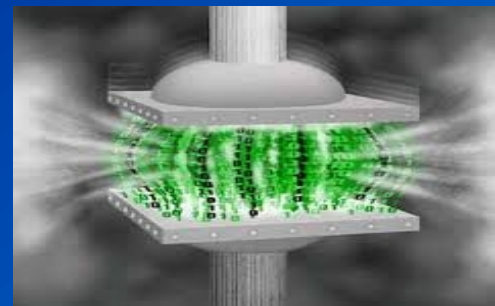
**NEIGHBOUR STRATEGY** for preprocessing and missing values

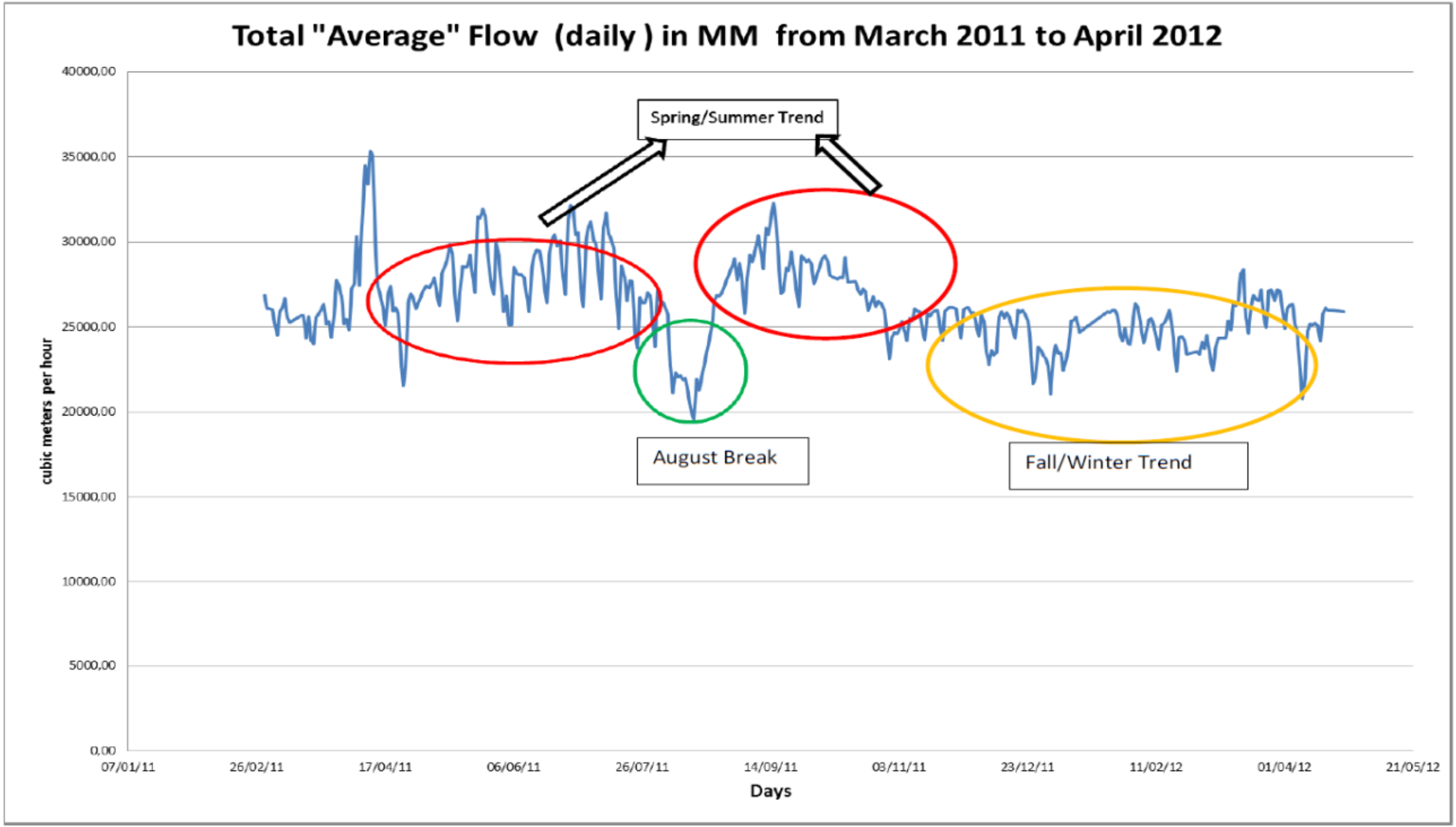
**TEMPORAL DATA MINING**
Frequent events and episodes

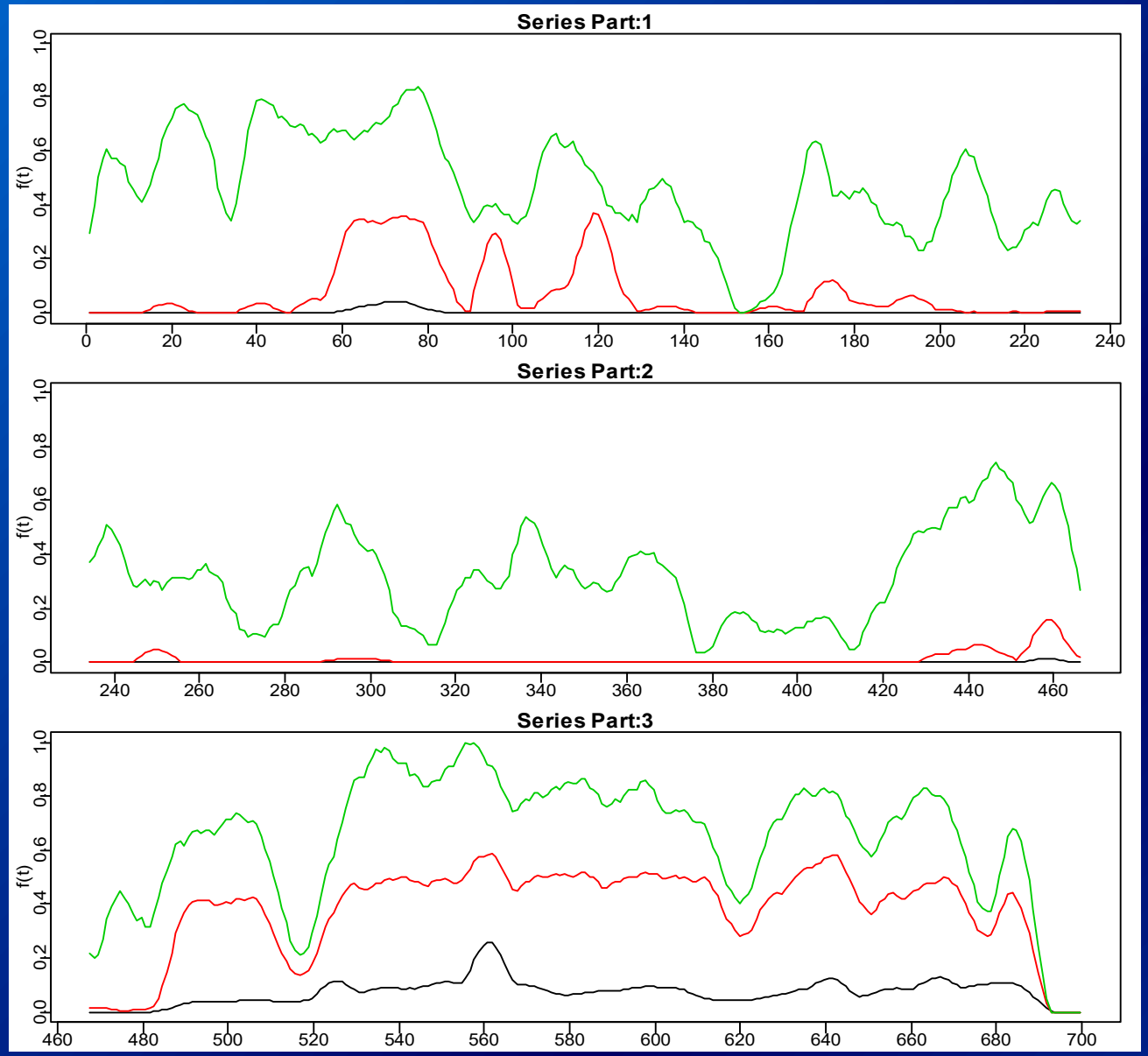
**FORECASTING**
ARMA,ARIMA, linear and non linear regression

**GRAPHICAL ANALYSIS** with Temporal Windows

**CLUSTERING ALGORITHMS**
Hierarchical; K-means; Partioning around mediods (PAM), stochastic clustering and bootstrap



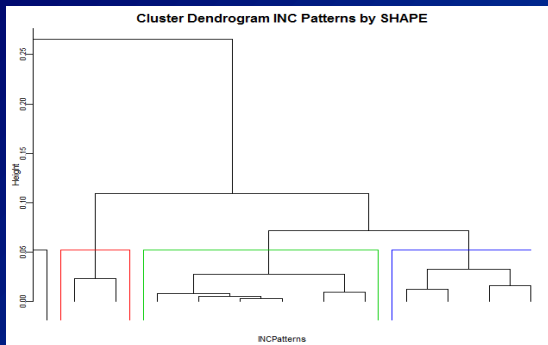

Total "Average" Flow (daily ) in MM from March 2011 to April 2012
40000,00
35000,00
30000,00
25000,00
20000,00
15000,00
10000,00
5000,00
0,00
cubic meters per hour
Spring/Summer Trend
August Break
Fall/Winter Trend
07/01/11
26/02/11
17/04/11
06/06/11
26/07/11
14/09/11
03/11/11
23/12/11
11/02/12
01/04/12
21/05/12
Days

