

## DtmVic - Tutorial C (new version, 2010)

# DtmVic and numerical data

**Five more examples to practise DtmVic with numerical data:  
Semiometry, Fisher's Iris data, Graphs, Images.**

*Each example corresponds to a directory included in the directory "DtmVic\_Examples\_NumDat" that has been downloaded with DtmVic.*

### Application examples C.1—C.5

[To select the examples, press the right button of the mouse](#)

#### **Example C.1.    EX\_C01. PrinCompAnalysis\_1** *(Visualization in Principal Components Analysis)*

Example C.1 aims at describing a set of numerical variables (an excerpt of semiometric data) through Principal Components Analysis. The principal axes visualisation is complemented by a clustering, with an automatic description of the clusters. Bootstrap procedures, Kohonen maps are followed by the various tools of visualisation provided in the sub-menu "**Visualization**" of the phase "VIC" : visualisation of clusters (or categories) using symbols or colours, convex hulls or density ellipses for clusters, Minimum spanning tree, drawing of various nearest neighbours graphs...

#### **Example C.2.    EX\_C02. PrinCompAnalysis\_2** *(Visualization in Principal Components Analysis)*

Example C.2 is a variant of Example C.1. It aims at describing the same set of numerical variables (the same excerpt of semiometric data) through Principal Components Analysis. The principal axes visualisation is complemented by a clustering of variables. Bootstrap procedures, Kohonen maps are followed by the various tools of visualisation provided in the menu "Contiguity view": visualisation of clusters of variables using symbols or colours, convex hulls or density ellipses for clusters, Minimum spanning tree drawn on variable-points, drawing of various nearest neighbours graphs...

#### **Example C.3.    EX\_C03. PCA-Contiguity** *(PCA and Contiguity Analysis on Fisher's Iris Data)*

Example C.3 aims at analysing a classical set of numerical variables (The Iris data set of Anderson and Fisher) through Principal Components Analysis, Classification, Contiguity Analysis, Discriminant Analysis. The principal axes visualisation is complemented by a clustering, with an automatic description of the clusters. At the outset, example C.3 is very similar to example C.2: Principal components analysis and classification (clustering) of a set of numerical data, with various tools of visualisation, involving also a specific categorical data. It presents then the improvements provided by Contiguity Analysis and its particular case: Linear Discriminant Analysis.

**Example C.4.    [EX\\_C04. Graphs](#)**  
*(Description of graphs through Correspondence Analysis)*

Example C.4 aims at describing three simple symmetrical planar graphs, mainly through correspondence analysis. Unlike the previous examples, the directory EX\_04.Graphs contains several sub-directories and examples. The three graphs are planar graphs: a chessboard shaped graph, a cycle, and an empirical graph supposed to roughly represent a map of the regions of Japan.

The examples provide a bridge between distinct facets of DtmVic: a same graph can lead to different input data : classical numerical data, textual data, and a specific “external format”.

**Example C.5.    [EX\\_C05. Images](#)**  
*(Structural Compression of Images through SVD and CA)*

Example C.5 could be viewed as a pedagogical appendix. It does not make use of data in DtmVic format, since it deals with digitalized images. A simple rectangular array of integers suffices: there is no need for identifiers of rows or column. A specialized interface is provided via the button “DtmVic Tools” of the main menu.

## Example C.1: **EX\_C01. PrinCompAnalysis\_1** (*Visualization in Principal Components Analysis*)

Example C.1 aims at describing a set of numerical variables (an excerpt of “semiometric data”) through Principal Components Analysis. The principal axes visualisation is complemented by a clustering, with an automatic description of the clusters. Bootstrap procedures, Kohonen maps are followed by the various tools of visualisation provided in the menu “Visualization” in the sub-window “Visualization, Inference, Classification”: visualisation of clusters (or categories) using symbols or colours, convex hulls or density ellipses for clusters, Minimum spanning tree, drawing of various nearest neighbours graphs.

### About Semiometric data:

In most surveys in the field of marketing research, it is customary to include information about lifestyles and values. Such information is generally obtained through a set of questions describing the attitudes and opinions towards a list of sentences or statements. "Semiometry" is a technique introduced by Jean-François Steiner, a writer interested in marketing research, to tackle that problem in a more general way.

The basic idea is to insert in the questionnaire a series of questions consisting uniquely of **words** (a list of 210 words is currently used, but we will be dealing here with an abbreviated lists containing a subset of 70 words). The interviewees must rate these words according to a seven levels scale, the lowest level (mark = 1) relating to a "most disagreeable (or unpleasant) feeling about the **word**", the highest level (mark = 7) relating to a "most agreeable (or pleasant) feeling" about the **word**.

The processing of the filled questionnaires (mainly through Principal Component Analysis) produces a stable pattern (up to 8 stable principal axes). Very similar patterns are obtained in ten different countries, despite the problems posed by the translation of the list of words.

For more information, please refer to the book: “La sémiométrie” (2003) [in French] by L. Lebart, M. Piron, J.F. Steiner; Publisher: Dunod, Paris. This book can be downloaded from the site: [www.dtm-vic.com](http://www.dtm-vic.com) (section “publication”).

To have a look at the data, search for the directory **DtmVic\_Examples**.

In this directory, open the sub-directory **DtmVic\_Examples\_C\_NumData** .

In that directory, open the directory of Example C.1, named **“EX\_C01. PrinCompAnalysis\_1”**.

It is recommended to use one directory for each application, since DtmVic produces a lot of intermediate txt-files related to the application. At the outset, such directory must contain 3 files :

- a) the data file,
- b) the dictionary file,
- c) the command file.

#### **a) Data file: “PCA\_semio.dat.txt”**

Our reduced-size example comprises 300 respondents (instead of 1000 or 2000 that are the usual sizes of semiometric survey samples) and 76 variables: 70 words (the marks given to the words are considered here as numerical variables) and 6 categorical variables describing the characteristics of the respondents.

The data file "PCA.dat.txt" comprises 300 rows and 76 columns (identifier of rows [between quotes] + 75 values [corresponding either to numerical variables or to item numbers of categorical variables] separated by at least one blank space).

**b) Dictionary file: "PCA\_semio.dic.txt"**

The dictionary file "PCA.dic.txt" contains the identifiers of these 76 variables. In this version of DtmVic, the identifiers of categories must begin at: "column 6" [a fixed interval font - also known as teletype font - such as "courier" can be used to facilitate this kind of format].

**c) Command file: "PCA\_semio.par1.txt"**

The computational phase of the analysis is decomposed into "steps". Each step requires some parameters briefly described in the main menu of DtmVic (button: "**Help about parameters**") and below.

Note that another "command file" similar (but not identical) to the "command file" **PCA\_semio.par1.txt** can be also generated by clicking on the button "**Create the command file**" of the main menu (Basic Steps). Proceed than as shown by the first example "EX\_A01.PrinCompAnalysis" of Tutorial A.

## Running the example C.1 and reading the results

- 1) Click on the button: "**Open an existing command file**" (panel *Basic Steps* of the main menu)
- 2) Search for the sub-directory "**DtmVic\_Examples\_C\_NumData**" in "**DtmVic\_Examples**".
- 3) In that directory, open the directory of Example C.1: "**EX\_C01.PrinCompAnalysis\_1**".
- 4) Open the command file: "**PCA\_semio.par1.txt**"

After identifying the two data files, 10 "steps" are identified: **ARDAT** (Archiving data), **SELEC** (selecting active and supplementary elements), **STATS** (some basic statistics), **PRICO** (Principal components analysis), **DEFAC** (Brief description of factorial axes), **RECIP** (Clustering using a hierarchical classification of the clusters - reciprocal neighbours method), **PARTI** (Cut of the dendrogram produced by the previous step, and optimisation of the obtained partition), **DECLA** (Automatic description of the classes of the partition), **SELEC** (selecting one categorical variable, in this case), **EXCAT** (extracting one categorical variable - selected by step **SELEC** - to be used in some graphical displays).

In this example, as in most applications, the step **SELEC** plays a fundamental role, in deciding which set of variables will be active, and which set will be illustrative or supplementary.

In that command file, the step **RECIP** performs a hierarchical clustering of the elements using the "reciprocal neighbour algorithm" and the step **PARTI** that follow cuts the obtained tree according to the *a priori* fixed number of clusters. **PARTI** optimizes afterwards the corresponding partition through k-means iterations. The methodology of this "hybrid algorithm" is presented in "Multivariate Descriptive Statistical Analysis" (L. Lebart, A. Morineau, K. Warwick; J. Wiley, New York, 1984). See also [in French] "Statistique Exploratoire Multidimensionnelle" (4<sup>th</sup> printing, L. Lebart, M. Piron, A. Morineau, Dunod, Paris, 2006).

We will comment later on this command file (Appendix of the section) which commands the basic computation steps. Instead of editing this file, we will content ourselves here in going back to the main menu and executing the basic computation steps.

**5) Return to the main menu (“return to execute”)**

**6) Click on the button: “Execute”**

This step will run the basic computation steps present in the command file.

**7) Click the button: “Basic numerical results”**

The button opens a created (and saved) html file named **“imp.html”** which contains the main results of the previous basic computation steps. After perusing these numerical results, return to the main menu. Note that this file is also saved under another name. The name **“imp.html”** is concatenated with the date and time of the analysis (continental notation). That file keeps as an archive the main numerical results whereas the file **“imp.html”** is replaced for each new analysis performed in the same directory.

This file is also saved under a simple text format, under the name **“imp.txt”**, and likewise with a name including the date and time of execution.

