Chapter 6. Softwares

by G. BOUXIN°

Contents

Introduction

This chapter presents programs and packages, mainly those for working in our developed conception of data analysis and out of ordination methods. We now present multivariate analysis software only in R language.

° rue des Sorbiers, 33 B.5101 Erpent Belgium mail : guy.bouxin@proximus.be

We first present several softwares and programs in R language [those proposed respectively by HUSSON et al. (2009), by the University of Lyon, by JEMBER (2012) and by LOMBARDO, R. & BEH (2019)]. Next come our personal programs that have been especially adapted to the analysis of phytosociological tables, with the lines corresponding to species and the columns to relevés, unlike other R programs that put variables in columns and individuals in columns. Programs are planned to easily transform relevé tables, with the usual abundance-dominance data (i, +, r, 1, 2, 3, 4 and 5) into numerical tables suitable for statistical analysis.

The GINKGO package from the Barcelona University is also illustrated. Some other packages are mentioned.

As examples, we use the file Royer taken from the literature (with the agreement of the author), a second personal file called Crupetvegenv113 with floristic abundance data and qualitative and quantitative environmental data. The easiest method is to enter the tables with a spreadsheet, here Excel and save the tables in .xls or in .txt format for example.

The R system

The R language is more and more used and is indispensable in data analysis (R Core Team, 2018). It offers a multitude of functions and graphical possibilities. It is free and can be downloaded from the following address:
http://cran.r-projec.org/

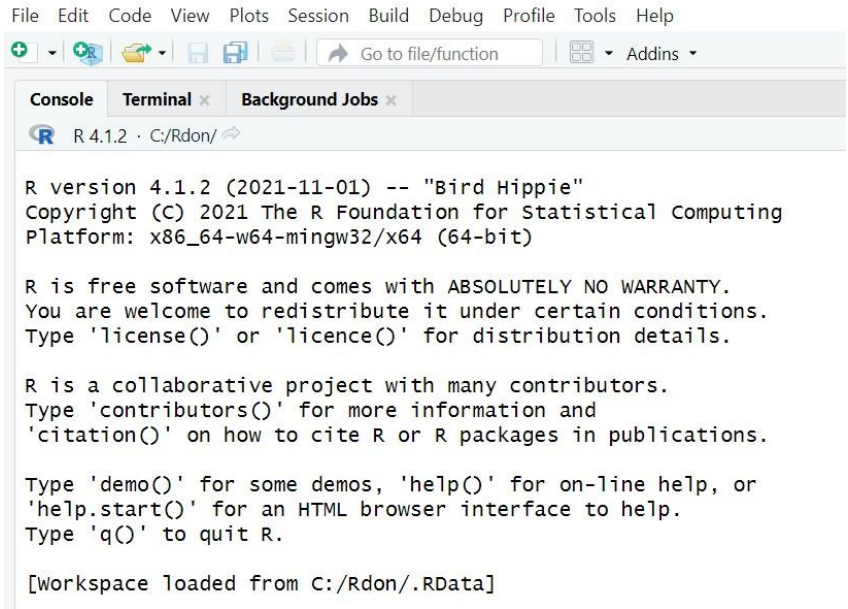


Figure 1. Console of the R software.

Usage requires that you load one or the other package. You have to go to "packages" and choose a CRAN mirror site, ex. France (Montpellier) and then load one or the other package.

Packages and programs of data analysis with « R »

Several packages and published programs are presented among others. Several personal programs are then proposed.

It should be noted that traditionally files are presented with variables in columns and individuals in rows. **In phytosociological tables, the species are placed in lines and the relevés in columns; the tables must first be transposed**.

Package Rcmdr (R commander)

The installation of R commander (R interface with menus) is easy. Just follow the five steps:

1- Open R.

2- Go to "Packages" and select "Install package (s)". Choose a CRAN mirror site (example: France (Montpellier), then OK.

3- Select Rcmdr from the list of available packages, then OK.

4- Start R command using the library (Rcmdr) command in the R console and the menu appears (figure 1).

5- Then go to "Tools" menu and click on RcmdrPlugin.FactoMineR.

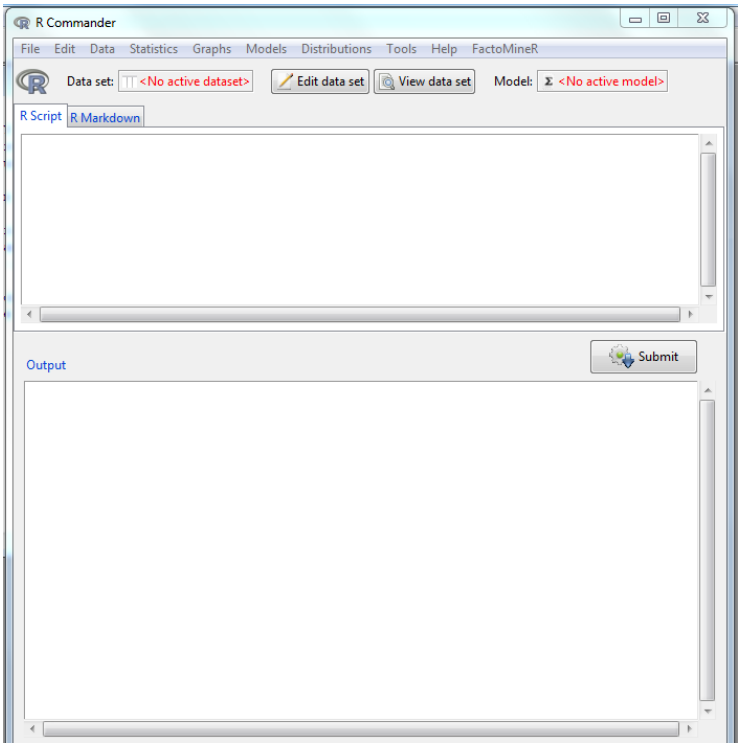


Figure 2. Menu of package Rcmdr.

You must first define the location of the data and results by going to File (change working directory).

It is then necessary to enter FactoMineR (figure 2) and define the parameters of the file (here a .txt file, figure 3).

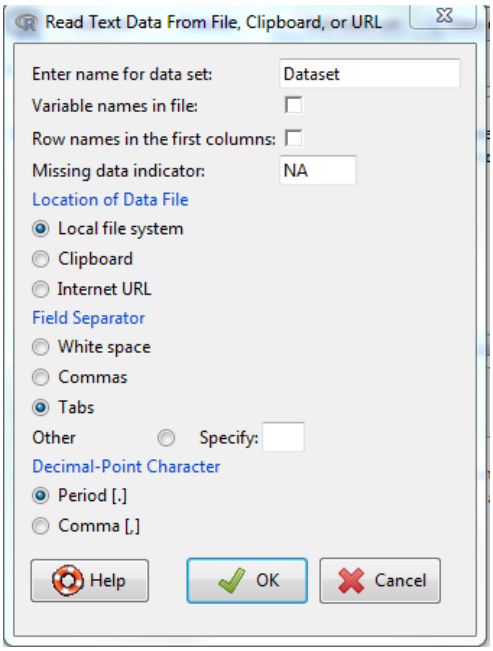


Figure 3. Window for loading a file with its parameters.

The file is visible as follows (figure 4). In this analysis, the row and column headings are not shown. They are put in separate files.

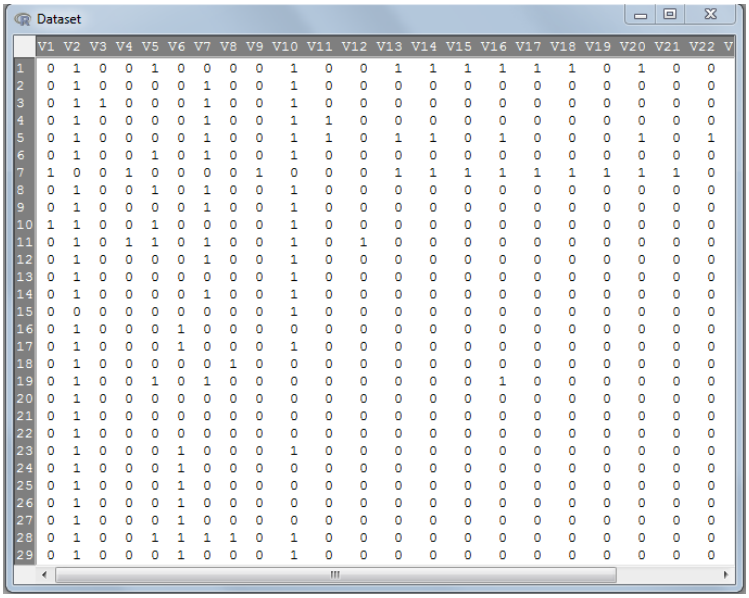


Figure 4. Loaded .txt file.

