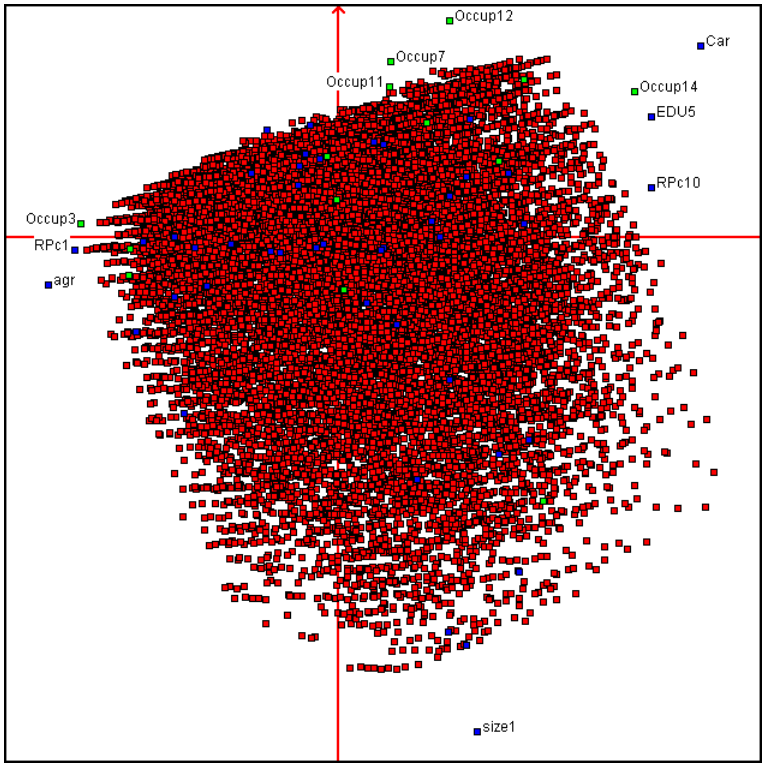
Silvio Griguolo

ADDATI for Windows

A Package for Exploratory Data Analysis
(Version 6.0 – February 2008)

User Guide



University IUAV in Venice – Dept. of Planning

SILVIO GRIGUOLO (*)

ADDATI

A Package for Exploratory Data Analysis

User Guide

(Version 6.0 – February 2008)

University IUAV in Venice – Dept. of Planning

6.1 – Geometric representation *2*
6.1.1 – The distance 3
6.1.2 – The centre of gravity of the cloud 3
6.1.3 – The Cloud's Inertia 4
6.1.4 – The interpretation of the relationships among the variables in R^n 4
6.2 – Introduction to the factorial analyses: PCA and ACORR *6*
6.3 – The Principal Components Analysis (PCA) *7*
6.3.1 – An example 9
6.3.2 - Entering the analysis control parameters 10
6.3.3 – How many Principal Components should be saved? 15
Principal Components to be used to cluster the statistical units 15
Description of the statistical units 15
Description of the variables 16
Projections onto the factorial planes 16
6.3.4 –The table of contributions and their interpretation 17
6.3.5 - Interpretation of the factors 20
6.4 - The Analysis of the Correspondences (ACORR) *22*
6.4.1 – Entering the control parameters 25
Tables of qualitative variables 25
Side-by-side contingency tables 25
6.4.2 – The table of contributions and their interpretation 26

7.1 – Some notes on numeric Classification *1*
7.1.1 –Hierarchical methods 2
7.1.2 – Non-hierarchical methods 3
Some definitions 3
7.2 – The clustering sequence in ADDATI *5*
7.2.1 – The non-hierarchical clustering method 5
7.2.2 – Non-hierarchical Clustering: the algorithm implemented in ADDATI 6
The exploratory stage 6
The Optimisation stage 7
7.3 – The NONGER dialogs *10*
7.3.1 – Controlling the exploratory stage 10
7.4 - NONGER – Optimisation stage and description of the partitions *12*
7.4.1 – Examining the profiles of the classes 14
Profile's components 15
NONGER – The interpretation of the results 15

Chapter 6. – Analysis Menu: the Factorial Analyses

6.1 – Geometric representation

Let us consider a data table ***X(n,p)***, in which the rows represent a set I of *n* units and the columns the values of *p* variables measured on those units. ***X*** is a descriptive table, and for sake of simplicity we will suppose that the variables are quantitative; however, the definitions and the concepts that will be given can be extended to any kind of table.

The behaviour of each row-unit (again, think of a district as an example) is represented by an array of *p* ordered real numbers (the values taken by the *p* variables in the unit). The *p* components of such an array can be thought as the co-ordinates of a point in a vectorial (geometrical) space $\mathbf{R}^p$ with *p* dimensions; the unit can be identified with that point.

The set I of *n* units can be represented as a **cloud of n massive points** (remember that there is a weight attached to each unit). Every other point in R^p can be thought of as a **virtual** unit (i.e., a combination of values of the *p* descriptive variables) that might possibly be encountered in another case, or another sample, or that can have a particular meaning for the given cloud, like its central point.

The Figure 6.1 gives an example in a simple case with two variables.

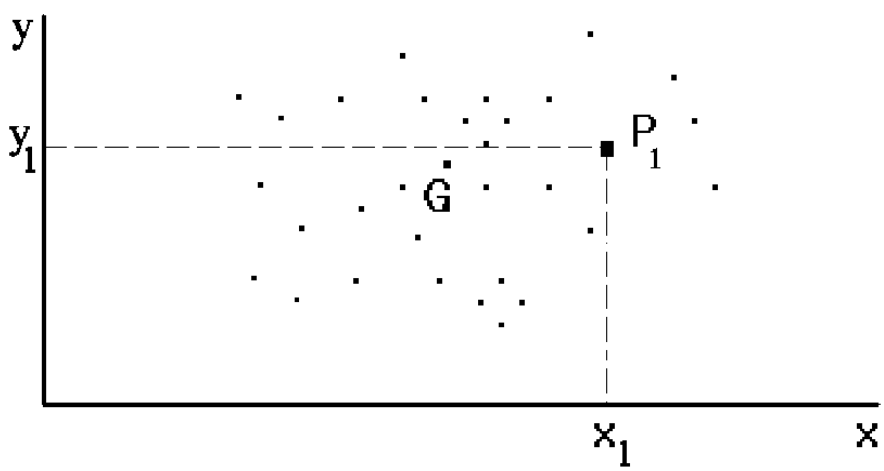


Figure 6.1 - Geometrical representation of a set of units described by two variables. G is the cloud's centre of gravity.

In the figure each of the two orthogonal axes carries the values of a variable; each geometrical point identifies unequivocally an array of two values (i.e., a unit) and vice versa. If the variables were three, three orthogonal axes would be necessary to represent them and they could still be visualised easily. When the variables are more than three, our three-dimensional mind cannot visualise the picture; however, all the mathematics that can be conceived for handling the two- or three-

dimensional case can be extended with no effort to the *p*-dimensional case. We shall therefore consider the general case of a cloud of *n* object-points in R^p, but you can focus intuitively on the representation with two dimensions without any loss of generality.

In a similar way, we can look at the table ***X*** *by columns*. Each column is an array of *n* numbers that represent the values assumed by a variable over the *n* units. It can be identified with a geometrical point in an *n*-dimensional space R^n. In this case, we have a cloud of *p* variable-points in R^n.

The table admits therefore two geometrical representations, respectively as a cloud of *n* unit-points in R^p or of *p* variable-points in R^n. As for their information contents, they are both perfectly equivalent to the numeric description given by the table **X**. It seems natural to focus on the cloud of *n* unit-points in R^p for analysing the differences existing amongst the statistical units with respect to the descriptive variables, but also the other representation could be used, and actually is, when there is a good computational reason.

The two spaces R^p and R^n are dual. It is generally convenient to study in R^p the relationships among the units (e.g., two units globally similar with respect to the *p* variables are represented by points in R^p close to one another, etc.) and focus on the cloud in R^n to study the relationships amongst the variables (two uncorrelated variables are represented by points that lie in orthogonal directions with respect to the origin, while two highly correlated variables lie in directions that form a small angle, etc.).

6.1.1 – The distance

We will assume as a ***global indicator*** of the ***dissimilarity*** between two statistical units the distance between the two points that represent them in R^p: all variables contribute to its determination. In the case of a quantitative descriptive table the *global dissimilarity* between units i and k is computed as

$$d^2(i,k) = (x_{i1}-x_{k1})^2 + \ldots + (x_{ip}-x_{kp})^2$$

using the Euclidean distance between point *i* and *k* in R^p (we will adopt a different metrics - i.e., *a different definition of distance* - in the case of a contingency table).

As the contributions to the distance coming from different variables are to be added up, they must be expressed in the same unit of measure, or be a-dimensional. It is also convenient to balance the contributions of different variables to the distance, so as to avoid the dominance of one or of a few of them, caused only by the unit of measure adopted. A way to obtain this is to standardise all the variables before analysing: this is performed automatically by the program. Once normalised (i.e., ***centred*** by subtracting from each value the variable's average, and ***reduced*** by dividing the values so obtained by the variable's standard deviation) The average of all variables is zero, and their variance is equal to 1.

6.1.2 – The centre of gravity of the cloud

The centre of gravity of the cloud of unit-points in R^p is the point **G** whose co-ordinates are the average values of the *p* variables. It represents a *virtual* unit with the system's overall average behaviour. Obviously, if the variables are centred (i.e., the value of the average is 0 for all of them), the centre of the cloud coincides with the origin of the reference system (**G ≡ O**); the cloud is said to be **centred**.

The analysis we wish to perform is focused on the differences existing amongst the *n* units, and to which variables these differences can be ascribed: from a geometrical point of view, we want to observe ***how much*** and ***in which way*** each unit differs from the average behaviour of the set I, represented by the cloud's centre **G** (or by the origin **O**, if the cloud is centred). It is reasonable to assume as an indicator of "*how much*" the **distance** of each unit-point from **G,** and to associate "*in which way*" with the direction of such elongation (i.e., to the variables which more contribute to that distance).

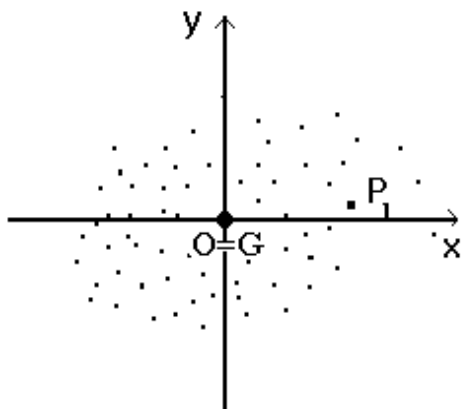


Figure 6.2 – The cloud of figure 6.1 centred.

6.1.3 – The Cloud's Inertia

Suppose the cloud to be centred. We call ***Inertia of the unit i*** with respect to the centre **G** ≡ **O** the product of the mass of i by the square of its distance from **O**:

$$\text{Inertia}\ (\ i\ )\ =\ m_i\ d^2(\mathbf{x_i},\mathbf{O})\ =\ \Sigma_j\, m_i\, x_{ij}^{\ 2}$$

As a measure of the cloud's dispersion we assume the ***total Inertia*** $\text{In}_{tot}(\ I\ )$ of its points.

$$\text{In}_{tot}(\ I\ ) = \Sigma_i\ m_i\ d^2(\mathbf{x}_i,\mathbf{O})$$

The Inertia of the cloud has a simple interpretation: it arises from the units' difference of behaviour, i.e. from the fact that the variables assume different values in different units, and have therefore a non-null variance in I. Were this not the case, the cloud would collapse onto its centre and its Inertia would be 0.

It is easy to verify that ***the total Inertia is equal to the sum of the variances of the p variables***

$$\text{In}_{tot}(\ I\ ) = \Sigma_i\ m_i\ d^2(\mathbf{x}_i,\mathbf{O}) = \Sigma_i m_i(\Sigma_j x_{ij}^{\ 2}) = \Sigma_j(\Sigma_i m_i x_{ij}^{\ 2}) = \Sigma_j \mathbf{var}(j)$$

In particular, if the p variables are standardised the contribution of each of them to the Inertia is 1, and $\mathbf{In_{tot} = p}$***.***

6.1.4 – The interpretation of the relationships among the variables in R^n

In R^n every point can be interpreted as a variable, i.e. an array of *n* values (its co-ordinates) measured on the *n* units. If the variables are centred, it can be proved that:

- **the distance of a variable-point *j* from the origin is equal to the variance of that variable**. Therefore, if the variables are standardised all the points representing them lie on the surface of a hypersphere centred on the origin and with radius 1;
- the correlation between two centred variables ***j*** and ***k*** is equal to the cosine of the angle formed by the segments that join their representative points with the origin. Remember that the correlation of two variables is a measure of the strength of their association over the set I: it varies between +1 (perfect positive association) and -1 (perfect negative association). If the variables are standardised, two variables having correlation +1 are represented by coincident points; two variables having correlation -1 are represented by points opposite with respect to the origin.

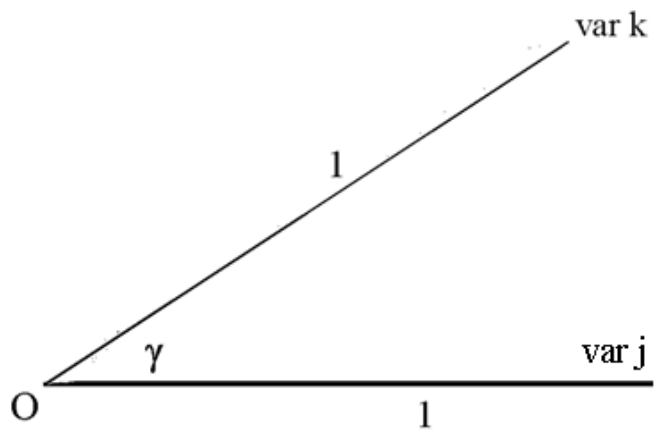


Figure 6.3 – Representation of variable-points in R^n. It can be proved that

cos γ = *corr*(var j, var k)

i.e., the cosine of the angular distance between the two variable-points *j* and *k* measures the correlation between the two variables.

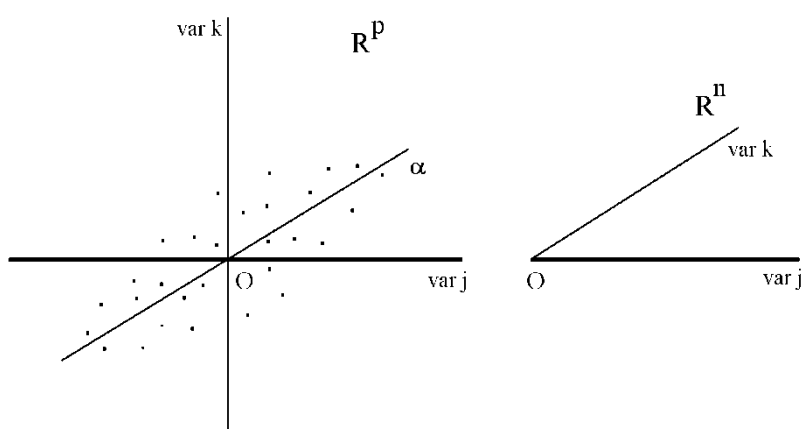


Figure 6.4 - The cloud in the two spaces R^p and R^n. In R^p the cloud is centred and its projection onto any axis passing through O (e.g., the axis α) is also centred. The two variables *j* and *k* are highly correlated. This can be seen from the cloud's shape in R^p and from the small angle between the two variable-points in R^n.

The properties of the two clouds (statistical units, variables) are therefore different: if the variables are centred, the origin in R^p is the centre of the cloud of the units. They are scattered about it, the level of the dispersion being measured by the cloud's Inertia. If the cloud is projected onto any axis passing through the origin - not necessarily a co-ordinate axis - the resulting uni-dimensional cloud is also centred (see figure 6.4).