**8) At this stage, we click on one of the lower buttons of the basic steps panel (Steps: “VIC”)**

**9) Click the button: “Axesview”** ... and follow the sub-menus. In fact, only two tabs are relevant for this example: **“Active variables”** [= the selected words], **“Individuals (observations) [= respondents]”** and **“supplementary categories”**. After clicking on **“View”** in each case, one obtains the set of principal coordinates along each axis.

Clicking on a column header produces a ranking of all the rows according to the values of that column. In this particular example, this is somewhat redundant with the printed results of the step **“DEFAC”**. Evidently, the use of the **Axeview** menu is justified when the data set is large, which is the case here.

In the case of this particular example, in which the first axes appear to be stable and to have an interpretation, the **AxesView** procedure is useful to observe at a glance the bundles of words occupying extreme locations along each axis. Example for active variables and axis 2: opposition between the words “sacred, God, perfection, soul” on the one hand, and “sensual, adventurer, nudity, island, desire” on the other. The supplementary categories characterize the respondents. The bootstrap, later on, will provide a validation of some aspects of the observed structure.

**10) Click the button: PlaneView ...** and follow the sub-menus.

In this example, four items of the menu are relevant **“Active columns (variables or categories)”** and **“supplementary categories”**, **“Active rows (individuals, observations)”**, **“Active columns + Active rows”**, **“Active individuals (density)”** and **“Active columns + Supplementary categories”**. The graphical display of chosen pairs of axes are then produced.

In the case of semiometric data, the so-called “first semiometric plane” is in fact the plane spanned by the axes 2 and 3. The first axis is referred to as a “purely methodological axis”, linked to a “size effect” common in many PCA applications (a whole chapter of the [downloadable] book quoted previously: “La sémiométrie” is devoted to this first axis).

In the case of PCA, the first menu item **“Active columns (variables or categories)”** may contain, in fact, both active numerical variables (in black) and supplementary numerical variables (in red). We have only active numerical variables in this particular example, but, later on, the reader can edit the command file (step **SELEC**) to withdraw some words from the active set and give them the status of “supplementary (or illustrative) elements”. He or she can also use the **“Create a command file”** menu, exemplified in Tutorial A, example A.1, to select more comfortably the active and supplementary elements.

Go back to the “VIC” set of buttons.

## 11) Click the button: “BootstrapView”

This button opens the DtmVic-Bootstrap-Stability windows.

11.1 Click “LoadData”. In this case (partial bootstrap), the two replicated coordinates file to be opened are named “ngus\_var\_boot.txt” and “ngus\_sup\_cat\_boot.txt” (see the small panel reminding the names of the relevant files below the menu bar).

In fact, “ngus\_var\_boot.txt” contains only active variables. The file “ngus\_sup\_cat\_boot.txt” contains only supplementary categories, for which the bootstrap procedure is also meaningful.

11.2 Click on “Confidence Ellipses” submenu, and choose the pair of axes to be displayed (choose axes 2 and 3, to begin with).

11.3 Click on “Loading” in the blue window that appears then, to obtain the dictionaries of variables. Tick the chosen white boxes to select the elements the location of which should be assessed, and press the button “Select”. Select for instance among others, the categories Male and Female

11.4 Click on “Confidence Ellipses” to obtain the graphical display of the active variable points (if the file ngus\_var\_boot.txt has been loaded), or of the supplementary category points (if the file ngus\_sup\_cat\_boot.txt has been loaded).

11.5 Close the display window, and, again in the blue window, press “Convex hulls”. The ellipses are now replaced by the convex hulls of the replicates for each point. The convex hulls take into account the peripheral points, whereas the ellipses are drawn using the density of the clouds of replicates. The two pieces of information are complementary.

Go back to the “VIC” set of buttons.

## 12. Click on “ClusterView ”

12.1 Choose the axes (2 and 3 to begin with), and “Continue”.

12.2 Click on “View”. The centroids of the 7 clusters of individuals (Step PARTI) appear on the first principal plane.

12.3 Activate the button “Categorical”, and, pointing with the mouse on a specific cluster, press the right button of the mouse. A description of the cluster involving the most characteristic response items appears. This description is somewhat redundant with that of the Step DECLA. But we can watch on this display the pattern of clusters and their relative locations. One can easily imagine the usefulness of the tool for a survey with thousands of individuals, hundreds of variables, and more clusters.

12.4 Activate the button “Numerical”. We will observe the link between the numerical variables (both active and supplementary variables) of the data file and the 5 clusters. Due to the small number of individuals, some clusters do not produce significant results.

Go back to the “VIC” menu.

## 13) Click on “Kohonen map”

Select the type of coordinate.

13.1 Select: “Active variables (columns)”: these active variables are the 70 words in this example.

13.2 Select a (4 x 4) map, and continue.

13.3 After clicking on two small check-boxes, press “draw” on the menu of the large green windows entitled Kohonen map.

13.4 You can change the font size (“Font”) and dilate the obtained Kohonen map (“Dilat.”) to make it more legible. The words appearing in the same cell are often associated in the same responses. This property holds, at a lesser degree, for contiguous cells.

13.5 Pressing **“AxeView”**, and selecting one axis allows one to enrich the display with pieces of information about a specified principal axis : large positive coordinates in red colour, large negative coordinates in green, with some transitional hues.

13.6 Go back to the main menu, click on **“Kohonen map”** and choose the item **“Observations”**

13.7 Select a (8 x 8) map, and redo the operations 13.3 to 13.5 for the observations.

Go back to the “VIC” set of buttons.

## 14. Click on **“Visualization”**

14.1 A new window entitled **“DTM-Visualization: Loading files, Selecting axes”** appears.

14.2 Click on **“Load coordinate”**

14.3 In the corresponding sub-menu, choose the file: **“ngus\_ind.txt”**. The principal coordinates of the individuals (rows) are selected.

14.4 Click then on **“Load or create Partition”**

14.5 In the corresponding sub-menu, choose **“Select a partition”**. The partition obtained previously from the computation step must then be loaded (its name: **“part\_cla\_ind.txt”**).

14.6 Click on **“MST”** (Minimum Spanning Tree). Choose then the number of axes that will serve to compute the Minimum Spanning Tree: 5 (for example).

14.7 Click on **“Load MST”**, to load the results for the forthcoming visualisation phase.

14.8 Click on **“N.N.”** (search for nearest neighbours – limited to 20 NN).

14.9 Click on **“Load N.N.”** (loading the nearest neighbours file)

14.10 Click on **“Visualisation”**.

14.11 Choose the axes 1 and 2 (default) in the small window **“Selection of Axes”** and click on **“Display”**.

14.12 In the new window entitled **“Contiguity\_Visualisation”** are displayed the individuals in the plane spanned by the selected axes. A random colour is attributed to each cluster. The button **“Change colour”** allows you to try a new set of colour. When you estimate that the colours are sufficiently contrasted, you can press **“Lock colour”**.

14.13 About the window **“Contiguity\_Visualisation”**

14.13.1 -- On the vertical tool bar, you can press each button to activate it (red colour), and press it again to cancel the activation (black colour)

14.13.2 -- The button **“Density”**, for sake of clarity, replace the identifiers of individuals by a single character reminding the cluster (the identifier and the cluster number can be obtained by clicking on the left button of the mouse in the vicinity of each point).

14.13.3 -- The button **“C.Hull”** (Convex hull) draws the convex hull of each cluster.

14.13.4 -- The button **“MST”** (Minimum Spanning Tree) draws the minimum spanning tree.

14.13.5 -- The button **“Ellipse”** perform a Principal Components Analysis of each cluster within the two-dimensional sub-space of visualisation and draws the corresponding ellipses (containing roughly 90% of the points).

14.13.6 -- The button **“N.N.”** (Nearest neighbours) joins each point to its nearest neighbours. Pressing afterwards the button **“N.N. up”** allows you to increment the number of neighbours up to 20

nearest neighbours.

14.13.7– We will see in example C.2 how to use the lower buttons of the left side vertical bar “IterKM”, “Mean”, “Clust” (useful to visualize the iteration of a k-means partition).

Go back to the “VIC” menu.

## 15. Click again on “Visualization”

15.1 We are going to redo the operation of paragraph 14, but instead of loading a partition provided by a clustering algorithm, we will load the partition induced by the categories of a specific categorical variable. Such partition correspond to the variable number 76 (gender), selected and extracted through the steps **SELEC** and **EXCAT** (at the end of the command file, see below).

15.2 In the window entitled “DTM-Visualization: loading files, selecting axes “, click on “**Load coordinate**”

15.3 In the corresponding sub-menu, choose again the file: “**ngus\_ind.txt**”. The principal coordinates of the individuals (rows) are selected.

15.4 Click then on “**Load or create Partition**”

15.5 In the sub-menu “**Load or create Partition**” choose the file “**part\_cat.txt**”. The partition induced by the categories of variable number 76 (gender) is loaded.

After loading that partition, all the operations from 14.6 to 14.13 can be carried out again. It is interesting to visualise the individuals in the plane spanned by the axes 2 and 3.

*Comment:* The two categories Male and Female are significantly linked to axis 3 (as it can be highlighted by looking at the bootstrap confidence areas). But this link is hardly visible when we look directly at the convex hulls of the two sub-clouds corresponding to these two categories of respondents. This (almost) paradoxical result exemplifies the difference between “statistically significant” (which is the case here) and “obviously different” (which is not the case).

## Appendix C.1 (for advanced users)

**A similar (but not identical) command file can be generated using the menu “Create parameters”. Therefore, beginners could skip this appendix**

The computational phase of the analysis is decomposed into "steps". Each step requires some parameters briefly described in the main menu of DtmVic (button: "**Help about parameters**").