We then choose the analysis always in FactoMineR and we define the parameters of the graphs and the output (figure 5).

The proposed analyzes are as follows:

Principal component analysis,

Correspondence analysis,

Multiple correspondence analysis,

Multiple factor analysis,

Hierarchical multiple factor analysis,

Dual multiple factor analysis,

Factor analysis of mixed data,

General Procrustes analysis,

Scatter plot with additional variables,

Description of categories,

Hierarchical clustering on principal components.

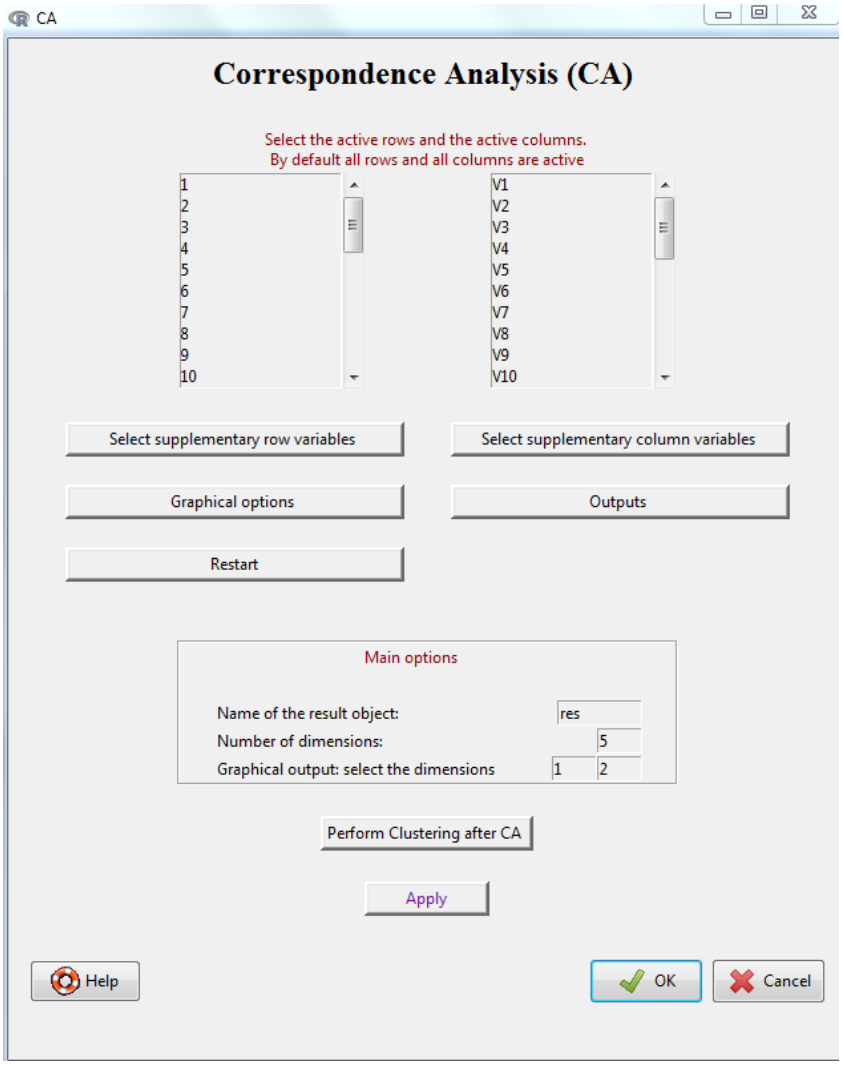


Figure 5. Window of the correspondence analysis.

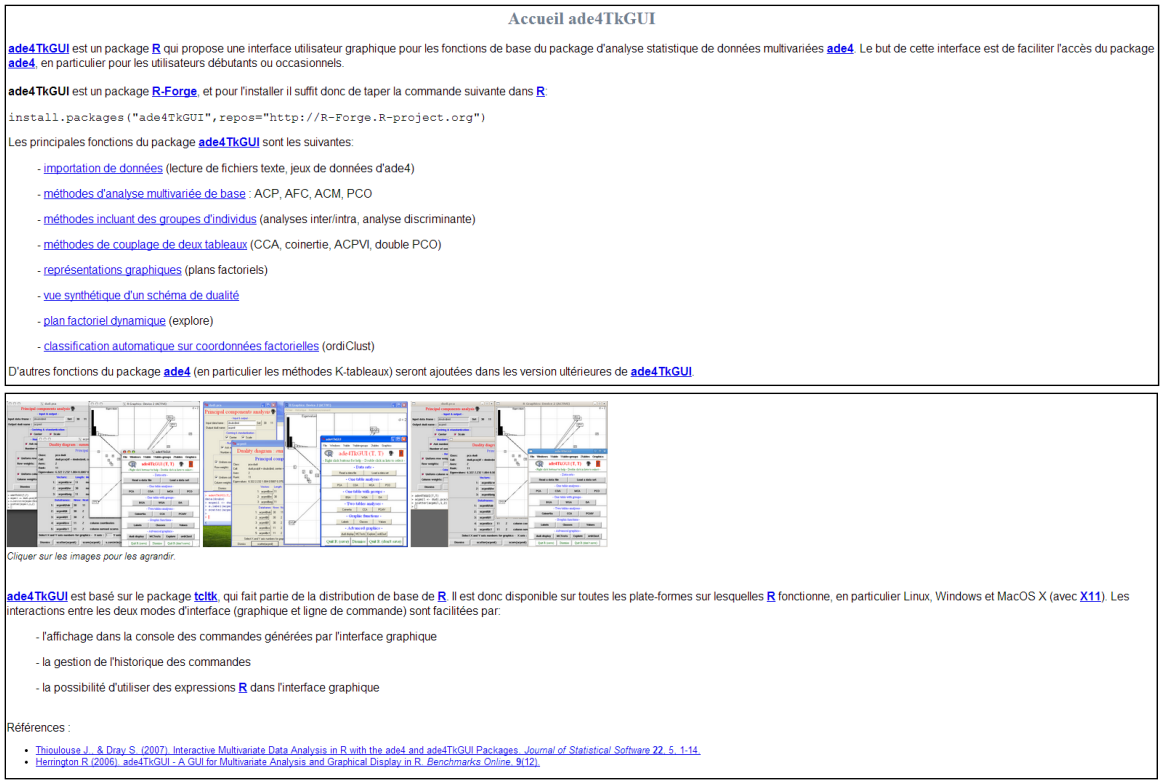
The results are saved in a .csv file and are transferable in other software (Excel for example). Graphics can also be saved or copied and introduced into a Word file. This set is quite complete and can be used with very large tables.

Package ADE4

This package developed by the University of Lyon1 is also very useful and rich in possible analyzes. It is accessed by the following URL:

http://pbil.univ-lyon1.fr/ade4TkGUI/

The home page (figure 6) gives an idea of the content.



Accueil ade4TkGUI

ade4TkGUI est un package **R** qui propose une interface utilisateur graphique pour les fonctions de base du package d'analyse statistique de données multivariées **ade4**. Le but de cette interface est de faciliter l'accès du package **ade4**, en particulier pour les utilisateurs débutants ou occasionnels.

ade4TkGUI est un package **R-Forge**, et pour l'installer il suffit donc de taper la commande suivante dans **R**:

```
install.packages("ade4TkGUI",repos="http://R-Forge.R-project.org")
```

Les principales fonctions du package **ade4TkGUI** sont les suivantes:

- importation de données (lecture de fichiers texte, jeux de données d'ade4)
- méthodes d'analyse multivariée de base : ACP, AFC, ACM, PCO
- méthodes incluant des groupes d'individus (analyses inter/intra, analyse discriminante)
- méthodes de couplage de deux tableaux (CCA, coinertie, ACPVI, double PCO)
- représentations graphiques (plans factoriels)
- vue synthétique d'un schéma de dualité
- plan factoriel dynamique (explore)
- classification automatique sur coordonnées factorielles (ordiClust)

D'autres fonctions du package **ade4** (en particulier les méthodes K-tableaux) seront ajoutées dans les version ultérieures de **ade4TkGUI**.