In the other space, as the angular distance between two points is related to the correlation of the variables they represent, the cloud is unevenly distributed around the origin: if all variables are highly positively correlated, the cloud lies on the same side of **O**, without any symmetry. This difference, that affects the interpretation of the analytical results in the two spaces, is a consequence of the different meaning of the rows and columns of the table and of the non-symmetry of the treatment to which the table is submitted (the average is computed by column and not by row, columns are standardised, etc.).

6.2 – Introduction to the factorial analyses: PCA and ACORR

This is not the right place for a detailed description of the statistical theory on which the two Factorial Analyses included in ADDATI (the Principal Components Analysis and the Analysis of the Correspondences) are based. These methods are very interesting and useful, but are well beyond the limits of a User Guide. However, a user that wants to master these powerful statistical techniques as exploratory tools, not just as a mean to reduce the dimensionality of the description before clustering the units, should spend some time having recourse to some specific textbook.

The **PCA** and **ACORR** are quite similar. Both accept as input a data table (also a very large one) and explore the relationships among its elements (rows and columns). The purpose is to simplify the representation by recognising (i.e., by suitably *constructing*) a **limited number of new underlying variables** (called **factors**) sufficient to summarise the more relevant aspects of the description, with a tolerable loss of details. This is obtained by rotating, in an optimal way with respect to the cloud, the reference system of the geometrical space where the phenomenon is represented. Remember that, according to the preceding section, every row and column of the table (respectively, statistical units and variables) can be represented as points in a suitably defined geometrical space.

The difference between the two analyses stem from the nature of the table processed:

- a **table of quantitative or binary variables** in the case of a **PCA**;
- a **contingency table** or a **table of categorical variables** for **ACORR.**

Both techniques carry out a preliminary transformation of the data table, different in the two cases.

We will describe in detail the parameters necessary for the correct control of a **PCA**, and how they are entered. **ACORR** puts more or less the same questions, so in that case the description will be more concise.

6.3 – The Principal Components Analysis (PCA)

Use	The program performs a Principal Components Analysis of a data table consisting of ***quantitative*** and/or ***binary categorical*** variables. Binary categorical variables, i.e. categorical variables with **exactly two categories**, are recoded *on the fly* converting each of them into a new variable, having value 1 for units that take the first category, and value 0 for the others (whatever the codes used for the original categorical variable). It can be proved that such variables can be processed with ACOMP. All input variables are normalised by the program: each of them will have mean 0 and standard deviation 1, and each of them will have the same importance in the analysis. The user can set the program so as to by-pass this normalisation step and diagonalise the matrix of covariances instead of using the table of correlations, which is the most frequent option. This decision should be taken only by expert users and in particular circumstances: better, in general, to stick to the default settings, using the correlations matrix.
Limits	No particular limit on the number of variables, active or supplementary, that can be process. The convenience not to use too many variables stems only from the need to obtain results interpretable with a minimum of clarity. Actually, when the number of variables is increased the interpretation tends to become more and more difficult.
Advice	The variables to be analysed should be carefully chosen, trying not to invalidate the exploratory power of the method with a careless selection. Again, we remark that the analyst's skill manifests itself also by constructing ***essential tables***.
Credits	The exercise on Kenya uses a data file prepared by Annalisa Conte, former FAO Consultant, for the ADDATI Tutorial she wrote for the IGADD EWFS Project back in November 1992.

The input data table describes a set of ***n*** statistical units (the rows) by means of *p* **quantitative** or **binary** variables (the columns). By default, the variables are standardised.

Note**:** *The program performs the standardisation automatically. The values of each variable are shifted in such a way that its mean becomes 0: i.e., each value is converted into the difference with respect to its average. Besides this, all variables are so scaled that the variance of each of them becomes* 1: *because of this operation, each variable assumes the same importance in the analyst.*

Consider the figure 6.5. It shows a simplified case, with only two (standardised) variables. Each unit can be represented as a point in a two-dimensional space R^2. The shape of the cloud is a stretched oval, as the variables are strongly correlated. This means that the value of one variable (e.g., *y*) can be inferred with a good approximation when the other is known, and vice versa. The second variable repeats part of the information already conveyed by the first one: only a little part of its information is actually original (i.e., independent from the other, or non-repeated).

Let us consider the bundle of all the straight lines passing through the origin O. The cloud of unit-points can be projected onto any such line: the resulting uni-dimensional cloud is dispersed about **O**, its dispersion being measured by its **Inertia** defined in section 6.1.

In particular, owing to the fact that the original variables have been standardised, the cloud projects onto each of the two co-ordinate axes *x* and *y* with an inertia equal to 1, that exactly represents the value of each normalised variable's variance. On any other line through **O**, however, **the cloud generally projects with an inertia different from 1**.

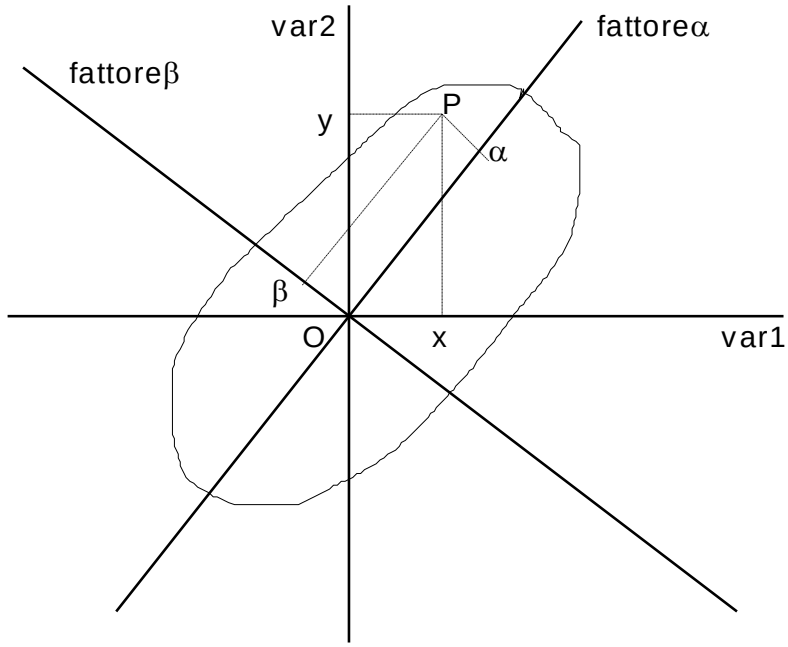


Figure 6.5 - A generic point P is represented by the pair of co-ordinates (*x*,*y*) with respect to the original variables, by the pair (α, β) on the new rotated reference system.

In general, an axis exists - indicated with α in the figure - onto which the cloud projects with a maximum inertia (we could also say: maintaining at best the distances amongst its points). This is called the **first factorial axis**, and the *signed* distance from **O** of the projection of each point is the first factorial co-ordinate of that point, or its first Principal Component.

The cloud projects onto the axis β perpendicular to α with a much smaller inertia. This completes the description when the dimensions are only two: it is easy to prove that the sum of the two inertias (on the axes α and β) **is exactly 2**, equal to the cloud's total inertia.

Every pair of orthogonal straight lines through **O** leads to a particular **decomposition of the total Inertia**. The advantage of the pair (α, β) with respect to (x, y) is that the small fraction of inertia "*explained*" by the axis β-can be easily ignored: this leads to an acceptable uni-dimensional simplification of the description of the original bi-dimensional phenomenon.

These simple considerations can easily extend to the case of *p* variables: the set of units is represented by a cloud of *n* points in a *p*-dimensional space. The value of the total Inertia, after normalising the variables, is *p*. It is always *possible - and convenient*, if at least some of the variables are sufficiently correlated - to determine an axis (called "***first principal axis***") onto which the cloud can be projected maintaining the maximum possible inertia. This amount of inertia is known as the ***eigenvalue*** associated with the axis. A second axis is then determined, orthogonal to the first that retains the maximum possible fraction of the residual inertia, and so on, until the description is completed.

These new axes can be assumed as a new reference system, alternative to the initial one. The phenomenon is the same, but our viewpoint, expressed by the axes, is changed; this allows us to focus on the most relevant aspects expressed by the first factors. As the rate of explained inertia progressively decreases from the first to the last factor - i.e., factors are ordered according to decreasing values of the associated eigenvalues – ignoring the last factors leads to a reduction in the dimensionality of the description at the cost of a minimal loss of information.

6.3.1 – An example

We will illustrate the operational use of the **PCA**, and the interpretation of its results, by means of a didactic example. The table processed describes some features of Kenya's 41 administrative districts. Of course, similar examples could be conceived for many other countries, but keep in mind that the variables depend on the objective of the analysis, and are generally not equivalent in different contexts. Statistical systems are designed differently in different countries, and collect differently defined variables. **The method is portable, but in general no fixed recipe about data is possible**.

The input file KENYA.DAT (documented in KENYA.TXT) consists of 41 records. Once loaded in ADDAWIN, the first step is usually the computation of some elementary statistics, correlations etc., before going through a Principal Components Analysis and a Classification.

Each record refers to one district, and contains the values of the following variables:

- the name of the district (used to label the district in the printouts)
- population (used as weight in the analysis - see below)
- average fertility rate
- average wage (in the formal economy)
- amount of high potential land per capita
- activity rate (in the formal economy)
- cereal production per capita
- cattle per 1000 inhabitants
- goats and sheep per 1000 inhabitants
- cash crop production per capita

The units of measure are not indicated. Some values are just rates, therefore pure numbers, but for some others the information on the used units of measure was lost over time. Anyway, as the same unit was used to measure a variable for all districts, this uncertainty does not subtract anything to the validity of the exercise.

In each record the first two fields are the label and population of the district; the latter is assumed as the district's weight throughout the analysis. The last seven variables, that describe some economic features of the district, are ***active*** in the analysis while the fertility rate, not homogeneous with the others, is used as ***supplementary***. This means that it will not contribute to determine the factors - computed on the basis of the active variables only - but its relationship with the active variables (and with the factors) will be investigated.

A table so conceived is aimed at exploring the economic features of the districts as well as the relationships existing amongst the variables.

The fertility rate does not make much sense as such, and is added here mainly to illustrate the use of supplementary variables, with little concern for the meaning. However, it will allow the user to single out the economic characters more correlated with a high or a low fertility rate.

The comments concerning the example will be indicated with the simble "🗁".

6.3.2 - Entering the analysis control parameters

Select the PCA (Analysis→Principal Components Analysis) from the Menu. The dialog shown in the figure 6.6 is displayed and the user is prompted to enter some control parameters.

First of all, highlight the Dataset (just one) that contains the variables to be processed. In our case only the Kenya Dataset has been loaded, but this may not be always the case. If the variables to be input to the PCA are spread over several Datasets, a new Dataset must be created to gather all of them.

In the centre of the dialog there are three listboxes: the one in the middle lists all the variables in the Datasets that can be processed by a PCA. This depends on their scale, as the note in the upper-right corner explains. The analyst chooses the variables to be used as active or supplementary, moving them to the appropriate list, to the left or to the right respectively. The variables left in the middle list **will be ignored** in the analysis.

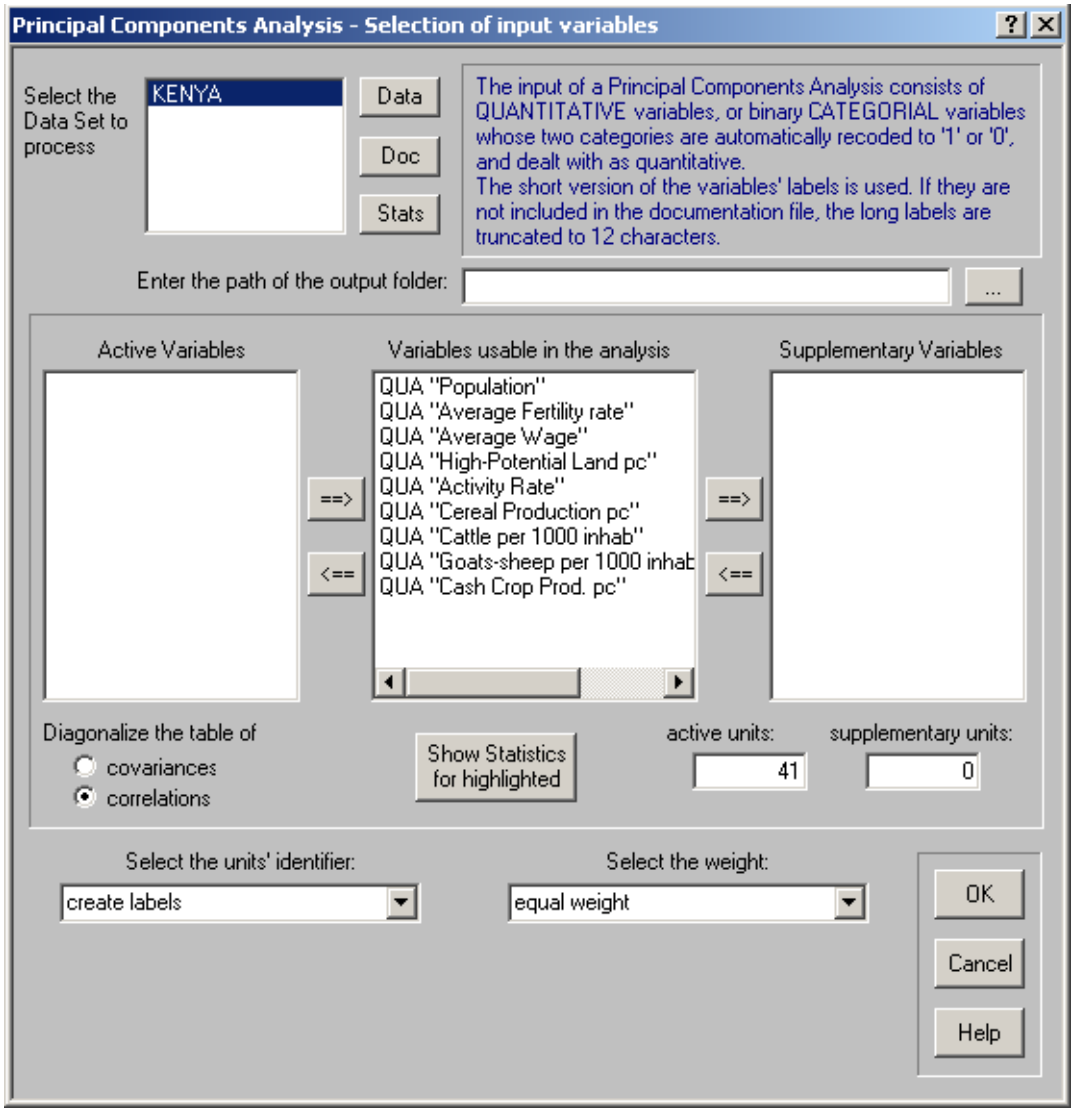


Figure 6.6 – PCA: the main dialog

- The path of the ***output directory*** must be provided, where all the ***output files*** will be written. In detail, they are:

  1. a file named ACOMPnnn.TXT ('nnn' is a progressive number to avoid overwriting), that will include the detailed numeric information on which the interpretation must be based;
  2. a file using as name the DS label and with extension .FPL (in our case, 'Kenya.FPL'), storing the information that the utility FACPLAN will use to display the projections onto the factorial planes;
  3. a file named as the DS label and with extension .PCS (in our case, 'Kenya.PCS', where PCS stands for Principal ComponentS), storing the number of principal components that the analyst decides to use for clustering (see below);
  4. a file named as the DS label and with extension .TMP (in our case, 'Kenya.TMP'), saved together with the PCS file, where the values of the input variables are stored in binary format. It will be used later to compute the average profiles of the classes.

- Indicate **an identifier** (a variable of type ID) existing in the records to label each units, or ask the program to generate automatically a list of progressive identifiers. If the statistical units are administrative areas, for which a cartographic file of borders is available (e.g., an ArcView shapefile), ***a natural id*** is the cartographic code associated with each unit. This will enable the analyst to easily produce a map of the resulting classification.