Let us remind that this set of commands comprises 10 steps:

**ARDAT** (Archiving data), **SELEC** (selecting active and supplementary elements), **PRICO** (Principal components analysis) **DEFAC** (Description of factorial axes) , **RECIP** (Clustering using a hierarchical classification of the clusters - reciprocal neighbours method), **PARTI** (Cut of the dendrogram produced by the previous step, and optimisation of the partition obtained), **DECLA** (Automatic description of the classes of the partition), **SELEC** (selecting one categorical variable, in this case), **EXCAT** (extracting one categorical variable - selected by step **SELEC** - to be used in some graphical displays). Now, we will display the command file that contains comments (preceded by #). Comments are also allowed without “#” in the (mandatory) line that immediately follows a statement "STEP xxxxx"

```
#-----  
# Example C.1 of principal component analysis  
#-----  
# continuation symbol = ">", Comments symbol = "#"  
# title mandatory immediately after each line "STEP"
```

```

LISTP = yes, LISTF = no, LERFA = yes # Global Parameters
NDICZ = 'PCA_semio.dic.txt'          # dictionary file
NDONZ = 'PCA_semio.dat.txt'          # data file

```

```

STEP ARDAT          #reading the dictionary
===== builds the Archive Dictionary
NQEXA = 76, NIDI = 1, NIEXA = 300
#----- Comments about step ARDAT
# - NQEXA = ... number of questions (or variables)
#   in both the dictionary and the data file
# - NIEXA = ... number of "individuals" (or rows)
#   in the data file.
# - NIDI = 1...indicate the presence of an identifier
#-----

```

```

STEP SELEC          # Selection of variables
===== Selecting active and supplementary variables
LSELI = TOT, IMASS = UNIF, LZERO = NOREC, LEDIT = short
CONT ACT 1--70
NOMI ILL 71--76
END
#----- Comments about step SELEC
# - LSELI = ... Parameter describing the selection
#   of individuals, the value FILT (or: 2) means that
#   the rows are selected through a "filter".
#   This filter is defined after the closing key-word "END"
# - IMASS = ... weight (or: mass) of the individuals
#(rows); the value 0 (or: UNIF) means "uniform"(same weights)
# - LZERO = REC ... means that the value 0, which
# indicates a missing value, will be recoded as an
# extra response item for the categorical variables.
# - CONT ILL (absent) means "continuous illustrative
# variable". This key-word is followed by the list
# of the variables numbers.
# - NOMI ILL means illustrative nominal (or:
# categorical) variable. This key-word is followed
# by the list of the numbers of the variables.
# - CONT ACT means active variable. This key-word
# is followed by the list of the numbers of the variables.
# Note that, for example, 6-12 means: 6 7 8 9 10 11 12.
# The key-word END indicates the end of the list.
#-----

```

```

STEP PRICO
===== Principal component analysis
LCORR = 2, NAXE = 12, LEDIN = 0, NAVIR = 2 nsimu = 10 nboot = 1
#----- Comments about step PRICO
# - LCOOR = 1 means that the variables will be standardized
# (the correlation matrix will be diagonalized)
# - NAXE = ... number of computed principal axes
# - LEDIN = 1: the principal coordinates of individuals are printed
# - NSIMU...number of bootstrap replication (<=30)
#   (0 = no bootstrap)
# - NBOOT = 1 : partial bootstrap.
#-----

```

```

STEP DEFAC          # Description of factorial axes
===== Describing axes

```

```

SEUIL = 40., LCRIM = VTEST, VTMIN = 2.
VEC = 1--6 / CONT / MOD
END
#----- Comments about step DEFAC
# SEUIL = ... Maximum number of elements that will
#   be sorted to describe each axis
# LCRIM = ... Criterion for sorting the elements
#   (here VTEST means "test-values" (signed number
#   of standard deviations)
# VTMIN = elements with a test-value < VTMIN are
#   not printed.
# VEC = ... list of axes to be described
# CONT = continuous variables , MOD = categories
#   The key-word END indicates the end of the list.
#-----

STEP RECIP          # hierarchical classification
==== Clustering of respondents using reciprocal neighbours
NAXU=7 LDEND=DENSE NTERM=20 LDESC=no

#----- Comments about step RECIP
# This step carries out a hierarchical clustering
# using the reciprocal neighbours technique (recommended
# when dealing with less than 1000 individuals.
# - NAXU...   number of axes kept from the previous MCA
# - NTERM...  number of kept terminal elements
#   TOT means that all the elements are kept.
# - LDESC...  describing nodes of the tree (0=no, 1=yes).
# - LDEND...  printing dendrogram (0=no, 1=dense, 2=large).
#-----

STEP PARTI          # partition
===== cutting the dendrogram and improving the partition
NITER = 10, LEDIN = 6
7 # list of the numbers of clusters required (here one cut, 5 clusters)
#----- Comments about step PARTI
# - NITER...  number of "consolidation" iterations (0=no).
# - LEDIN...  printing the correspondences classes-
# individuals (3 = printing of the
# correspondence classes->individuals and the
# correspondence individuals-->classes).
# The line immediately following the command must
# contain the sizes of the desired final partition
# (here: 7).
#-----

STEP DECLA          # Description of partitions
===== Systematic description of clusters
CMODA = 5.0, PCMIN = 2.0, LSUPR = yes, CCONT = 5.0 >
LPNOM = NO, EDNOM = NO, EDCON = NO
7 # list of partitions (characterised by they numbers of clusters)
#----- Comments about step DECLA
# - CMODA...  describing classes with categories
# (0=no; CMODA = 5.0 means a p-value <= 0.05 for
# the selection of characteristic categories).
# - PCMIN...  minimum relative ( % ) weight for a
# category (categories whose relative weight is
# <=2% are discarded).

```

```

# - LSUPR...   characteristic category if
#   %(cat./class) > %(cat./total) (0=no,1=yes).
# (LSUPR = yes means that only characteristic
# elements will be printed)
# - CCONT...   describing classes with numerical
# variables. (0=no; CCONT = 5.0 means a
# p-value <= 0.05 for the selection of
# characteristic variables).
# - LPNOM...   describing partition with questions
# (i.e: whole set of categories) (0=no, 1=yes).
# - EDNOM...   printing the tables crosstabulating
# (classes * questions) (0=no).
# - EDCON...   describing partition globally with
# numerical variables (0=no, 1=yes),

#-----

STEP SELEC
===== Selecting one categorical variable
LSELI = TOT,  IMASS = UNIF, LZERO = NOREC, LEDIT = short
NOMI ACT   76
END
#----- Comments about this second call to SELEC
# The variable 76 is selected, to feed the following step.
#-----

STEP EXCAT
===== Extracting categorical variable 76 (on file part_cat.txt)
#-----
# This particular step without parameter serves to enrich some displays
# in the menu Visualization.
#
#-----
STOP                               # End of command file.

```

---

**End of example C.1**

## **Example C.2: [EX\\_C02.PrinCompAnalysis\\_2](#)** **(Principal Components Analysis with clustering of variables)**

Example C.2 is a variant of example C.1. It aims at describing the same set of numerical variables (an excerpt of “semiometric data”) through Principal Components Analysis. The principal axes visualisation is now complemented by a clustering of variables, instead of a clustering of individuals.

Another clustering of variables (or of observations/individuals) through a simple k-means method can be obtained and visualized from the sub-menu “Visualization”.

**Please, refer to example C.1 for a description of the data.**

### **Running the example C.2 and reading the results**

- 1) Click on the button: **“Open an existing command file”** (panel *Basic Steps* of the main menu)
- 2) Then, search for the sub- directory **“DtmVic\_Examples\_C\_NumData”** in the directory: **“DtmVic\_Examples”**.
- 3) In that directory, open the directory of Example C.2 : **“EX\_C02. PrinCompAnalysis\_2”**.
- 4) Open the command file: **“PCA\_semio.par2.txt”**

After identifying the two data files, seven "steps" are performed:

[ARDAT](#) (Archiving data), [SELEC](#) (selecting active and supplementary elements), [PRICO](#) (Principal components analysis), [DEFAC](#) (Brief description of factorial axes), [PERMU](#) (the data matrix is transposed), [RECIP](#) (Clustering of variables using a hierarchical classification of the clusters - reciprocal neighbours method), [PARTI](#) (Cut of the dendrogram produced by the previous step, and optimisation of the partition obtained). There are again 70 active numerical variables in this case : marks given to the words of the reduced semiometric list..

We will comment later on this command file (Appendix of the section) which commands the basic computation steps. Instead of editing this file, we will content ourselves here in going back to the main menu and execute the basic computation steps.

- 5) Return to the main menu (**“return to execute”**)

- 6) Click on the button: **“Execute”**

This step will run the basic computation steps present in the command file.

## 7) Click the button: **“Basic numerical results”**

The button opens a created (and saved) html file named **“imp.html”** which contains the main results of the previous basic computation steps. After perusing these numerical results, return to the main menu. Note that this file is also saved under another name. The name **“imp.html”** is concatenated with the date and time of the analysis (continental notation). That file keeps as an archive the main numerical results whereas the file **“imp.html”** is replaced for each new analysis performed in the same directory.

This file is also saved under a simple text format, under the name **“imp.txt”**, and likewise with a name including the date and time of execution.

8) At this stage, we click on one of the lower buttons of the basic steps panel (Steps: “VIC”)

## 9) Click directly on: **“Visualization”**

9.1 A new window entitled **“Visualization, Loading files, Selecting axes”** appears.

9.2 Click on **“Load coordinate”**

9.3 In the corresponding sub-menu, choose the file: **“ngus\_var\_act.txt”**. The principal coordinates of the **active variables** are selected.

9.4 Click then on **“Load or create Partition”**

9.5 In the corresponding sub-menu, choose **“Select a partition”**. The partition obtained previously from the computation step must then be loaded (its name: **“part\_cla\_var.txt”**).