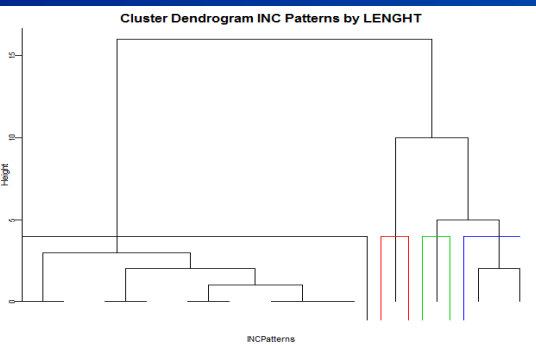


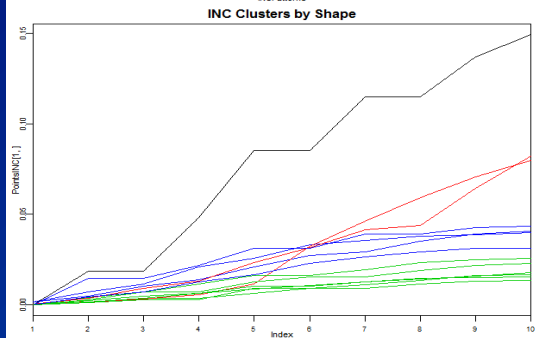

Green= Peak Hours Demand

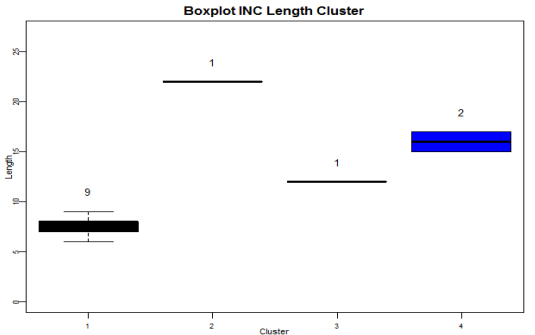
Red= Medium level for Demand

Black = Low level demand



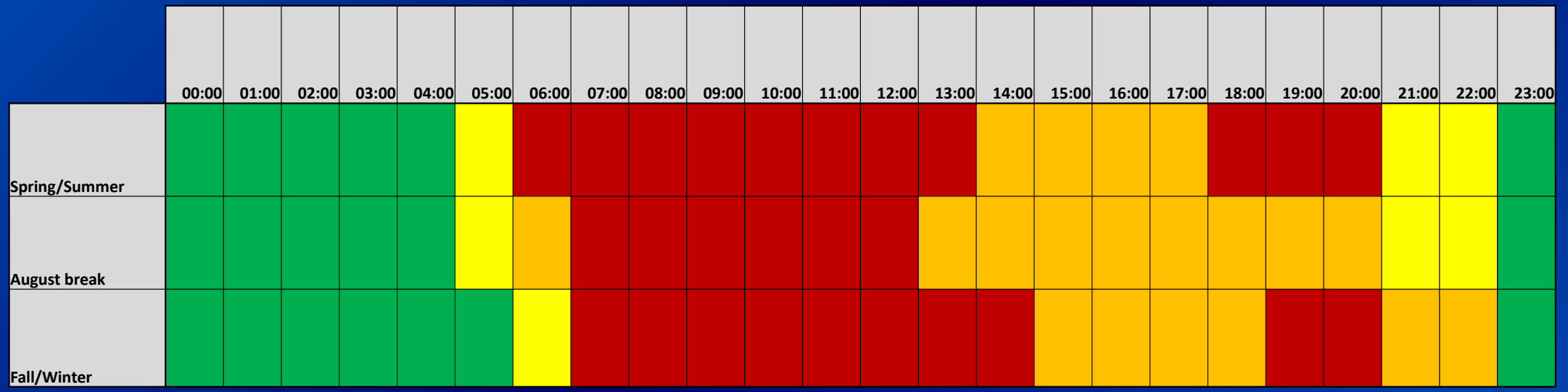



# • Temporal Clustering & Postprocessing

## Time Scheduling versus levels of hourly water Flow




RED= High Level of Water Flow

GREEN= Low Level of Water Flow

YELLOW= Middle level of Water Flow

ORANGE= Middle-High level of Water Flow

# Time-Series Clustering

**Main Strategies for Clustering**

**Whole time series clustering**
Roelofsen (2018): whole (complete) time series of equal length are compared

**Subsequence time series clustering**
Roelofsen (2018): a given subsequence is compared with the subsequences of other whole time series

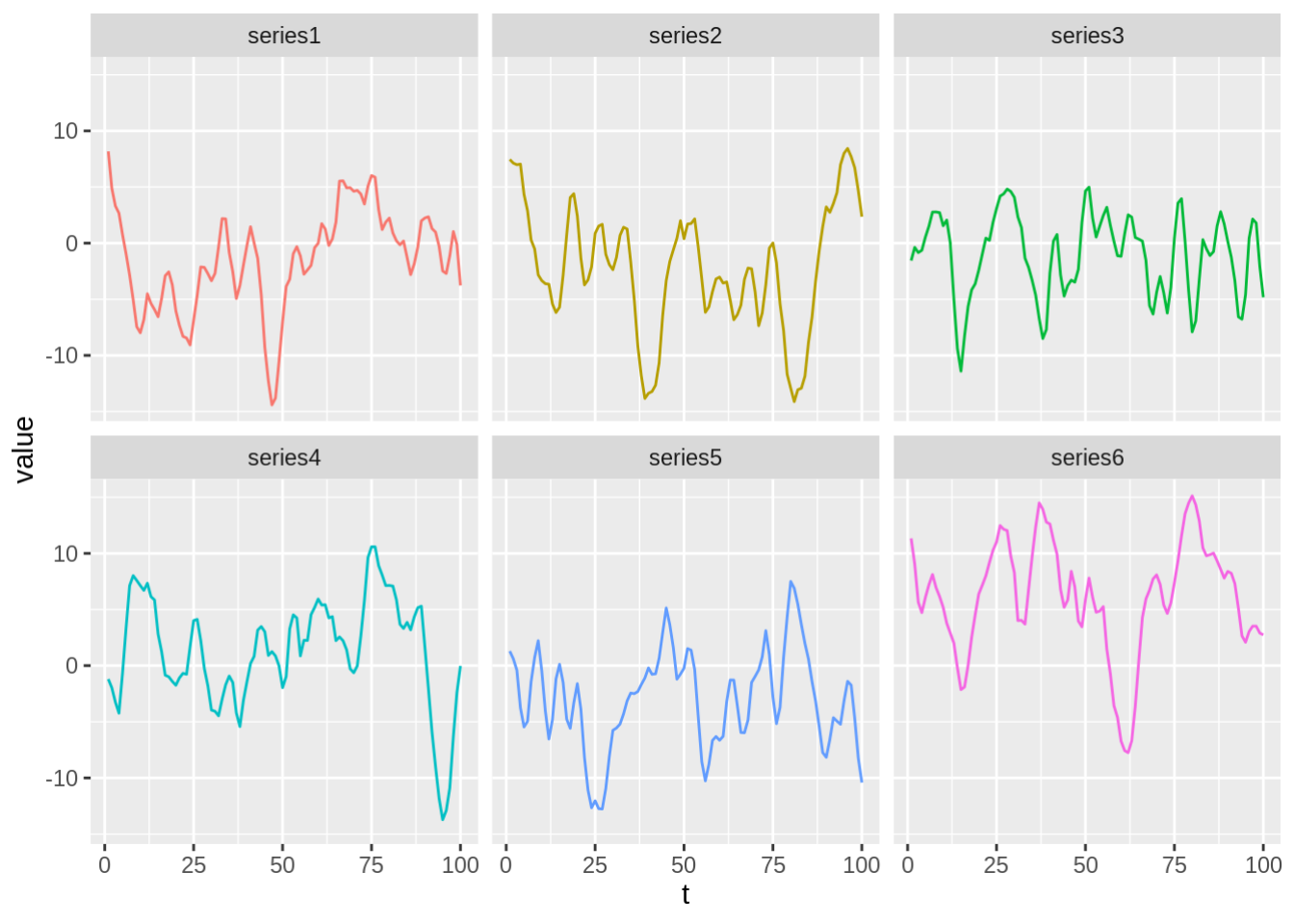
Marques, A., 2018, at http://rstudio-pubs-static.s3.amazonaws.com/398402_abe1a0343a4e4e03977de8f3791e96bb.html.
Roelofsen, P., 2018, Time series clustering: Vrije Universiteit Amsterdam, Amsterdam.

https://cran.r-project.org/web/views/TimeSeries.html

http://www.timeseriesclassification.com/index.php

# Time-Series Clustering

**Time point clustering**
**Distance / Similarity Measures**




***Use distances or similarity metrics as in Classical Clustering***

# Time-Series Clustering

**Time point clustering**
**Distance / Similarity Measures**

**https://cran.r-project.org/web/packages/TSdist/TSdist.pdf**

1. Shape-based distances

Compare the overall shape of time series based on the actual (scaled) values of the time series.

i. Lock-step measures ($n = m$)

These distance measures require both time series to be of equal length ($n = m$) and compare time point $i$ of time series $x$ with the same point $i$ of time series $y$.

- Minkowski distance $L_p$-norma of the difference between two vectors of equal length ($n = m$).

$$d_{min}(x, y) = \left( \sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

It is the generalization of the commonly used Euclidean distance ($p = 2$), Manhattan distance ( $p = 1$) and Chebyshev distance ($p = \infty$). The time complexity for the Minkowsky distance is $O(n)$ and thus determining the distance matrix with this measure takes $O(nN^2)$ time.

$$d_{euc}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

## Time point clustering Distance / Similarity Measures

- Pearson correlation distance

Takes into account the linear association between two vectors of variables. The Pearson correlation coefficient is defined as:

$$\rho(x,y) = \frac{Cov(x,y)}{\sigma_x \sigma_y} = \frac{\mathbb{E}[(x-\mu_x)(y-\mu_y)]}{\sigma_x \sigma_y} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}$$

where $\mu_x$ and $\mu_y$ are the means of $x$ and $y$ and $\sigma_x$ and $\sigma_y$ are the standard deviations of $x$ and $y$, respectively. The values of $\rho$ lie within the range [-1,1], where $\rho = 1$ indicates a perfect positive relationship between $x$ and $y$, $\rho = -1$ indicates a perfect negative relationship between $x$ and $y$, and $\rho = 0$ indicates no relationship between the two variables. The Pearson correlation distance is defined as:

$$d_{cor}(x,y) = 1 - \rho(x,y)$$

This distance measure can take values in the range [0,2]. The time complexity is $O(n)$ and thus determining the distance matrix with this measure takes $O(nN^2)$ time.