Cliquer sur les images pour les agrandir.

ade4TkGUI est basé sur le package **tcltk**, qui fait partie de la distribution de base de **R**. Il est donc disponible sur toutes les plate-formes sur lesquelles **R** fonctionne, en particulier Linux, Windows et MacOS X (avec **X11**). Les interactions entre les deux modes d'interface (graphique et ligne de commande) sont facilitées par:

- l'affichage dans la console des commandes générées par l'interface graphique
- la gestion de l'historique des commandes
- la possibilité d'utiliser des expressions **R** dans l'interface graphique

Références :

- Thioulouse J., & Dray S. (2007). Interactive Multivariate Data Analysis in R with the ade4 and ade4TkGUI Packages. *Journal of Statistical Software* **22**: 5, 1-14.
- Herrington R (2006). ade4TkGUI - A GUI for Multivariate Analysis and Graphical Display in R. *Benchmarks Online*, 9(12).

Figure 6. Home page of the package ADE-4.

This package is quite complete and is also recommended for large tables.

In the window R, you have to load the package ade4TkGUI, then enter library (ade4TkGUI) and finally the function ade4TkGUI (). The following window appears:

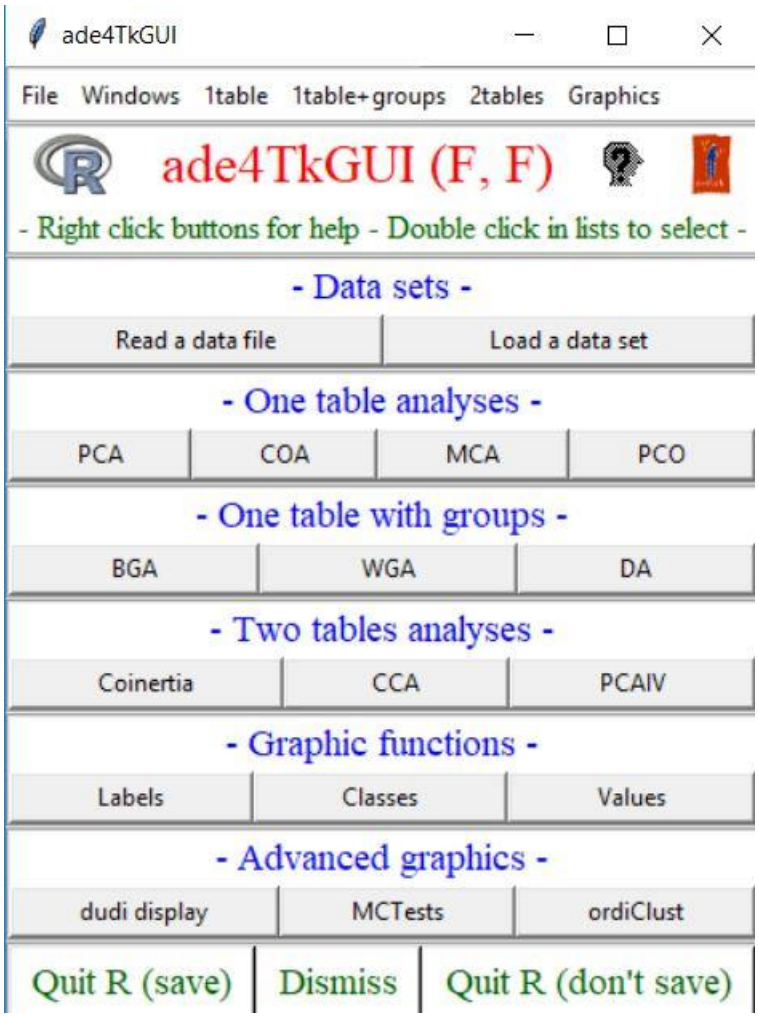


Figure 7. Window of the package Ade4TkGUI.

R Programs proposed by Tessema Genanew Jember

Several useful programs are provided by the author (JEMBER, 2012) in his book "Multivariate Data Analysis Using R Software".

In addition to correlation, regression, mean comparison and variance analysis programs, there is a set of classical classification programs, with its graphic appendices (*e.g.* dendrograms), and ordination such as principal component analysis, principal coordinate analysis, non-metric multidimensional analysis and constraint analysis such as the so-called "canonical correspondence analysis". These programs use many R packages and functions; they are easily implemented.

Package CAvariants

This package, published in 2019, offers simple solutions for six multivariate analyzes:
It calculates:

- Simple correspondence analysis (catype = "CA")
- The analysis of doubly ordered matches (catype = "SOCA")
- Simply ordered correspondence analysis (catype = "DOCA")
- Non-symmetric correspondence analysis (catype = "NSCA")
- Non-symmetric correspondence analysis doubly ordered (catype = "NSCA")
- Non-symmetric correspondence analysis simply ordered (catype = "DONSCA").

Several functions are available for the representation of the results.
In the Rstudio window, for example, starting from an x file:
library (CAvariants) and enter
resfile <-Avariants (x, catype = "CA", firstaxis = 1, lastaxis = 2)
summary (resFile).

Personal softwares

Why write new programs when there are already enough? The following personal programs are especially for phytosociologists. **The species are placed in lines and the relevés in columns**; it is therefore not necessary to transpose the tables as in the other R programs.

The detailed presentation illustrates the calculation steps and leads to a better understanding of the algorithms. Without a minimum understanding, their use is not very productive and often disappointing. The programming is probably not as elaborate as that of a computer scientist, but it runs, even with very large tables. It is not a black box that I present but an analysis tool. In addition, all the useful results are grouped together in a single .csv array, which can be changed by means of another software such as Excel, for example.

For our own programs, we use Rstudio (figure 8), that is an integrated development environment (IDE) created specifically to work with R (GOULET, 2016). We load it from the following address:
https://www.rstudio.com/products/rstudio/download/

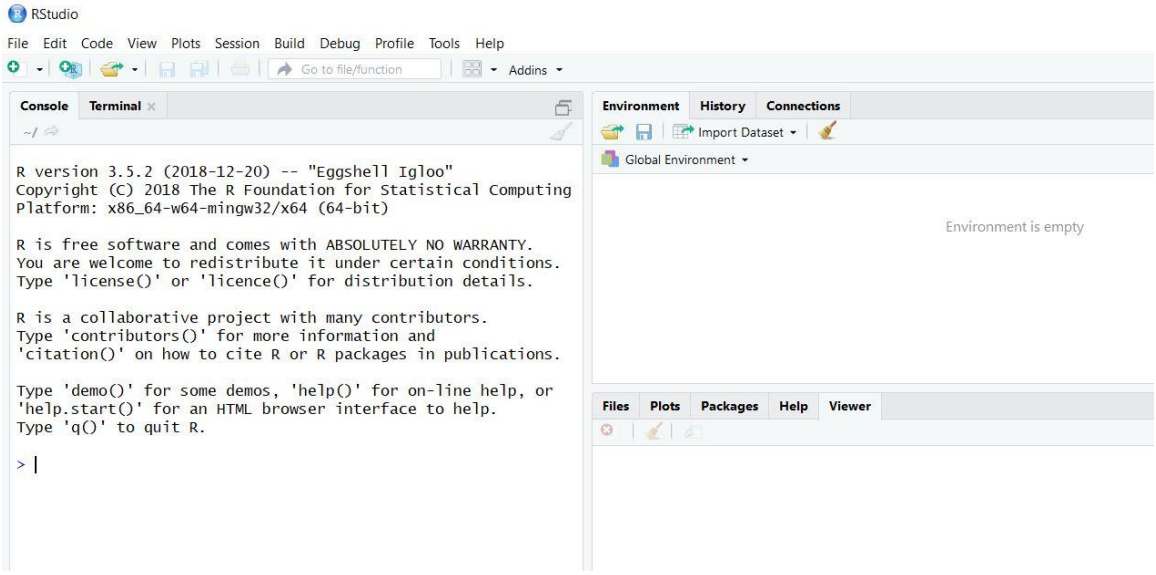


Figure 8. Window Rstudio

It allows to consult in a user-friendly interface, its script files, R command lines, help topics, graphics. We also recommend that you also load the Visual Studio Code software, which allows you to elegantly view and transform programs in .R format.