9.6 Click on **“MST”** (Minimum Spanning Tree). Choose then the number of axes that will serve to compute the Minimum Spanning Tree: 5 (for example).

9.7 Click on **“Load MST”**, to load the results for the forthcoming visualisation phase.

9.8 Click on **“N.N.”** (search for nearest neighbours – limited to 20 NN).

9.9 Click on **“Load N.N.”** (loading the nearest neighbours file)

9.10 Click on **“Visualisation”**.

9.11 Choose the axes 1 and 2 (default) in the small window **“Description of classes”** and click on **“Display”**.

9.12 In the new window entitled **“Visualisation”** are displayed the individuals in the plane spanned by the selected axes. A random colour is attributed to each cluster. The button **“Change colour”** allows you to try a new set of colour. When you estimate that the colours are sufficiently contrasted, you can press **“Lock colour”**.

Go back to the “VIC” sub-menu.

## 10) Direct computation of a partition within the menu **“Visualization”**

DtmVic makes it possible now to build on line (i.e. outside the “command file”) a “k-means partition” of variables (or of individuals as well).

### Click on **“Visualization”**

10.1 A new window entitled **“Visualization, Loading files, Selecting axes”** appears.

10.2 Click on **“Load coordinate”**

10.3 In the corresponding sub-menu, if you want a clustering of variables, choose the file: **“ngus\_var\_act.txt”**. The principal coordinates of the **active variables** are selected. If you want a clustering of individuals, select the file: **“ngus\_ind.txt”**.

10.4 Click then on **“Load or create Partition”**. In the corresponding sub-menu, select the item **“Create a new k-means partition”**.

10.5 You have then to select the number of desired clusters, the number of principal coordinates to compute the distances, the maximum number of iteration (generally 12 iterations suffice).

You can also choose to visualize the iterations (answer: **yes** at the question **“Do you want to look at the intermediate iterations?”**).

10.6 The obtained partition will be automatically loaded and visualized. You can then redo the previous steps 9.6 to 9.12.

10.7 If you want to visualize the different steps, in the window **“Contiguity\_Visualisation”**, click on **“IterKM”**, then click alternately on **“Means”** (computation of the centroids of the clusters) and on **“Clust”** (assignment of the elements to the new centroid) until the convergence is reached.

Note that the partition obtained through the classical k-means algorithm generally will not coincide with the partition induced by the parameters of the command file. In that command file, as already explained in example C.1 (section 4), the step **RECIP** performs a hierarchical clustering of the elements using the “reciprocal neighbour algorithm” and the step **PARTI** that follow cuts the obtained tree according to the *a priori* fixed number of clusters. **PARTI** optimizes afterwards the corresponding partition through k-means iterations.

## Appendix C.2 (for advanced users)

The computational phase of the analysis is decomposed into "steps". Each step requires some parameters briefly described in the main menu of DtmVic (button: **“Help about parameters”**).

Let us remind that this set of commands comprises seven steps:

**ARDAT** (Archiving data), **SELEC** (selecting active and supplementary elements), **PRICO** (Principal components analysis), **DEFAC** (Brief description of factorial axes), **PERMU** (the data matrix is transposed), **RECIP** (Clustering of variables using a hierarchical classification of the clusters - reciprocal neighbours method), **PARTI** (Cut of the dendrogram produced by the previous step, and optimisation of the partition obtained). There are 70 active numerical variables in this case : marks given to the words of the reduced semiometric list..

Now, we will exhibit the command file that contains comments (preceded by #). Comments are also allowed in the (mandatory) line that immediately follows each statement "STEP xxxxx"

```
#-----  
# Example C.2 of principal component analysis  
#-----  
# continuation symbol = ">", Comments symbol = "#"  
# title mandatory immediately after each line "STEP"  
  
LISTP = yes, LISTF = no, LERFA = yes # Global Parameters  
  
NDICZ = 'PCA2_semio.dic.txt'           # dictionary file  
NDONZ = 'PCA2_semio.dat.txt'         # data file
```

```

STEP ARDAT          #reading the dictionary
===== builds the Archive Dictionary
  NQEXA = 76, NIDI = 1, NIEXA = 300
#----- Comments about step ARDAT ----- see Appendix C.1

STEP SELEC          # Selection of variables
===== Selecting active and supplementary variables
LSELI = TOT, IMASS = UNIF, LZERO = NOREC, LEDIT = short
CONT ACT 1--70
NOMI ILL 71--76
END
#----- Comments about step SELEC ----- see Appendix C.1

STEP PRICO
===== Principal component analysis
LCORR = 2, NAXE = 12, LEDIN = 0, NAVIR = 2  nsimu = 10  nboot = 1
#----- Comments about step PRICO ----- see Appendix C.1

STEP DEFAC          # Description of factorial axes
===== Describing axes
SEUIL = 40., LCRIM = VTEST, VTMIN = 2.
VEC = 1--6 / CONT / MOD
END
#----- Comments about step DEFAC ----- see Appendix C.1
#-----

STEP PERMU
===== permuting roles of individuals and variables
LEDIT = 1

#----- Comments about step PERMU
# ledit:  printing the number of permuted elements
#         (0=no, 1=yes).
#-----
NGUS = 'NFIC.999'
#-----
# The permuted file is named 'NFIC.999' (name purely
# conventional). The previous assignment replace the classical
# coordinate file NGUS with the permuted file.
#-----

STEP RECIP          # hierarchical classification
==== Clustering of respondents using reciprocal neighbours
NAXU=7 LDEND=DENSE NTERM=tot  LDESC=no
#----- Comments about step RECIP ----- see Appendix C.1

STEP PARTI          # partition
===== cutting the dendrogram and improving the partition
NITER = 10, LEDIN = 6
12 # list of the numbers of clusters required (here one cut, 5 clusters)
#----- Comments about step PARTI ----- see Appendix C.1
#-----
STOP                # End of command file.

```

---

**End of example C.2**

## Example C.3: [EX\\_C03.PCA\\_Contiguity](#) (PCA and Contiguity Analysis on Fisher's Iris Data)

Example C3 aims at analysing a classical set of numerical variables (The Iris data set of Anderson and Fisher) through Principal Components Analysis, Classification, Contiguity Analysis, Discriminant Analysis. The principal axes visualisation is complemented by a clustering, with an automatic description of the clusters.

Up to subsection 14 of Section I, Example C.3 is very similar to Example C.1 : Principal components analysis and classification (clustering) of a set of numerical data, with various tools of visualisation, involving also a specific categorical data.

Subsection 15 and the following subsections present the improvements provided by Contiguity Analysis.

### Reminder about Contiguity Analysis

In Contiguity analysis, we consider the case of a set of multivariate observations, ( $n$  objects described by  $p$  variables, leading to a  $(n, p)$  matrix  $\mathbf{X}$ ), having an *a priori* graph structure. The  $n$  observations are the vertices of a symmetric graph  $G$ , whose associated  $(n, n)$  matrix is  $\mathbf{M}$  ( $m_{ii'} = 1$  if vertices  $i$  and  $i'$  are joined by an edge,  $m_{ii'} = 0$  otherwise).

Such situation occurs when vertices represent time-points, geographic areas. Contiguity Analysis, confronting local and global variances, provides a straightforward generalization of Linear Discriminant Analysis. It enables to point out the levels responsible of the observed patterns (*local* versus *global* level).

In this example, we will deal with the situation in which  $\mathbf{M}$  and the graph structure are not external, but derived from the data matrix  $\mathbf{X}$  itself,  $\mathbf{M}$  being for example the *k-nearest neighbours graph* derived from a distance between observations.

Some interesting possibilities of exploration of data are sketched. Note that the idea of deriving a metric likely to highlight the existence of clusters dates back to the works of Art et al. (1982) and Gnanadesikan et al. (1982).

### Some references

- Art D., Gnanadesikan R., Kettenring J.R. (1982) Data Based Metrics for Cluster Analysis, *Utilitas Mathematica*, **21** A, 75-99.
- Burtschy B., Lebart L. (1991) Contiguity analysis and projection pursuit. In : *Applied Stochastic Models and Data Analysis*, R. Gutierrez and M.J.M. Valderrama, Eds, World Scientific, Singapore, 117-128.
- Gnanadesikan R., Kettenring J.R., Landwehr J.M. (1982) Projection Plots for Displaying Clusters, in *Statistics and Probability, Essays in Honor of C.R. Rao*, G. Kallianpur, P.R. Krishnaiah, J.K.Ghosh, eds, North-Holland.
- Lebart L. (1969) Analyse statistique de la contiguit . *Publications de l'ISUP*. XVIII, 81-112.
- Lebart, L. (2000): Contiguity Analysis and Classification, In: W. Gaul, O. Opitz and M. Schader (Eds):*Data Analysis*. Springer,Berlin, 233--244.
- Lebart L. (2006): Assessing Self Organizing Maps via Contiguity Analysis. *Neural Networks*, 19, 847-854.

## Looking at the data

To have a look at the data, search for the directory **DtmVic\_Examples**.

In this directory, open the sub-directory **DtmVic\_Examples\_C\_NumData**.

In that directory, open the directory of Example C.1, named **“EX\_C03. PCA\_Contiguity”**.

It is recommended to use one directory for each application, since DtmVic produces a lot of intermediate txt-files related to the application. At the outset, such directory must contain 3 files :

- a) the data file,
- b) the dictionary file,
- c) the command file.

### a) Data file: **“iris\_dat.txt”**

Our example comprises 150 observations and 4 variables: 4 measurements (these numerical variables are the lengths of various constituents of the flowers: *Sepal Length*, *Sepal Width*, *Petal Length*, *Petal Width*) and one categorical variable describing the characteristics of the observations (three species of plants : *setosa*, *versicolor*, *virginica*).