Alternative correlation measures include Spearman's Rank and Kendall's Tau correlation coefficients. These coefficients indicate correlation based on rank, whereas the Pearson coefficient is based on a linear relationship between two vectors. This makes Spearman's Rank and Kendall's Tau less sensitive to noise and outliers. However, this comes with an increase in time complexity: $O(n\log(n))$ for Spearman's Rank and $O(n^2)$ for Kendall's Tau.

Spearman correlation indicates the direction of association between X (the independent variable) and Y (the dependent variable). If Y tends to increase when X increases, the Spearman correlation coefficient is positive.

$$\rho_{spearman} = \frac{Cov(rg_x, rg_y)}{\sigma_{rg_x}\sigma_{rg_y}}$$

where $rg_x$ is the rank of the vector $x$, which requires sorting the vector.

In the same way, Kendall correlation between two variables will be high when observations have a similar rank (i.e. relative position label of the observations within the variable) between the two variables.

$$\tau = \frac{concordant_{pairs} - discordant_{pairs}}{\binom{N}{2}}$$

Any pair of observations $(x_i, y_i)$ and $(x_j, y_j)$ are said to be concordant if the ranks for both elements agree. In order to count the number of concordant and discordant pairs, it is necessary to compare all pairs of observation, thus the time complexity $O(n^2)$.

# IMPORTANT

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}},$$

The resulting similarity ranges from -1 meaning exactly opposite, to 1 meaning exactly the same, with 0 indicating orthogonality or decorrelation, while in-between values indicate intermediate similarity or dissimilarity.

$$\text{cosine distance} = D_C(A, B) := 1 - S_C(A, B)\,.$$

$$\sqrt{2(1 - S_C(A, B))}$$

θ = arccos(SC(A, B)).

$$\text{angular distance} = D_\theta := \frac{\arccos(\text{cosine similarity})}{\pi} = \frac{\theta}{\pi}$$

$$\text{angular similarity} = S_\theta := 1 - \text{angular distance} = 1 - \frac{\theta}{\pi}$$

# Time point clustering
# Distance / Similarity Measures

## Cosine Distance

The term **cosine distance**[3] is commonly used for the complement of cosine similarity in positive space, that is

$$\text{cosine distance} = D_C(A, B) := 1 - S_C(A, B).$$

It is important to note that the cosine distance is not a true distance metric as it does not exhibit the triangle inequality property—or, more formally, the Schwarz inequality—and it violates the coincidence axiom. One way to see this is to note that the cosine distance is half of the squared Euclidean distance of the $L_2$ normalization of the vectors, and squared Euclidean distance does not satisfy the triangle inequality either. To repair the triangle inequality property while maintaining the same ordering, it is necessary to convert to angular distance or Euclidean distance. Alternatively, the triangular inequality that does work for angular distances can be expressed directly in terms of the cosines; see below.

## Angular distance and similarity

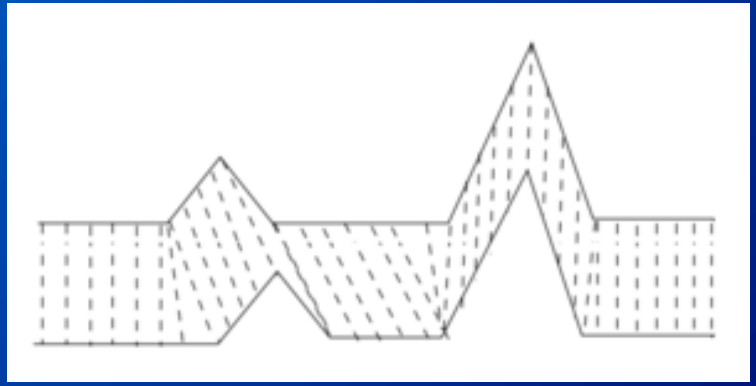
The normalized angle, referred to as **angular distance**, between any two vectors $A$ and $B$ is a formal distance metric and can be calculated from the cosine similarity.[4] The complement of the angular distance metric can then be used to define **angular similarity** function bounded between 0 and 1, inclusive.

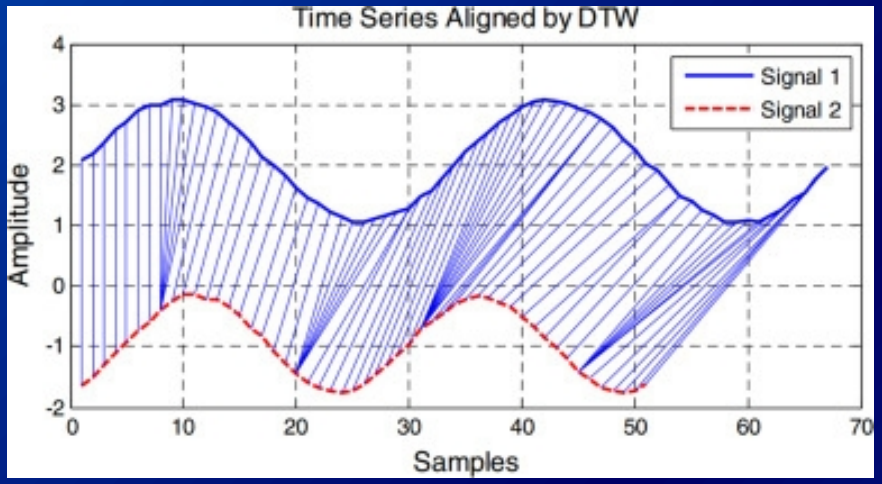
When the vector elements may be positive or negative:

$$\text{angular distance} = D_\theta := \frac{\arccos(\text{cosine similarity})}{\pi} = \frac{\theta}{\pi}$$

$$\text{angular similarity} = S_\theta := 1 - \text{angular distance} = 1 - \frac{\theta}{\pi}$$

# Time point clustering Distance / Similarity Measures






ii. Elastic measures ($n \neq m$)

Elastic measures allow one-to-many and one-to-none matching, and are able to warp in time and be more robust in handling outliers. The main disadvantage, however, is that elastic distance measures generally come with and increase in time complexity.

- Dynamic time warping

DTW warps two sequences $x$ and $y$ non-linearly in time in order to cope with time deformations and varying speeds in time dependent data. When determining the DTW distance between two time series, first and $(nxm)$ local cost matrix (LCM) is calculated, where element $(i, j)$ contain the distance between $x_i$ and $y_j$. This distance is usually defined as the quadratic difference: $d(x_i, y_j) = (x_i, y_j)^2$ . Next, a warping path $W = w_1, w_2, \ldots, w_K$ is determined, where $\max(n, m) < K < m + n - 1$. This path transverses the LCM under three constraints:

a. Boundary condition: path must start and end in the diagonal corners $w_1 = (1, 1), w_k = (n, m)$.
b. Continuity: only adjacent elements in the matrix are allowed for steps in the path.
c. Monotonicity: subsequent steps in the path must be monotonically spaced in time.

The total distance for path $W$ is obtained by summing the individual elements (distances) of the LCM that the path traverses. To obtain the DTW distance, the path with minimum total distance is required. This path can be obtained by an $O(nm)$ algorithm that is based on dynamic programming:

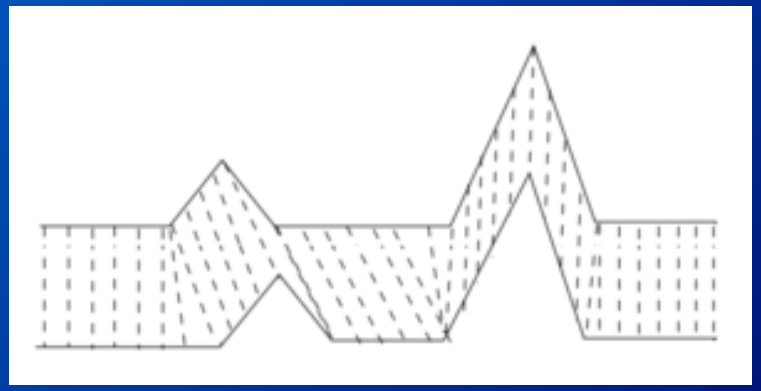
$$d_{cum}(i, j) = d(x_i, x_j) + \min\{d_{cum}(i-1, j-1), d_{cum}(i, j-1), d_{cum}(i-1, j)\}$$

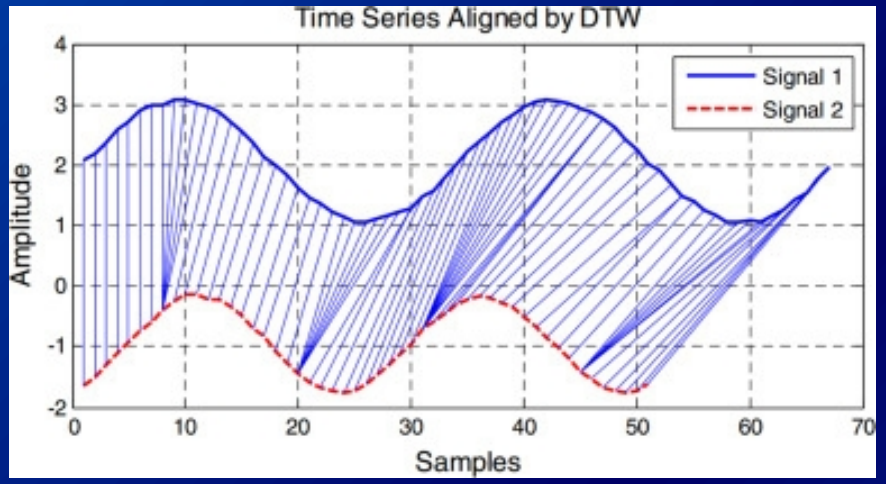
$$d_{DTW}(x, y) = min\sqrt{\sum_{k=1}^{K} w_k}$$