- Select a variable, among those listed in the low-right combo, to be assigned as weight to the statistical units. Any QUANTITATIVE or COUNT variable can be used as weight (the population, which is a COUNT variable, is typically used when administrative areas are processed). Alternatively, all units can be assigned the same weight, which is typical when working with individual units, like persons or households, though the value of a "raising factor" is often used as weight when carrying out an analysis on data from a sample survey, and results valid for the universe are desired.

- Besides the **active** units, whose relationship with the variables will contribute to determine the factors, it is also possible to include in the analysis a set of so-called **supplementary** units, which have no part in constructing the factorial space, but can be projected onto the factorial planes, and for which the **relative contributions** are computed (the **absolute contributions** are obviously null, see below). The aim is to get more information from the way these supplementary objects are placed with respect to the active ones and the variables.
  By default, all the units in the Data Set are processed as active. You can change the number of active units and add some supplementary units, but keep in mind that the **first units in the DS**, in the requested number, are assumed as active; the units taken as supplementary, if any, are those that immediately follow them. This condition, quite strict, might be relaxed in future releases of ADDAWIN.

Example *If the active rows describe the behaviour of a group of districts in a given year, the supplementary rows may describe the same districts in a different year, thus permitting a qualitative visualisation and interpretation of the variations occurred.*

- By default, the analysis is performed on the correlation matrix. Do not change this setting if you are not an expert, or you have no good reason to do it.

The figure 6.7 shows the dialog filled with the control values for the Kenya analysis.

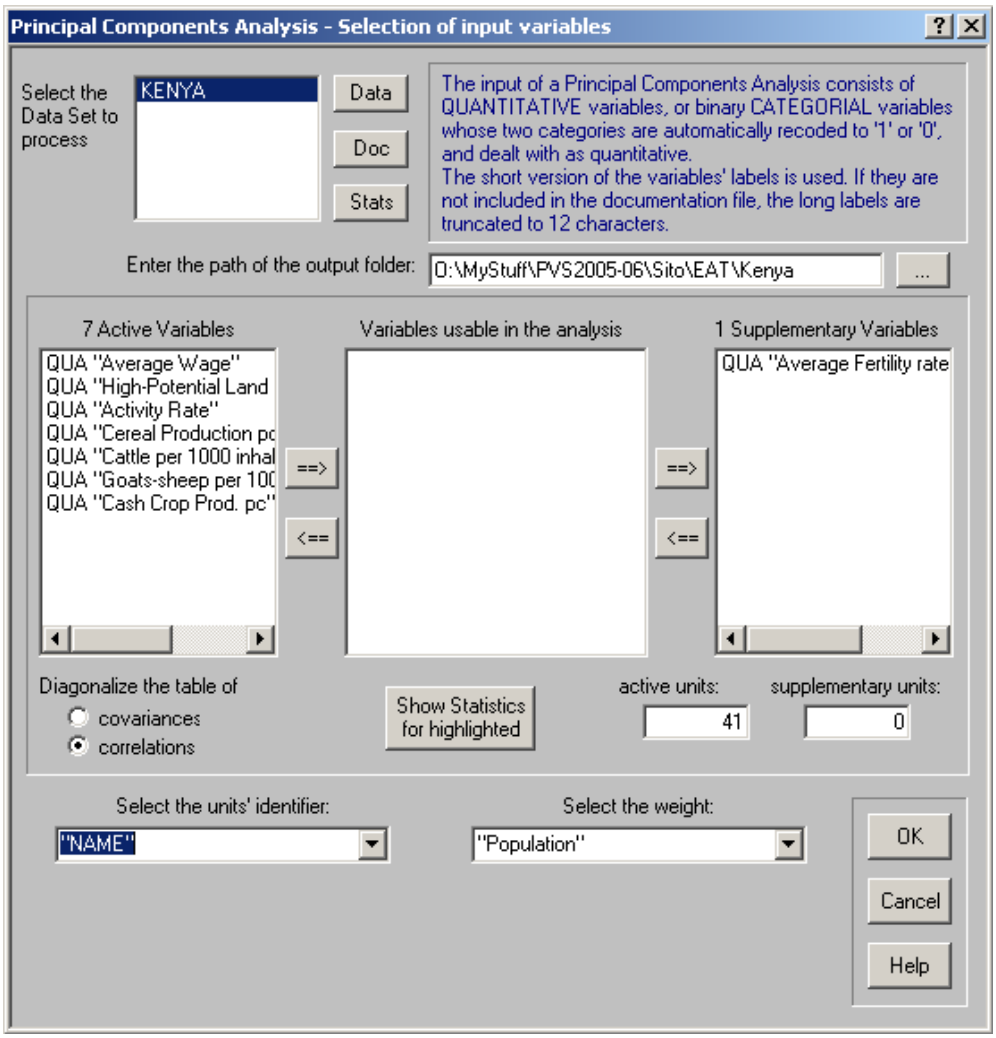


Figure 6.7 – The dialog of figure 6.6, suitably filled.

Active variables contribute to the construction of factors, while the **supplementary** ones are used only for descriptive purposes. The total variance of the table, to be split amongst the factors, is only the one contributed by the active variables.

In general, only part of the variance of the supplementary variables can be "explained" in the (factorial) space spanned by the active ones. In other words, in general a supplementary variable cannot be exactly expressed as a linear combination of the active ones.

There is only one supplementary variable (the fertility rate) in our example.

After filling the dialog, press the OK button. An editor window like the one in figure 6.8 will appear. It consists of some lines of comment, headed by a '#' character, followed by some other lines that contain, in double quotes, the **long and short names** of the variables to be processed.

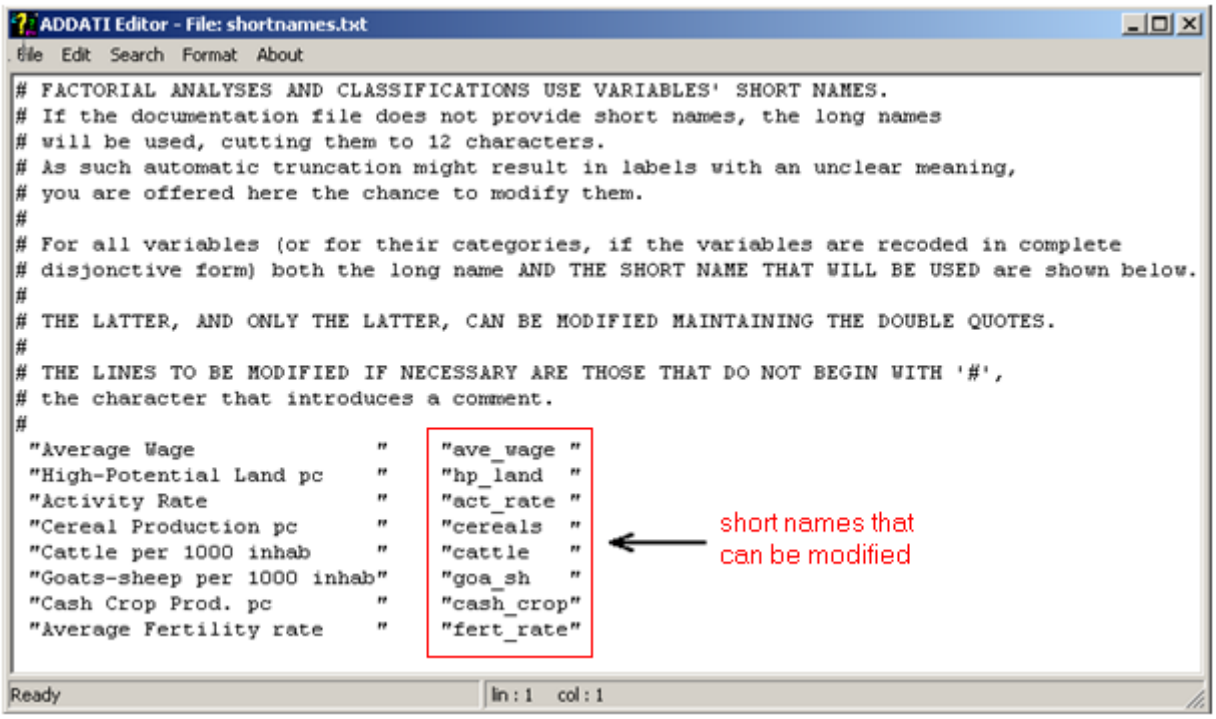


Figure 6.8 – The edit window to confirm or modify the variables' short names.

Note	For each variable, both a ***long*** and a ***short*** label can be specified in the Dataset Documentation file. Short labels are used in multivariate and clustering procedures. If for a variable no short label is provided, one is automatically created in ADDATI by truncating its long name to its **first twelve characters**. As what results is not guaranteed to have a clear meaning, the names (long and short, supplied or created) are displayed in the editor. Short labels can then be modified by the user, maintaining the twelve character limitation and the double quotes.

Actually, the role of the editing window is more complex and depends on the scale of the variables (whether they are QUANTITATIVE or BINARY, and PCA is used; or COUNT or CATEGORIAL, requiring an Analysis of the Correspondences). Until you become sufficiently expert, **read carefully the lines of comment, which depend on the variables' scale**.

Save the file if you have changed something, and close the window. The modified short labels, if any, are stored in the Dataset documentation: remember to save the Dataset before exiting ADDAWIN if you wish to keep these changes for future use.

At this point the variables are standardised and their correlation matrix is computed and saved to the output file (conventionally named ACOMPnnn.TXT). Starting from the correlation matrix the factorial axes and the associated eigenvalues are computed.

Two windows are then displayed, that give the analyst the possibility to decide how many Principal Components to save for various purposes.

The first one (see figure 6.1) displays the contents (at least, the part written so far) of the output file ACOMPnnn.TXT. It shows the ***eigenvalues*** in the case of our Kenya example: they are a measure of the explanatory power of the various Principal Components (their Inertia). After examining it, the analyst must fill the dialog of figure 6.9, deciding how many Components to save for clustering, for the interpretation of the analysis or for the visualisation of the factorial planes.

```
7 SIGNIFICANT FACTORS DETERMINED - SHARE OF INERTIA:

TOTAL INERTIA   = 7.000000

|    |           |EXPLAIND|CUMULATD|
|  # | EIGENVALUE| INERTIA| INERTIA|
|    |           |   (%)  |  (%)   |
|----|-----------|----------------|
|  1 | 2.5354424 | 36.221 | 36.221 |*******************************************
|  2 | 1.7289826 | 24.700 | 60.920 |******************************
|  3 | 1.0887727 | 15.554 | 76.474 |******************
|  4 | 0.8229769 | 11.757 | 88.231 |**************
|  5 | 0.4085767 |  5.837 | 94.068 |*******
|  6 | 0.2909169 |  4.156 | 98.224 |*****
|  7 | 0.1243322 |  1.776 |100.000 |**
```

Table 6.1 – The eigenvalues associated with the Principal Components measure their explanatory power.

The value of the total inertia, distributed over the Principal Components, is 7, as there are 7 active variables and each of them, being normalised, contributes a variance equal to 1. The cloud maintains an Inertia equal to 2.54 when it is projected onto the first factorial axis: this is the 36.22 per cent of the overall Inertia. The last column of the table shows the cumulated inertia explained by all the principal components considered up to that point: it can be seen that the first five factors summarise 94.07% of the total inertia. This means that in the 5-dimensional space spanned by the first 5 factors the cloud maintains 94.07 per cent of its inertia, while the residual 5.93% is lost.

This could be an acceptable simplification; yet, as the variables are only seven (and so are the factors) we will decide in this case to retain all of them for clustering.

Moving up on the file displayed, just before the eigenvalues, the correlation matrix can be inspected: the table 6.2 shows it for our example on the 41 Kenya districts. The matrix includes all the correlations among active and/or supplementary variables.

```
CORRELATIONS (*1000)
-------------------------------------------------------------
             ave_  hp_l  act_  cere   cat   goa cash_ fert_
             wage   and  rate   als   tle   _sh  crop  rate
-------------------------------------------------------------
 ave_wage    1000
  hp_land    -429  1000
 act_rate     698  -263  1000
  cereals    -529   579  -157  1000
   cattle    -125   233  -257   100  1000
   goa_sh     -63   105  -240  -165   631  1000
cash_crop    -310   116   162   323    -6   -96  1000
fert_rate    -462   302  -193   625  -103  -330   264  1000
```

Table 6.2 - The correlation matrix as it appears in the PCA printout.

🗁 The highest correlation is between "average wage" and "activity rate", mainly due to the urban districts of Nairobi and Mombasa; a high correlation also exists between "cattle" and "goat_sheep", meaning that these two activities tend to be present in the same districts. The "fertility rate" appears to be positively correlated with the cereal production, meaning that it specially characterises rural districts.

These considerations can in general be further developed and refined, leading to a first identification of groups of correlated variables.

6.3.3 – How many Principal Components should be saved?

The user decides how many factors fit his/her needs, then the factorial co-ordinates are computed and saved to file for interpretation and clustering.

According to the values entered in the dialog of figure 6.9, the Clustering Analysis that follow will assume as input all the seven Principal Components, while the first three of them, summarising 76.47% of the overall variance, will be used to illustrate the characteristics of the districts, six for the variables, and three for their projections onto the factorial planes. Six is the maximum number of PCs that can be used for these last operations.

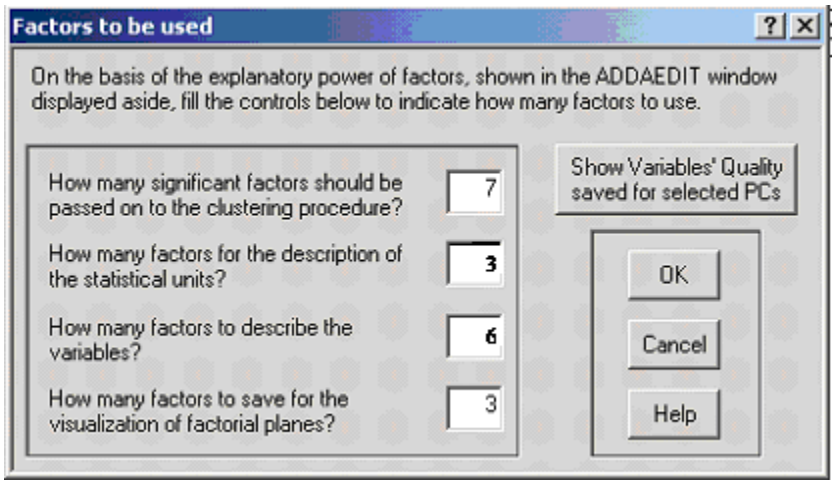


Figure 6.9 - The dialog on how many PCs to use

In general:

Principal Components to be used to cluster the statistical units

It is necessary to save the factorial co-ordinates before proceeding to a non-hierarchical classification of the units. For a PCA, it is generally convenient to save a number of factors sufficient to explain 80% or 90% of the overall inertia. When working with categorical variables (and ACORR) a lower rate is generally sufficient.

There is however an uncertainty: even though the rate of global Inertia explained by the selected factors is sufficiently high, the representation of some particular variable can be unacceptably low. In order to assess this, and increase the number of factors if necessary, use the "**Show Variables' Quality**" button.

🗁 Kenya example: we have already decided (see above) to use all the factors for clustering (though the last, explaining only 1.78% of the inertia, could easily be ignored). Therefore the value '7' is entered.

Description of the statistical units

It is convenient to have these contributions saved **only when** one is interested in the behaviour of some particular units. Skip it (i.e., enter '0') if the units are too many, or when you are not particular interested in their individual behaviour.

For each unit and each requested factorial axis the following information is saved on request:

- the **factorial co-ordinate** of the unit on the axis;
- the **relative contribution** (fraction of the unit's inertia explained by that factor);

- the **absolute contribution** (fraction of the factor's overall inertia contributed by the unit);

Their meaning will be illustrated further on.