The data file "iris\_dat.txt" comprises 150 rows and 6 columns (the identifier of rows [between quotes] + 5 values [corresponding to 4 numerical variables and one categorical variable] separated by at least one blank space).

[Reference: Anderson, E. (1935). The irises of the Gaspé Peninsula, *Bulletin of the American Iris Society*, **59**, 2–5.]

### b) Dictionary file: **“iris\_dic.txt”**

The dictionary file "iris\_dic.txt" contains the identifiers of these 5 variables. In this version of DtmVic dictionary, the identifiers of categories must begin at: "column 6" [a fixed interval font - also known as teletype font - such as "courier" can be used to facilitate this kind of format].

### c) Command file: **“iris\_par.txt”**

The computational phase of the analysis is decomposed into "steps". Each step requires some parameters briefly described in the main menu of DtmVic (button: **"Help about parameters"**) and below.

Note that another "command file" similar (but not identical) to the "command file" **“iris\_par.txt”** can be also generated by clicking on the button **“Create the command file”** of the main menu (Basic Steps). Proceed than as shown by the first example **“EX\_A01.PrinCompAnalysis”** of Tutorial A.

## Running the example C.3 and reading the results

- 1) Click on the button: **“Open an existing command file”** (panel *Basic Steps* of the main menu)
- 2) Search for the sub-directory **“DtmVic\_Examples\_C\_NumData”** in **“DtmVic\_Examples”**.
- 3) In that directory, open the directory of Example C.3 named **“EX\_C03. PCA\_Contiguity”**.
- 4) Open the command file: **“iris.par.txt”**

In that command file, we can read that after identifying the two files (data and dictionary) , 9 "steps" are performed: **ARDAT** (Archiving data), **SELEC** (selecting active and supplementary elements), **PRICO** (Principal components analysis), **DEFAC** (Brief description of factorial axes), **RECIP** (Clustering using a hierarchical classification of the

clusters - reciprocal neighbours method), **PARTI** (Cut of the dendrogram produced by the previous step, and optimisation of the obtained partition), **DECLA** (Automatic description of the classes of the partition), **SELEC** (selecting one categorical variable, in this case), **EXCAT** (extracting one categorical variable: the species of iris - selected by step **SELEC** - to be used in some graphical displays).

We will comment later on this command file (Appendix of the section) which commands the basic computation steps. Instead of editing this file, we will go back to the main menu and execute the basic computation steps.

#### 5) Return to the main menu (“return to execute”)

#### 6) Click on the button: “Execute”

This step will run the basic computation steps present in the command file.

#### 7) Click the button: “Basic numerical results”

The button opens a created (and saved) html file named “**imp.html**” which contains the main results of the previous basic computation steps. After perusing these numerical results, return to the main menu. Note that this file is also saved under another name. The name “**imp.html**” is concatenated with the date and time of the analysis (continental notation). That file keeps as an archive the main numerical results whereas the file “**imp.html**” is replaced for each new analysis performed in the same directory.

This file is also saved under a simple text format, under the name “**imp.txt**”, and likewise with a name including the date and time of execution.

#### 8) At this stage, we click on one of the lower buttons of the basic steps panel (Steps: “VIC”)

#### 9) Click directly on the button: “BootstrapView”

This button opens the DtmVic-Bootstrap-Stability windows.

9.1 Click “**LoadData**”. In this case (partial bootstrap), the two replicated coordinates file to be opened are named “**ngus\_var\_boot.txt**” and “**ngus\_sup\_cat\_boot.txt**” (see the small panel reminding the names of the relevant files below the menu bar). In fact, “**ngus\_var\_boot.txt**” contains only active variables. The file “**ngus\_sup\_cat\_boot.txt**” contains only supplementary categories, for which the bootstrap procedure is also meaningful.

9.2 Click on “**Confidence Ellipses**” submenu, and choose the pair of axes to be displayed (choose axes 1 and 2, to begin with).

9.3 Click on “**Loading**” in the blue window that appears then, to obtain the dictionaries of variables. Tick the chosen white boxes to select the elements the location of which should be assessed, and press the button “**Select**”. Select all the four variables when you open the file “**ngus\_var\_boot.txt**”, and the three species when you open the file “**ngus\_sup\_cat\_boot.txt**”.

9.4 Click on “**Confidence Ellipses**” to obtain the graphical display of the active variable points (if the file **ngus\_var\_boot.txt** has been loaded), or of the supplementary category points (if the file **ngus\_sup\_cat\_boot.txt** has been loaded). We can observe, for the variables, that the “petal lengths” seem to be somewhat redundant, since their ellipses markedly overlap. We can see also that the three categories are significantly distinct (that does not mean that they can be linearly separated...).

9.5 Close the display window, and, again in the blue window, press “**Convex hulls**”. The ellipses are now replaced by the convex hulls of the replicates for each point. The convex hulls take into account the peripheral

points, whereas the ellipses are drawn using the density of the clouds of replicates. The two pieces of information are complementary.

[Go back to the “VIC” menu.](#)

## 10. Click on **“Visualization”**

A new window entitled **“Visualization, Loading files, Selecting axes”** appears.

**We are going to visualise the different species of flowers (categorical variable n° 5) in the plane spanned by the first principal components.**

10.1 Click on **“Load coordinate”**

10.2 In the corresponding sub-menu, choose the file: **“ngus\_ind.txt”**. The principal coordinates of the individuals (rows) are selected.

10.3 Click then on **“Load or create Partition”**

10.4 In the corresponding sub-menu, choose **“Select a partition”**. The partition obtained previously from the computation step must then be loaded (its name: **“part\_cat.txt”**). The partition induced by the 4 categories of variable number 5 (species of irises) is loaded. As in example C.2, this partition has been selected and extracted through the steps **SELEC** and **EXCAT** (at the end of the command file, see below).

10.5 Click on **“MST”** (Minimum Spanning Tree). Choose then the number of axes that will serve to compute the Minimum Spanning Tree: 5 (for example).

10.6 Click on **“Load MST”**, to load the results for the forthcoming visualisation phase.

10.7 Click on **“N.N.”** (search for nearest neighbours – limited to 20 NN).

10.8 Click on **“Load N.N.”** (loading the nearest neighbours file)

10.9 Click on **“Visualisation”**.

10.10 Choose the axes 1 and 2 (default) in the small window **“Description of classes”** and click on **“Display”**.

In the new window entitled **“Visualisation”** are displayed the individuals in the plane spanned by the selected axes. A random colour is attributed to each cluster. The button **“Change colour”** allows you to try a new set of colour. When you estimate that the colours are sufficiently contrasted, you can press **“Lock colour”**.

On the vertical tool bar, you can press each button to activate it (red colour), and press it again to cancel the activation (black colour)

-- The button **“Density”**, for sake of clarity, replace the identifiers of individuals by a single character reminding the cluster (the identifier and the cluster number can be obtained by clicking on the left button of the mouse in the vicinity of each point).

-- The button **“C.Hull”** (Convex hull) draws the convex hull of each cluster.

-- The button **“MST”** (Minimum Spanning Tree) draws the minimum spanning tree.

-- The button **“Ellipse”** perform a Principal Components Analysis of each cluster within the two-dimensional sub-space of visualisation and draws the corresponding ellipses (containing roughly 95% of the points).

-- The button **“N.N.”** (Nearest neighbours) joins each point to its nearest neighbours. Pressing afterwards the button **“N.N. up”** allows you to increment the number of neighbours up to the 20 nearest neighbours.

At this step, we have obtained a display of the 150 individuals, with either the convex hulls (or the ellipses) corresponding to the three species. This is the classical display in the principal plane of PCA, showing that the first species (number < 51) are well separated from the species 2 and three, species that are overlapping.

Go back to the “VIC” menu.

## 11. Click again on “Visualization”

We are going to redo the operation of paragraph 10, but instead of loading a partition induced by the 4 categories of variable number 5 (species of irises), we will load a partition produced by a clustering algorithm. Such partition correspond to the steps **RECIP** and **PARTI** (see the command file, below).

11.1 Click on “**Load coordinate**”

11.2 In the corresponding sub-menu, choose the file: “**ngus\_ind.txt**”. The principal coordinates of the individuals (rows) are selected.

11.3 Click then on “**Load or create Partition**”

11.4 In the corresponding sub-menu, choose “**Select a partition**”. The partition obtained previously from the computation step must then be loaded (its name: “**part\_cla\_ind.txt**”). The partition derived from the steps **RECIP** and **PARTI**.

**After loading that partition, all the operations from 10.6 to 10.10 can be carried out again.**

It is interesting to visualise the individuals in the plane spanned by the axes 1 and 2.

**As suspected, the partition obtained directly from the numerical measurements, ignoring the species, is unable to separate the three species. Only the species “setosa”, well separated from the two other species, coincides with a cluster of the partition.**

Go back to the “VIC” menu.

## 12. Click now on the button: “Contiguity”

**We are now going to perform a “Contiguity analysis” using a “nearest neighbours graph” derived from the data. The partition into species is not taken into account.**

12.1 Click on “**Parameters/Edit**”

Choose the item “**Create**”

We are going to enter the parameters needed by a contiguity analysis:

- In the first block entitled “**ncoord = input coordinate file**”, tick “1” (File **ngus\_ind.txt**: coordinates of individuals). The contiguity analysis will use the coordinates of individuals as input data.

- In the second block entitled “**npart = partition file**”, tick “1” (no partition)

- In the third block entitled “**meth = method**”, tick “2” (Contiguity graph defined by nearest neighbours).