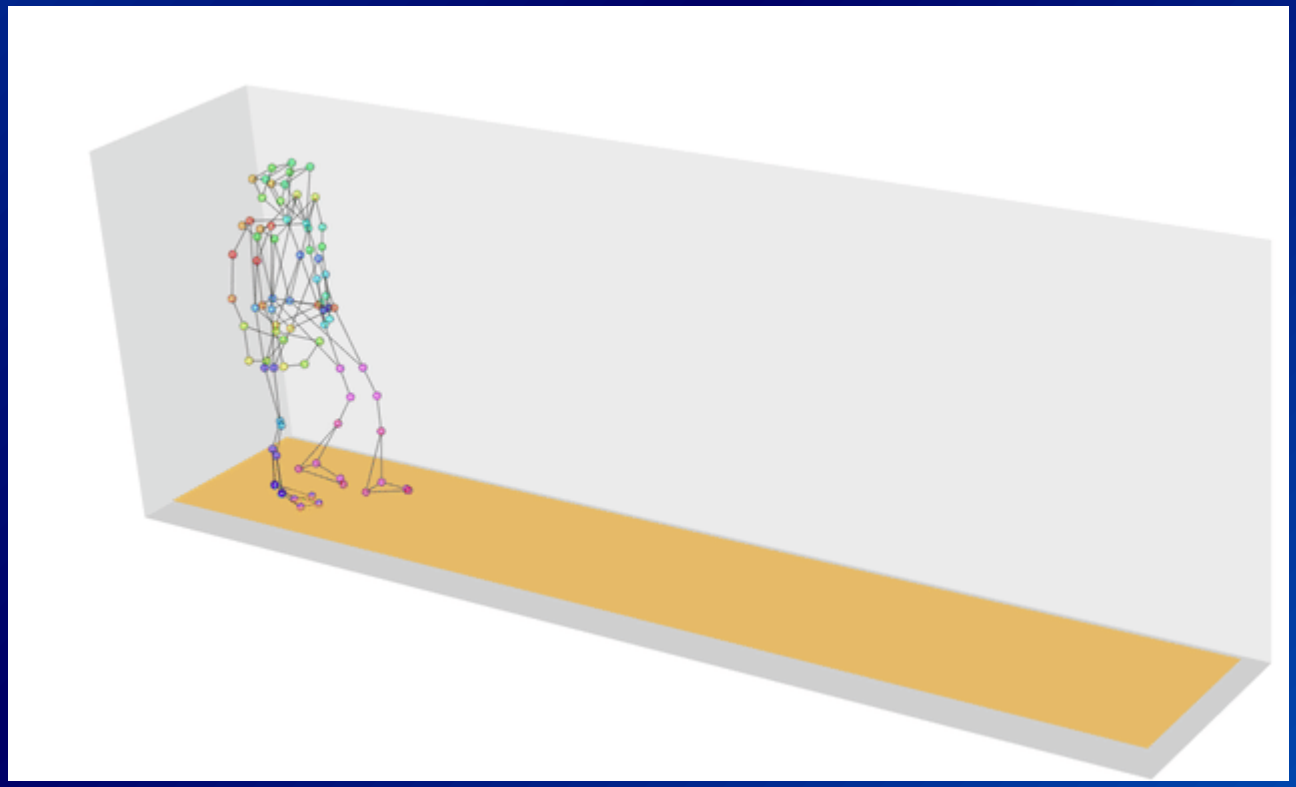
This distance is equal to the Euclidean distance for the case where $n = m$ and only the diagonal of the LCM is traversed. DTW does not satisfy the triangle inequality, even when the local distance measure is a metric.

# Time point clustering
# Distance / Similarity Measures

**IMPORTANT**





Two repetitions of a walking sequence recorded using a motion-capture system. While there are differences in walking speed between repetitions, the spatial paths of limbs remain highly similar

# Time point clustering
# Distance / Similarity Measures

- Longest Common Subsequence

Finding the longest subsequence that is common to two or more sequences. Here a subsequences is define as a sequence that appears in the same relative order, but where the individual elements are not necessarily contiguous.

$$L(i,j)\begin{cases} 0 & \text{if } i=0 \text{ or } j=0 \\ 1+L(i-1,j-1) & \text{for } |x_i-y_j|<\epsilon \\ max\{L(i-1,j),L(i,j-1)\} & \text{otherwise} \end{cases}$$

This distance only regards similar points, so it is robust to noise and outliers. The time complexity is $O(mn)$, but often a warping threshold $\delta$ is added to restrict the matching to a maximum difference in time ($|i-j|<\delta$), improving the time complexity to $O((n+m)\delta)$.

$$d_{LCSS}(x,y)=\frac{n+m-2L(n,m)}{n+m}$$

# Time point clustering
# Distance / Similarity Measures

Transforms the time series from the "time-domain" $x(t)$ to a "frequency-domain" representation $X(f)$. The Fourier transform decomposes a time series into all different cycles (amplitude, offset and rotation speed). DFT is calculated by the inner product of the time series and a sine wave:

$$X(f) = \sum_{t=0}^{n-1} x_t \, e^{-i\frac{2\pi f}{n}t}$$

The resulting vector $X(f)$ if a vector of $n$ complex numbers. The inverse DTF transform a collection of frequencies $X(f)$ back to the time-domain:

$$x(t) = \frac{1}{n} \sum_{f=0}^{n-1} X(f) \, e^{i\frac{2\pi f}{n}t}$$

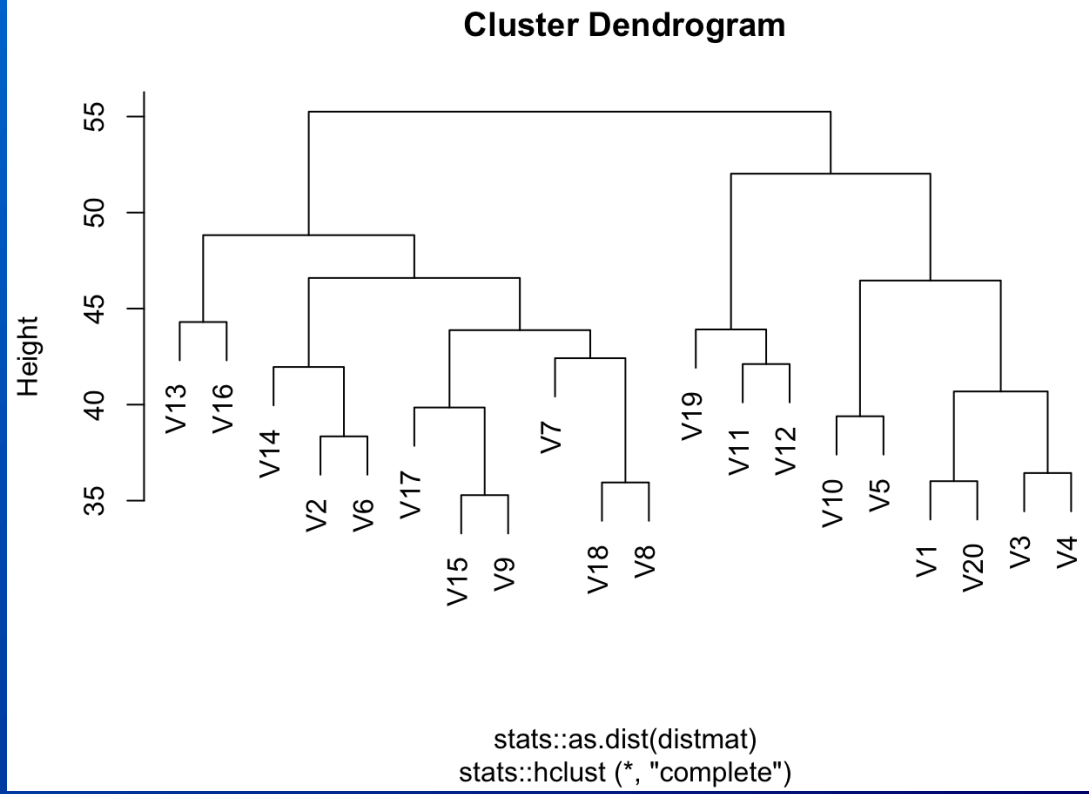
Parseval's theorem: the DTF preserved the Euclidean distance between two time series: When all the frequencies are use, the Euclidean distance between two Fourier transforms is equal to the Euclidean distance between the original time series. This is caused by the fact that DFT is a linear transformation.

