Tutorial 4 : PCA and cluster

I hope this tutorial will help you better understand PCA and clustering with R! Don't hesitate to ask me questions if you need further clarification. That's what I'm here for.

Learning objectives:
Familiarize yourself with R's functions for PCA and clustering

ACP

Part 0 : Packages downloading and loading

I let you install and load the following packages :

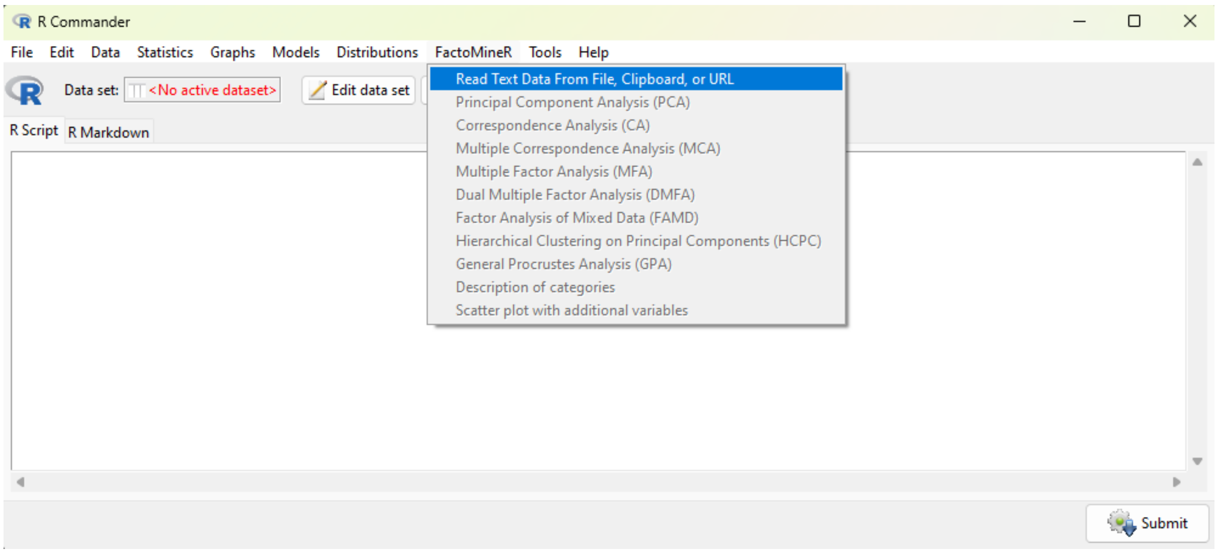
- Rcmdr
- FactoMineR
- RcmdrPlugin.FactoMineR

Part1 : Data loading

Download the dataset « data_PCA_got.csv » with a small change that is the name of rows need to be the characters' name. It will be better for graphs.

CLICK BUTTON TEAM

In the FactoMineR tab of Rcmdr.

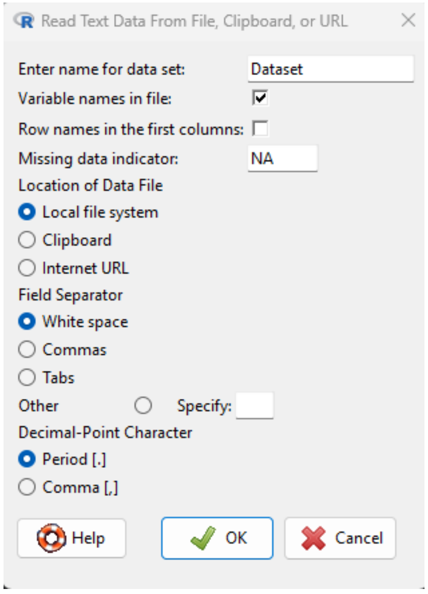

R Commander
File Edit Data Statistics Graphs Models Distributions FactoMineR Tools Help
Data set: <No active dataset>
Edit data set
R Script R Markdown
Read Text Data From File, Clipboard, or URL
Principal Component Analysis (PCA)
Correspondence Analysis (CA)
Multiple Correspondence Analysis (MCA)
Multiple Factor Analysis (MFA)
Dual Multiple Factor Analysis (DMFA)
Factor Analysis of Mixed Data (FAMD)
Hierarchical Clustering on Principal Components (HCPC)
General Procrustes Analysis (GPA)
Description of categories
Scatter plot with additional variables
Submit

Change dataset name, indicate 'got'.