The programs and functions presented here were first written from a Word page, then entered into the Rstudio environment by doing File → New File → R Script, and then saved in .R format. They can also be viewed and transformed from Visual Studio Code (free download) which gives an elegant presentation of the program.

In this chapter, we insist on the calculation aspect, but we add graphical presentation software in R, thus taking advantage of the power of this language in this area. All our programs in .R format. are grouped together in a single .docx file present in the appendix. Just take them back and place them in Visual Studio Code by a simple copy-paste and send them back to the chosen directory. They are executed from small scripts in Word, of a few lines, provided in this chapter. You must first define a working directory, in which the programs and functions are located and in which the results of the calculations are saved. For example, c:/Rdon/, but each defines its own directory. Some programs work with personal functions. You must first load them into the console by doing Code → Source file and choose the function.

The phytosociological tables have this peculiarity of showing many empty cells (often of the order of 80 - 90%). These empty cells are sometimes occupied by a dot. In the abundance-dominance scale, we find the abundances of 1 to 5 but also the characters +, i, r. It is important to first transform them into numbers, replacing the empty cells with 0 and the +, i, r (or other symbols) with 1 (or other numeric values), which is

easily done with a small program. Remember that this is an ordinal scale. As we have seen in previous chapters, directly submitting these tables to multivariate analyzes, such as correspondence analysis or principal component analysis, gives only poor results. Several transformations are necessary. The first is to create a presence table (0-1), the second would be to create a complete disjunctive table (one line for "0", one for "1", and so on), but this type is not appropriate to floristic tables with far too many empty cells. A third solution, first tried in a purely empirical way, seems to bring the best results; it consists in creating, for each floristic variable, three lines: one for presence data, one for abundances greater than 1 and a third for abundances greater than 3. This last proposition seems to prove itself and we call these tables "simplified disjunctive tables". Cutting the lines further brings too much data dispersion and produces less interesting results. However, this last transformation may seem a bit cumbersome and multiplies the number of rows in the table. In non-symmetrical correspondence analysis programs, with usual abundance-dominance tables, it is also possible to perform a logarithmic transformation of the coefficients 1, 2, 3, 4, 5 with the log1p function. This transformation is integrated in the mfanscaVP program (weighting by the first eigenvalues of the subroutines).

Here are the programs with their small implementation scripts. The files Royer and Crupetenv113 files are useful for testing programs.

Program phyto1 for the transformation of a table .+ri12345 into a presence table

It is necessary to import the file .txt, by going in File, then Import Data set (see figure with Heading "Yes" and Row names "Use first column") and by loading it. It appears in the top bar for each subsequent use. We can keep it in memory by saving it. The program is intended only for symbols ., +, r, i, 1, 2, 3, 4 and 5 (figure 9).

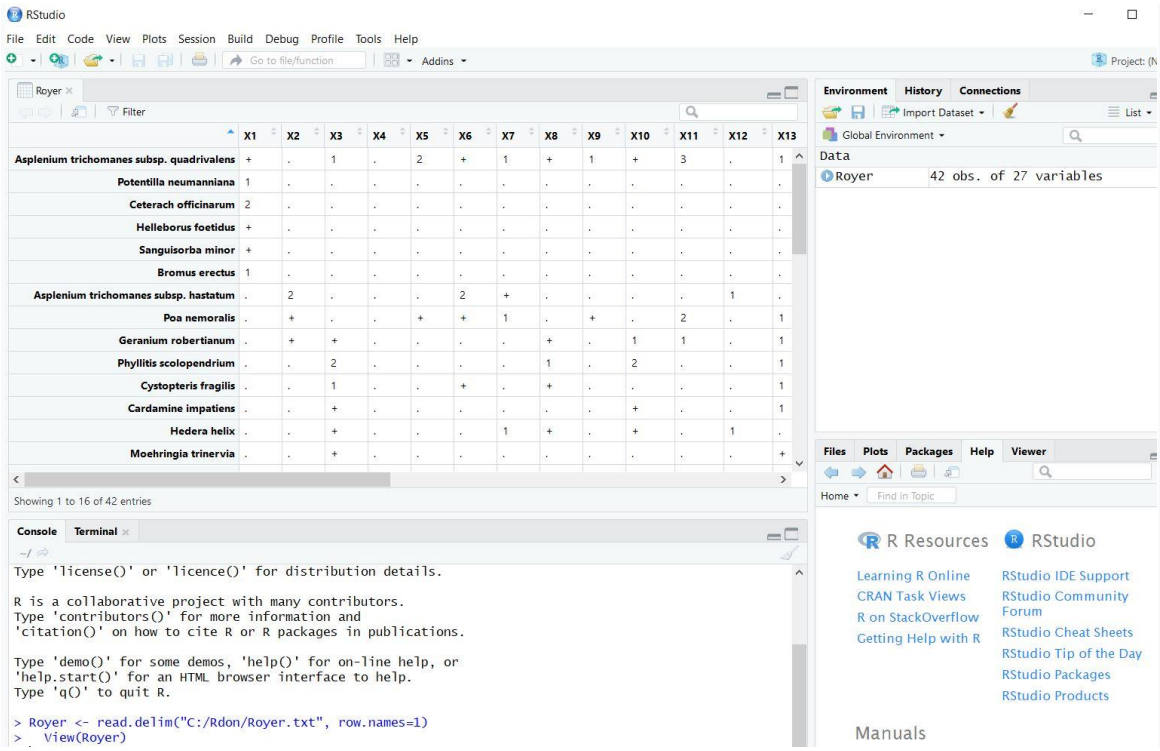


Figure 9. Window Rstudio with the table to be transformed.

Here is the script :

#Transformation of a table ., +, r, i, 1, 2 , 3, 4, 5 into a presence table Guy BOUXIN 2021
filename<-XXX

phyto1<-source("c:/Rdon/phyto1.R")

write.csv2(resul, "c:/Rdon/XXX1.csv")

In the script, we replace XXX with the name of the file (twice). We obtain the following figure (figure 10), thus produced and saved in .txt (table 1), can be used directly in the multivariate analyzes.

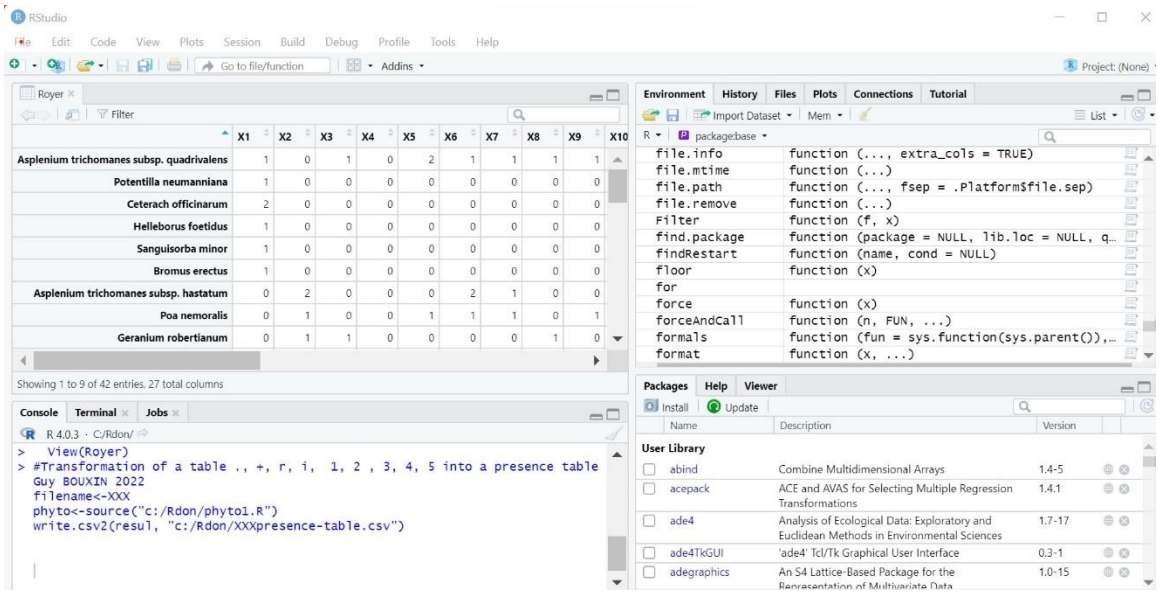


Figure 10. Window Rstudio with the transformation program.

