Training Fiche Template

Title	Cluster Analysis	
Keywords (meta tags)	Statistical units, cluster, intra-cluster, inter-cluster, dissimilarity index, merge distance, dendogram.	
Language	English	
Objectives / Goals / Learnig outcomes	**The aim of this module is to introduce and explain the technique of Cluster Analysis.** **At the end of this module you will be able to:** - **Know the logic of Cluster Analysis** - **Know the requirements** - **Conduct a Cluster Analysis**	
Training course:		
Data Science Literacy		
Data Visualisation and Visual Analytics Module		X
Introduction to Data science for Human & Social Sciences		
Data Science for good		
Data Journalism and Storytelling		
Description	In this training module you will be presented the multidimensional analysis technique called Cluster Analysis, also called automatic group analysis. Cluster analyses are used to group statistical units that have characteristics in common and assign them to categories not defined a priori. The groups that are formed must be as homogeneous as possible inside (intra-cluster) and heterogeneous outside (inter-cluster). The application of this type of analysis is manifold: computer science, medicine, biology, marketing. The last part of the module will be dedicated to the application of cluster analysis with the R software.	

Co-funded by the Erasmus+ Programme of the European Union

With the support of the Erasmus+ programme of the European Union. This document and its contents reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Contents arranged in 3 levels

1. INTRODUCTION

Cluster analyses are used to group statistical units that have characteristics in common and assign them to categories not defined a priori. The groups that are formed must be as homogeneous as possible inside (intra-cluster) and heterogeneous outside (inter-cluster).

Cluster analyses are procedures that essentially consist of four phases:

- Choice of variables
- Data collection
- Data processing
- Verify and use results

2. CLUSTERED ANALYSIS REQUIREMENTS

Several types of variables can be used in cluster analysis:

- Descriptive variables (example: demographic, socio-economic, geographical)
- Behavioral variables (i.e. those variables that answer questions: what, when, where, how and why)

So let's talk about both qualitative and quantitative variables.

The sample available for cluster analysis shall be sufficiently numerous, identifiable, stable enough, easily accessible and sufficiently profitable.

3. How to Conduct Cluster Analysis

3.1 Dissimilarity matrix (or Distance matrix), D

We start from our **X data matrix**, with nxp dimensions, and transform it into a **D dissimilarity matrix**, with nxn dimensions. This last is useful to know how many statistical units are different from each other and therefore useful to choose which variables should be considered in the analysis.

$$\mathbf{X} = \begin{pmatrix} x_{1,1} & & x_{1,p} \\ & x_{i,k} & \\ x_{n,1} & & x_{n,p} \end{pmatrix} \Longrightarrow \mathbf{D} = \begin{pmatrix} d_{1,1} & & d_{1,n} \\ & d_{i,j} & \\ d_{n,1} & & d_{n,n} \end{pmatrix}$$

As we can see the matrix **D** is a symmetric matrix that along the major diagonal has all 0, since the distance of a point with itself is zero.

To calculate the distances between the points is used the index $d_{i,j}$, i.e. the measure of the degree of similarity between i and j,.

There are different indices to be able to calculate these distances, depending on the type of variable you are using.

3.2 Distances

- When using **quantitative variables** we refer to the **degree of dissimilarity**, there are several ways to calculate it:
  Euclidean Distance:
  It refers to the Pythagorean theorea, it turns out to be sensitive to outlier. It is calculated:

$$d_{i,j} = \left[\sum_k (x_{i,k} - x_{j,k})^2 \right]^{\frac{1}{2}}$$

  Manhattan distance:
  Also called City Block, it turns out to be more robust than the Euclidean distance and therefore when possible it is preferred to use this. It is calculated:

$$d_{i,j} = \sum_k |x_{i,k} - x_{j,k}|$$

In the calculation of distances the units of measurement of the variables are always taken into account. The effect of the measurement can be eliminated through the standardization of the **X matrix** in the **Z matrix**, which will be given by:

$$Z_k = \frac{(\boldsymbol{X}_k - \boldsymbol{M}_k)}{\boldsymbol{S}_k}$$

Once the matrix is standardized, of course, we will use it to calculate the dissimilarity index. The distance to Manhattan will be:

$$d_{i,j} = \sum_{k} \frac{1}{S_k} |z_{i,k} - z_{j,k})|$$

Where $\frac{1}{S_k}$ is the weighting.

Standardization is performed if we want to give all variables the same weight; If, on the other hand, it is considered appropriate that a variable should have a greater weight than the others, then standardization will not be carried out.