- Then we will have to enter the following numerical values :

- **Npas** = 2 (increment for the number of nearest neighbours)
- **Nmin** = 4 (minimum number of nearest neighbours)
- **Nmax** = 8 (maximum number of nearest neighbours)

- Three contiguity analysis will be performed for the three (symmetrised) graphs corresponding

respectively to 4, 6, 8 neighbours (from  $N_{min} = 4$  up to  $N_{max} = 8$ , with an increment of  $N_{pas} = 2$ ).

Then: Click on: **“Validate”**. A summary of the parameters appears. This summary can be saved, to be able to repeat an identical analysis, if needed .

Click on **“Save the command file”** (choose any name with the extension “txt”).

12.2 In the upper bar of the window, Click on **“Execute”**. The computations are carried out.

The item **“Results”** of that bar contains technical details about the computations involved in Contiguity analysis.

12.3 Click on **“Contiguity View”**. We are led to the same window of visualisation than previously.

In the menu **“Load coordinates”**, of the new window, choose the item: **ngus\_contig.txt** file. Instead of using the principal coordinates of PCA (**ngus\_ind.txt** as done previously), we use now the result of the Contiguity Analysis **ngus\_contig.txt**.

From the menu **“Select Partition”**, choose the file: **part\_cat.txt**. (we will identify the species)

We cannot compute the Minimum Spanning Tree nor the Nearest Neighbours from the **“ngus\_contig.txt”** coordinates, but we can load the results (MST and NN) obtained previously (section 10) that are automatically saved.

12.4 Click on **“Visualization”**.

Then choose the axes 1 and 2 (default values)

Choose (tick) the contiguity level number 2, that correspond to 6 nearest neighbours. (level 1 corresponds to 4 nearest neighbours, and level 3 to 8 nearest neighbours).

Click on **“Display”**.

Change the colours if necessary.

Click on **“Convex Hull”** (vertical bar)

The three species are now better separated.

**That means that the (“symmetrised”) graph of 6 nearest neighbours allows for computing a “local covariance matrix” that can act as a “within covariance matrix”. In this example, the principal plane of a contiguity analysis is similar to the principal plane of a Fisher Discriminant Analysis.**

**We must keep in mind that the contiguity analysis did not use the *a priori* knowledge about the species. It is an *unsupervised* method.**

[Go back to the “VIC” menu.](#)

### 13. Click again on **“Contiguity”**

**We are now going to perform a “Contiguity analysis” that coincide exactly with a classical Linear Discriminant Analysis.**

(Linear Discriminant Analysis is a particular case of Contiguity Analysis. In such a case, the graph involved in Contiguity Analysis is made of k cliques (complete graphs) corresponding to the k classes of the Discriminant Analysis).

13.1 Click on **“Parameters/Edit”**

Choose the item **“Create”**

We are going to enter the parameters needed by a contiguity analysis:

- In the first block entitled **“ncoord = input coordinate file”**, tick “1” (File: **ngus\_ind.txt**: coordinates of individuals). The contiguity analysis will use the coordinates of individuals as input data.

- In the second block entitled “**npart = partition file**”, tick “2” (**part\_cat.txt**, categorical) (this partition will now be used to derive a graph).

- In the third block entitled “**meth = method**”, tick “3” (Classical Discriminant Analysis).

- In this case, the following parameters are meaningless. DtmVic asks you to skip them.

- The contiguity analysis will be performed using the graph associated with the partition into species. (all pairs of individual belonging to the same species are joined by an edge; no edge between individuals belonging to different species)

Then: Click on: “**Validate**”. A summary of the parameters appears. This summary can be saved, to leave the possibility of performing again an identical analysis.

Click on “**Save the command file**” (choose any name with the extension “txt”).

13.2 In the upper bar of the window, Click on “**Execute**”. The computations are carried out.

The item “**Results**” of that upper bar contains technical details about the computations involved in Contiguity analysis. The matrix associated with the graph with its three diagonal blocks of “1” and with the value “0” elsewhere is visible in this listing of results.

13.3 Click on “**Contiguity View**”.

In the menu “**Load coordinates**”, of the new window, choose the file: **ngus\_contig.txt**.

In the menu “**Select Partition**”, choose the file: **part\_cart.txt** (we will identify the “species of iris”)

We cannot compute the Minimum Spanning Tree nor the Nearest Neighbours from the “**ngus\_contig.txt**” coordinates, but we can load the results obtained previously (section 13) that are automatically saved.

13.4 Click on “**Visualisation**”.

Then choose the axes 1 and 2 (default values)

Click on “**Display**”.

Change the colours of the display if necessary to obtain a good contrast between classes, then lock the colours.

Click on “**Convex Hull**” (vertical bar)

**The three species of iris are well separated, too. But this is less of a surprise, since the Linear (Fisher) Discriminant Analysis aims precisely at separating the classes. We are here in a supervised case. The method uses the *a priori* knowledge of the species of iris to exhibit the coordinates (discriminant functions) that induce the best separation of the classes.**

## Appendix C.3 (for advanced users)

An identical command file can be generated using the menu “**Create parameters**” (see Tutorial A, exemple A.1) Therefore, beginners could skip this appendix.

Since the steps are the same as those of Example C.1, the listing of parameters will not be commented here.

Let us remind that this set of commands comprises 9 steps:

ARDAT (Archiving data), SELEC (selecting active and supplementary elements), DEFAC (Description of factorial axes), RECI (Clustering using a hierarchical classification of the clusters - reciprocal neighbours method), PARTI (Cut of the dendrogram produced by the previous step, and optimisation of the partition obtained), DECLA (Automatic description of the classes of the partition), SELEC (selecting one categorical variable, in this case), EXCAT (extracting one categorical variable - selected by step SELEC - to be used in some graphical displays).

```

#-----
# Example C.3 of principal component analysis
#-----
# continuation symbol = ">", Comments symbol = "#"
# title mandatory immediately after each line "STEP"
#-----
#
  LISTF = NO, LERFA = yes          # Global Parameters

NDICZ = 'iris_dic.txt'           # dictionary file
NDONZ = 'iris_dat.txt'           # data file
#-----
# Description of IRIS DATA through PCA, then, classification (unsupervised)
# into 3 groups (evidently, these groups will not coincide with the 3 real
# groups [species], as described by the categorical variable number 5).
#-----
# SEE COMMENTS IN APPENDIX C.1
#-----
STEP ARDAT                #reading dictionary and data
===== builds the Archive Dictionary
  NQEXA = 5, NIDI = 1,  NIEXA = 150

STEP SELEC                # Selection for PCA
===== Selects active, supplementary variables and observations
LSELI = 0, IMASS = UNIF, LZERO = REC, LEDIT = short
NOMI ILL 5
CONT ACT 1--4
end

STEP PRICO                # Principal Components Analysis
=====
LCORR = 1, NAXE =4, LEDIN = 1,  NSIMU = 10 nboot = 1

STEP DEFAC                # Description of factorial axes
===== Principal Component Analysis
SEUIL = 40., LCRIM = VTEST, VTMIN = 2.
VEC = 1--2 / CONT / MOD
end

STEP RECI                 # hierarchical classification
===== Clustering using reciprocal neighbours algorithm
NAXU = 4, NTERM = TOT, LDESC = NO, LDEND = DENSE

STEP PARTI                # partition
===== Cut of the dendrogram and optimization
NITER = 4, LEDIN = 3
3 # list of the numbers of clusters required (here one cut, 3 clusters)

STEP DECLA                # Description of partitions
===== Systematic description of clusters
CMODA = 5.0, PCMIN = 2.0, LSUPR = yes, CCONT = 5.0 >
LPNOM = NO, EDNOM = NO, EDCON = NO
  3 # list of partitions (characterised by they numbers of clusters)

```

```
STEP SELEC          # Selection of one nominal variable
===== Selects active, supplementary variables and observations
LSELI = 0, IMASS = UNIF, LZERO = REC, LEDIT = short
NOMI act 5
end

STEP EXCAT         # Selection of the previous nominal variable
=====
# no parameter

STOP              # End of command file.
```

---

**End of example C.3**

## Example C.4: EX\_C04.Graphs (Description of graphs)

Example C.4 aims at describing four simple symmetrical planar graphs from their associated matrices, mainly through correspondence analysis. Unlike the previous example directories, the directory EX\_C04.Graphs contains several sub-directories and examples.

### Section 1 : Overview of the different directories and files

#### 1.1 Search for the examples directory **DtmVic\_Examples**

#### 1.2 In that directory, open the directory of Example C.4, named **EX\_C04.Graphs**.

This directory comprises four sub-directories. The sub-directory named **“Chessboard”** relates to the description of a “chessboard shaped graph” (49 nodes corresponding to a square chessboard with 7 rows and 7 columns, the associated matrix being a 49 x 49 binary matrix).

The sub-directory named **“Cycle”** similarly relates to the description of a “cycle shaped graph” (49 nodes).

The sub-directory named **“Geography”** concerns the description of graphs associated with geographical maps (graphs of contiguous regions in Japan recorded under *textual form*, graphs of contiguous “departments” of France recorded under both *textual form* and “external form”).

#### 1.3 Open the sub-directory named **“Chessboard”**.

##### 1.3.1 Open the sub-sub-directory **“Chessboard\_numerical”**:

The file: **“Chessboard\_7x7\_dat.txt”** contains the data set representing the incidence matrix of the graph, with 49 rows and 49 columns. Like any classical data set of DtmVic, each row begins with its identifier. The entry cell  $m(i, j)$  of such a matrix **M** has the value 1 if the nodes  $i$  and  $j$  are joined by an edge, 0 otherwise. The identifiers of columns are to be found in the associated dictionary file: **“Chessboard\_7x7\_dic.txt”**.