Change field separator: check 'other' and specify semicolon '; ' as it is a csv file.

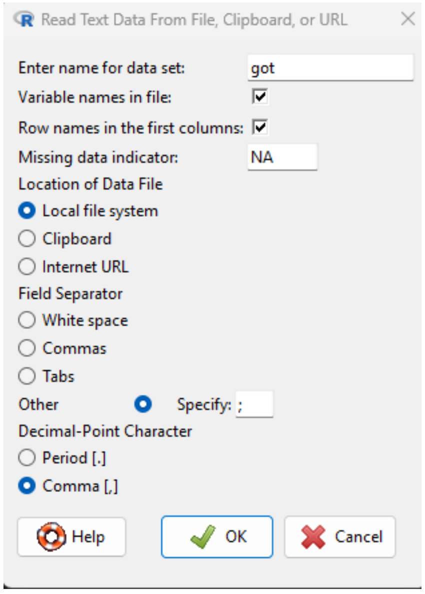
Change filed decimal-point character if needed.

Also check that the1st line corresponds to the characters' names (not compulsory, but visually more interesting).



Read Text Data From File, Clipboard, or URL

Enter name for data set:	Dataset
Variable names in file:	✓
Row names in the first columns:	☐
Missing data indicator:	NA
Location of Data File	[✓ Local file system] Clipboard, Internet URL
Field Separator	[✓ White space] Commas, Tabs, Other (Specify:)
Decimal-Point Character	[✓ Period [.]] Comma [,]



Read Text Data From File, Clipboard, or URL

Enter name for data set:	got
Variable names in file:	✓
Row names in the first columns:	✓
Missing data indicator:	NA
Location of Data File	[✓ Local file system] Clipboard, Internet URL
Field Separator	White space, Commas, Tabs, [✓ Other] Specify: ;
Decimal-Point Character	Period [.] [✓ Comma [,]]

Go and find your file in the directory structure of your computer.

CODING TEAM

row.names = ‘Name’ means that the column ‘Name’ will be taken as row names.

stringsAsFactors = TRUE convert all the columns of text into factors, which allow using some functions.

```
got<-read.csv2("data_PCA_got.csv",row.names = "Name", stringsAsFactors = TRUE)
got
```

```
summary(got)
```

Part2 : Principal Component Analysis (PCA) :

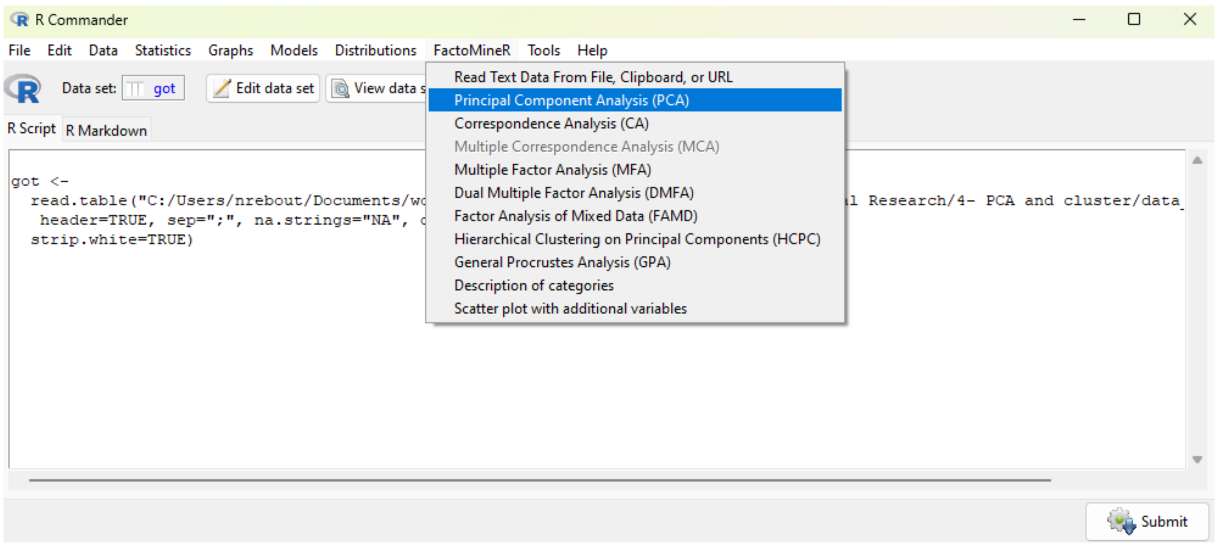
1) Creation of PCA and first conclusions