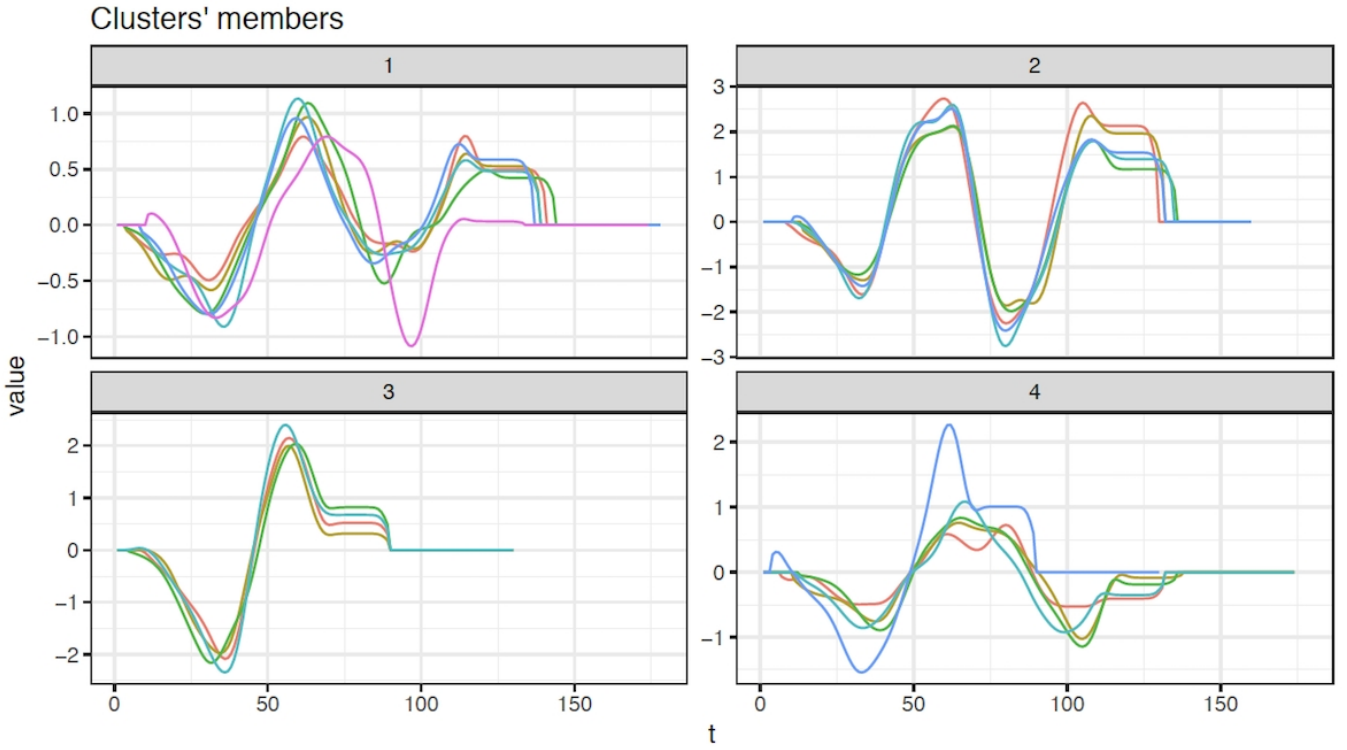
- Discrete Wavelet Transform (wavelets)

DWT is a dimensionality reduction method that also reduces noise. It decomposes a time series into a set of bases functions that are called wavelets. A wavelet is a rapid decaying, wave-like oscilation with mean zero, and have finite duration. Wavelets are defined by two functions: the wavelet function (mother) $\psi$ and the scaling function (father) $\varphi$. The mother defines the basic shape, and the father the scale (frequency).

The DWT is obtained by successively passing a time series through high-pass and low-pass filters, which produces detail and approximation coefficients for different levels of decomposition.

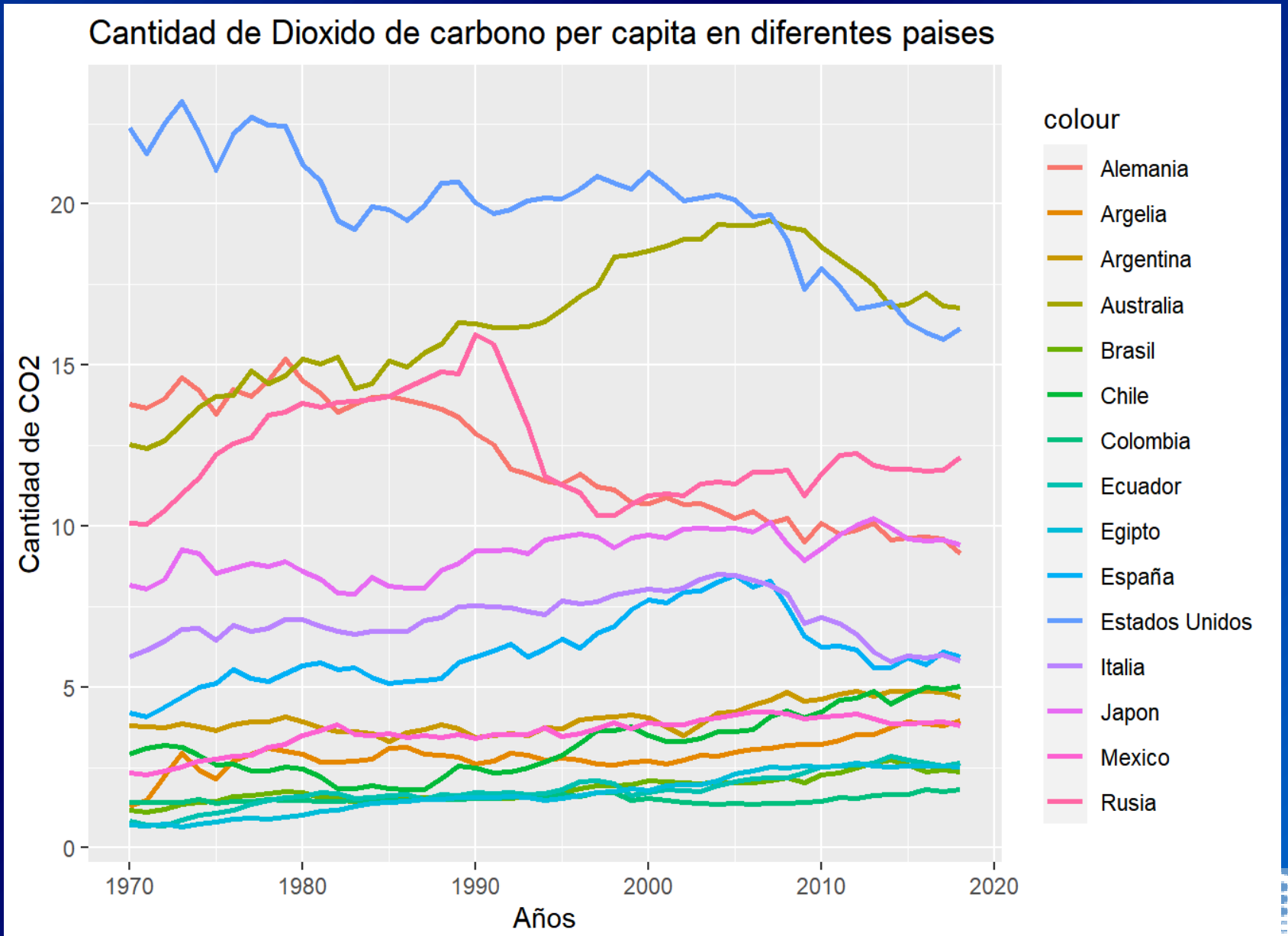
Main advantage over DFT: captures frequency and location in time.