- When using **Binary variables**, that is**,** variables that have only two modes (when we talk about modes it means that the variables available to us are **qualitative variables**). Binary variable modes are assigned status 0 and 1. With this type of variables we calculate **the degree of similarity,** i.e. the similarity between i and j.
  Binary variables are divided into:
  Simmetric variables BS, **BS:** the two states (0 and 1) have the same importance.
  Asymmetric Binary Variables, BA: more importance is given to state 1 than to state 0.

  Zubin index:
  It is used for **variables binarie simmetric**, it is calculated by adding the concordance frequencies and the discordance frequencies, then it is divided by the total.

  $$s = \frac{(a+d)}{p}$$

  Jaccard index:
  It is used for **asymmetric binary variables**, it is calculated by dividing the concordance frequency by the difference between the total and the discordance frequency.

  $$s = \frac{a}{(p-d)}$$

Co-funded by the
Erasmus+ Programme
of the European Union

3.3 Types of Clusters

There are different types of clusters depending on the approach you want to use in creating groups.

Hierarchical algorithms perform successive mergers or divisions of data, once an object has joined a cluster its assignment is irrevocable.

- **Agglomerative or aggregative clusters (bottom-up)**:
  The goal is to group the many clusters and obtain a single claster that contains all those present from the beginning.

- **Split clusters or splitters (top-down):**
  In this case we start from a single cluster and the ultimate goal is to divide it into many clusters.

3.4) Types of Links between Statistical Units

Clusters can be formed through different types of links:

- **Single** or simple linkage
- **Complete** linkage
- **Average** linkage

The **simple linkage** uses the technique "of the nearest neighbor". The degree of proximity between two groups is established taking into account the minimum minimum distance between the points. In other words, you take into account the units that are closest to each other. This link, however, despite being the fastest to achieve at the computational level, creates groups that are too homogeneous between them.

The **complete linkage** uses, instead, the technique of the "farthest neighbor". Considers the similarities / distances between the most distant groups (therefore those less similar to each other). In practice, the minimum maximum distance between the points takes into account. This link, despite being the slowest from a computational point of view, creates very heterogeneous groups on the outside and homogeneous on the inside.

The **average link** in the creation of clusters uses the minimum average distance. In practice first the average distance between all observations is calculated and then the smallest one is taken into account. This binding is also slow from a computational point of view but it is robust, it is less sensitive to outliers.

Ward link can be used with quantitative data. This technique minimizes variance within groups by homogenizing them, in practice this method maximizes internal homogeneity (or minimizes internal heterogeneity) and maximizes external heterogeneity.

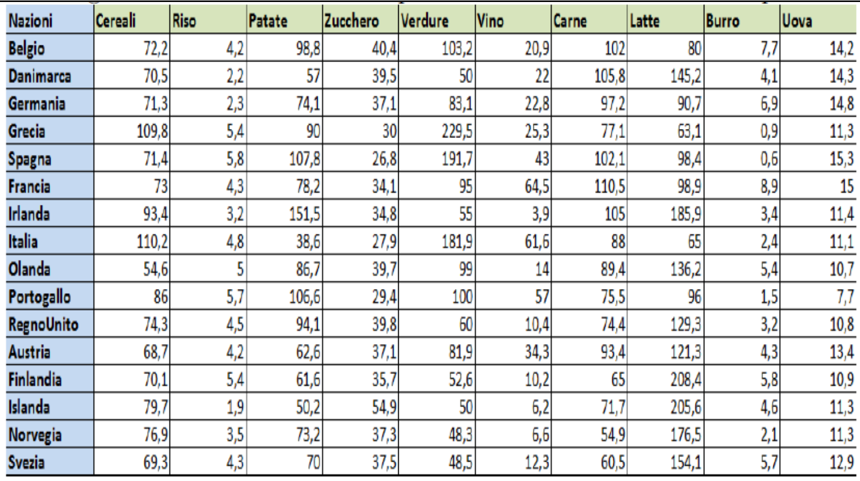
3.5 Dendogram and Melting Distance

Once the link that best represents the data in our possession has been chosen, the **dendogram** will be obtained. We can visualize through a **tree graph** how the statistical units have been distributed. At each step the distance between the clusters tends to increase and therefore it is necessary to choose a **stop rule.** This rule allows us to choose the number of groups we want to get. You can use the tree cutting technique through the graph of **blending distances** (or heights), which indicates where clusters are created. Graphically we observe the point at which we register a greater surge. This part will then be taken up in the part of the module dedicated to the R software.