You will run PCA on the got dataset and store the result in a variable you'll call 'res' (for result).

We'll consider the following quantitative variables: number of chapters prepared ('Nb_chp_narr'), Gender, Married, Age, number of deaths in relationships ('Nb_death_relationships'), Popularity.

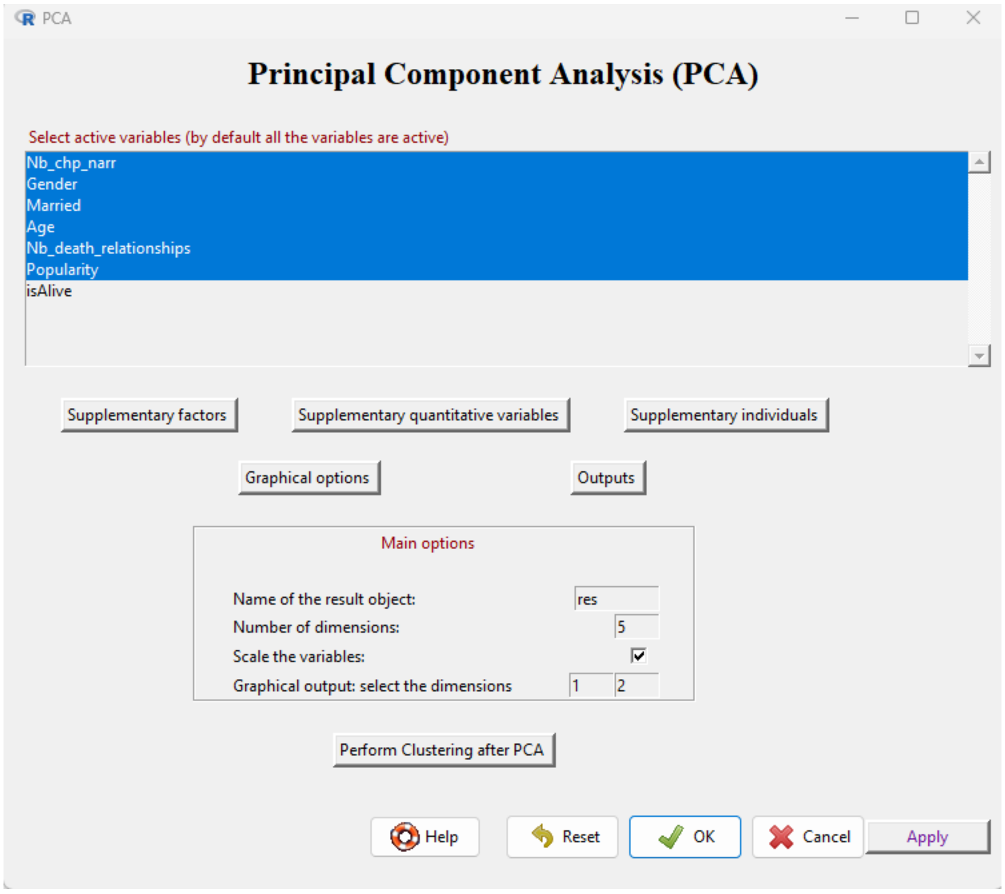
We don't consider here survival ('IsAlive').