🗁 Kenya example: as units are only 41, it might be interesting to examine thoroughly the behaviour of some of them. Let us request the first three factors to be saved, that explain 76.47 % of inertia.

Description of the variables

It is always convenient to request it for the most relevant factors, as the meaning of factors is derived from the variables' contributions. For each variable and each requested factor the following information is saved to file ACOMPnnn.TXT:

- **the factorial co-ordinate** of the variable on the axis;
- **the relative contribution** (share of the variable's inertia explained by that factor);
- **the absolute contribution** (share of the factor's overall inertia contributed by the variable);

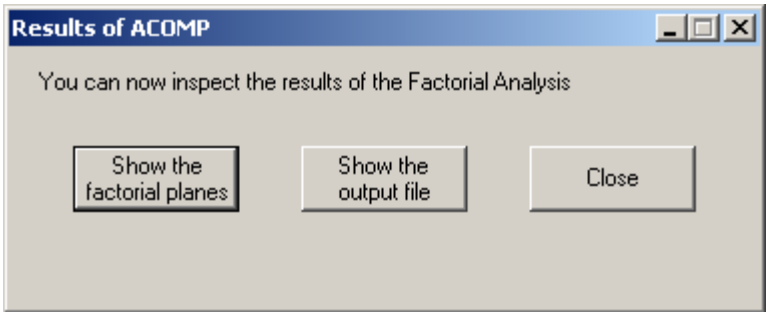
🗁 For our Kenya example we request the contributions on 6 factors, sufficient to explain 98.22% of the overall variability (the explanatory power of the last factor is negligible). We will use these contributions to interpret the meaning of factors.

Projections onto the factorial planes

They may help the interpretation when a plane accounts for a high fraction of the overall variance. One must, however, limit oneself to consider only those points (units or variables) that are **well represented** on that plane.
FACPLAN offers a specific option to select well represented points.
Even when the projections are used as a starting point, it is expedient to insist that the interpretation **should always be based on the relative and absolute contributions**.

🗁 Kenya example: three factors are sufficient to display the most interesting factorial planes.

When the OK button is pushed, the requested principal Components are computed and save for variables and statistical units, then the dialog shown below is displayed.



Note *It is possible to display on the screen the projections of the cloud onto several factorial planes, visualizing at will any combination of units and variables, both active and supplementary. Any part of the plane can be zoomed, and the image can be saved and printed.*

6.3.4 –The table of contributions and their interpretation

The information described in detail below is stored (on request) in the output file ACOMPnnn.TXT, separately for active and supplementary units and variables.

The table 6.3 shows the information on the variables' contributions as it appears in ACOMPnnn.TXT**.** The meaning is detailed for the ***hp_land*** variable and the first two factors in the table 6.4 here below.

QLT (quality of the representation): this is the fraction of the variable's inertia globally explained by **all** the factors for which information is printed (six factors were requested in this case).

It sums up the variable's relative contributions on the printed factors. The table shows that the first six factors explain 1000/1000 of the inertia of the variable *hp_land*.

INR (variable's total inertia): as all variables are standardised, they contribute equally to the cloud's overall inertia which is exactly 7 (there are 7 active variables all having variance 1). INR is here expressed as a fraction of the total inertia: 143/1000, corresponding to 1/7.

In the case of a Correspondence Analysis, the variables (i.e., the columns of the table) have generally different values of INR, which depends on the point's weight and on its distance from the cloud centre, which represents the overall average profile.

WEIG **(weight)** Importance of the variable in the analysis. As variables are standardised, WEIG has conventionally the same value for all.

#	ACT VAR	QLT	WEIG	INR	DIS	FAC 1	REL CON	ABS CON	FAC 2	REL CON	ABS CON	FAC 3	REL CON	ABS CON
1	ave_wage	939	1	143	1000	-850	723	285	-180	33	19	282	80	73
2	hp_land	1000	1	143	1000	722	521	206	136	18	11	136	19	17
3	act_rate	957	1	143	1000	-671	450	177	288	83	48	608	370	340
4	cereals	992	1	143	1000	692	479	189	487	237	137	171	29	27
5	cattle	997	1	143	1000	451	204	80	-671	450	261	411	169	155
6	goa_sh	999	1	143	1000	282	80	31	-816	666	385	289	84	77
7	cash_crop	991	1	143	1000	281	79	31	491	241	140	582	339	311

#	SUP VAR	QLT	WEIG	INR	DIS	FAC 1	REL CON	ABS CON	FAC 2	REL CON	ABS CON	FAC 3	REL CON	ABS CON
8	fert_rate	487	1	143	1000	437	191	0	486	236	0	-77	6	0

Table 6.3 - The contributions for the variables on the first three factors.

	#	ACT VAR	QLT	WEIG	INR	DIS	FAC 1	REL CON	ABS CON	FAC 2	REL CON	ABS CON
Variable's → ordinal number	5	hp_land	1000	1	142	1000	722	521	206	136	18	11
		↑ alphanumeric label					↑	↑	↑	↑	↑	↑
							information on factor # 1			information on factor # 2		

Table 6.4 - The contributions for the variable ***hp_land***.

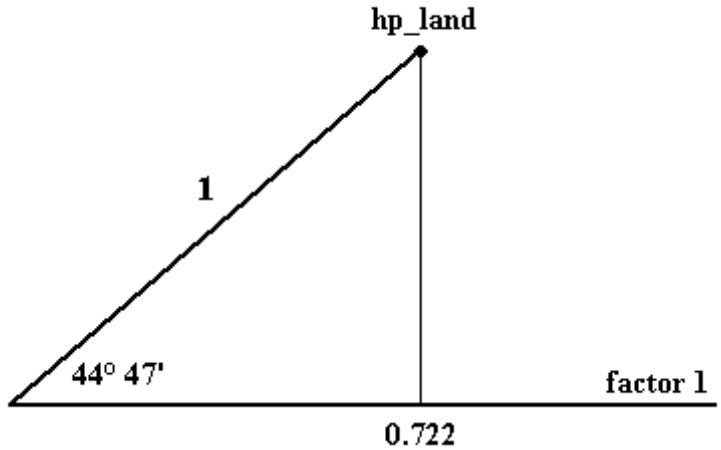


Figure 6.10 - The co-ordinate of a variable-point is the correlation between the variable and the factor: 0.722 = cos (44° 47') = correlation (hp_land, factor 1).

DIS is the square of the distance of the variable-point from the origin. **It measures the variable's variance**. Here it is 1 for all variables, as they are standardised. In the variable space, all points representing variables lie on the surface of a hyper-sphere with radius 1.

FAC1 is the co-ordinate of the variable-point on the first factorial axis (it must be read as 0.722). As the distance of each variable-point from the origin is exactly 1, FAC1 is equal to the cosine of the angle formed by the segment joining the point with the origin and the first factor axis (see figure 6.10). It can be shown that this is a measure of the correlation between the variable and the first factor (considered as a new variable constructed as the maximum-variance linear combination of all the original variables).

REL CON **relative contribution** (of the factor to the variable): this is the fraction (x 1000) of the variable's inertia explained by the factor.

Here the first factor explains 52.1 per cent of the variance (over districts) of the high-potential land rate. It can be easily shown that for a Principal Component Analysis the relative contribution is the square of the corresponding factor co-ordinate (FAC1); it is therefore equal to the square of the correlation between the variable and the factor.

ABS CON **absolute contribution** (of the variable to the factor's variance): this is the fraction (x 1000) of the factor's inertia contributed by the considered variable. Here the 20.6 per cent of the variance of the first factor is contributed by the variable *hp_land*.

The table 6.5 shows the **information concerning the units** as it appears in ACOMPnnn.TXT.

The meaning is the following:

QLT (quality of the representation): this is the rate of the unit's inertia globally explained by the factors requested (they are three in this case). It cumulates the unit's relative contributions on the printed factors.

INR (**unit's total inertia**): this is the fraction (x 1000) of the total inertia that is contributed by the unit-point:

INR = (unit's inertia) / (total inertia)

where the total inertia is the sum of the eigenvalues. The unit's inertia with respect to the origin (coincident with the centre of the cloud) is defined as the product of the point mass (WEIG) by the square of its distance from the origin (DIS).

WEIG **Weight** of the unit in the analysis (so scaled that all weights sums up to 1000); it represents the unit's relative importance.

Here the Kericho district has a population (assumed as weight) that amounts to 41/1000 of the Kenya population, while its inertia is 64/1000 of the total inertia.

#	ACT OBJ	QLT	WEIG	INR	DIS	FAC 1	REL CON	ABS CON	FAC 2	REL CON	ABS CON	FAC 3	REL CON	ABS CON
1	Busia	378	19	6	2177	-379	66	1	-120	7	0	-815	305	12
2	Bungoma	573	33	19	3999	1154	333	17	844	178	14	-500	63	8
3	Kakamega	684	67	12	1216	-175	25	1	281	65	3	-850	593	45
4	Siaya	750	31	7	1479	53	2	0	-215	31	1	-1030	717	30
5	Kisumu	295	31	5	1156	-373	120	2	-57	3	0	-446	172	6
6	S.Nyanza	218	53	16	2087	342	56	2	-148	11	1	-562	151	15
7	Kisii	748	57	17	2131	322	49	2	451	96	7	-1134	604	67
8	Turkana	610	9	36	27314	5	0	0	-4054	602	88	484	9	2
9	W.Pokot	552	10	6	3723	650	114	2	-1151	356	8	-555	83	3
10	T.Nzoia	789	17	40	16411	2710	448	49	2322	329	53	464	13	3
11	Nandi	960	20	47	16726	2977	530	68	1868	209	39	1926	222	67
12	U.Gishu	678	20	35	12653	2026	324	32	1957	303	43	807	51	12
13	El.Marakwet	439	10	5	3506	281	23	0	-1172	392	8	-293	24	1
14	Baringo	628	13	34	17968	1339	100	9	-3028	510	71	563	18	4
15	Samburu	743	5	48	1280	4143	257	34	-4960	368	71	2812	118	36
16	Kericho	903	41	64	10870	2434	545	97	628	36	9	1870	322	133
17	Nakuru	964	34	17	3403	663	129	6	1430	601	40	891	233	25
18	Laikipia	756	9	14	11139	2412	522	20	987	87	5	1277	146	13
19	Narok	915	14	34	17515	2096	251	24	-3130	559	78	1358	105	23
20	Kajiado	781	10	56	40181	794	16	2	-4829	580	131	2724	185	66
21	Nyandarua	520	15	23	10572	2251	479	30	655	41	4	-51	0	0
22	Nyeri	438	32	3	667	-289	126	1	457	313	4	-3	0	0
23	Kirinyaga	914	19	3	995	-212	45	0	191	37	0	-910	833	14
24	Murang'a	184	42	13	2124	-99	5	0	582	159	8	-207	20	2
25	Kiambu	393	45	60	9329	-373	15	2	1503	242	58	1125	136	52
26	Marsabit	804	6	40	44459	213	1	0	-5721	736	119	1719	66	17
27	Isiolo	569	3	11	27661	-265	3	0	-3614	472	21	1613	94	7
28	Meru	943	54	9	1124	-138	17	0	-105	10	0	-1015	916	51
29	Embu	963	17	3	1292	-731	413	4	-168	22	0	-826	528	11
30	Machakos	478	67	12	1285	-387	116	4	-480	179	9	-485	183	14
31	Kitui	749	30	16	3631	183	9	0	-1616	719	46	-271	20	2
32	Mandera	767	7	7	7129	-1866	488	9	-1332	249	7	-463	30	1
33	Wajir	600	9	5	4125	-689	115	2	-963	225	5	-1037	261	9
34	Garissa	622	8	8	6548	-726	81	2	-1879	539	17	-114	2	0
35	Tana_River	439	6	6	6604	194	6	0	-1673	424	10	249	9	0
36	Lamu	783	3	1	2869	-979	334	1	-80	2	0	-1132	447	3
37	Taita/Taveta	807	10	1	791	-378	181	1	-12	0	0	-704	626	4
38	Kilifi	964	28	8	1933	-702	255	5	49	1	0	-1170	708	35
39	Kwale	788	19	3	1263	-22	0	0	-876	608	8	-476	180	4
40	Mombasa	922	22	40	12597	-3304	866	96	470	18	3	689	38	10
41	Nairobi	960	54	212	27437	-4708	808	472	545	11	9	1966	141	192

Table 6.5 - The units' contributions on the first three factors.

Generally speaking, the greater INR in comparison to WEIG, the more peculiar the unit's behaviour. In fact, if all unit-points were located at the same distance from the origin their inertia would be exactly proportional to their weight. In this case, as both INR and WEIG are scaled to 1000, the two quantities would have exactly the same value.

The meaning is detailed in the table 6.6 below for the **Kericho** district and the first two factors.

	#	ACT OBJ	QLT	WEIG	INR	DIS	FAC 1	REL CON	ABS CON	FAC 2	REL CON	ABS CON
unit's ordinal. number →	5	Kericho	903	41	64	10870	-2434	545	97	628	36	9
		↑ unit's alpha-numeric label					↑	↑	↑	↑	↑	↑
							information on factor # 1			information on factor # 2		

Table 6.6 - The contributions on the first two factors for the Kericho district.

A value of INR greater than WEIG (as in the case of Kericho) means that the point has a distance from the origin greater than the average distance, and that its behaviour is quite particular (it must be remembered that **the origin represents the system's average behaviour**). A more careful inspection is needed in order to identify the profile components involved; i.e., on which aspects the district is peculiar. The results of the Principal Component Analysis could be used for this purpose, but it is far more simple to consider the profile of the class to which the district is assigned in the subsequent cluster analysis that follows: if the variability within the class is neglected, the class profile can be assumed to represent the behaviour of all the units assigned to it.

It must be kept in mind that the distance between the unit-point and the origin is a measure of the global difference between the unit's behaviour and the overall average behaviour of the system (i.e., the set of all the units considered, in our case Kenya as a whole). A Principal Component and a Correspondence Analysis always assume as reference the average behaviour of the whole system, and analyse in which way and how much the various units differ from it (and from one another). This admits a straightforward interpretation in most cases.

DIS is the **square of the distance** of the unit-point from the origin. The greater DIS, the more the unit's profile globally differs from the average behaviour, represented by the cloud centre.

FAC1 is the **co-ordinate** of the geographical unit on the first factorial axis (for Kericho, it is to be read as 2.434).

REL CON **relative contribution** (of the factor to the unit): this is the fraction (x 1000) of the unit's inertia explained by the factor. In the case of Kericho, the first factor is sufficient to explain 54.5 per cent of the district's inertia.

ABS CON **absolute contribution** (of the unit to the factor's variance): this is the fraction (x 1000) of the factor's inertia contributed by the unit. Here, 9.7 per cent of the first factor's variance is contributed by the Kericho district.

6.3.5 - Interpretation of the factors

The meaning of the factors, considered as new variables, can be derived from their correlation with the initial variables, i.e. by determining which variables more contribute to them.

Factor 1

Look at the table 6.3. The highest absolute contributes to the factor come from ***ave_wage*** and ***act_rate*** (which lie on the negative side of the axis) and from ***hp_land*** and ***cereals*** which lie on the positive side. This represents the most relevant set of relationships existing in the correlation matrix. *Ave_wage* and *act_rate* are positively correlated with one another, and so are *hp_land* and *cereals*; the two variables in the first group are negatively correlated with those in the second group. Thus,

the first factor captures the main variability present in the system (at least, according to the set of variables we chose), namely the one between *rural districts* - with low values of activity rate and average wage associated with values of *hp_land* and cereal production above the average - and *urban districts*, characterised by opposite features.

An exam of table 6.5 shows which districts are more affected by this opposition: it is sufficient to single out those receiving a strong relative contribution (REL CON) from the first factor. They are Mombasa, Nairobi and, to a less extent, Mandera on one side and T.Nzoia, Nandi, Kericho, Laikipia, Nyandarua and Embu on the other. The other appear much less involved: their behaviour cannot be reduced to the above opposition and seems to depend on other relationships existing among the variables.