4. Example with R software

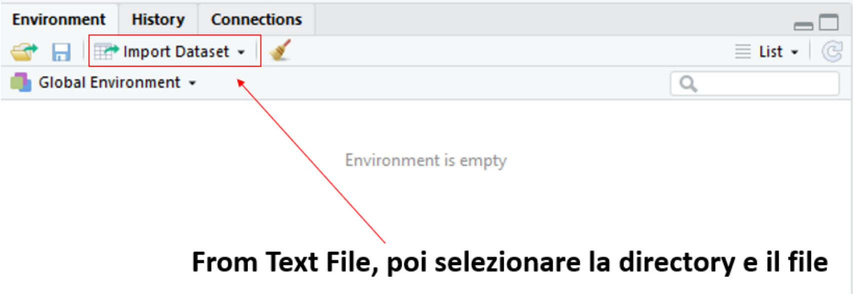
The Cluster analysis aims to identify the best possible distribution, in terms of number and composition, of a set of elements in groups so that these are: as homogeneous as possible within them and as different as possible from each other. These constructions can be carried out both according to the choice of grouping strategies, and in relation to the criterion chosen for the measurement of similarity/dissimilarity.

Dataset:



Nazioni	Cereali	Riso	Patate	Zucchero	Verdure	Vino	Carne	Latte	Burro	Uova
Belgio	72,2	4,2	98,8	40,4	103,2	20,9	102	80	7,7	14,2
Danimarca	70,5	2,2	57	39,5	50	22	105,8	145,2	4,1	14,3
Germania	71,3	2,3	74,1	37,1	83,1	22,8	97,2	90,7	6,9	14,8
Grecia	109,8	5,4	90	30	229,5	25,3	77,1	63,1	0,9	11,3
Spagna	71,4	5,8	107,8	26,8	191,7	43	102,1	98,4	0,6	15,3
Francia	73	4,3	78,2	34,1	95	64,5	110,5	98,9	8,9	15
Irlanda	93,4	3,2	151,5	34,8	55	3,9	105	185,9	3,4	11,4
Italia	110,2	4,8	38,6	27,9	181,9	61,6	88	65	2,4	11,1
Olanda	54,6	5	86,7	39,7	99	14	89,4	136,2	5,4	10,7
Portogallo	86	5,7	106,6	29,4	100	57	75,5	96	1,5	7,7
RegnoUnito	74,3	4,5	94,1	39,8	60	10,4	74,4	129,3	3,2	10,8
Austria	68,7	4,2	62,6	37,1	81,9	34,3	93,4	121,3	4,3	13,4
Finlandia	70,1	5,4	61,6	35,7	52,6	10,2	65	208,4	5,8	10,9
Islanda	79,7	1,9	50,2	54,9	50	6,2	71,7	205,6	4,6	11,3
Norvegia	76,9	3,5	73,2	37,3	48,3	6,6	54,9	176,5	2,1	11,3
Svezia	69,3	4,3	70	37,5	48,5	12,3	60,5	154,1	5,7	12,9

We import the dataset:



From Text File, poi selezionare la directory e il file

In the ***row names*** select the wording: "***use first column***" in order to have the labels of both individuals and variables on the graphs.

In the ***decimal*** we select: "***comma***".

With the command:

X<-as.matrix(nome_del_dataset)

Awe attribute to **X**, as an object, the dataset used in the analysis.

We standardize the **X** matrix:

Z<-scale(X)

Next we calculate the distance between the elements, we can use either the Euclidean distance or the Manhattan distance.

Respectively the commands are:

```
d<-dist(Z)

D<-round(D,2)

d_m<-dist(Z, method="manhattan")

d_m<-round(d_m, 2)
```

NB: the round command allows us to round up to the significant figure we prefer, in this case to the second.

Then we move on to the choice of the link between the elements.

Let's start with **the single linkage**:

```
hc_s<-hclust(d,method="single")
```

We can display a **summary of the results of** the single bond with the command:

```
summary(hc_s)
```

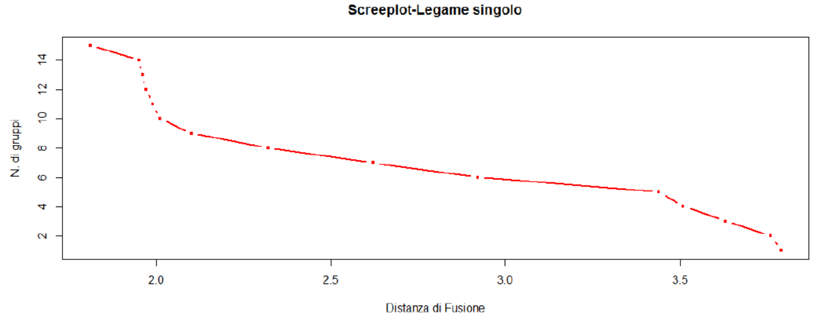
We can visualize the **dendogram** with the plot function:

```
plot(hc_s)
```