CLICK BUTTON TEAM


R Commander
File Edit Data Statistics Graphs Models Distributions FactoMineR Tools Help
Data set: got
Edit data set
View data s
R Script R Markdown
Read Text Data From File, Clipboard, or URL
Principal Component Analysis (PCA)
Correspondence Analysis (CA)
Multiple Correspondence Analysis (MCA)
Multiple Factor Analysis (MFA)
Dual Multiple Factor Analysis (DMFA)
Factor Analysis of Mixed Data (FAMD)
Hierarchical Clustering on Principal Components (HCPC)
General Procrustes Analysis (GPA)
Description of categories
Scatter plot with additional variables
Submit

```
got <-
  read.table("C:/Users/nrebout/Documents/wo                                                l Research/4- PCA and cluster/data_
   header=TRUE, sep=";", na.strings="NA", c
  strip.white=TRUE)
```

Then select the variables you wish to include in the PCA. Note: they must be quantitative. Then click on OK.


PCA
Supplementary factors
Supplementary quantitative variables
Supplementary individuals
Graphical options
Outputs
Perform Clustering after PCA
Help
Reset
OK
Cancel
Apply

Principal Component Analysis (PCA)

Select active variables (by default all the variables are active)

Nb_chp_narr
Gender
Married
Age
Nb_death_relationships
Popularity
isAlive

Main options

Name of the result object: res
Number of dimensions: 5
Scale the variables: ✓
Graphical output: select the dimensions 1 2

The disadvantage of Rcmdr is that it is gonna write all the lines of code at once and will superimpose all the graphs.

```
got <-
  read.table("C:/Users/nrebout/Documents/worshop Easy Statistics to Shine in Biological Research/4- PCA and cluster/data_
   header=TRUE, sep=";", na.strings="NA", dec=",", row.names=1,
  strip.white=TRUE)
got.PCA<-got[, c("Nb_chp_narr", "Gender", "Married", "Age",
  "Nb_death_relationships", "Popularity", "isAlive")]
res<-PCA(got.PCA , scale.unit=TRUE, ncp=5, graph = FALSE)
print(plot.PCA(res, axes=c(1, 2), choix="ind", habillage="none",
  col.ind="black", col.ind.sup="blue", col.quali="magenta", label=c("ind",
  "ind.sup", "quali"),new.plot=TRUE))
print(plot.PCA(res, axes=c(1, 2), choix="var", new.plot=TRUE,
  col.var="black", col.quanti.sup="blue", label=c("var", "quanti.sup"),
  lim.cos2.var=0))
summary(res, nb.dec = 3, nbelements=10, nbind = 10, ncp = 3, file="")
remove(got.PCA)
```

I encourage you to isolate blocks of code as following :

```
got <-
  read.table("C:/Users/nrebout/Documents/worshop Easy Statistics to Shine in Biological Research/4- PCA and cluster/data
   header=TRUE, sep=";", na.strings="NA", dec=",", row.names=1,
  strip.white=TRUE)

got.PCA<-got[, c("Nb_chp_narr", "Gender", "Married", "Age",
  "Nb_death_relationships", "Popularity", "isAlive")]

res<-PCA(got.PCA , scale.unit=TRUE, ncp=5, graph = FALSE)

print(plot.PCA(res, axes=c(1, 2), choix="ind", habillage="none",
  col.ind="black", col.ind.sup="blue", col.quali="magenta", label=c("ind",
  "ind.sup", "quali"),new.plot=TRUE))

print(plot.PCA(res, axes=c(1, 2), choix="var", new.plot=TRUE,
  col.var="black", col.quanti.sup="blue", label=c("var", "quanti.sup"),
  lim.cos2.var=0))

summary(res, nb.dec = 3, nbelements=10, nbind = 10, ncp = 3, file="")

remove(got.PCA)
```

And then run the blocks of code again to get the different graphs.

TEAM CODAGE :

First, select the quantitative variables in your dataset. To do this, we create a sub-dataset from got, called got.PCA.

```
got.PCA<-got[, c("Nb_chp_narr", "Gender", "Married", "Age",
  "Nb_death_relationships", "Popularity")]
```

Then run the PCA function and store the result in 'res' (the calculations of PCA).

```
res<-PCA(got.PCA, scale.unit=TRUE, ncp=5, graph =  FALSE)
```

Then build the graphs based on the calculations :

```
    # plot of individuals
    plot(res,choix="ind")

    # plot of variables
    plot(res,choix="var")
```

What do the % values on the PCA axes correspond to ? What can we conclude?