We obtain the following table:

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16
Asplenium trichomanes subsp. *quad.*	1	0	1	0	1	1	1	1	1	1	1	0	1	1	0	0
Potentilla neumanniana	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Ceterach officinarum	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Helleborus foetidus	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sanguisorba minor	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bromus erectus	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Asplenium trichomanes subsp. *hast.*	0	1	0	0	0	1	1	0	0	0	0	1	0	0	0	1
Poa nemoralis	0	1	0	0	1	1	1	0	1	0	1	0	1	0	0	0
Geranium robertianum	0	1	1	0	0	0	0	1	0	1	1	0	1	1	0	1

Table 1. Portion of the Royer presence table produced by the R program.

Program phyto12345 for the transformation of a table .+ri12345 into 012345 table

The program is always intended only for signs +, r, i, 1, 2, 3, 4 and 5. In each row of the table, the signs +, r, i are replaced by 1. It therefore does not make any distinction between these three signs. The numbers 1, 2, 3, 4 and 5 are unchanged. This transformation of the +, r and i signs (as well as possibly others) is susceptible of various adaptations. These last three signs may be optionally given a score of 0.1 to give them less importance in relation to abundance 1.

Here is the script :

#Transformation of a table ., +, r, i, 1, 2 , 3, 4, 5 into a 12345 table Guy BOUXIN 2021
filename<-XXX

phyto12345<-source("c:/Rdon/phyto12345.R")

write.csv2(resul, "c:/Rdon/XXX12345.csv").

In the script, we replace XXX with the name of the file (twice). We obtain the following table 2, thus produced and saved in .txt, can be used directly in the analyzes multivariate.

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16
Asplenium trichomanes subsp. *quad.*	1	0	1	0	2	1	1	1	1	1	3	0	1	1	0	0
Potentilla neumanniana	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Ceterach officinarum	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Helleborus foetidus	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sanguisorba minor	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bromus erectus	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Asplenium trichomanes subsp. *hast.*	0	2	0	0	0	2	1	0	0	0	0	1	0	0	0	1
Poa nemoralis	0	1	0	0	1	1	1	0	1	0	2	0	1	0	0	0
Geranium robertianum	0	1	1	0	0	0	0	1	0	1	1	0	1	1	0	1

Table 2. Portion of the Royer table transformed into a digital table.

Program disj12345 for the transformation of a table 012345 into a simple disjunctive table

This program is used in the same way as the previous ones. Each row of the table is replaced by as many rows as there are different digits in the row, except the zero which is not taken into account. This is not a complete disjunctive table that would not be suitable for vegetation data, as explained in Chapter 2. The result is shown in Table 3.

Here is the script :

#Transformation of a numerical table into a disjunctive file 12345. Guy BOUXIN 2022
filename<-XXX

disj12345<-source("c:/Rdon/disj12345.R")

write.csv2(resul, "c:/Rdon/XXXdisj12345.csv")

In the script, we replace XXX with the name of the file (twice). We obtain the following table 3, thus produced and saved in .txt, can be used directly in the analyzes multivariate.

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16
Asplenium trichomanes subsp. *quad.*	0	0	1	0	0	0	1	0	1	0	0	0	1	0	0	0
Asplenium trichomanes subsp. *quad.* 2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Asplenium trichomanes subsp. *quad.* 3	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Potentilla neumanniana	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ceterach officinarum 2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bromus erectus	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Asplenium trichomanes subsp. *hastatum*	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
Asplenium trichomanes subsp. *hastatum* 2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Poa nemoralis	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
Poa nemoralis 2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Geranium robertianum	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0

Table 3. Portion of the Royer table transformed into a simple disjunctive table.

Program disj113 for the transformation of a table 012345 into a simplified disjunctive 113 table

This program is used in the same way as the previous ones. It is only intended for data types 0, 1, 2, 3, 4 and 5.

Each row of the table is replaced by one, two or three rows:

- if there is only data 0 and 1, the line remains unchanged,
- if there are data 0, 1 and abundances 2 and 3, the original line is replaced by two lines: one indicating the presence of the variable, whatever the abundance and a second replacing abundances two and three by the number 1; the variable is followed by the sign "> 1",
- if there are data 0, 1 and abundances 2, 3, 4 and 5, a third row must be added for abundances 4 and 5; the variable is followed by the sign "> 3".

However, if a variable shows only abundances 2, 3, 4 and 5 without abundances 1, the program only predicts two rows, one for abundances> 1 and one for abundances> 3; The same is true if there are only abundances 4 and 5; then there is only one line marking "> 3". This way of proceeding avoids giving too much weight to the abundant species, which end up with two or three identical lines.

Here is the script :

#**Transformation of a numerical table 12345 table into a disjunctive table 1>1>3.** Guy BOUXIN 2022
filename<-XXX
disj113<-source("c:/Rdon/disj113.R")
write.csv2(resul, "c:/Rdon/XXXdisj113.csv")

In the script, we replace XXX with the name of the file (twice). We obtain the following table 4, thus produced and saved in .txt, can be used directly in the analyzes multivariate.

Here is an example of a .txt file.

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16
Asplenium trichomanes subsp. *quad.* >1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
Asplenium trichomanes subsp. *quad.* >1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0
Potentilla neumanniana	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ceterach officinarum >1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ceterach officinarum	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bromus erectus	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Asplenium trichomanes subsp. *hast.* >1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Asplenium trichomanes subsp. *hast.* >1	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1
Poa nemoralis >1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Poa nemoralis	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0
Geranium robertianum	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0

Table 4. Simplified disjunctive table, ready for the multivariate analyzes.

Programmes PCoA et NMDS

These programs are used in the same way as the previous ones..

```
#PCoA
filename2<-as.matrix(XXX)
library(vegan)
library(permute)
k=6
nbrrel= ncol(filename2)
nomrel<-list(1 :nbrrel)
nomcol<-list(1 :k)
nomcol2<-list(1 :2)
nomrel<-colnames(filename2)
nomrel
k=6
filename<-as.matrix(t(filename2))
#Matdis<-dist(filename, method = "euclidean", diag=TRUE, upper=TRUE)
Matdis<-vegdist(filename, method = "chord", diag=TRUE, upper=TRUE)
Matdis
for(j in 1:k)                    #
{
nomcol[1+(j-1)]<-paste("coord",j)
}
nomcol2[1]<-paste("eigenvalues")
nomcol2[2]<-paste("% of trace")
hist(Matdis, breaks=40, include.lowest = TRUE, right = TRUE,
labels=FALSE, border = NULL,
main = paste("Simple disjunctive table – chord distance "), plot=TRUE,
axes = TRUE,
warn.unused = TRUE)
x<-cmdscale(Matdis, eig=TRUE, k=6)
resulpcoacoord<-matrix(1: nbrrel*k,nrow=nbrrel,ncol=k,byrow=TRUE)
colnames(resulpcoacoord)<- nomcol
for(i in 1 :nbrrel)
{
for(j in 1:k)
{
resulpcoacoord[i,j]<-x$points[i,j]
}
}
rownames(resulpcoacoord)<-nomrel
trace=sum(x$eig)
```

```
Lam<-matrix(1:nbrrel*1,nrow=nbrrel,ncol=1)
Lam<-x$eig
Lam
for(i in 1:nbrrel)
{
Lam[i]<-Lam[i]/trace*100
}
resulpcoaeig<-matrix(1: nbrrel*1+1,nrow=nbrrel,ncol=2,byrow=TRUE)
colnames(resulpcoaeig)<-nomcol2
for(i in 1 :nbrrel)
{
resulpcoaeig[i,1]<-x$eig[i]
resulpcoaeig[i,2]<-Lam[i]
}
write.csv2(resulpcoacoord, "c:/Rdon/ XXXPCoAchordcoordrel.csv")
write.csv2(resulpcoaeig, "c:/Rdon/XXXPCoAchordeig.csv")
```