To decide where to cut the tree graph you use the **cutree** command. The choice of how many groups to get by displaying the melting point through the **scree-plot** of the melting distance linkage. The commands are:

```
n<-nrow(X)

n_clus<-seq(n-1,1)

hc_s$merge

hc_s$height

d_fus_s<-hc_s$height

plot(d_fus_s,n_clus,"b", main="Screeplot Single bond", xlab="Melting Distance", ylab="Number of groups",cex=0.6, col="red",lwd=2.5)
```

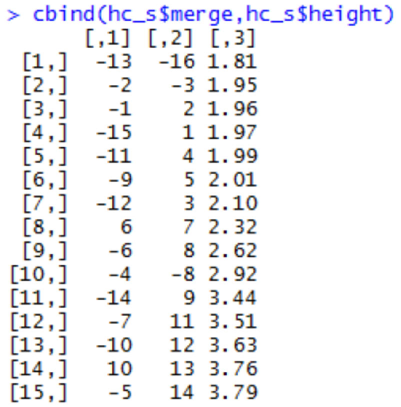
Graphically:



Now you want to see the melting points (hc_s$merge) and heights (hc_s$height), to be able to visualize them together you use **cbind**. The command $merge shows, for each step, of the grouping algorithm, the pair of elements merged according to the chosen link. Values preceded by "-" indicate the single element, while positive values represent clusters formed in previous steps.

Thus, at the first step, the first cluster will be formed consisting of the pair (13, 16), corresponding to the Finland and Sweden models, while the third cluster (step 10) will be formed by the elements of cluster 2 (Greece, Italy) plus element 1 (France). The $height field shows the distance considered for the merger between elements/groups.

cbind(hc_s$merge,hc_s$height)



```
> cbind(hc_s$merge,hc_s$height)
       [,1] [,2] [,3]
  [1,]  -13  -16 1.81
  [2,]   -2   -3 1.95
  [3,]   -1    2 1.96
  [4,]  -15    1 1.97
  [5,]  -11    4 1.99
  [6,]   -9    5 2.01
  [7,]  -12    3 2.10
  [8,]    6    7 2.32
  [9,]   -6    8 2.62
 [10,]   -4   -8 2.92
 [11,]  -14    9 3.44
 [12,]   -7   11 3.51
 [13,]  -10   12 3.63
 [14,]   10   13 3.76
 [15,]   -5   14 3.79
```

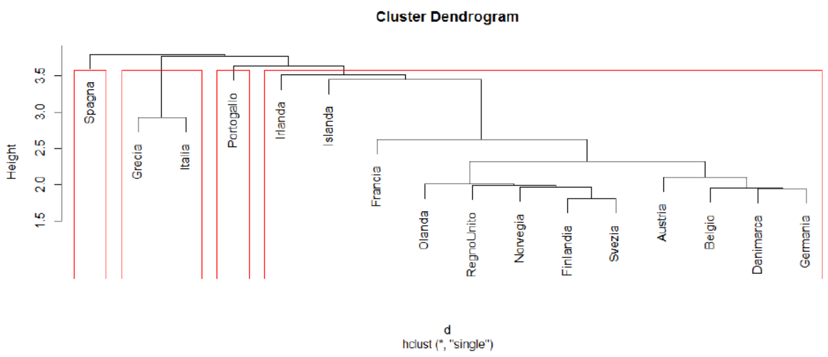
For cutting the tree graph we use the cutree command, at k we put the point where the melting distance takes a horizontal trend:

groups <- cutree(hc_s, k=4)

plot(hc_s)

```
rect.hclust(hc_s, k=4, border="red")
```

The dendogram will be:



We can say that this type of bond is not good, because there are clusters that contain single elements and a cluster that is too homogeneous within it.

We proceed with the other links in the same way.