Observe the cloud of variables. What can we conclude?

Observe the cloud of individuals. What can we conclude?

Considering the two types of information above, what can we conclude?

2) Because customers/researchers want « significant results », want p-values.

Use the dim.desc() function directly in the console.

```
dimdesc(res)
```

In the function, you specify the result of your PCA (here called ‘res’).

The dimdesc() function helps to interpret the dimensions of a PCA, MCA, CFA, MFA or HFA.

It allows dimensions to be described by both categorical and continuous variables.

It shows which variables the axes are most related to: which continuous variables are most correlated with each axis, and which categorical variables and modalities best describe each axis.

In the first part of the output, the correlation coefficient between each continuous variable and each PCA dimension is calculated. Correlation coefficients significantly different from zero are kept and sorted.

Conclude.

3) BONUS : use of FactoShiny

What is Factoshiny?

It's a graphical interface that lets you parameterize methods and modify graphs interactively.

No programming skills required.

Modifications to graphical options are instantly visible on graphs

Results (graphs and listings) are modified instantly.

Graphs can be downloaded, along with the lines of code needed to redo the analysis.

Factoshiny's result object can be reused to modify graphs. The interface is reopened with the latest settings and graphical options that can be modified.

First, install the Factoshiny package.

Then run this line of code:

```
RES<-PCAshiny(as.data.frame(got))
RES
```

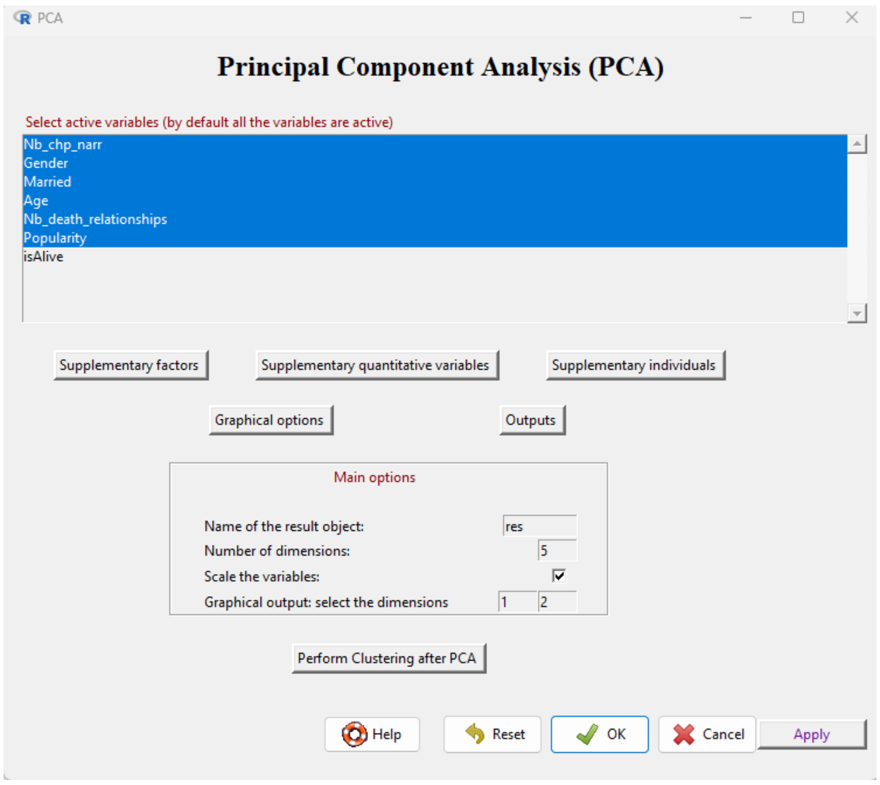
An 'Internet' application window opens. All you have to do is explore and click.

Warnings : Don't forget to quit the application when you're finished, as R waits for you to do so before continuing.

Partie 3 : Hierarchical ascending classification

CLICK BUTTON TEAM

Let's go back to the creation of the ACP.