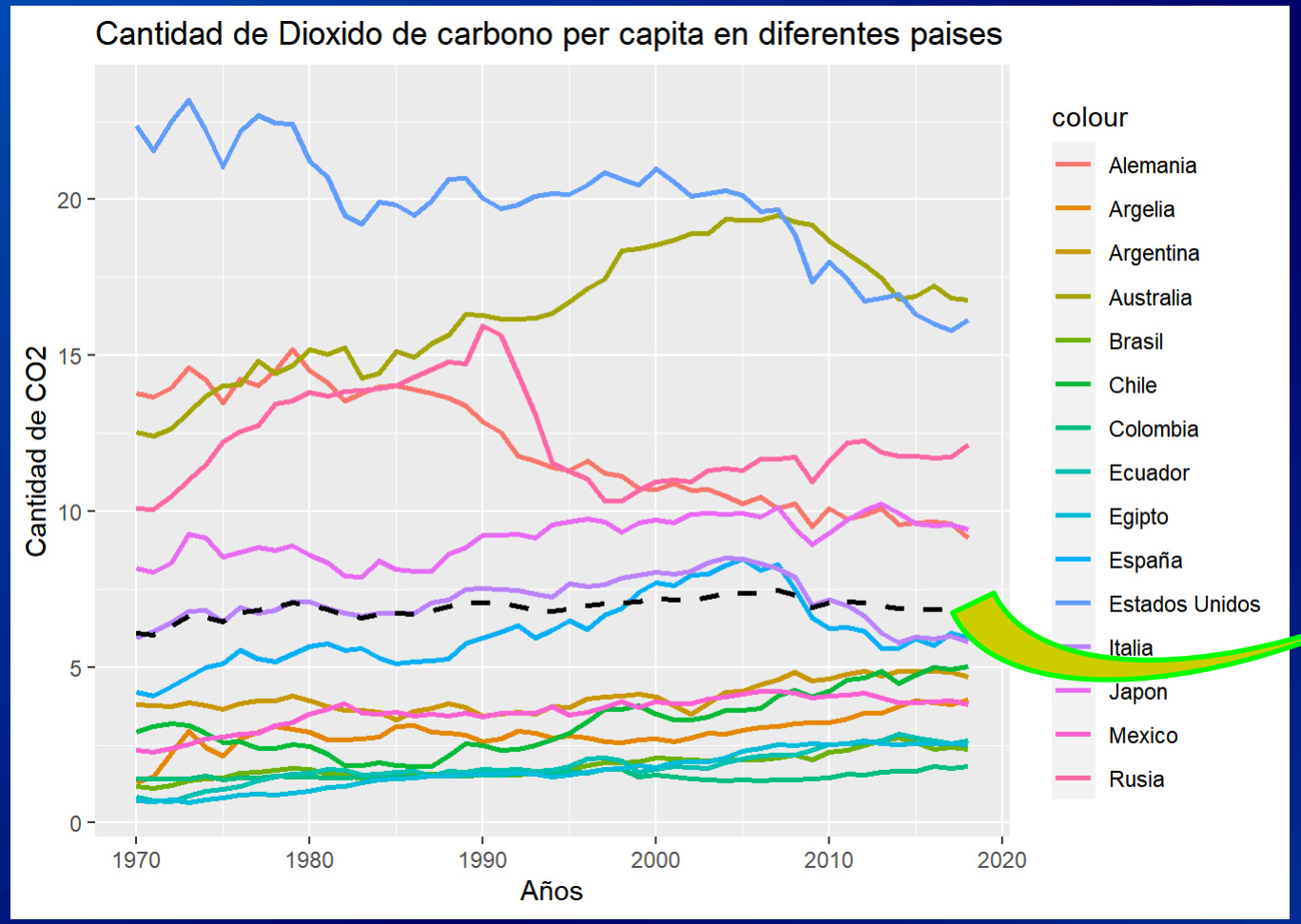

***Check results and extract profiles***

# Time-Series Clustering Statistical Approach (Extraction of trends – Non parametric Methods



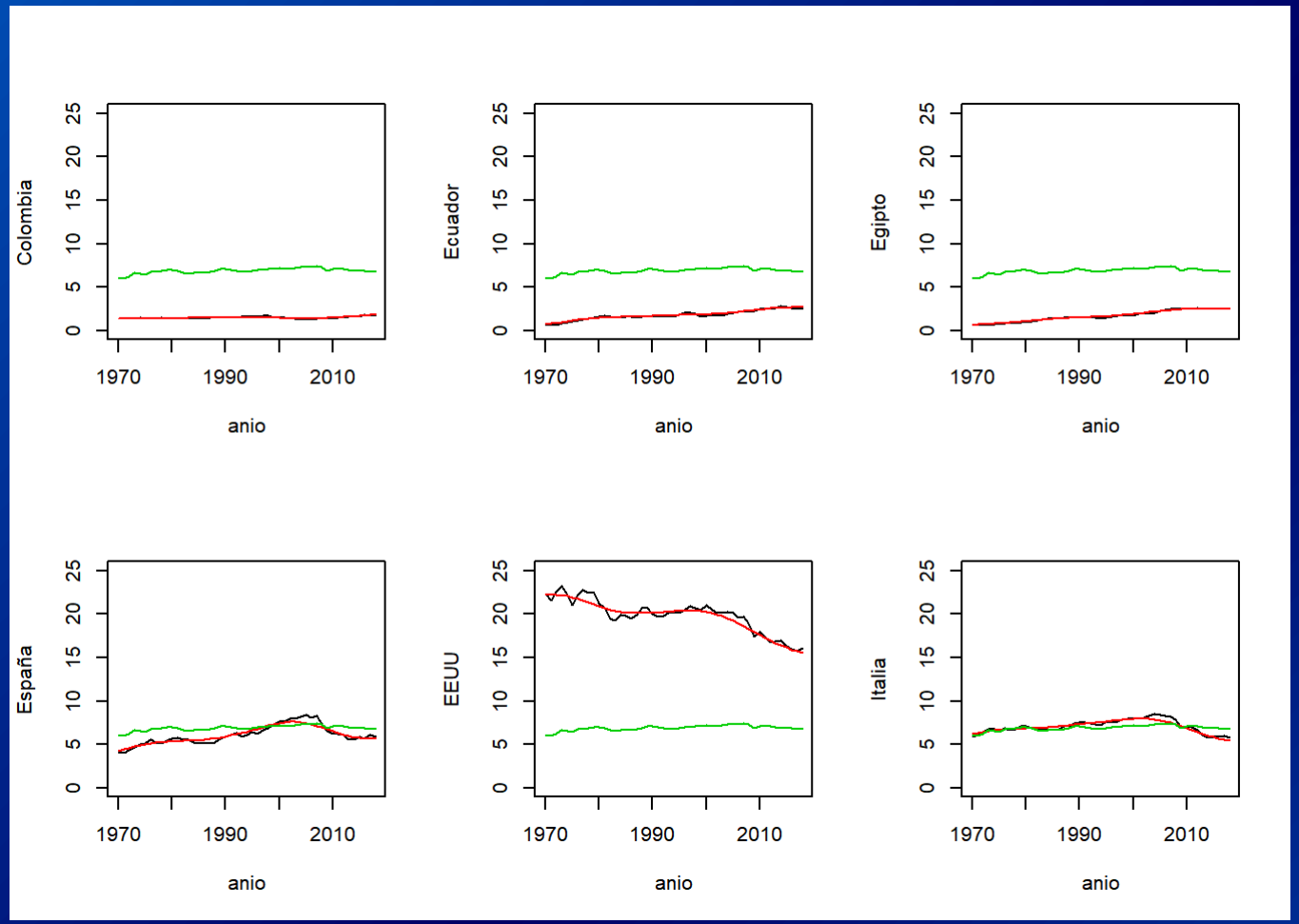

https://rpubs.com/Edison-D/615477
(Case study in Spanish language)