Note *The first factor captures and summarises a relation among the four mentioned variables in which the above mentioned districts are involved. Some other district might have a high value of act_rate, but not associated with a high value of ave_wage and with a low level of cereals production and hp_land.*

Factor 2

It summarises the positive association existing between the presence of **goats_and_sheep** and **cattle** in some districts (those showing in table 5 a high REL CON value for factor 2, associated with a negative value of the FAC2 co-ordinate: Turkana, Baringo, Narok, Kajiado, Marsabit, Kitui, Garissa, Kwale and others to a minor extent). From table 3, a negative correlation appears to exist also between the presence of cattle/goat_sheep and that of cereals/cash_crop (characterising a few districts; e.g., Nakuru) showing on the second factor a high REL CON value associated with a positive FAC2 co-ordinate.

The interpretation could continue for the other factors, which become anyway less and less relevant.

The files output by ACOMP

- **ACOMPnnn.TXT** is the text file that contains the information to be interpreted.
- **Label.PCS** is a binary file, written on user's request, that contains the factorial co-ordinates (together with some extra information) used by NONGER to cluster the units.
- **Label.TMP** (that accompanies the .PCS file) is a binary file that stores the original values of the processed variables, and will be used by the Clustering procedure when it computes the average profiles of the classes.
- **Label.FPL** (written on user's request) is a text file that contains the information passed over to FACPLAN to display the projections onto the factorial planes.

In the filenames **'Label'** stands for the label that identifies the Dataset on whose variables the analysis is carried out.

6.4 - The Analysis of the Correspondences (ACORR)

Use, limits and advices

What was said for **ACOMP** holds also here.

The standard input table for a Correspondence Analysis is a **contingency table**, i.e. a table obtained by cross-tabulating two categorical variables.

If the two variables have *n* and *p* categories respectively, the resulting contingency table has *n* rows and *p* columns. The table's generic cell (i,j) counts the units that simultaneously take the i-th category of the first variable ***and*** the j-th category of the second one.

Some ***only apparently different*** tables can be actually thought of as contingency tables, and dealt with via a Correspondence Analysis:

- tables obtained by setting side-by-side several contingency tables that count the same basic units (households, individuals, firms, etc.);
- binary tables obtained from qualitative descriptive tables by converting the qualitative (categorical) variables to (binary) ***complete disjunctive form***.

The latter case is less intuitive, but quite interesting in practice for its possible applications to the analysis of Survey data.

Example

Think of a table where the n rows represent n statistical units, described by p categorical variables, some of which may have been obtained by recoding to categorical form some variables directly observed as quantitative. When the p variables are converted to complete disjunctive form, each of them produces a table of zeros and ones, ***having as many columns as the variable has categories****.*

Exactly p side-by-side binary tables are obtained, each of which can be seen as a contingency table cross-tabulating the variable "unit" with one of the descriptive variables: for each row (unit), a cell corresponding to a category not assumed by that unit contains 0 (i.e., it includes no unit), and a cell corresponding to a category assumed contains 1 (i.e., ***it counts exactly one unit****). The structure is quite banal, but it can be thought of as a multiple contingency table. In this case, as the rows represent a set of units, their mutual similarity structure can be analysed via* ***ACORR****, followed in case by a classification.*

In a contingency table **rows and columns have a similar role** and are dealt with symmetrically in **ACORR**. The purpose of the method is to analyse the similarity among rows (with respect to columns), the similarity among columns (with respect to rows) and the relationships existing between rows and columns.

Owing to the table's symmetry, the analytical treatment can focus on rows as well as on columns,. The table 6.7 shows a small didactic example: a system consists of three geographical units, each described by the amount of cultivated land per type of crop (only three crops are considered explicitly, and the fourth category summarises all the rest). **ACORR** would convert the initial table 6.7a) to that of **row profiles** 6.7b), or that of **column profiles** 6.7c). Actually only one of these tables needs to be analysed; the results for the other are derived through simple transformations.

	teff	mais	sorghum	other		
unit 1	40	60	50	100	250	
unit 2	100	100	200	200	600	**a)**
unit 3	80	120	100	200	500	
	220	280	350	500	1350	

	teff	mais	sorghum	other		
unit 1	.16	.24	.20	.40	250	
unit 2	.17	.17	.33	.33	600	**b)**
unit 3	.16	.24	.20	.40	500	
	.16	.21	.26	.37	1350	

	teff	mais	sorghum	other		
unit 1	.18	.21	.14	.20	.19	
unit 2	.45	.36	.57	.40	.44	**c)**
unit 3	.36	.43	.29	.40	.37	
	220	280	350	500	1350	

Table 6.7

a) An example of a contingency table: each cell counts the area (in thousands of acres) per administrative unit and type of crop.
The totals (***marginal values*** of rows and columns) are also shown.

b) The ***row profiles*** computed from the table **a)**. Each row shows, for the concerned unit, the percentage of the cultivated land dedicated to each type of crop; the last row gives the same information for the whole system and is assumed as the "normal" (or average) behaviour.

The relative specialisation of a unit is determined by comparing its profile with the overall profile. In **ACORR**, the weight of each unit is proportional to its marginal (i.e., to the total cultivated land in the unit) and is computed by the program itself.

c) The ***column profiles*** computed from the table **a)**. Each column shows, for the concerned crop, how the cultivated land is distributed in percentage on geographical areas. The last column gives the same information for the whole system and is assumed as the "normal" (average) crop distribution on geographical units.

The relative concentration of a crop is determined by comparing its distribution with the overall profile. In **ACORR**, the weight of each column is proportional to its marginal (i.e., to the total cultivated land for the concerned crop) and is computed by the program itself.

Owing to the symmetrical role of the two variables, the analysis can focus indifferently on the rows or the columns. The table 6.7 shows a little didactic example: a system consists of three geographic units, each described by the area dedicated to some types of crops (only three types are considered explicitly, while the fourth counts all residual crops). **ACORR** converts the initial table 6.7a) into a table of ***row profiles*** 6.7b) or one of ***column profiles*** 6.7c). It is sufficient to analyse only one of

them: the program automatically determines which is computationally the more convenient. The results relative to the other table are then derived through some simple transformations.

According to the table 6.7b, as there are four descriptive variables, each unit is represented by a point in a four-dimensional space, whose co-ordinates are the components of the profile. The point is assigned a weight proportional to the cultivated land in the unit (the row's marginal value). All together, there are three weighted profile patterns in a four-dimensional space.

Symmetrically, according to the table 6.c each crop can be represented as a point in a three-dimensional space (there are three geographical units), whose co-ordinates are the components of the corresponding column-profile. The crop-point is assigned a weight proportional to the overall area where that crop is cultivated. In this case we have four weighted profile patterns in a three-dimensional space.

Consider the table 6.b. The profiles of the first and the third units are identical; the reason is that the two corresponding lines in table a) are proportional. The two units have a different amount of cultivated land (and therefore a different importance for the analysis) but an identical percentage distribution of that land amongst crop types. The two corresponding unit-points are coincident in the representation space.

Were all the lines in the a) table proportional, all the unit-points would be coincident: there would be no cloud, only a point, and no variability to be analysed. On the contrary, when the units have different behaviours the corresponding points are scattered about the cloud centre, which represents the system's average behaviour (i.e., the overall crop mix, given by the marginal row in the table 6.7b.

Similar considerations can be developed for the table 6.7c.

The distance between two points-profile in R^p is computed according to a modification of the usual Euclidean formula, and is known as "***chi-square distance***". For the definition the user is referred to a multivariate statistics textbook.

ACORR processes the table of profiles in a way very similar to what was already explained for the **PCA**. *Eigenvalues*, *factorial co-ordinates* and *contributions* are determined, on which the interpretation is based. The following differences must be remarked:

- in **ACORR**, differently than in **PCA**, rows and columns play a totally symmetrical role. The tables of row and column contributions, saved to file ACORRnnn.TXT, are interpreted in exactly the same way. In **ACORR** no standardisation of variables is performed, no correlation table is printed and we prefer to speak of a stronger or weaker association of two given lines (two rows or two columns) with respect to the lines of the other set.
- in **ACORR the first eigenvalue** (called "***trivial***" or "***banal***") **is always 1**: it is of no interest, as it is a mere consequence of the transformation to which the original table a) has been submitted to compute the profile table b); therefore, it is ignored. All other eigenvalues (the meaningful ones) **lie between 1 and 0**.
- The number of non-null eigenvalues is generally less than what could be expected from the size of the data table. The dimensionality related to the number of the columns is only apparent (and redundant): in every line of each single contingency table the values of the cells add up to the same total, i.e., to the number of the counted units. Therefore, the columns are not independent, and this reduces the actual dimensionality of the feature space.
- The total Inertia of the cloud can be computed in a very simple way from the number of the categorical variables and that of their categories:

$$Inertia = \frac{n.of\ \ categories}{n.of\ \ \text{var}iables} - 1$$

Thus, for example, if the table to be processed consists of five side-by-side contingency tables (i.e., if there were five initial categorical variables), with 4, 4, 3, 4 and 4 categories respectively (19 categories altogether), then the total Inertia of the cloud is 2.8 = (19/5) –1.

- If the original data table consisted of categorical variables, automatically recoded to binary form by ACORR, the explanatory power of the first factors is less than what could be expected in **PCA** (or even in **ACORR**, when processing a normal contingency table). This is an effect of the conversion to complete disjunctive form, which has increased the number of the columns of the table passed on to **PCA**, introducing some fictitious inertia. Even if the fraction of inertia loaded on the first factors seems to be low, their importance when interpreting results is still relevant.

6.4.1 – Entering the control parameters

ACORR and **PCA** need almost the same parameters to be specified in order to carry out an analysis, and similar dialogs must be filled. Therefore, the reader is referred to the full description already given for **PCA**. We limit ourselves here to illustrate the only question specifically concerning **ACORR** .

An **Analysis of the Correspondences** usually takes as input a table obtained by setting side by side one or several contingency tables, which count elementary units of the same type (households, dwellings, individuals, etc.).

Tables of qualitative variables

While working with urban or regional data (especially data drawn from Census or surveys) each line of the table often describes an elementary unit (e.g., a firm or a household), by means of some ***qualitative*** (categorical) variables. In such case, **ACORR automatically converts the qualitative variables** to **binary** (or **complete disjunctive**) form: a column for each category, with value 1 if that category is assumed by the unit of interest, 0 otherwise. The resulting binary table can be submitted to an Analysis of Correspondences, known in this case as a **Multiple Correspondence Analysis.**

ACORR, like the **PCA**, uses short names for variables and categories: an edit page is presented to the user, in which labels with up to 12 characters for the categories are proposed. **Please read carefully** the instructions on the way to fill it, as they vary according to the particular case.

In order to ease the interpretation of projections onto factorial planes, **use a common prefix** for all the categories of the same qualitative variables. For example: 'TEN_owner' and 'TEN_rent' for the tenure; 'EDU_sup', 'EDU_secund', 'EDU_primary', 'EDU_litter' and 'EDU_illitt' for the education level achieved; 'TV_yes' and 'TV_no' for the presence of a TV set, 'RADIO_yes' and 'RADIO_no' for the radio set, etc. , The categories are often indicated with the same label (e.g., 'yes' and 'no'), and this label style prevents confusion.

Side-by-side contingency tables

Data tables that describe administrative units often consist of variables of type COUNT: for each administrative unit, the frequency of some underlying elementary units (households, individuals, buildings…) is determined for the categories of some suitable categorical variables, describing aspects of interest. For example, in each administrative unit:

- the population can be counted for a given number of age classes;

- the population can be counted for a given number of education levels achieved;
- dwellings can be counted according to the levels of some type of internal service…

The result is a set of contingency tables written side-by-side, each of them describing the way elementary units are split according to a particular descriptive aspect (age, education, etc.).

In order to compute correctly the class profiles in the non-hierarchical clustering stage that will follow, ADDAWIN needs to know exactly of how many contingency tables the data table is formed, and which they are. The user is requested to provide this information by separating the sets of variables belonging to the same contingency table by means of a line consisting only of an asterisk '*'. This is done in the same editing page used to provide the short names.

When presented the page, read its comments carefully.

6.4.2 – The table of contributions and their interpretation

The information detailed below is stored (on request) in the output file ACORRnnn.TXT, separately for active and supplementary rows and columns. The interpretation of results is the same for both rows and columns.

QLT (**quality of the representation**): fraction of the point's inertia globally explained by the factors on which information is printed. It sums up the point's relative contributions on the printed factors.

INR (**point's total inertia**): fraction (∗1000) of the total inertia that is contributed by the point:

INR = (point's inertia) / (total inertia) where the total inertia is the sum of the eigenvalues. The point's inertia with respect to the origin (coincident with the centre of the cloud) is defined as the product of the point's mass (WEIG) by the square of its (chi-square) distance from the origin (DIS).

WEIG **Weight** of the point in the analysis (so scaled that the total of the weights of all the points in a set - rows or columns - sums up to 1000); it represents the point's relative importance.

Generally speaking, the greater INR in comparison to WEIG, the more peculiar the point's behaviour. In fact, if all points were located at the same distance from the origin their inertia would be exactly proportional to their weight. In this case, as both INR and WEIG are scaled to 1000, the two quantities would have exactly the same value.

A value of INR greater than WEIG means that the point has a distance from the origin greater than the average distance, and that its behaviour is quite particular (it must be remembered that the origin represents the system's average behaviour). A more careful inspection is needed in order to identify the profile components involved; i.e., in which aspects the point is peculiar. It must be kept in mind that the distance between a point and the origin is a measure of the global difference between the point's behaviour and the overall average behaviour of the system (e.g., if each row represents a district, the cloud's centre of gravity represents the Country's average behaviour.

A Correspondence Analysis always assumes as reference the average behaviour of the whole system, and analyses in which way and how much the various units differ from it (and from one another). This admits a straightforward interpretation in most cases.

DIS is the **square of the distance** of the point from the origin. The greater DIS, the more the point's profile globally differs from the system's average behaviour, represented by the cloud centre.

FAC1 is the **co-ordinate** of the concerned point on the first factorial axis.

REL CON **relative contribution** (of the factor to the point): this is the fraction (x 1000) of the point's inertia explained by the factor.

ABS CON **absolute contribution** (of the point to the factor's variance): this is the fraction (x 1000) of the factor's inertia contributed by the considered point.

The files written by ACORR

- **ACORRnnn.TXT** is the output file, to be printed and interpreted.
- **Label.PCS** (written on user's request) contains the factor co-ordinates (together with some more information) needed by NONGER to cluster the units.
- **Label.TMP** (that accompanies the .PCS file) is a binary file that stores the original values of the processed variables, and will be used by the Clustering procedure to compute the average profiles of the classes.
- **Label.FPL** (written on user's request) contains the information passed over to FACPLAN to display the projections onto factor planes.

In the filenames **'Label'** stands for the label that identifies the Dataset on whose variables the analysis is carried out.

Cap. 7. – Non-hierarchical Clustering

7.1 – Some notes on numeric Classification

The purpose of a numeric classification is to group sufficiently similar statistical units into a limited number of groups (also called **classes** or **clusters**). The *similarity* between two units can be directly observed (e.g. in a survey, by asking specific questions) or it can be **computed** on the basis of a set of observed variables that offer a suitable description of the units of interest.

Consider for example the provinces of a Country, described by the series of their per capita income over some years. Which provinces have a similar behaviour? There is no *absolute* answer: the result depends on the method used, and includes some elements of subjectivity. For example, we could perform all possible pairwise comparisons of districts, and rank pairs in order of decreasing perceived similarity.

The similarity depends upon the variables considered; i.e., it is relative to the particular description adopted. Two provinces can have a very similar demographic structure, but they can be very different for what concerns the educational or occupational levels.

The similarity level of two units can be defined in several ways.