Remarque : pour calculer la distance « chord » il faut installer les packages vegan et permute et utiliser la fonction vegdist()

#NMDS

```
library(MASS)
filename2<-as.matrix(XXX)
library(vegan)
library(permute)
k=3
nbrrel= ncol(filename2)
nomrel<-list(1 :nbrrel)
nomcol<-list(1 :k)
nomcol2<-list(1 :2)
nomrel<-colnames(filename2)
k=6
filename<-as.matrix(t(filename2))
#Matdis<-dist(filename, method = "euclidean")
Matdis<-vegdist(filename, method = "chord", diag=TRUE, upper=TRUE)
for(j in 1:k)
{
nomcol[1+(j-1)]<-paste("coord",j)
}
nomcol2[1]<-paste("eigenvalues")
nomcol2[2]<-paste("% of trace")
hist(Matdis, breaks=40, include.lowest = TRUE, right = TRUE,
labels=FALSE, border = NULL,
main = paste(" Abundance table- distance matrix"), plot=TRUE,
axes = TRUE,
warn.unused = TRUE)
x<-isoMDS(Matdis, k=6, maxit = 50)
resulNMDScoord<-matrix(1: nbrrel*k,nrow=nbrrel,ncol=k,byrow=TRUE)
colnames(resulNMDScoord)<- nomcol
for(i in 1 :nbrrel)
{
for(j in 1:k)
{
resulNMDScoord[i,j]<-x$points[i,j]
```

```
}
}
rownames(resulNMDScoord)<-nomrel
write.csv2(resulNMDScoord, "c:/Rdon/ XXXNMDScoordrel.csv")
```

Programs pca, caf et nscaf

The numerical results of the pca and caf programs correspond to those produced by RCmdr, those of nsca to those provided by the nscaf program of CAvariants ().

These three programs work identically. Just replace XXX with the name of the file, specify the number of permutations (nper) and the number of axes to calculate (nax).

They produce a result file in columns with,

- for the first row, successively for each axis, the corresponding eigenvalue, the percentage with respect to the trace of the correlation or covariance matrix and the probability associated with the eigenvalue,
- For the following lines and for each species or relevé, the coordinate on the corresponding axis, the relative contribution (or cos2 in the case of PCA) and the probability associated with the relative contribution (or cos2).

In the programs that follow, you must first load the covarcaR and covarnsca.R functions in Rstudio, by going to Code (top bar), then GoTo Fike/Function.

Here are the scripts :

1. For principle component analysis

```
#pca with permutations Guy BOUXIN 2022
filename<-XXX
nper=n1
nax=n2
pca<-source("c:/Rdon/pca.R")
write.csv2(resulpca, "c:/Rdon/XXXpca.csv")
```

2. For correspondence analysis

```
#caf with permutations and personal functions Guy BOUXIN 2022
filename<-XXX
nper=n1
nax=n2
caf<-source("c:/Rdon/caf.R")
write.csv2(resulca, "c:/Rdon/XXXcaf.csv")
```

3. For non-symmetrical correspondence analysis

```
#nscaf with permutations and personal functions Guy BOUXIN 2022
filename<-XXX
nper=n1
nax=n2
nscaf<-source("c:/Rdon/nscaf.R")
write.csv2(resulnsca, "c:/Rdon/XXXnscaf.csv")
```

We replace, in the script, XXX by the name of the file (twice), n1 by the desired number of permutations (1000 at least, 10000 if possible, according to the size of the number of columns of the table), n2 by the number of calculated eigenvalues.

This produces a table as follows:

	coord 1	Cr% 1	P 1	coord 2	Cr% 2	P 2
Vp,Vp% et P	5.74790056	13.6854775	2.8	4.34982694	10.3567308	8.5
Asplenium trichomanes subsp. *quadrivalens*	0.27685687	0.07664973	47	0.047633	0.0022689	90.4
Potentilla neumanniana	-0.44413484	0.19725576	21	-0.57490655	0.33051755	6.9
Ceterach officinarum	-0.3209514	0.1030098	31	-0.69205829	0.47894467	2.9
Helleborus foetidus	-0.28100306	0.07896272	24	-0.76686229	0.58807777	2.5
Sanguisorba minor	-0.28100306	0.07896272	25	-0.76686229	0.58807777	2.4
Bromus erectus	-0.28100306	0.07896272	24	-0.76686229	0.58807777	2.1
Asplenium trichomanes subsp. *hastatum*	-0.19001115	0.03610424	63	0.02792301	0.00077969	94.2

Table 5. Principal component analysis of the file “Royerpres”, results for the first two axes and the first seven species.

These programs also display, in the viewer window of RStudio, the histogram of the distribution of the correlation or covariance coefficients, which tells us about the quality of the analysis. This histogram can be saved.

The coordinates of species and records are likely to be used to construct graphs for species or records. The R language offers a lot of possibilities, and we present two softwares to draw these graphs.

Programs mfapca, mfacaf and mfanscaf of multiple factor analysis

These programs always work like the previous ones, except that it is necessary to specify the respective numbers of lines in the sub-tables.

1. For multiple factor analysis, based on principal component analysis,the weighting is based on the first eigenvalues of the subtables.

```
#mfapca Guy BOUXIN 2022
filename<-XXX
```

```
co<-c(n1,n2,…)
nax=n3
mfapca<-source("c:/Rdon/mfapca.R")
write.csv2(resulmfapca, "c:/Rdon/XXXmfapca.csv")
```

2. For multiple factor analysis based on correspondence analysis.

The weighting is based on the first eigenvalues of the subtables.

#mfacaf with personal functions Guy BOUXIN 2022

```
filename<- XXX
co<-c(n1,n2, …)
nper=n3
nax=n4
mfacaf<-source("c:/Rdon/mfacaf.R")
write.csv2(resulmfaca, "c:/Rdon/ XXX.csv")
```

3. For multiple factor analysis based on non-symmetrical correspondence analysis, variante 1.

The weighting is based on the density of sub-tables. That program works with presence data (for instance table 113).

#mfanscaf with personal functions and permutations, weighting by the densities of the subtables Guy BOUXIN 2022

```
filename<- XXX
co<-c(n1,n2)
nper=n3
nax=n4
mfanscaf<-source("c:/Rdon/mfanscaf.R")
write.csv2(resul1mfansca, "c:/Rdon/ XXX-mfansca -original table & sub-tablesdensity.csv")
write.csv2(resul2mfansca, "c:/Rdon/ XXX-mfansca-density.csv")
```

4. For multiple factor analysis based on non-symmetrical correspondence analysis, variante 2

The weighting is based on the first eigenvalues of the respective sub-tables. That program works directly with abundance data. A log transformation of the abundance data is necessary and included in the program.

#mfanscaf with personal functions and permutations, log1p transformation and weighting by the first eigenvalues of the subtables Guy BOUXIN 2023

```
filename<-XXX
co<-c(n1,n2,…)
nper=n3
nax=n4
mfanscafEV<-source("c:/Rdon/mfanscafEV.R")
write.csv2(resul1mfansca, "c:/Rdon/ XXX-mfansca-EV -original table & sub-tables eigenvalues.csv")
write.csv2(resul2mfansca, "c:/Rdon/ XXX-mfansca-EV.csv")
```

We replace XXX with the file name, specify the respective numbers of rows of the sub-tables (n1, n2, n3, ...), the number of simulations (nper) for only mfansca and the number of eigenvalues to be calculated (nax).

In these last two programs, the number of axes to be calculated cannot exceed the smallest number of rows among the sub-arrays. The results appear in two files, a usual one and a second one with the numbers of lines, the densities or the eigenvalues, for the sub-tables and for the transformed table.

The mfacaf and mfanscaf programs in R are somewhat different from those offered in the previous versions of this site or in BOUXIN (2016). The ancient programs were written in such a way as to produce the same numerical results as those provided by the ADE-4 software (version used in 2006). In this software, the calculation of the eigenvalues of the sub-tables is done without changing the sums on the columns and therefore the sum of the traces of the sub-tables is equal to the trace of the complete table, as in multiple factor analysis based on principal component analysis. In the latter, the rows of the sub-tables are identical to those of the complete table. In our own programs in R, the eigenvalues of the sub-tables are calculated by taking account of the sums on their respective columns which are, of course, different from one sub-table to another.