That file will be analysed through Correspondence analysis (command file: **“Chessboard\_CA.Par.txt”**) and also, through Principal Component Analysis (command file: **“Chessboard\_PCA.Par.txt”**) for the sake of a comparison. The comparison is not favourable to PCA in this particular case. [see, e.g.: *Exploring textual Data* (1998), by L. Lebart, A. Salem, L. Berry, Kluwer Academic Publisher].

##### 1.3.2 In the sub-sub-directory **“Chessboard\_textual”**:

The file: **“Chessboard\_textual\_7x7.txt”** contains the same basic information under a quite distinct form: the format relates to responses to open ended questions. Each node of the graph is considered as a respondent, answering to the fictitious open-ended question: “ Please, tell me which are your neighbours”. Instead of a binary matrix **M**, we are dealing here with a much smaller data matrix containing the address (column numbers) of the “1” in the matrix **M**. The command file **“Chessboard\_Textual.Par.txt”** leads to the same results as those from the correspondence analysis of the previous paragraph, using however a quite distinct sequence of DtmVic steps. It is a “pedagogical example” of bridge between numerical and textual steps of DtmVic. In this type of data, the numbers are not considered as numbers in the mathematical meaning of the term, but as mere sequences of characters. [See below the example of the maps of Japan and France ].

1.3.3 The file: “**Chessboard\_Extern\_7x7.txt**” is another possible coding of the Chessboard graph, similar to the previous textual file. But in this case, the number are effectively read as integers, not as simple sequences of characters. The first line of the data set contains the number of nodes (49), then the length of the identifiers (4) and the maximum degree of the graph (upper bound of the numbers of edges adjacent to a single node) (10). Note that each row terminates with the dummy value 0.

Such specific format, the most compact one, can lead directly to a description of the graph in the sub-menu “Contiguity” of DtmVic, without command file.

## 1.4 Open the sub-directory named “**Cycle**” .

This sub-sub-directory is the counterparts of the one relating to the chessboard graph. Only the shape of the graph is different. The textual coding and the PCA command file are omitted in this case.

## 1.5 Open the sub-directory named “**Geography**” .

The three sub-sub-directories files are the counterparts of those relating to the chessboard textual example. The directories “**Japan\_text**” and “**France\_text**” exemplify the “textual coding” in the case of a maps describing the different regions of Japan and the departments of France.

In the case of Japan, for example, the two first lines of the file Japan\_map\_text.txt set indicate that the provinces of *Akita* and *Iwate* are contiguous to the province of *Aomori*, etc. The file “**Japan\_text\_par.txt**” is the corresponding command file. It is identical to the file “**Chessboard\_Textual.Par.txt**”, except for the name of the input data set.

## Section 2 : Running the example “**Chessboard\_numerical**”

Click on the button “**Open an existing command file**” (panel *Basic Steps* of the main menu)

### 2.1 Reach again the “sub-sub-directory”: “**Chessboard\_numerical**”:

We are in the framework of either a classical correspondence analysis or a Principal Components Analysis.

- a) **Data file:** “**Chessboard\_7x7\_dat.txt**”
- b) **Dictionary file:** “**Chessboard\_7x7\_dic.txt**”.
- c) **Command file:** “**Chessboard\_CA.Par.txt**” [Correspondence Analysis]  
“**Chessboard\_PCA.Par.txt**” [Principal Components Analysis]

Note that other “command files” similar to the previous ones, can be easily generated by clicking on the button “**Create the command file**” of the main menu (Basic Steps). A window “**Choosing among some basic analysis**” appears. Click then either on the button : **SCA** – Simple correspondence analysis– or on the button **PCA** –Principal components analysis - both of them located in the paragraph “**Numerical data**”, and follow the instructions as shown in Tutorial A.

We will start with correspondence analysis.

### 2.2 Open the command file: “**Chessboard\_CA.Par.txt**”

After identifying the two data files, four "steps" are performed:  
**ARDAT** (Archiving data), **SELEC** (selecting active and supplementary elements), **AFCOR** (Correspondence analysis). (See, e.g., : Example A.2)

## 2.3 Return to the main menu (“return to execute”)

### 2.4 Click on the button: “Execute”

This step will run the basic computation steps present in the command file: archiving data and dictionary, selection of active elements, correspondence analysis of the selected table.

### 2.5 Click the button: “Basic numerical results”

The button opens a created (and saved) html file named “**imp.html**” which contains the main results of the previous basic computation steps. After perusing these numerical results, return to the main menu. Note that this file is also saved under another name. The name “**imp.html**” is concatenated with the date and time of the analysis (continental notation). That file keeps as an archive the main numerical results whereas the file “**imp.html**” is replaced for each new analysis performed in the same directory.

This file is also saved under a simple text format, under the name “**imp.txt**”, and likewise with a name including the date and time of execution.

2.6 At this stage, we click on one of the lower buttons of the basic steps panel (Steps: “VIC”)

2.7 Click directly on the button: “**Visualization**” (we skip here the buttons **AxeView**, **PlaneView**, etc.)

#### We are going to visualise the graph

2.7.1 A new window named “**DTM-Visualization: Loading files, Selecting axes**” appears.

2.7.2 Click on “**Load coordinate**”

2.7.3 In the corresponding sub-menu, choose the file: “**ngus\_ind.txt (individuals)**”. The principal coordinates of the individuals (rows) are selected. [Since the data matrix is symmetrical, it is equivalent to choose “**ngus\_var\_act.txt**”].

2.7.4 Click then on “**Select or create a Partition**”

2.7.5 In the corresponding sub-menu, choose “**No Partition**”.

2.7.6 Click on “**MST**” (Minimum Spanning Tree). Choose then the number of axes that will serve to compute the Minimum Spanning Tree: 8 (for example).

2.7.7 Click on “**Load MST**”, to load the results for the forthcoming visualisation phase.

2.7.8 Click on “**N.N.**” (search for nearest neighbours – limited to 20 NN).

2.7.9 Click on “**Load N.N.**” (loading the nearest neighbours file)

2.7.10 Click on “**Visualisation**”.

2.7.11 Choose the axes 1 and 2 (default) in the window “**Selection of axes**” and click on “**Display**”.

2.7.12 A new window entitled “**Contiguity-Visualisation**” is displayed.

2.7.13 About the window “**Contiguity-Visualisation**”

In the window entitled “**Contiguity-Visualisation**” are displayed the nodes in the plane spanned by the selected axes. A random colour is attributed to the display. The button “**Change colour**” allows you to try a new set of colour. When you are satisfied with the colours, you can press “**Lock colour**”.

On the vertical tool bar, you can press each button to activate it (**red** colour), and press it again to cancel the activation (**black** colour)

- ❑ The button “**Density**”, for sake of clarity, replace the identifiers of nodes by a single character.
- ❑ The button “**C.Hull**” (Convex hull) is irrelevant here.

- ❑ The button **“MST”** (Minimum Spanning Tree) draws a possible minimum spanning tree.
- ❑ The button **“Ellipse”** is not relevant here.
- ❑ The button **“N.N.”** (Nearest neighbours) joins each point to its nearest neighbours. Pressing afterwards the button **“N.N. up”** allows you to increment the number of neighbours up to the 20 nearest neighbours.
- ❑ The button **“Graph”** is irrelevant here. It will be used later (Chessboard Extern and France Extern).

Important: in this particular application, the Minimum Spanning Tree and also the nearest neighbours are computed from the coordinates of the nodes in a space spanned by the first components.

## 2.8 Go back to the main menu.

## 2.9 Redo all the operations 2.2 to 2.7, opening now, during step 2.2, the command file: **“Chessboard\_PCA.Par.txt”** (Principal Components Analysis).

It will be seen through this example that PCA is less faithful than CA vis-à-vis the description of the graph structure.

## Section 3 : Running the example **“Chessboard\_textual”**

Click on the button **“Open an existing command file”** (panel *Basic Steps* of the main menu)

### 3.1 Open the **“sub-sub-directory”**: **“Chessboard\_textual”**:

We are in the framework of a textual analysis similar to the one of example which aimed at describing the responses to an open ended question in a sample survey (examples A.4, A.5, B.1 to B.4).

We find in this directory the text file and the command file.

(in this particular context, there are neither data file nor dictionary file: the questionnaire comprises one pseudo open-ended questions, posed to each node: ” Which are your neighbouring nodes ?”)

#### 3.1.1) Text file: **Chessboard\_textual\_7x7.txt**

The format is the same as in Example B.4. Since the responses may have very different lengths, separators are used to distinguish between individuals (or: respondents). Individuals (here: nodes) are separated by the chain of characters “----“ (starting column 1) possibly followed by an identifier.

#### 3.1.2) Command file: **Chessboard\_Textual.Par.txt**

The computational phase of the analysis is decomposed into "steps". Each step requires some parameters briefly described in the main menu of DtmVic (button: **“Help about parameters”**).

### 3.2) Open the command file: **Chessboard\_Textual.Par.txt**

After identifying the input textual data file, four "steps" are performed:

**ARTEX** (Archiving texts), **SELOX** (selecting the open question), **NUMER** (numerical coding of the text), **ASPAR** (correspondence analysis of the [sparse] contingency table “respondents - words”).

We will not comment on this command file which commands the basic computation steps (see Example B.1 and B.4). Instead of editing this file, we will content ourselves here in going back to the main menu and execute the basic computation steps.

Note that such a command file can be generated by clicking on the button **“Create the command file”** of the main menu (Basic Steps). A window **“Choosing among some basic analysis”** appears. Click then either on the button : **VISURESP**, both of them located in the paragraph **“Textual data”**, and follow the instructions as shown in Tutorial A.