Complete linkage:

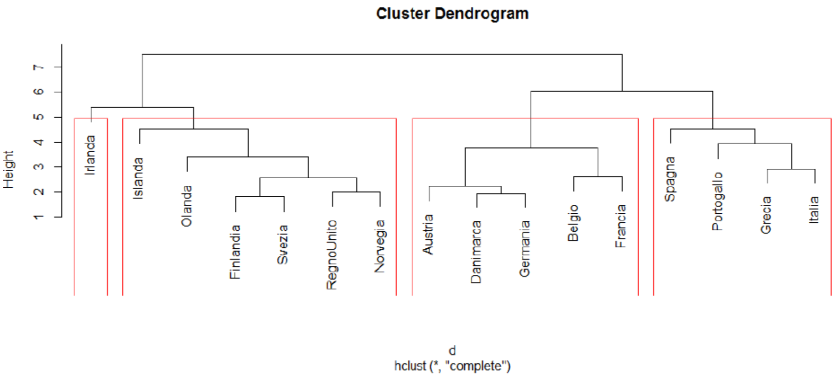
```
hc_c<-hclust(d,method="compl")

summary(hc_c)

plot(hc_c)

n<-nrow(X)

n_clus<-seq(n-1,1)

hc_c$merge

hc_c$height

d_fus_c<-hc_c$height
```

Screeplot of melting distances for the Full bond:

```
plot(d_fus_c,n_clus,"b", main="Screeplot Full Bond", xlab="Melting Distance", ylab="N. of groups",cex=0.6, col="red",lwd=2.5)

cbind(hc_c$merge,hc_c$height)
```

Cutting the tree graph for the Complete linkage, to k we will attribute the figure according to the screeplot of melting distances:

```
groups <- cutree(hc_c, k=4)

plot(hc_c)

rect.hclust(hc_c, k=4, border="red")
```



Average Linkage:

```
hc_a<-hclust(d,method="average")

summary(hc_a)

plot(hc_a)

n<-nrow(X)

n_clus<-seq(n-1,1)

hc_a$merge

hc_a$height

d_fus_a<-hc_a$height
```

Screeplot of melting distances for the average linkage:

```
plot(d_fus_a,n_clus,"b", main="Screeplot Mean bond", xlab="Melting Distance", ylab="N. of groups",cex=0.6, col="red",lwd=2.5)
```

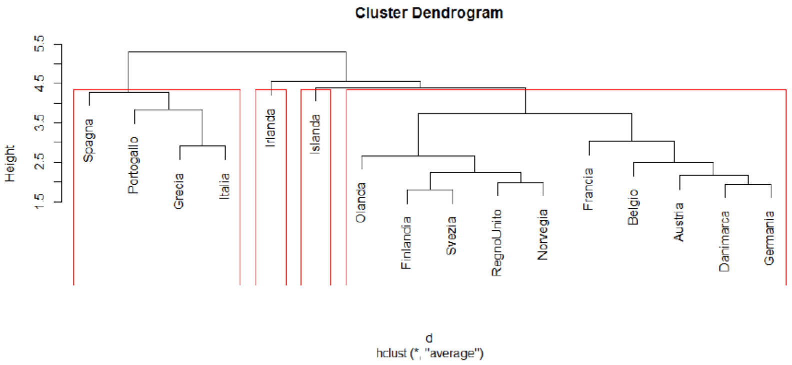
```
cbind(hc_a$merge,hc_a$height)
```

Cutting the shaft for the average linkage, to k we will assign the figure according to the screeplot of melting distances:

```
groups <- cutree(hc_a, k=4)
```

```
plot(hc_a)
```

```
rect.hclust(hc_a, k=4, border="red")
```



Self-assessment (multiple choice queries and answers)

1. The distance matrix:
    A) Has on the greater diagonal all 0
    B) Has on the largest diagonal all 1
    C) Has on the largest diagonal the distances between i and j

2. Which of these distances is more robust, or insensitive to extreme values?
    A) Jaccard index
    B) City block
    C) Euclidean Distance

3. Standardisation shall make it possible to:
    A) Eliminate higher frequencies
    B) Eliminate the effect of the unit of measurement

	C) Give different weight to variables
Resources (videos, reference link)	
Related material	
Related PPT	
Bibliography	Johnson, S. C. (1967). Hierarchical clustering schemes, Psychometrika, 32, 241-254. Pollice, A. (2013). Statistica multivariata, http://www.uniba.it/ricerca/dipartimenti/dse/dipartimento/personale/personale-docente/pollice/stat_mult/disp10.pdf Ward, J. H., Jr. (1963). Hierarchical Grouping to Optimize an Objective Function, Journal of American Statistical Association, 58, 236-244.
Provided by	[Unisalento]

Co-funded by the
Erasmus+ Programme
of the European Union

With the support of the Erasmus+ programme of the European Union. This document and its contents reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.