In this new version of factor analysis, it is important that there are no empty columns in one or the other sub-table, at least in the versions based on dual or non-symmetrical correspondence analysis.

The results appear in table 6.

The Crupetvegenv113 file includes a floristic subfile with 57 lines and an environmental subfile with 44 lines.

	coord 1	CR 1	P 1
Vp, Vp%, P	0,21664321	13,2424065	0
Agrostis stolonifera	-0,2408397	5,80037589	10,05
Alnus glutinosa	-0,049166	0,24172958	41,57
Alnus glutinosa >1	-0,16093688	2,59006786	30,73
Alnus glutinosa >3	0,00905918	0,00820688	72,11
Alnus incana	0,15348452	2,35574976	33,09
Alnus incana >1	0,14713428	2,16484949	16,29
Angelica sylvestris	-0,04813769	0,23172372	72,4
Apium nodiflorum	-0,04966777	0,2466887	27,48
Caltha palustris	0,05544199	0,30738139	38,83
Calystegia sepium	0,00992352	0,00984762	91,2
Cardamine amara	0,24977337	6,23867355	10,06

Table 6. Multiple factor analysis based on non-symmetric correspondence analysis of file "Crupetvegenv113". Results for the first axis and the first eight species, *Alnus glutinosa* being divided into three variables and *Alnus incana* into two variables.

For the first lines (two plus the number of subfiles), the three programs provide the eigenvalues for each axis, the corresponding relative contributions and the probabilities (only for mfansca). For the following lines, the presentation is the same as in the simple analyzes, with or without the probabilities.

The mfacaf and mfanscaf programs are slightly different, in that they replace the computation of the covariance matrix with a function. This should make the calculations faster. The script is less explicit about the calculations. You must first transform the covarca.docx and covarnsca.docx scripts into .R format via Rstudio, Code tab then choosing Source File, depending on the directory chosen to load or receive the files resulting from the analyzes.

Program Cs for the definition of character-species

This program makes it possible to establish, from a presence file, the list character-species of a grouping or association, within a table comprising several syntaxa. This definition only applies to the scale of the used large table used. This program requires that the relevés of the same group be juxtaposed. Just enter the name of the complete file with the rearranged columns and the respective numbers of relevés in each grouping and the number of permutations (nper, 10,000 recommended). That definition is only valid at the scale of the large table.

 # Definition of character species of a presence table. Guy BOUXIN 2022

```
filename<-XXX
co<-c(n1,n2,…….)
nper=10000
cs<-source("c:/Rdon/cs.R")
write.csv2(resul, "c:/Rdon/XXXcs.csv")
```

Program Graph for drawing graphs from coordinates of multivariate analyses

There are two programs: one when there is only one kind of variables, such as floristic variables, and a second when mixing, for example, floristic and environmental variables.

For the first program, you must output a file comprising a first column with the list of variable names and as many columns as there are coordinates considered, indicated coord1, coord2, ... in the first line. An example is given in Table 7.

c	coord1	coord2	coord3
1	-0.134601	-0.460529	1.277298
2	1.050046	0.398986	0.937062
3	-0.118671	-0.327056	1.805153
4	-0.270912	-0.216169	3.389558
5	2.972145	0.288958	1.202091
6	2.623666	0.032145	-0.418913
7	-0.656541	-0.272471	1.00179
8	1.581773	0.184523	-0.196533
9	2.203882	0.262407	0.420033
10	2.989747	0.539129	0.600225

Table 7. Coordinates for 10 relevés, on the first three axes of a multivariate analysis.

With the second program, you must specify the type of character corresponding to the variables, on additional columns. Here is an example of a file with 6 columns, one with the variables, three with the coordinates, the fourth specifying the color and the fifth, the font.

coord1	coord2	coord3	colo	cara	
Syzygium rowlandiii G	0,044090554	-0,061651151	-0,122712784	a	3
Syzygium rowlandiii G 20-39	-0,087414582	-0,096412416	-0,011908992	a	3
Syzygium rowlandiii G >40	0,023628466	0,047747795	-0,142272082	a	3
Syzygium rowlandii p	0,025185604	-0,028021546	-0,060402434	a	3
Syzygium rowlandii p 20-39	0,095857426	-0,06464239	0,011357952	a	3
Syzygium rowlandii +	0,153844472	-0,043228559	-0,069207629	a	3
Syzygium rowlandii + 20-39	0,102597138	-0,089898555	0,022400078	a	3
Carapa grandiflora G	-0,002141596	-0,008193828	-0,067967865	a	3
Carapa grandiflora G 20-39	-0,01233258	0,0084035	-0,016967498	a	3
Carapa grandiflora p	-0,04500537	0,133986642	-0,072509291	a	3
pente0	0,08301115	-0,034781038	-0,127867489	b	2
pente 1-9	-0,03090512	-0,044024429	-0,023049512	b	2
pente 10-19	-0,012415985	0,016079666	0,151575951	b	2
pente>19	-0,015671775	0,029473705	0,054554763	b	2
profondeur du sol 1	-0,068887046	0,037281395	0,189510058	b	2
profondeur du sol 2	-0,011419242	0,115827687	0,010251562	b	2
profondeur du sol 3	0,104324559	-0,186361179	-0,144547907	b	2
couvert 1	0,102261718	-0,099365189	0,161534433	b	2
couvert 2	-0,015350527	0,084104893	0,070700006	b	2
couvert 3	-0,094069771	-0,06330587	-0,145327554	b	2
couvert 4	0,03117685	0,04531407	-0,031693173	b	2
hydrographie 1	-0,017978603	-0,000285474	-0,03427732	b	2
hydrographie 2	-0,000100038	-0,020634889	-1,19E-05	b	2

Table 8. File .txt used in the drawing of the graph with two colors and two fonts.

The construction of the graph is based on the ggplot () function. The position of variable labels is optimally arranged with the instruction "gridExtra :: grid.arrange (p1, ncol = 1)". We can play on the colors (a, b or others or even a single letter) and on font: 1 = normal text, 2 = text in bold, 3 = text in italics, 4 = text in bold italics and 5, Greek font.

Here is the script of the first program :

```
library(ggplot2)
library(ggrepel)
dat<-XXX
nom = rownames(dat)
nom[dat$coord1 <= 0.05 & dat$coord1> -0.05 & dat$coord2<= 0.05 & dat$coord2> -0.05] = ""
p<-ggplot(dat, aes(x = coord1, y = coord2, label = nom, fontface = 1)) + geom_point() +
geom_hline(yintercept=0) +
geom_vline(xintercept=0) +
ggtitle("Analysis **********") +
xlab("coord1 - % inertie : ****") +
```

```
ylab("coord2 - % inertie : ****") +
theme_bw()+
theme( axis.text.x = element_text(size = 24), axis.text.y = element_text(size = 24), plot.title = element_text(size=24, face=1), axis.title.x = element_text(size=24, face=1), axis.title.y = element_text(size=24, face=1)) +
geom_text_repel(box.padding = 0.25, min.segment.length= 0.25, segment.curvature = -0.1, segment.ncp = 3, segment.angle = 20, max.overlaps = getOption("ggrepel.max.overlaps", 10),
size = 8)
gridExtra::grid.arrange(p, ncol = 1)
ggsave("c:/Rdon/ XXX.png", plot = p, width = 12, height = 12)
```

It is necessary to specify the type of analysis in ggtitle and the percentages of inertia of the axes. Depending on the size of the file, and in particular the number of variables to be placed, you must adapt, in geom_text_repel, the maximum number of overlapping variables (getOption("ggrepel.max.overlaps", 10 or 20), the size of the letters (8 or 6).

In the second program, the fifth line of the script is changed as follows :

```
p<-ggplot(dat, aes(x = coord1, y = coord2, label = nom, fontface = cara)) +
```

You can also play on the colors of the variables with the fifth line.

```
p<-ggplot(dat, aes(x = coord1, y = coord2, label = nom, colour=colo, fontface = cara)) + geom_point()+ sca
le_color_manual(values = c("black","black")) + labs(title = "mfansca")
```

You must specify the axes considered (coord1, coord2, etc.), the type of analysis in the ggtitle function and the percentages of the axes.