Consistently with the geometrical representation adopted so far, according to which each statistical unit is considered as a point in a space that has as many dimensions as there are active variables (see section 6.1), we will adopt also for clustering the same notion of distance already introduced for the factorial analyses: **an Euclidean distance** (after standardisation) for quantitative variables (treated with **PCA**); a **chi-square distance** in the case of COUNT variables or qualitative descriptions (dealt with by **ACORR**). The **distance** is **a complex indicator** that takes into account contributions coming from all the variables. We conventionally assume the distance as an indicator of **dissimilarity,** considering two units more similar than two others when their representative points are closer to one another in R^p than the representative points of the other pair. This seems a good assumption, on which there can be consensus.

Even if we agree on the definition of similarity, some other operational problems arise:

- how can we measure the optimality of a partition, and how can we compare partitions with the same number of classes and decide which is the best?
- how many classes should we construct? How can we be sure that this number fits the structure of the set to be clustered?
- which clustering algorithm should we adopt?

We can identify two large groups of clustering methods, known respectively as **hierarchical** and **non-hierarchical**. Both methods work iteratively: they repeat a given sequence of operations that depend on the selected algorithm, until a final satisfactory configuration is reached. Both have advantages and drawbacks.

7.1.1 –Hierarchical methods

The ***Ascending*** (aggregative) ***Hierarchical Methods*** perform iteratively the following operations on a set of *n* elementary units, or on groups previously formed:

- compute the similarity for each pair of items (units or groups);
- merge the two most similar items, thus reducing to *n-1* the number of the groups.

At starting there are as many groups as there are elementary units, each consisting of exactly one unit. At the end of the process, after n-1 aggregation steps, all the units are grouped in one global cluster. An acceptable solution lies somewhere between these two extremes, as we would like to have the elementary units regrouped in a number of clusters small enough to guarantee synthesis, but large enough to save a consistent fraction of the information.

The aggregation process is graphically represented by means of an **aggregation tree**.

The units to be aggregated are shown on the left, at the base of the tree (figure 7.1). From left to right the units are progressively aggregated, at a distance proportional to their dissimilarity (i.e., the more similar they are, the more to the left the aggregation point lies). A partition can be obtained by cutting the tree vertically at some intermediate level. Moving to the right the number of the resultant classes becomes smaller and smaller, but also smaller becomes their internal homogeneity.

A compromise criterion is needed to decide how to cut the tree conveniently.

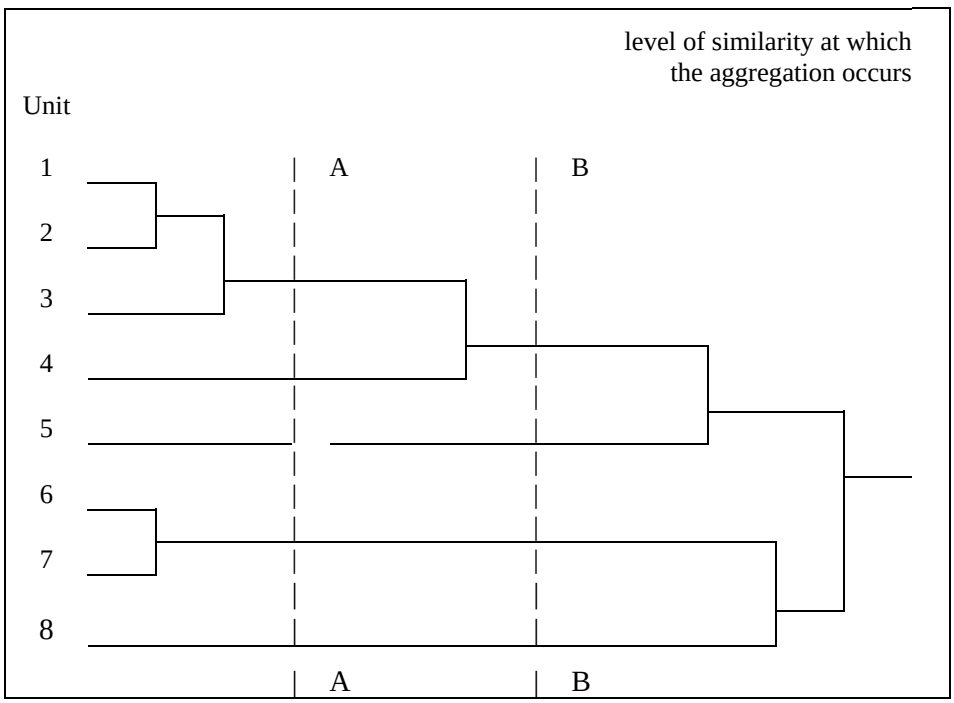


Figure 7.1 - A hierarchical aggregation tree. Eight units are clustered, and the number of the resulting groups decreases moving to the right of the tree, corresponding to an increasing dissimilarity. The section AA produces a partition with 5 classes; the section BB produces 4 clusters.

A hierarchical procedure is handy when the number of units does not exceed some tens. If they are more numerous (let us say, over a hundred) the aggregation tree becomes burdensome to compute

and difficult to read. Beyond this, if n is the number of groups currently existing, ***n(n-1)/2*** similarity values must be computed at each step: the time needed grows quadratically with *n*.

Another severe drawback of these methods is the irreversibility of the choice made at each step: when two units are joined, this is forever. Anyway, the algorithm selects the pair to be joined only on the basis of local considerations, with no global concern: it is like a chess player that steadily chooses the move which gives the best ***immediate*** advantage, with no consideration for what can occur in the next moves, i.e., with no concern for strategy.

In the case of hierarchical methods, the overall aggregation path might sometimes evolve in a more satisfactory way if an aggregation other than the locally optimal one were selected at the current step. As a consequence of this, a partition obtained by cutting the tree at any intermediate level is usually far from optimal.

7.1.2 – Non-hierarchical methods

Suppose that an initial partition, with the user-requested number of classes, has been determined somehow. Its quality is then improved iteratively, by moving some elements from one class to another if this increases the value of the **objective function**, which is a suitable measure of the partition optimality.

The process continues until a final configuration is reached that cannot be further improved through a local re-assignment of units. The partition obtained is ***a local optimum***: this means that small changes in the allocation of the units to the groups are unable to improve it. However, some better partitions with the same number of classes might exist, unreachable from the current one via small changes.

The partition eventually obtained depends on the configuration assumed at start and on the number of the requested groups.

Some definitions

When we introduced the representation of a set of statistical units as points in a multidimensional space we assumed the Inertia (defined in section 6.1), to which all units contribute, as a measure of the overall variability of the data table (or, of its information contents). We will here act consistently with that concept.

Let $In_{tot} = \Sigma_i\, m_i * d_i^2$ be the total Inertia of the cloud with respect to its overall centre, and let us consider a generic ***partition*** of the cloud in *k* clusters ('*partition*' means that each unit belongs **to one and only one group**, with no superposition between groups). Let $\mathbf{G}_j$ represent the centre of the *j*-th cluster: its co-ordinates are the average values of the *p* variables, computed keeping into account only the units belonging to the j-th class.

The generic class *j* of the partition has an ***Internal*** *(or intra-class)* ***Inertia*** defined as

$$\mathbf{In_{int}}(\,j\,) = \Sigma_{i \in Ij}\, m_i * d^2(i,\mathbf{G}_j)$$

where the sum extends only to the units belonging to the *j*-th class and the distances are computed from the $\mathbf{G}_j$, the centre of the class.

The ***internal inertia*** of a class is a measure of the dispersion of its elements about the class centre. A good partition should consist of groups as homogeneous as possible, i.e. with a low internal inertia.

The ***Internal Inertia of the partition*** (also called Inertia within Classes) is the sum of the internal inertia of all its classes. This value should be as low as possible, which means that all the classes should be as homogeneous as possible. The average characteristics of the units included in a class are represented by the co-ordinates of the class' centre $\mathbf{G}_j$.

The purpose of clustering is to offer a simplified view of a phenomenon, in which all units belonging to the same class are identified with the class' centre, neglecting as irrelevant the differences existing amongst them. The initial cloud is thus reduced to a new cloud formed by the *k* centres of class, spread about the cloud's global centre. Its inertia is the ***External Inertia*** of the partition (or Inertia between Classes):

$$\mathbf{In_{ext}} = \sum_j M(j) * d^2(\mathbf{G}_j,\mathbf{G})$$

where M(j) is the mass of the *j*-th class (equal to the sum of the masses of the units belonging to it) and $d^2(\mathbf{G}_j,\mathbf{G})$ is the square of the distance between $\mathbf{G}_j$ and the overall centre **G**.

Let us suppose that, in a way or another, the cloud has already been split into *k* clusters, whose centres are $\mathbf{G}_1, \mathbf{G}_2 ... \mathbf{G}_k$. The well-known Huyghens' theorem proves that the Total Inertia can be decomposed as follows:

$$In_{tot} = In_{ext} + In_{int}$$

where $\mathbf{In_{tot}}$ represents the cloud's Total Inertia, $\mathbf{In_{int}}$ and $\mathbf{In_{ext}}$ are the Internal and External Inertia defined above.

The ***objective function*** assumed for non-hierarchical clustering in ADDATI is

$$\max (In_{ext} / In_{tot}) \quad \text{equivalent to} \quad \min (In_{int} / In_{tot})$$

corresponding to a set of clusters which are globally as compact as possible. The value of the objective function varies between 0 and 1 (the highest the best, if the number of the clusters is the same).

In a hierarchical aggregation there are at start as many clusters as there are units. For this situation, $In_{ext} = In_{tot}$ and $In_{int} = 0$.

When the units are progressively aggregated the Internal Inertia is increased, while the External Inertia is decreased exactly by the same amount. When at the end of the process all the units are regrouped in one class, $In_{int} = In_{tot}$ and $In_{ext} = 0$. At each step, those two units are joined for which the unavoidable increase of the Internal Inertia is minimal.

Diday's dynamical clouds method

The clustering strategy used in ADDATI is quite complex, and will be illustrated in some steps. An important component is the non-hierarchical clustering method proposed by Erwin Diday in 1971.

Diday's method requests the user to enter the number of the groups that should be constructed (tentatively, the number of groups he would like eventually to obtain) and to provide in one way or another an equivalent number of points $\{S_1, S_2,...S_k\}$ in the feature space, to be assumed as initial centres (or ***seeds***) for the aggregation procedure.

The method repeats iteratively the two steps shown in the figure 7.2. The distance from all *k* seeds is computed for each unit, and the unit is assigned to the class associated with the closest seed. This generates a provisional partition with *k* classes: each unit belongs to one and only one class.

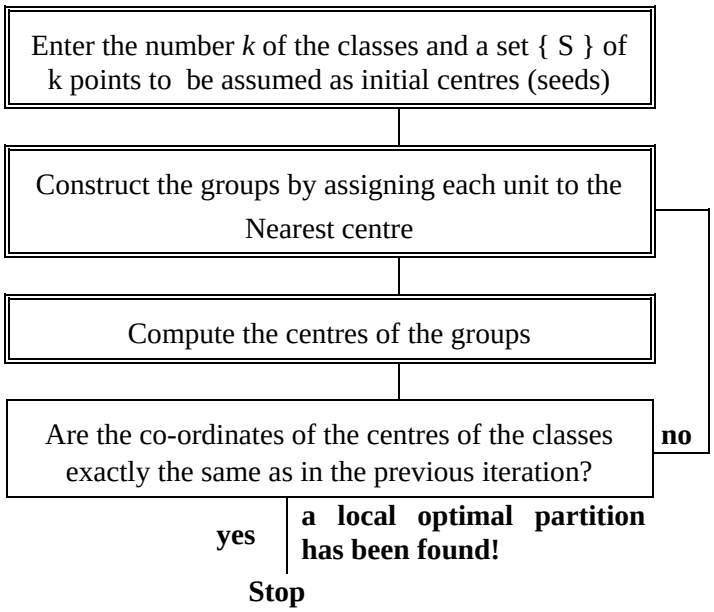


Figure 7.2 Scheme of Diday's non-hierarchical clustering algorithm.

The centres of these classes are then computed. They replace the initial centres. The assignment procedure is then repeated; the centres are recomputed, and so on. At each iteration some units change class, until a stable configuration is reached.

It can be proved that with each iteration ***the Internal Inertia of the partition cannot increase*** (it actually decreases, otherwise a minimum is reached and the procedure stops). This means that the groups become more and more compact.

The final partition corresponds to a minimum of the Internal Inertia $\mathbf{In_{int}}$ or, because of Huyghens' theorem, to a maximum of the External Inertia $\mathbf{In_{ext}}$. It is only a ***local optimum***: this means that the partition cannot be improved by changing the assignment of a few units, but might be improved by a more radical re-attribution. **We are never sure that we have found the global optimum**, i.e. ***the best*** of all the possible partitions with that number of classes: owing to the size of the problem, such certainty would generally require an enormous computing time.

Once the number of the groups has been chosen, the final partition depends only on the set of the initial seeds $\{S_1,...,S_k\}$, as the algorithm is totally deterministic.

7.2 – The clustering sequence in ADDATI

7.2.1 – The non-hierarchical clustering method

The input to the clustering routine is a table of factorial co-ordinates saved by **ACORR** or **ACOMP** after processing a table of quantitative or qualitative observed variables, or a set of contingency tables.

Diday's algorithm, that iteratively re-assigns the units to the groups in order to achieve an optimal partition, requires from the user an initial decision about the number of the groups. When the

number indicated by the user mirrors poorly the structure of the set to be segmented, the quality of the resulting partition is generally unsatisfactory.

The partition obtained, that represents a local optimum, not the global one, depends on the choice of the *seeds* (the initial centres) around which the units are aggregated according to a criterion of minimum distance. Various strategies can be conceived for choosing the seeds; in general, changing the seeds leads to different results.

In order to cope at least partly with such problems, ADDATI implements a classification strategy that uses in an integrated way both hierarchical and non-hierarchical procedures. The currently implemented analytical path is the evolution of a different strategy used in past, indicated here as *Method 1*. Here we will limit ourselves to describe with some detail the two methods (the one used in the preceding versions of the package and the one used currently) and their relationships.

We think that the hierarchical ascending method, internally used in the overall non-hierarchical procedure, is simple and does not need a detailed explanation. In the past versions of the package it was implemented as an option in the analysis menu (AHC: Ascending Hierarchical Clustering), but we decided to eliminate it, as the interpretation gets very difficult already with a hundred statistical units.

7.2.2 – Non-hierarchical Clustering: the algorithm implemented in ADDATI

In order to produce a satisfactory partition, two steps were neatly distinguished in the former releases: an **exploratory phase**, that yielded information about the most suitable number of groups and suggests a good choice of the initial seeds, followed by an **optimisation phase**, that generated the final optimal partition

The sequence was improved with version 4.0, under the name of *Method 2*. For a while *methods 1 and 2* have both been available, and the user could decide which to use. The comparison of their performances in many analytical exercises has led us to decide to eliminate the *method 1* starting from version 5.2.

We will only limit ourselves to a description of the *method 2*, still distinguishing for convenience an exploratory from an optimization phase, even though the two phases are now implemented as a continuous process, without any need of a user's intervention.

The exploratory stage

Instead of one, several partitions (say, some tens) are constructed with exploratory purposes. In line of principle the requested number of classes (for all partitions) is the same that one would like to obtain in the final optimal partition. The initial centres are usually randomly chosen (though some alternatives are possible).

The two or three partitions with the highest values of the objective-function (minimal internal inertia of the groups, i.e. maximal homogeneity within the groups), which are the best from the statistical point of view, are cross-tabulated. The number of groups in the ***product-partition*** is a-priori unknown. By construction, the units in the same group (i.e., belonging to the same cell of the rectangular table resulting from the cross-tabulation) have been clustered together in all the cross-tabulated basic partitions. Therefore, we can have a reasonable confidence on their similarity. For this reason, the groups of the product-partition are called **stable classes** or **strong forms**. Even though they are often too numerous for the purpose of the research, they offer a detailed and often exhaustive description of the most important behaviours emerging in the context of that analysis.