# Time-Series Clustering Statistical Approach – Non parametric Methods



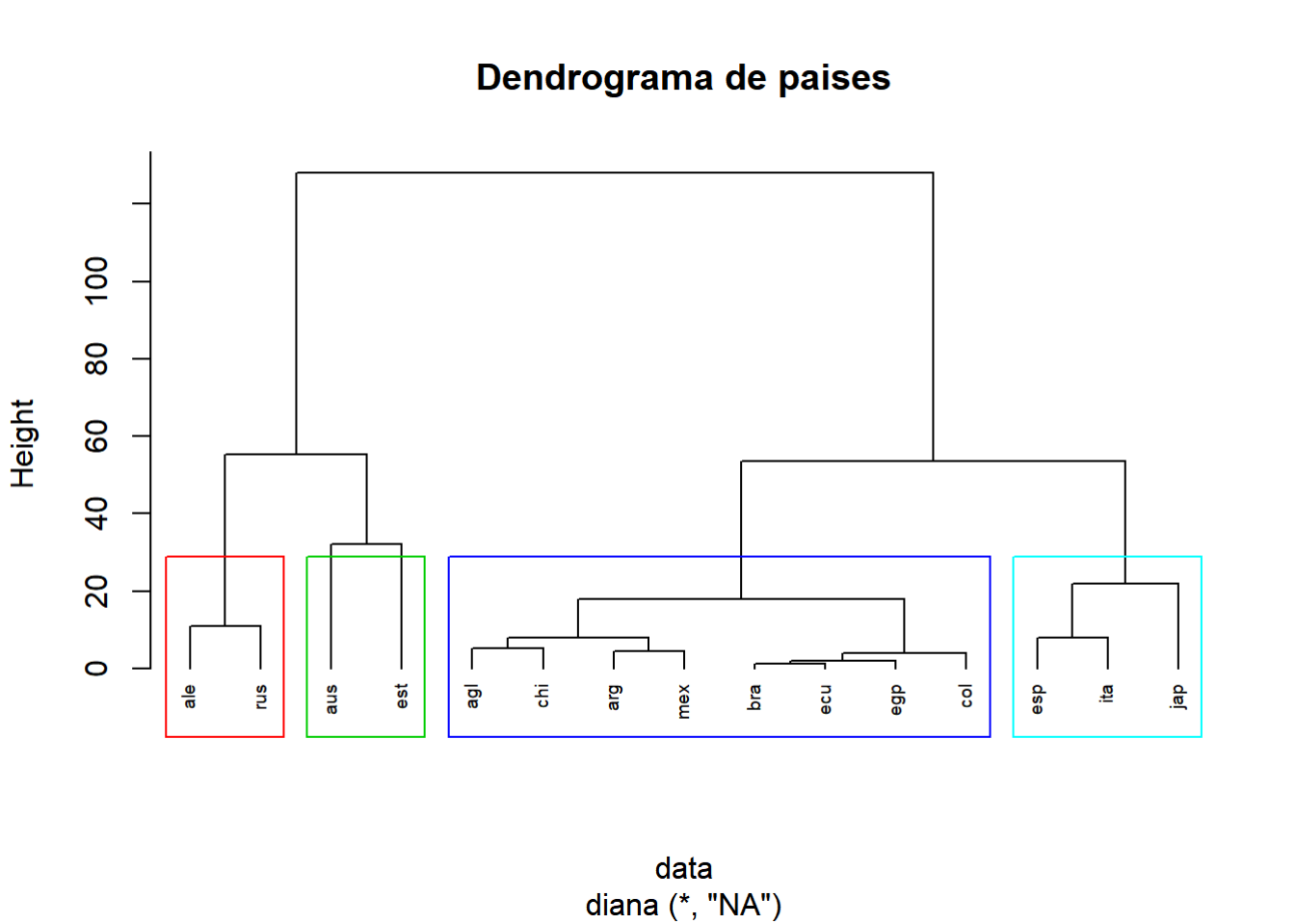

**Global Trend**

# Time-Series Clustering Statistical Approach – Non parametric Methods

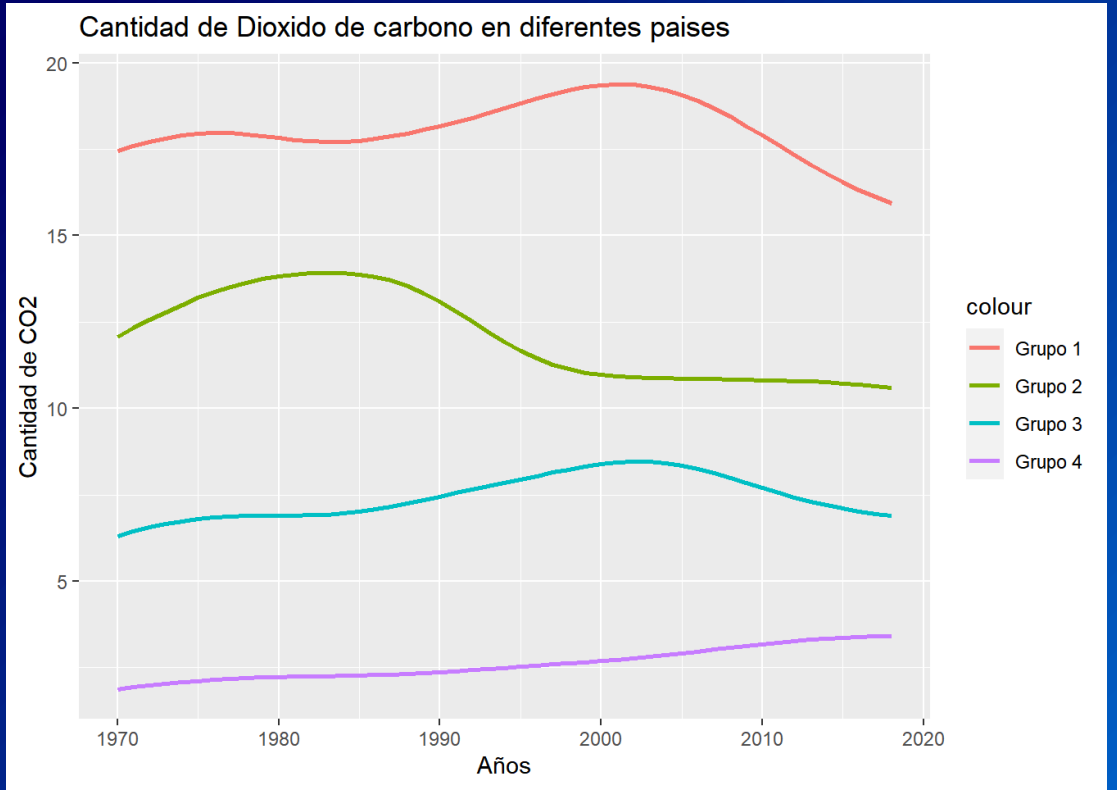



**Global Trend versus each of the Time-Series (smooth trends)**

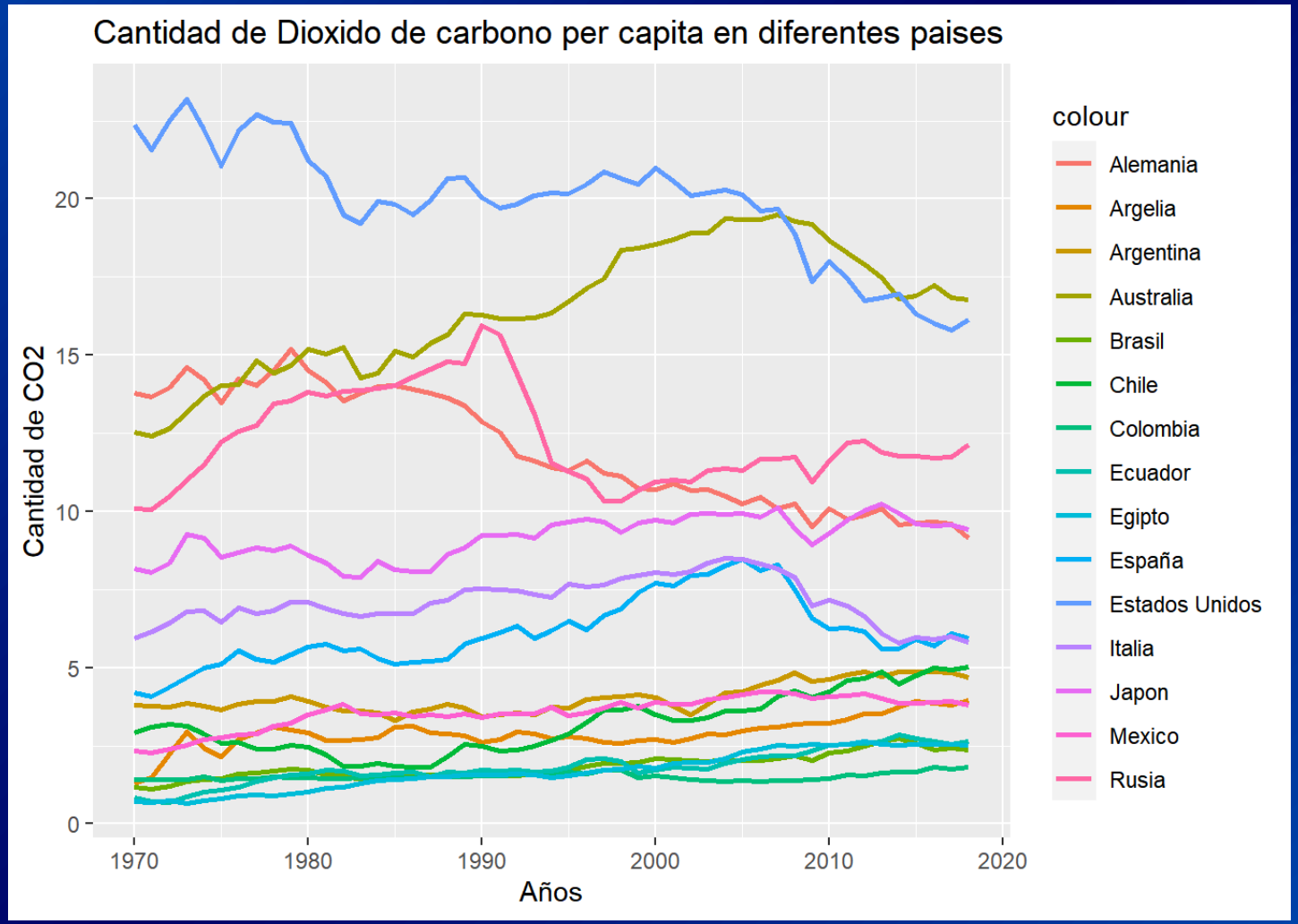
# Time-Series Clustering Statistical Approach – Non parametric Methods




## Clustering by using a smoothing representation of the trends and Profiling



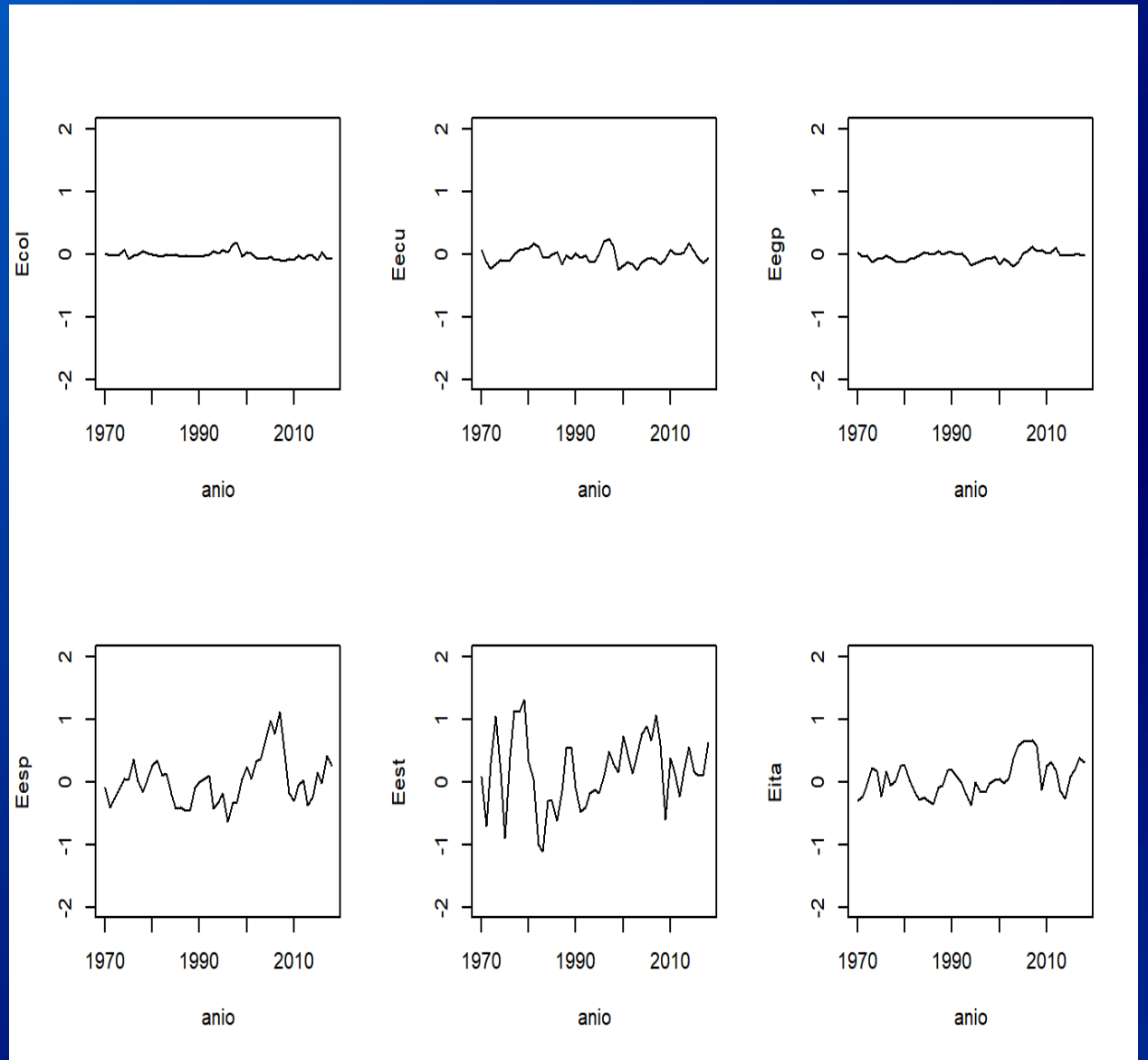
# Time-Series Clustering
# Statistical Approach (Extraction of random component – Non parametric Methods