Clustering programs

Several programs are presented on the same page, they are strongly inspired by those presented by JEMBER (2012) and they benefit from the information provided by Claire Della Nova's blog. The program "Cluster analysis using base R and euclidean distance as resemblance measure" is used to classify vegetation relevés. It works from two kinds of files: the coordinates of the relevés or the abundances of the species. If we use the coordinates of the relevés, the data are presented as in Table 7. If we use the abundances of a phytosociological table, the latter must first be transposed. Just replace, as usual, XXX with the name of the file loaded in RStudio. A complement makes it possible to draw a fixed number of clusters in the dendrogram.

The "Kmeans with R" program is another classic, but it is no longer a classification in the proper sense but rather an arrangement or classification of the relevés in a predetermined number of clusters.

In principle, the Elbow, Silhouette and Gap programs allow you to set the ideal number of clusters, based on the Kmeans function, but this should not be a panacea. The Elbow program does not produce very clear results, unlike the Silhouette program. The Gap program often provides different results than the other two. In our analyzes, I mainly focused on the results of the silhouette program.

Finally, a last program draws ellipses around clusters represented in a two-dimensional space. This allows you to draw beautiful figures.

In all the programs, it is rcommanded to use transformed variables, as in table 8. XXX must be replace by the table name.

1. Classification based on euclidean distance as similarity measurement

```
newdat<-as.matrix(XXX)
distance =dist(newdat, method= "euclidean") #distance matrix
structure =hclust(distance, method="ward.D")
dendrogram=as.dendrogram(as.hclust(structure))
plot(dendrogram, nodePar = list(lab.cex = 0.8, lab.col= "black", pch = NA, axes = T), cex.axis=1)
mtext("", side = 1, line=1, cex=0.8,font = 2)
mtext("Dissimilarity", side=2,line=2.5,cex=1,font=2)
title(main="HAC",cex.main=1)
#facultative option drawing the clusters
#k must be defied, for instance after a previous classification
k= *
range(distance)
cor(distance,cophenetic(structure))
#division of the dendrogram into specified number of clusters can be done after inspection
rect.hclust(structure, k=k,border=(1:k))
ClusterID=cutree(structure,k=k)
write.csv2(ClusterID, "c:/Rdon/XXXclusters.csv")
```

The k variable (number of clusters) must be entered.

2. Kmeans avec R

#You must first install the packages tidyverse, factoextra et NbClust

```
library(tidyverse)
 str(XXX)
km.out = kmeans(XXX,centers=*,nstart =20)
km.out$cluster
resul<-as.matrix(km.out$cluster)
write.csv2(resul, "c:/Rdon/ XXX.csv")
```

Inside the line "km.out = kmeans(XXX,centers=*,nstart =20) ", the chosen number of centers must be specified (*).

3. Méthods Nbclust et kmeans Elbow

```
library(factoextra)
```

```
library(NbClust)# Elbow method
fviz_nbclust(XXX, kmeans, method = "wss") + geom_vline(xintercept = 5, linetype = 2)+
  labs(subtitle = "Elbow method")
```

4. **Nbclust et kmeans Silhouette method**

```
library(factoextra)
library(NbClust)# Silhouette method

fviz_nbclust(XXX, kmeans, method = "silhouette")+
labs(subtitle = "Silhouette method"
```

5. **Program with ellipses.**

```
# Gap statistic
# nboot = 50 to keep the function speedy.
# recommended value: nboot= 500 for your analysis.
# Use verbose = FALSE to hide computing progression.
set.seed(123)
fviz_nbclust(XXX, kmeans, nstart = 25, method = "gap_stat", nboot = 50)+
labs(subtitle = "Gap statistic method")
km.out=kmeans(XXX,centers=5,nstart =20)
str(km.out)
pairs(XXX, col=c(1:2)[km.out$cluster])
library(factoextra)
fviz_cluster(km.out, XXX, ellipse.type = "norm")
```

The software GINKGO of VegAna

The Ginkgo software, produced by the University of Barcelona, makes it possible to easily perform several multivariate analyses, which have been developed in a context of numerical ecology. It is easy to use. This software is part of the "package" VegAna, a working environment that provides several tools for editing and analyzing flora and vegetation. The program is written in Java language.

Ginkgo is accessible from the link:

biodiver.bio.ub.es/ginkgo/Ginkgo.htm

The following window (figure 11) shows the possibilities of the program.

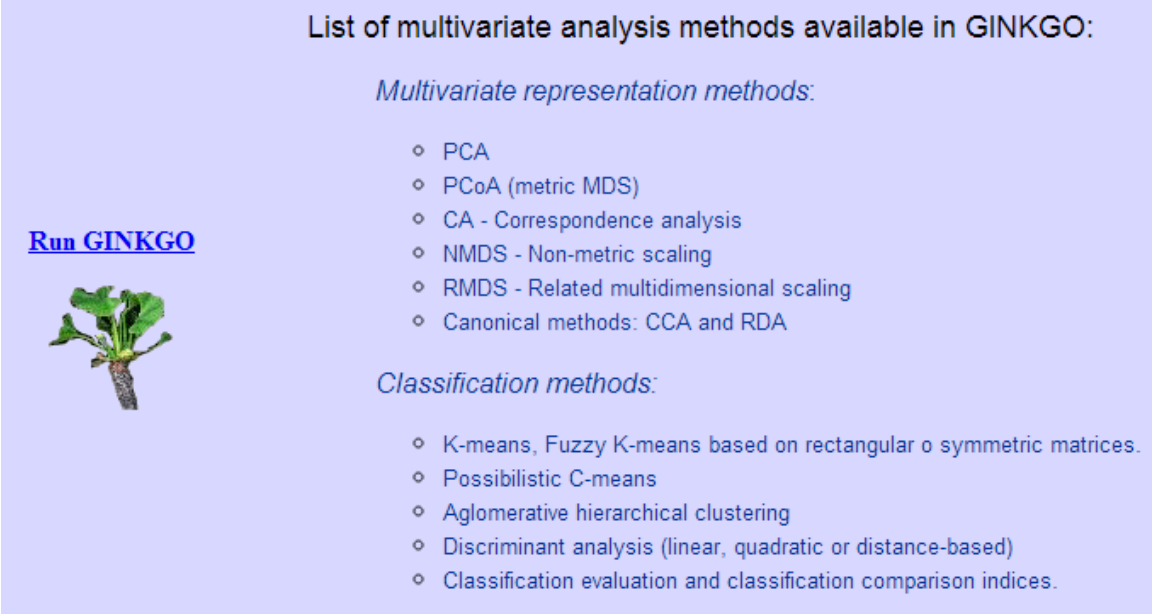


Figure 11. Starting window of software Ginkgo.

Two other references

We also mention the very detailed works of BOCCARD, GILLET & LEGENDRE (2011) and WILDI (2013), which offer a lot of R-softwares in numerical ecology.

References

BERTHET, P., FEYTMANS, E., STEVENS, D. & GENETTE, A. (1976). A new divisive method of classification illustrated by its applications to ecological problems. *Proc. Ninth Int. Biom. Conf., invited papers*. **Vol. II** : 366-382.

BORCARD, D., GILLET, F. & LEGENDRE, P. (2011). *Numerical Ecology with R*. Springer. 306 pp.

BOUXIN, G. (2005). Ginkgo, a multivariate analysis package. *Journal of vegetation Science* **16** : 355-359.

DIXON, P. (2003). VEGAN, a package of R functions for community ecology. *Journal of vegetation Science* **14** : 927-930.

GOULET, V, avec la collaboration de L. CARON (2016). Introduction à la programmern en R, cinquième édition. Université Laval, Québec. 202 pp.

HUSSON, R., LÊ, S. & PAGÈS, J. (2009). *Analyse des données avec R*. Presses Universitaires de France. 224 pp.

JEMBER, T. G. (2012). Multivariate Data Analysis using R software. Practical exercices for multivariate software. LAP Lambert Academic Publishing. 104 pp.

LOMBARDO, R. & BEH, E.J. (2019). Package 'CAvariants'. The R Foundation. https://www.R-project.org

R Core Team (2018). R: *A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

ROUX, M. (1985). *Algorithmes de classification*. Masson, Paris. 151 p.

WILDI, O (2013). Data analysis in vegetation ecology. Second edition. WILEY-BLACKWELL. 301 pp.