Note also that in this simple data case (only one fictitious “open question”), it is possible to consider each response as a text. In such a case, the separators “----“ should be replaced with “\*\*\*\*\*”, as in example A.4 of Tutorial A. Instead of the analysis **“VISURESP”**, it is then necessary to perform the analysis **“VISUTEX”**.

### 3.3) Return to the main menu (**“return to execute”**)

### 3.4) Click **“Execute”**

This step will run the basic computation steps present in the command file: archiving text, correspondence analysis of the lexical table.

### 3.5) Click the **“Basic numerical results”** button

The button opens a created (and saved) html file named **“imp.html”** which contains the main results of the previous basic computation steps. After perusing these numerical results, return to the main menu. Note that this file is also saved under another name. The name **“imp.html”** is concatenated with the date and time of the analysis (continental notation). That file keeps as an archive the main numerical results whereas the file **“imp.html”** is replaced for each new analysis performed in the same directory.

This file is also saved under a simple text format , under the name **“imp.txt”**, and likewise with a name including the date and time of execution.

From the step NUMER, we learn for instance that we have 49 responses, with a total number of words (occurrences or token = here: edges of the graph) of 217, involving 49 distinct words (here: neighbours). Note that each node has been considered as its own neighbour.

### 3.6 Click the **PlaneView** button ...

and follow the sub-menus...

In this example, four items of the menu are relevant **“Active columns (variables or categories)”**, **“Active rows (individuals, observations)”**, **“Active columns + Active rows”**, **“Active individuals (density)”**. The graphical display of chosen pairs of axes are then produced.

### 3.7 Click on **“Visualization”**

All the steps of the previous section 2.7 could be carried out likewise.

## Section 4 : Running the example **“Chessboard\_Extern”**

There are *neither command file, nor dictionary file* in this directory, since the specific type of coding of the graph (**“external coding”**) provides a direct entry into the **“Contiguity”** menu.

In the menu **“ Visualization, Inference, Classification”**, click on the button: **“Contiguity”**.

#### 4.1 Click on “Parameters/Edit”

Choose the item “Create”

We are going to enter the parameters needed by a graph description:

- In the first block entitled “**ncoord = input coordinate file**”, tick “0”: “No coordinate file (simple description of an external graph)”.
- In the second block entitled “**npart = partition file**”, tick “1” (No partition)
- In the third block entitled “**meth = method**”, tick “4” (External contiguity graph ).

Then: Click on: “Validate” (as prompted by a message).

Click on “Save the command file” (choose any name with the extension “txt”). The parameter should be saved in the same directory as the external graph file (as suggested by a pop up message).

#### 4.2 In the upper bar of the window, Click on “Execute”.

A new window appears, and you are asked to choose the external graph file. It is in this example the file:

“Chessboard\_Extern\_7x7.txt”.

The computations are carried out.

The item “Results” of that bar contains some technical details about the computations involved in the correspondence analysis of the associated matrix  $M$  (These results are saved in the file “imp\_contig.txt”).

#### 4.3 Click on “Visualisation ”.

In the menu “Load coordinates”, of the new window, choose the file: **anagraf.txt**. (graph view through Correspondence Analysis)

In the menu “Select Partition”, choose the item: **No partition**.

We can compute and load the Minimum Spanning Tree or the Nearest Neighbours from the “anagraf.txt” file coordinates, choosing for instance 12 axes (maximum number allowed in this version).

#### 4.4 Click on “Visualisation”.

Then choose the axes 1 and 2 (default values)

Click on “Display”.

Change the colours if necessary.

**Once again, all the steps of the previous section 2.7 could be carried out likewise.**

#### 4.5 About the window “Contiguity-Visualisation”

To represent the edges of the original graph, click on the button “ExtG” (External Graph) of the vertical bar.

Open then again the file “Chessboard\_Extern\_7x7.txt”.

Click on the button “Graph”.

The button “Graph” produces the original graph as recorded in the file.

This allows you to observe the distortion of the planar graph in the spaces spanned by the axes 3 to 12. It is the multidimensional Guttman effect [See Benzécri, (1973),(in French) “L’analyse des données”, Tome II B, Chapter 10, “Sur l’analyse de la correspondance définie par un graphe”, pp 244-261].

## **Section 5 : Running the example “Cycle\_Numerical”**

This section is identical to Section 2 (Running the example “Chessboard\_Numerical”). The graph has now the shape of a cycle, with the same number of nodes.

The homologues of the files “Chessboard\_7x7\_dat.txt” “Chessboard\_7x7\_dic.txt” and “Chessboard\_CA\_Par.txt” are now respectively “Cycle\_49\_dat.txt”, “Cycle\_49\_dic.txt” and “Cycle\_CA\_Par.txt”. They can be found within the directory “Cycle” in the directory “Cycle\_numerical”.

## **Section 6 : Running the example “Cycle\_Extern”**

This section is identical to Section 4 (Running the example “Chessboard\_Extern”). The graph has now the shape of a cycle, with the same number of nodes.

The homologue of the directory “Chessboard\_Extern” is : “Cycle\_Extern” whereas the homologue file of “Chessboard\_7x7\_Extern.txt” is “Cycle\_Extern\_49.txt”.

## **Section 7 : Running the example “Japan\_map”**

This section is identical to Section 3 (Running the example “Chessboard\_Textual”). The graph is now a sketch of a map of Japan, presented as a set of responses to the open question “Which are your neighbouring regions”, the “respondents” being the same regions of Japan...

The homologue of the directory “Chessboard\_Textual” is : “Japan\_map” whereas the homologue files of “Chessboard\_textual\_7x7.txt” and “Chessboard\_textual\_Par.txt” are respectively : “Japan\_map\_Textual.tex.txt” and “Japan\_map\_Textual.Par.txt”.

## **Section 8 : Running the example “France\_map”**

This section is identical to Section 3 (Running the example “Chessboard\_Textual”). The graph is now a sketch of a map of Japan, presented as responses to the open question “Which are your neighbouring regions”, the “respondents” being the same regions of Japan...

The homologue of the directory “Chessboard\_Textual” is : “France\_map” whereas the homologue files of “Chessboard\_textual\_7x7.txt” and “Chessboard\_textual\_Par.txt” are respectively : “France\_map\_Textual.tex.txt” and “France\_map\_Textual.Par.txt”.

---

**End of example C.4**

## Example C.5: EX\_C05. Images

### ***(Structural Compression of Images through SVD and CA)***

Example C.5 is mainly a pedagogical example which serves as an illustration for the compression effect of principal axes techniques (Singular Value Decomposition and Correspondence Analysis) in the (rather unexpected for DtmVic users) domain of image analysis.

It does not make use of data in internal DtmVic text format, since it deals with digitalized images. A simple rectangular array of integers suffices: there is no need for identifiers of rows or column.

In fact, three particular formats will be used: rectangular arrays of levels of gray (simple text format), plain “pgm” format (acronym derived from "Portable Gray Map") and for color images, plain “ppm” format (acronym derived from "Portable Pixel Map")

A specialized interface is provided via the button “DtmVic Tools” of the main menu.

#### ***About the data***

To have a look at the data,

1) Search for the directory “DtmVic\_Examples”.

2) Search for the sub-directory “DtmVic\_Examples\_C\_NumData” in “DtmVic\_Examples”.

3) In that directory, open the directory of Example C.5: “EX\_C05. Images”.

Five sub-directories correspond to five examples:

“1\_Cheetah\_txt”,

“2\_Baalbeck\_pgm”,

“3\_Cardinal\_ppm\_color”,

“4\_Chessboard\_like\_txt”,

“5\_Extra\_pgm\_ppm”

All these file can be examined via a text editor (such as “notepad” included in Windows, or a free software such that “notepad++”, or “TotalEdit”, etc.).

--> For grayscale images, two input format are available:

1) Simple text format :

The data table contains positive integers  $\leq 255$  that are the values of the level of grey for each pixel (no identifiers). Examples 1 and 4: the image “cheetah.txt” (adapted from "The Data Compression Book", Mark Nelson, M&T Publishing Inc., 1992) and the artificial image “Chessboard\_4x4\_txt. Such a format that does not contain explicitly the size of the image is the simplest one. Because of its rusticity, it is neither used nor provided by the usual image processing software.

2) The pgm txt. format (portable grayscale map) (look at the example: "Baalbeck.pgm", using a text editor or a notepad)

The PGM format is a simple and transparent grayscale file format.

The difference in the plain format is:

There is one image in a file (general pgm format can cope with several images).  
The first line contains the format identifier: P2.  
The second and the third lines contain three integers: number of columns, number of rows, and the maximum value (255).  
Then the table is displayed row-wise.  
Each pixel in the table is represented as an ASCII decimal number (<255).  
Each pixel in the table has at least one white space before and after it.  
No line should exceed 72 characters.

For more information about such a format, please consult (e.g.): <http://netpbm.sourceforge.net/doc/pgm.html>

Here is an example of a small image in the plain PGM format.

--> For color (small) images, the input format is the ppm text format (portable pixel map). Look at the example "Cardinal.ppm",  
via a text editor or a notepad.

-----  
Both pgm and ppm files can be obtained through an exportation from the free software "Open office", using a jpeg file as an input.

The PPM format is a lowest common denominator color image file format.

The name "PPM" is an acronym derived from "Portable Pixel Map." Images in this format (or a precursor of it) were once also called "portable pmaps."

**4) Open the sub- directory: "Cheetah\_txt".**

**5) Within "Cheetah\_txt": open the file: "Cheetah.txt".**

- 1) Click on one of the "read" button and open the image file .
- 2) Select the number of axes.
- 3) Select a method (CA or SVD)
- 4) Click on the "Image" button adapted to the selected image.

Note that all the created bitmap files are saved in the folder that contains the data.

-----

---

**End of example C.5**

**End of tutorial C**