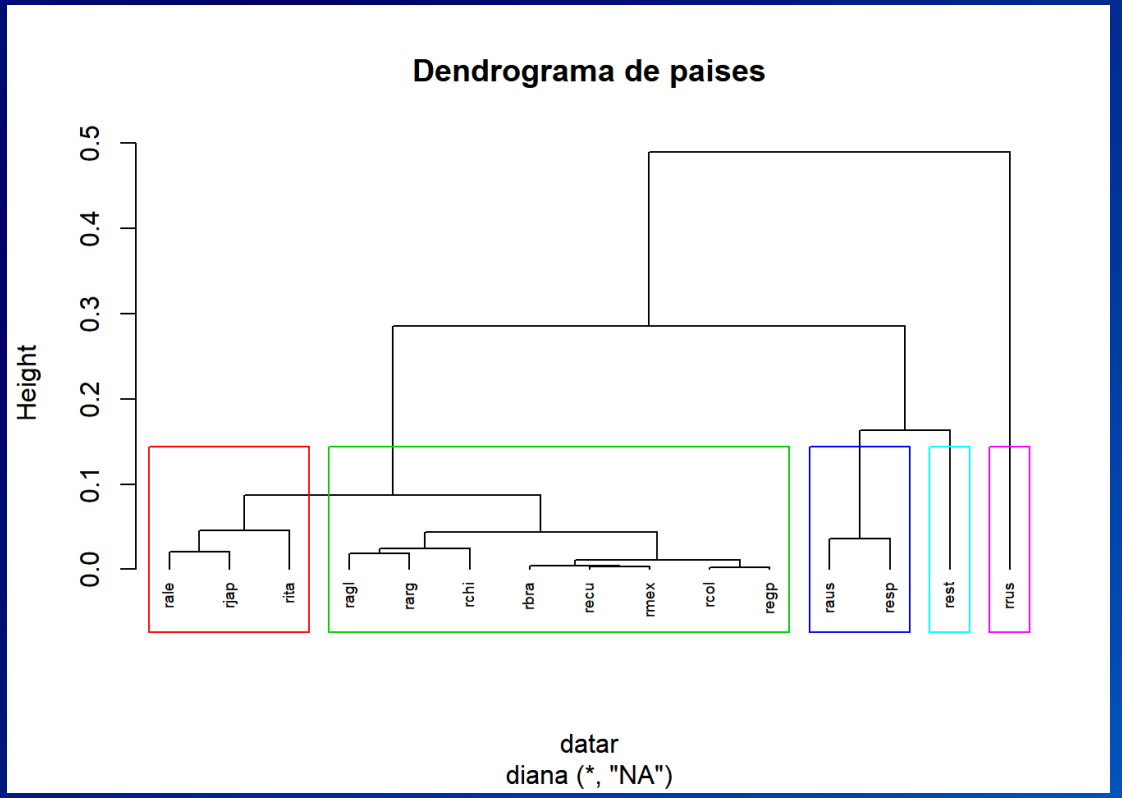



https://rpubs.com/Edison-D/615477 (Case study in Spanish language)

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

# Time-Series Clustering Statistical Approach (Extraction of random component – Non parametric Methods




UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

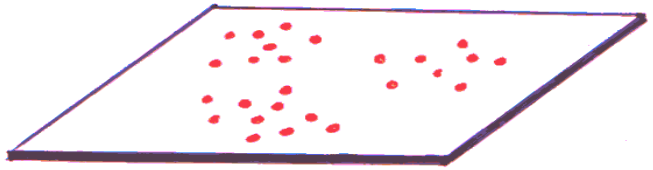
FIB UPC UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

# Advanced Clustering – Next Steps...

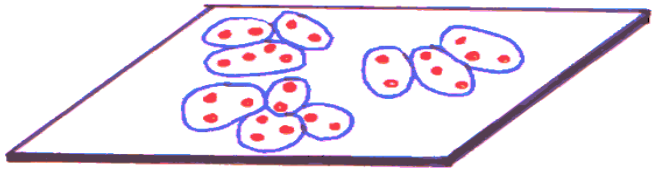
# *Clustering very large data sets*

1. Perform *m* (=2 or 3) times a k-means algorithm (with a large value for K, K≈14)
2. Form the crosstable of m obtained partitions
3. Calculated the centroids of the (non empty) cells of the crosstable
4. Perform a Hierarchical Clustering of the centroids weighted with the number of individuals per cell
5. Decide the number of classes present in your data
6. Consolidate your clustering

# *Clustering very large data sets ( CURE as STRATEGY)*

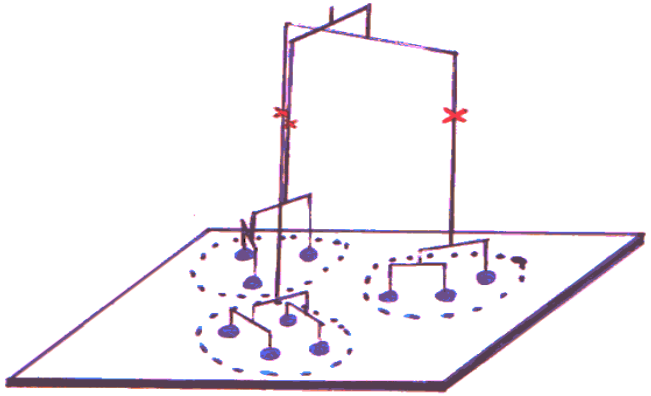


DATOS A CLASIFICAR



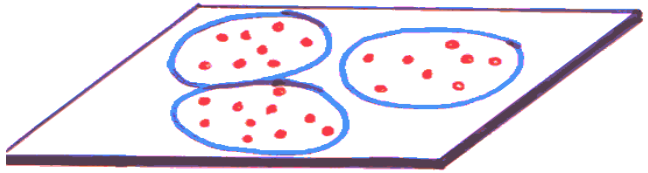
ETAPA 1

PARTICION CON UN NUMERO
MUY GRANDE DE CLASES



ETAPA 2

CLASIFICACION JERARQUICA
DE LOS CENTROS DE LAS CLASES
Y CORTE DEL ARBOL



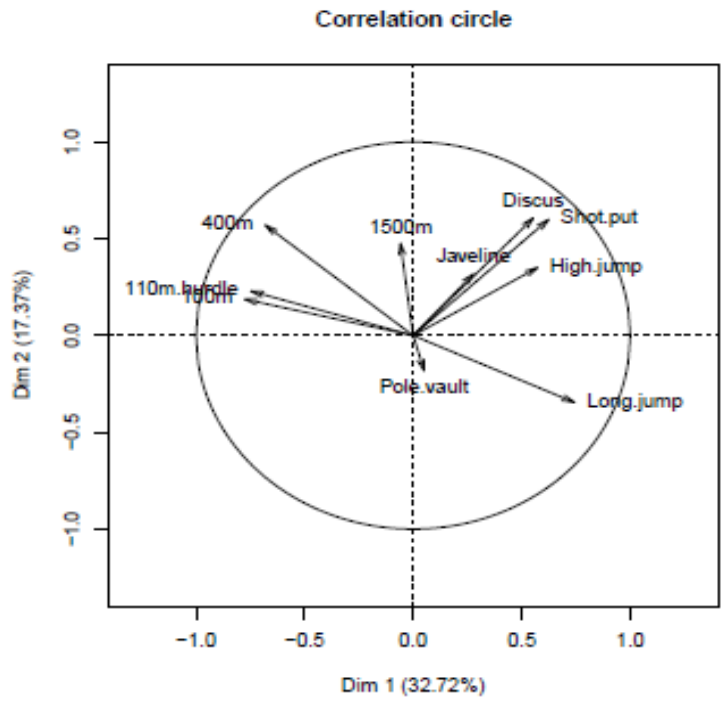
DATOS CLASIFICADOS

# *Clustering large data sets in R*

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

```
# CLUSTERING OF LARGE DATA SETS
# FIRST 2 KMEANS WITH K=14

n1 = 14
k1 <- kmeans(Psi,n1)
k2 <- kmeans(Psi,n1)

table(k2$cluster,k1$cluster)
clas <- (k2$cluster-1)*n1+k1$cluster

freq <- table(clas)    # WHAT DO WE HAVE IN VECTOR freq?
cdclas <- aggregate(as.data.frame(Psi),list(clas),mean)[,2:(nd+1)]

# SECOND HIERARCHICAL CLUSTERING UPON THE CENTROIDS OF CROSSING THE 2 KMEANS PARTITIONS

d2 <- dist(cdclas)
h2 <- hclust(d2,method="ward.D2",members=freq)   # COMPARE THE COST

plot(h2)
barplot(h2$height[(nrow(cdclas)-40):(nrow(cdclas)-1)])   # PLOT OF THE LAST 39 AGGREGATIONS

nc = 7    # for instance
c2 <- cutree(h2,nc)
cdg <- aggregate((diag(freq/sum(freq)) %*% as.matrix(cdclas)),list(c2),sum)[,2:(nd+1)]   # WHY WEIGHT

# CONSOLIDATION

k6 <- kmeans(Psi,centers=cdg)
Bss <- k6$betweenss
Wss <- k6$tot.withinss)

Ib6 <- 100*Bss/(Bss+Wss)
Ib6
```

# Advanced Clustering – Clust of Var



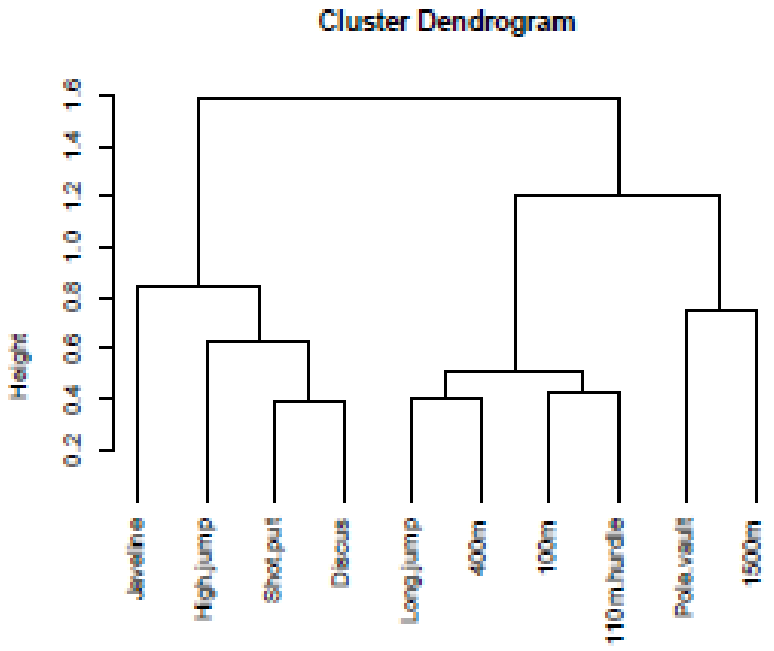




Figure 1: Correlation circle of the two first PCA dimensions.

FIB UPC UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

# Advanced Clustering – Density based clustering (DBSCAN & OPTICS)