PCA

Principal Component Analysis (PCA)

Select active variables (by default all the variables are active)

- Nb_chp_narr
- Gender
- Married
- Age
- Nb_death_relationships
- Popularity
- isAlive

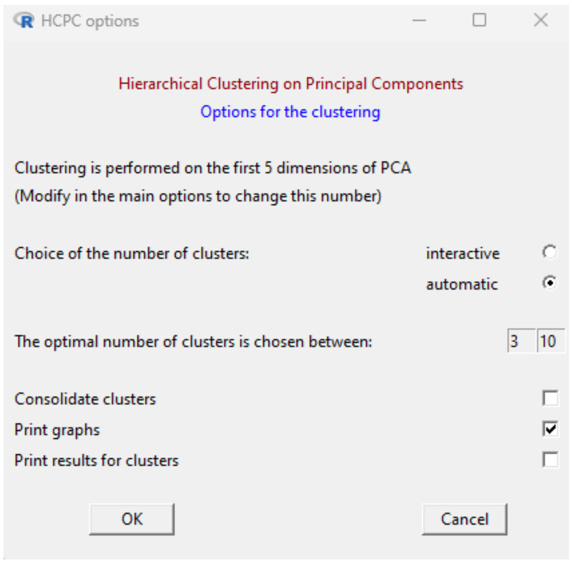
Supplementary factors
Supplementary quantitative variables
Supplementary individuals
Graphical options
Outputs

Main options

Name of the result object:	res
Number of dimensions:	5
Scale the variables:	✓
Graphical output: select the dimensions	1 2

Perform Clustering after PCA
Help
Reset
OK
Cancel
Apply

Then click on 'perform clustering after PCA'.



HCPC options

Hierarchical Clustering on Principal Components

Options for the clustering

Clustering is performed on the first 5 dimensions of PCA

(Modify in the main options to change this number)

Choice of the number of clusters: interactive [✓ automatic]

The optimal number of clusters is chosen between: 3 10

Consolidate clusters	☐
Print graphs	✓
Print results for clusters	☐

Then click on OK.

Once again, you may be confronted with an overlay of all the graphics.

```
got.PCA<-got[, c("Nb_chp_narr", "Gender", "Married", "Age",
  "Nb_death_relationships", "Popularity")]
res<-PCA(got.PCA , scale.unit=TRUE, ncp=5, graph = FALSE)
res.hcpc<-HCPC(res ,nb.clust=-1,consol=FALSE,min=3,max=10,graph=TRUE)
print(plot.PCA(res, axes=c(1, 2), choix="ind", habillage="none",
  col.ind="black", col.ind.sup="blue", col.quali="magenta", label=c("ind",
  "ind.sup", "quali"),new.plot=TRUE))
print(plot.PCA(res, axes=c(1, 2), choix="var", new.plot=TRUE,
  col.var="black", col.quanti.sup="blue", label=c("var", "quanti.sup"),
  lim.cos2.var=0))
summary(res, nb.dec = 3, nbelements=10, nbind = 10, ncp = 3, file="")
remove(got.PCA)
```

I advise you to isolate blocks of code.

```
got.PCA<-got[, c("Nb_chp_narr", "Gender", "Married", "Age",
  "Nb_death_relationships", "Popularity")]

res<-PCA(got.PCA , scale.unit=TRUE, ncp=5, graph = FALSE)

res.hcpc<-HCPC(res ,nb.clust=-1,consol=FALSE,min=3,max=10,graph=TRUE)

print(plot.PCA(res, axes=c(1, 2), choix="ind", habillage="none",
  col.ind="black", col.ind.sup="blue", col.quali="magenta", label=c("ind",
  "ind.sup", "quali"),new.plot=TRUE))

print(plot.PCA(res, axes=c(1, 2), choix="var", new.plot=TRUE,
  col.var="black", col.quanti.sup="blue", label=c("var", "quanti.sup"),
  lim.cos2.var=0))

summary(res, nb.dec = 3, nbelements=10, nbind = 10, ncp = 3, file="")

remove(got.PCA)
```

CODING TEAM

```
res.hcpc<-HCPC(res ,nb.clust=-1,consol=FALSE,min=3,max=10,graph=TRUE)

  plot(res.hcpc)
  plot(res.hcpc,choice="tree")
  plot(res.hcpc,choice="map")
```