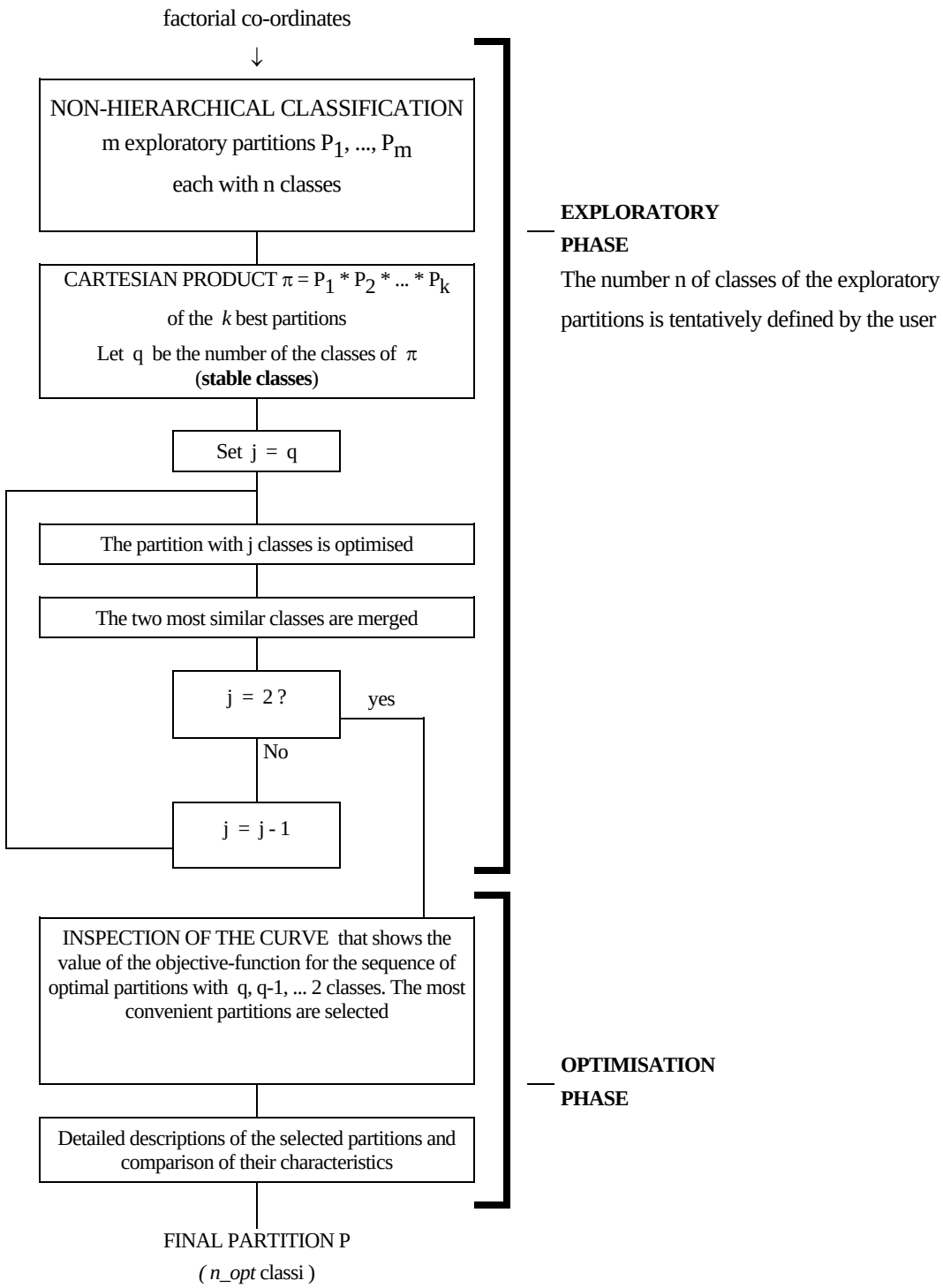


Figure 7.3 - Scheme of the clustering strategy currently implemented in ADDATI.

The Optimisation stage

The product-partition thus obtained, that almost always consists of a too high number of classes when confronted with the purpose of the research, is assumed as the starting configuration for an

optimisation sequence. It has been constructed in such a way that its groups should represent in some detail the different behaviours emerging in the set to be clustered.

Let *q* be the number of the ***stable classes*** (the classes of the *product-partition*). At this point, two routines are called: the first optimises this partition with *q* classes and saves to a temporary file an essential description, sufficient to reconstruct it easily; the second decreases by one the number of the groups by merging the two most similar ones. We have now *a non-optimal* partition with *q-1* classes.

The optimising/merging procedure is repeated, and a non-optimal partition with *q-2* classes is obtained. This procedure continues iteratively until all groups have been merged.

At this point a dialog like that in the figure 7.4 below is displayed.

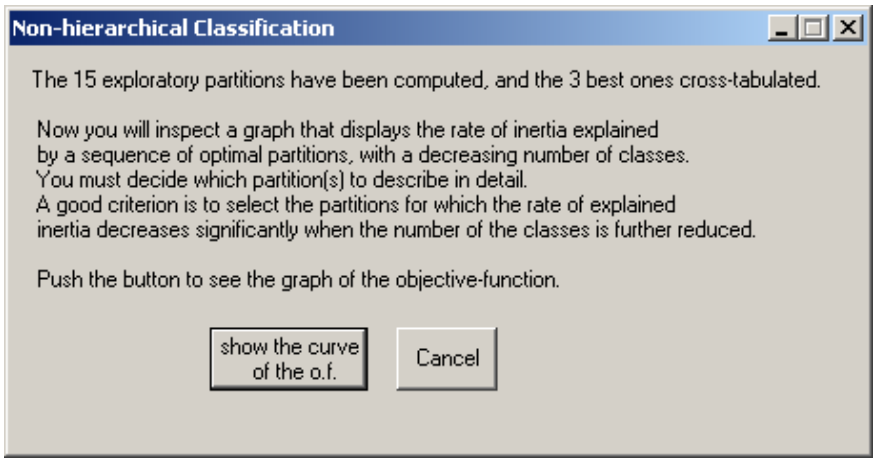


Figure 7.4 – The dialog for the selection of the partitions to be described

When the button is pushed a graph like that in Figure 7.5 is displayed. It plots the value of the objective function vs. the number of classes for the sequence of partitions with a progressively decreasing number of classes (remember that in this case **all partitions are optimal**). It is possible to focus on the most promising ones, i.e. those for which the decrease of the o.f. starts becoming significantly large when the number of the groups is further decreased by one. On user's request, the selected partitions are completely described; their comparison leads to the final selection.

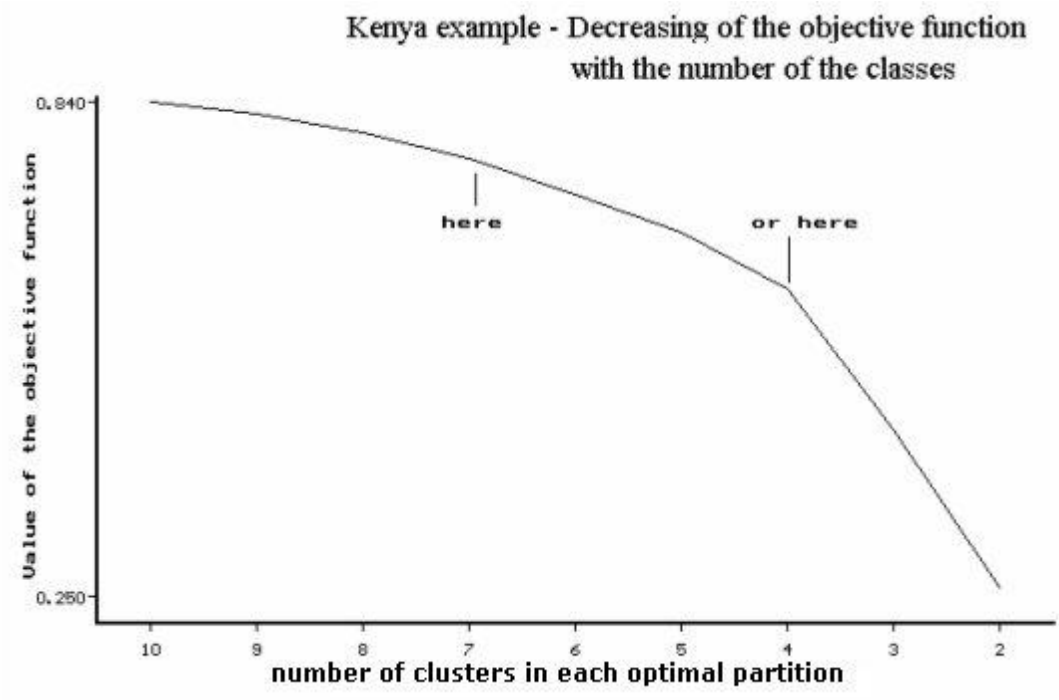


Figure 7.5 - The graph of the values of the o.f. when the number of the clusters is progressively decreased by iteratively merging and optimising.

It is worth remarking that the number of the classes of the exploratory partitions initially suggested by the analyst ***is used only to construct the cross-tabulated partition***, that is the starting configuration for the next step, in which a sequence of partitions with a progressively decreasing number of groups are optimised.

The number of the classes in the finally selected partition - chosen after inspecting the graph that displays the values of the objective function and after comparing the features of the candidate partitions if they are more than one - should really represent an intrinsic property of the set to be clustered.

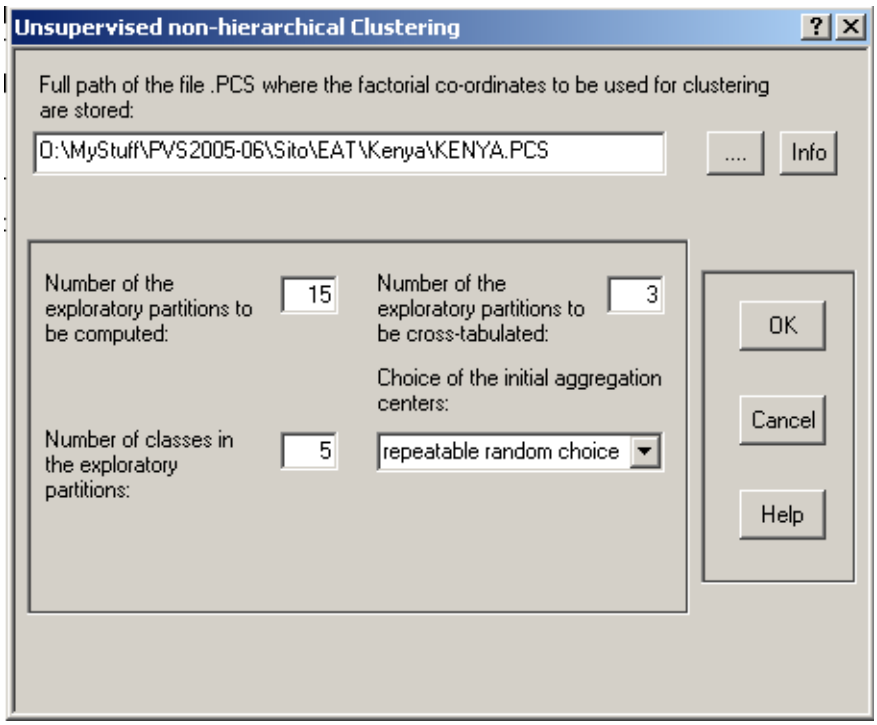
Of course, we can never be sure to have achieved the globally optimal partition with a given number of classes (the so called **optimum optimorum**) and probably we have not. Both methods are heuristic, and yield a partition of good quality, not the absolutely best one. But this is well known for combinatorial reasons, and we have to accept it.

7.3 – The NONGER dialogs

Use	It computes an optimal partition of the set of units to be clustered, using a non-hierarchical method.
Limits	In order to speed up the computation the data table is stored in central memory. To this purpose all the available core memory can be used, but remember that if the set of units to be clustered is very numerous, Windows will have recourse to virtual memory on disk, and things can become painfully slow. A maximum of **three exploratory partitions** can be cross-tabulated. The reason of this limitation is to prevent the user from creating a product-partition with a too high number of stable classes. To optimise iteratively a partition with 200 classes is something really slow! Try it to convince yourself... For what is possible, **try also not to exaggerate with the number of the classes you request**.

7.3.1 – Controlling the exploratory stage

The control parameters are entered in the following dialog.



Path of the .PCS file…

If the clustering immediately follows a factorial analysis, the path of the file where the requested factorial co-ordinates have been saved is displayed automatically in the control on top of the dialog. If the classification operates on factorial co-ordinates saved during a previous work session, the analyst must browse to point to the appropriate file.

From the PCS file NONGER will also read

- the name of the .TMP file that contains the values of the observed variables submitted to the factorial analysis (PCA or ACORR). They are necessary to compute the profiles of the classes after constructing them.
- the .FPL file with the information necessary to draw the projections onto factorial planes, that NONGER will modify adding some points representing the centres of the classes. The little squares that mark the location of the statistical units on the plane are replaced by the number of the class to which each unit has been assigned.

How many exploratory partitions?

Indicatively, 10-20. A higher number of ***base*** (or ***exploratory***) partitions is more likely to produce some good quality partitions to be cross-tabulated, but requires a proportionally longer computing time (anyway, this is no longer a problem with modern computers, unless your Dataset is very numerous). Memory occupation is not affected.

Obviously, if the units to be clustered are relatively few it is unnecessary to compute many partitions; if, on the contrary, the units are several thousands, it is convenient to increase the number of the partitions, in the hope to find a good local optimum. Unfortunately, this will increase the computational time…

How many exploratory partitions (the best ones) are to be cross-tabulated?

Amongst the base partitions, computed for exploratory purposes, the best ones (i.e. those with the highest value of the objective-function, their number being decided by the analyst) will be cross-tabulated in order to determine the most homogeneous groups emerging from the analysis (the so-called ***stable classes*** or ***strong forms***).

The choice about how many partitions should be cross-tabulated depends on the level of detail desired for the stable classes. For instance, if some seven-class partitions are requested, cross-tabulating two of them could produce (at least theoretically) up to 49 stable classes (i.e., 49 combinations of the two classes to which a unit has been assigned in the two partitions).

In general, the more strongly-structured the set, the more consistent the two partitions will be, and the number of stable classes will decrease accordingly. They would be seven if the two cross-tabulated partitions were identical.

In order to avoid an excessive fragmentation of the *product-partition*, which would make the interpretation more difficult, ADDATI accepts to cross-tab no more than 3 base partitions..

How many classes in each partition?

This number is tentative, and should represent the ideal number of groups that the user would like to obtain finally.

The number of the classes of the final partition will be eventually decided after a careful inspection of the diagram that shows how the objective function decreases when the number of the classes is reduced.

Options for the choice of the starting clustering centres:

1. **repeatable random choice**
2. **non repeatable random choice**

Some more ways of choosing the initial seeds, present in ADDATI 5.2, will probably be added in the future. Anyway, a random choice of the starting seeds is a good solution (the programme itself takes care of that). The two current alternatives have the following meaning.

Repeatable random choice

If n units are to be clustered, for each partition as many values between 1 and n are generated as there are classes, and the units with those ordinal numbers are assumed as the initial aggregation centres. The random generation starts from a **fixed seed** (a value that determines the sequence of the numbers drawn randomly). If the analysis is repeated, the same clustering centres are determined and the same sequence of partitions is produced.

Non repeatable random choice

In this case the "seed" that determines the sequence of random numbers varies, as it is derived from the computer's internal clock. Repeating the analysis will therefore generate different aggregation centres and will in general produce different results.

7.4 - NONGER – Optimisation stage and description of the partitions

Let us suppose that the product-partition generated in the exploratory stage consists of *q stable classes*: **NONGER** starts optimising this-partition by suitably re-allocating some units (Diday method). Then the two closest (most similar) groups are merged, and the resulting partition is also optimised. An optimal partition with *q-1* classes is thus produced. The same operation is iterated on this latter partition, yielding another partition with *q-2* classes, and so on until eventually an optimal partition with only two classes is obtained.

At this point ADDAWIN calls an internal utility to display the graph that plots, for this sequence of optimal partitions, the value of the objective function vs. the number of the classes. The value of the o.f. obviously decreases when the number of the classes decreases. The figure 7.5 refers to the Kenya example, in which 10 *stable classes* were determined.

By inspecting this graph the user can focus on just one or on several *promising partitions*, with an acceptable number of classes and a sufficiently high value of the objective function. Two possible partitions - with seven and four classes respectively - are pointed out in figure 7-4.

When choosing the final partition the user should consider the **trade-off** between the level of synthesis that can be achieved (few classes are always more convenient) and the value of the objective function, that represents the rate of information maintained. The number of the clusters should be reduced as much as possible, but the value of the o.f. should not decrease too much. It is advisable to merge further if the o.f. level of the resulting partition (the next on the right along the graph) is still satisfactorily high. Otherwise, the price to be paid in terms of information loss may result unaffordable.

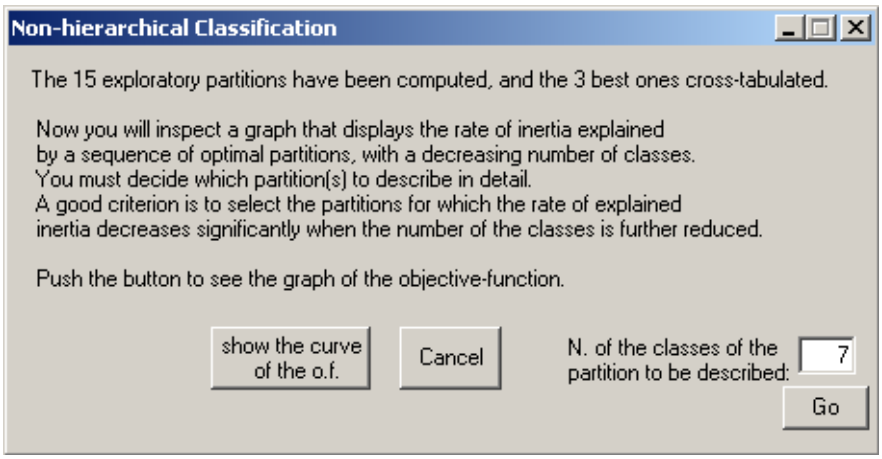
It is therefore convenient to select a partition such that the value of the objective function decreases significantly by further merging (i.e., the **slope of the curve changes quite abruptly** moving one step to the right).

The decision should take into account the three following aspects.

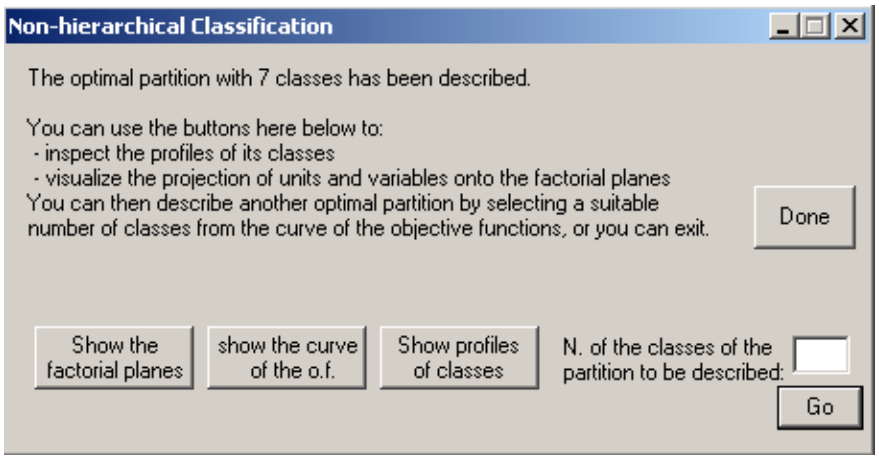
- The number of clusters that better fit the analyst's objectives should orient him/her towards the part of the curve to be more carefully considered.
- The decreasing of the value of the o.f. with each merging step should lead to focus on few ***candidate partitions***, to be described in detail.
- The final decision should be taken after examining carefully the features of the candidate partitions – specially their class profiles – choosing the one that better mirrors the objectives of the analysis.

After choosing one or more candidate partition(s) to be further investigated, the user closes the graph by selecting the option ***File→Exit*** from the Menu.

The dialog shown below is displayed, and the user is prompted to enter the number of classes of the optimal partition to be described. This request is repeated again and again, until all the partitions are described whose features are to be compared to reach the final decision.



Here we have requested to describe the partition with 7 classes. Immediately thereafter, the following dialog appears.



It is then possible to inspect the features of the requested partition (especially the profiles of the classes), to view the projections on the main factorial planes, to describe a partition with another number of classes, to review the graph, or to terminate.

7.4.1 – Examining the profiles of the classes

The profile of each class must be compared with the overall profile in order to assess which variables characterise it more, because of their values significantly higher or lower than the overall average. Some aids are written under the numeric values of the variables: their purpose is to ease the interpretation by focusing the user's attention on the most important variables in each cluster's profile. One of the following alphanumeric strings is written under each profile component:

"----", "--", "~~~", "++", "++++"

The string actually written is determined by the **ratio** between the concerned component and the corresponding component of the overall profile. The ratio **is computed for each component** of all class profiles, and compared with a set of thresholds values. The aid string is chosen according to the scheme shown in the two tables 7.1 and 7.2. The thresholds shown are the programme's default; suitable in most cases.

When the variables are QUANTITATIVE, the '+' and '-' signs used to facilitate the inspection of the class profiles have the following meaning. Let

xm(j,i) be the **average value** of variable j in class i

xg(j) be the **overall average value** of the same variable j.

The variable j is relevant for cluster i when the difference xm(j,i) - xg(j) is far from 0. In order to evaluate its significance, the difference is compared with the variable's standard deviation **σ(j)**. The value of the ratio

R = [xm(j,i) - xg(j)] / σ(j)

is then compared with four ***suitably chosen thresholds*** s1,..., s4 whose current values are shown here below. The value of the aid string is then determined according to the following scheme:

```
      ----        --        ~~~        ++       ++++     value
  ---------->|<---------->|<---------->|<---------->|<-------------   of the
          -1.00       -0.20       0.20       1.00                ratio R
```

Table 7.1 - The default thresholds proposed to help interpreting the profiles in the case of quantitative variables.

The + and - signs are used to facilitate the inspection of the class profiles. They are to be interpreted as follows.

The reference is to the **ratio** between the frequency of each variable in the cluster and its overall frequency. With the current threshold values such ratio is represented as follows:

```
----          --          ~~~          ++          ++++        value
-------------->|<--------------->|<--------------->|<--------------->|<----------------  of the
     -1.00        -0.20        0.20        1.00                ratio R
```

Table 7.2 - The default thresholds proposed to help interpreting the profiles in the case of one or more contingency tables.

Note: *If the table submitted to the analysis consists of quantitative variables, the default thresholds are automatically computed by the programme and depend,* ***for each variable****, from its standard deviation. While the classes are determined from the factorial co-ordinates, the profiles are computed using the original variables and the initial unit of measures. This makes the interpretation easier.*

Profile's components

The profile's components have a different meaning in the two cases.

- **Quantitative variables** (tables of measure): for each group, a profile component is the average value of a variable in the class; the corresponding component of the overall profile is the overall average of that variable.
- **Contingency tables**: a profile component represents ***the frequency*** in the class of a category of a qualitative variable. The corresponding component of the overall profile is the overall frequency of that variable.

As a general guideline, for quantitative variables the way of determining the aids to interpretation seems appropriate: their values should not be modified by the user.

For contingency tables it may sometimes happen that the various units (and therefore also the clusters) are little different, and the default thresholds do not seem appropriate (e.g., it may occur that in almost no group the variables' frequencies differ from the corresponding values of the overall profile by more than 20 per cent). ADDAWIN should then allow the analyst to change the thresholds conveniently (not yet implemented).

NONGER – The interpretation of the results

🗁 We shall illustrate the output with reference to the Kenya example used with the PCA (see section 6.2). Suppose to have clustered the districts on the basis of seven factorial co-ordinates (that explain 100 per cent of inertia), and enter the following values of the clustering parameters in the exploratory step:

- exploratory partitions to compute: 10
- no. of exploratory partitions (the best ones) to be cross-tabulated: 2
- no. of classes in each exploratory partition: 7
- repeatable random choice of the initial aggregation centres

In our Kenya analysis the exploratory step produces 10 stable classes, and the graph of figure 7.5 is displayed. The user can request the description of any optimal partition determined, ranging from 10 to 2 classes.

The results are saved to a file named **NGnn.TXT**.

After a summary of the parameters enter by the user to control the analysis, the output file reports the number of iterations that were necessary to achieve convergence for each partition computed, and the final value of the objective function.

For each partition whose description has been requested, the following information is written:

- a table like 7.3, which summarises the number of the groups in the partition, the number of units in each group and its weight. Remember that the weight was assumed proportional to the district's population.

10 STABLE CLASSES IN THE CROSS-TABULATED PARTITION											
CLASS	1	2	3	4	5	6	7	8	9	10	TOT
UNITS	10	7	6	6	3	2	2	2	2	1	148
WEIGHT (%)	31.7	19.5	12.8	11.8	2.9	1.3	6.1	7.6	1.9	4.5	100.0

Table 7.3 - Kenya example: the stable classes obtained in the exploratory stage.

- a detailed description of the classes (the table 7.4 shows an example)

```
**************
* CLASSE   6 *
**************

UNITS : 2    WEIGHT:  1.26 %
UNITS ASSIGNED TO THE CLASS :
Kajiado     Isiolo

UNIT CLOSEST TO THE CLASS CENTRE (d2 =  0.4111) : Kajiado
UNIT FARTHEST AWAY FROM THE CLASS CENTRE (d2 =  4.8285) : Isiolo
CLASS RADIUS : 1.18697
DISTANCE OF THE CLASS CENTRE FROM THE OVERALL CENTRE : 5.99533
```

Table 7.4 - An example of the information printed for each class

The units included in each class are listed, together with the values of the following indicators:

- **the class' radius**, which is an indicator of class compactness.

  Let $In_{int}(j)$ be the internal inertia of the *j*-th class, obtained by adding up the inertia of all the units belonging to the class, computed with respect to the class centre $\mathbf{G}_j$. This value can be written as

$$In_{int}(j) = \Sigma_i\, m_i * d^2(i,\mathbf{G}_j) = M_j * d_j^2$$

  where M_j is the weight of the class and d_j its average radius, i.e. the distance from G_j at which the class' mass should be distributed in order to have an Inertia equal to $In_{int}(j)$. It can be derived that

$$d_j = [In_{int}(j) / M_i]^{1/2}$$

- **the distance of the class centre from the overall centre** of the cloud, as an indicator of the peculiarity of the class' features: the greater this distance, the more different the average features of the class from the overall mean characters represented by the overall centre.

After describing the clusters, some more parameters are output:

- **the total Internal Inertia** In_{int};
- **the total external Inertia** In_{ext} ;
- **the overall Inertia**, equal to the sum of the eigenvalues associated with the factorial co-ordinates taken into account when clustering. For our Kenya example this value is 8.0, as we had eight normalised variables, and all the Principal Components had been passed on to the clustering procedure.
- the value of **the objective function**, that measures the quality of the partition and is equal to the ratio between the external and the total Inertia. Its maximum is 1, achieved only when there are as many classes as there are units.

The profiles of the classes follow (table 7.5 offers an example)

The descriptions of all the requested partitions are written sequentially to file **NGnnn.TXT**. For each of them, the information on the class to which the statistical units have been assigned is stored in **a text file** (in .CSV format) named '**NGCLASnn.CSV**', where 'nn' stands for the number of the classes. In case there should be any need to load this file in ADDATI and modify something, the documentation file that describes it, named NGCLAnn.TXT, is saved along. The file .CSV is compatible with ARC/VIEW, and can be used to draw a classified map when geographical units (districts, Census tracts, regions and similar) are clustered.

The profiles of the classes of each partition described are saved to a file named **NGCLASnn_XLS.CSV**, where as usually 'nn' is the number of the classes. The values are comma-separated, and the file can be directly loaded by EXCEL as a text file, to further process the profiles if necessary.

CLASS	NUM	fert_ rate	ave_ wage	hp_land	act_ rate	cereals	cattle	goa_sh	cash_ crop
1	23	6.804	69.490	0.033	4.092	0.138	0.051	0.040	0.150
		~~	~~	~~	--	~~	~~	~~	--
2	6	7.618	59.532	0.075	9.385	0.516	0.049	0.025	0.609
		++	--	++	~~	++	~~	~~	~~
3	4	6.321	81.480	0.056	3.832	0.210	0.389	0.237	0.000
		--	++	~~	--	~~	++++	++	--
4	3	5.705	71.839	0.041	3.517	0.000	0.117	0.412	0.000
		--	~~	~~	--	--	~~	++++	--
5	2	7.470	40.957	0.066	9.362	0.451	0.143	0.078	2.152
		++	----	~~	~~	++	++	~~	++++
6	2	5.861	126.930	0.003	38.011	0.000	0.001	0.000	0.193
		--	++++	--	++++	--	--	~~	--
7	1	6.920	67.470	0.019	11.090	0.100	0.013	0.017	2.460
		~~	~~	~~	++	--	~~	~~	++++
OVERALL PROFILE	41	6.835	71.217	0.038	7.963	0.191	0.063	0.053	0.428

Table 7.5 - Kenya example: the classes' profiles of the optimal partition with 7 classes.

The files written by NONGER

NONGER saves the following files, which record in various forms the results of an analysis.

- **NGnnn.TXT**, where '**nnn**' is a progressive number aimed at avoiding overwriting existing files. The file describes all the partitions requested. It lists the units assigned to each class, describes the profiles of the classes and offers some other information useful to the interpretation.
- **NGCLASnn.CSV** records the class to which each statistical unit has been assigned. '**nn**' is the number of the classes of the partition to which the file refers: a different file is saved for each partition.

  These files have as many records as there are units, plus one record of headings. Thanks to their format, they can be directly loaded as text files by ARCVIEW. Each record contains the identifier of the geographical unit **both as a string and as a number**, followed by the number of the class to which the unit is assigned. The information can be added to the original data archive or – in case geographical units have been clustered – can be used to create a map of the classification.

  If ArcMap is used instead of ArcView to create the map, the file should be an .XLS (EXCEL) binary file, not a .CSV one. To convert it, load the .CSV file in EXCEL by double-clicking on it, and save it immediately thereafter in .XLS format.

  It may occur that EXCEL does not load correctly the .CSV file saved by NONGER, which uses commas as field delimiters. Should this happen, load NGCLASnn.CSV in ADDAWIN (use its documentation file NGCLASnn.TXT, automatically written by NONGER), then save it choosing as separators semicolons. EXCEL, depending on its settings, interprets as separators COMMAS or SEMICOLONS, and should then load it correctly.
- **NGCLASnn_XLS.CSV** stores just the profiles of the classes in format .CSV, which should be accepted by EXCEL as a text file for further elaborations. A file of this type is written for each partition described.
- **NGnn.FPL** is a file written by **NONGER** for **FACPLAN**. This file is written only if the factorial analysis that precedes the clustering (**PCA** or **ACORR**) has saved the file necessary to visualise the projections. If NONGER finds this file, it reads its contents, adds some information and saves it as **NGnn.FPL**, where 'nn' is as usual the number of the clusters. When **FACPLAN** displays its contents, the location of each unit is no longer represented by a small square but by the number of the class to which the unit was assigned.

  Also the centres of the clusters are shown. This is another way of visualising how the classes (visible as sub-clouds of numbers '1', '2', etc.) are located with respect to the variables.