



## dynGraph: interactive visualization of “factorial planes” integrating numerical indicators

S. Lê

Agrocampus Ouest

65 rue de Saint-Brieuc, Rennes, France

J. Durand

Agrocampus Ouest

65 rue de Saint-Brieuc, Rennes, France

---

### Abstract

**dynGraph** is a visualization software that has been initially developed for the **FactoMineR** package, an R package dedicated to multivariate exploratory methods such as principal components analysis, (multiple) correspondence analysis and multiple factor analysis. The main objective of **dynGraph** is to allow the user to explore interactively graphical outputs provided by multidimensional methods by visually integrating numerical indicators.

*Keywords:* visualization, multivariate data analysis, **FactoMineR**.

---

## 1. Introduction

**dynGraph** (Lê and Durand (2010)) is a visualization software that has been initially developed for the **FactoMineR** package (Lê *et al.* (2008) and Husson *et al.* (2008)), an R package dedicated to multivariate exploratory methods such as principal components analysis, (multiple) correspondence analysis and multiple factor analysis; **dynGraph** has been extended to allow the visualisation of data frames. The main objective of **dynGraph** is to allow the user to explore interactively graphical outputs provided by multidimensional methods by visually integrating numerical indicators. The first basic feature of **dynGraph** is the connecting line that appears whenever the user moves a label associated with an object, *i.e.* an individual, a variable or a category. Labelling of the different objects displayed on the graph can be easily set. Colours can be assigned to individuals according to a categorical variable of interest (external information). One of the main features of **dynGraph** is the way objects are displayed. Objects are displayed according to their quality of representation (a geometrical indicator that lies between 0 and 1). Of course the amount of information to be displayed can be easily set by the user with a cursor: graphical outputs can be analyzed interactively

from the most general piece of information to the most relevant one. Moreover, the font size of each label associated with an object is proportional to the importance of the object in the analysis which facilitates tremendously the interpretation of the results. Besides, different criteria can be used to assess the importance of an object and this information is calculated via R and the **FactoMineR** package. Finally, by clicking on one of the dimensions provided by the analysis, the user gets a list of the variables that may explain the dimension significantly that will help him to interpret the data.

## 2. An illustrative example

We will develop an example throughout this paper using the “*decathlon*” dataset of the **FactoMineR** package which refers to athletes’ performance during two athletic meetings. The data set is made of 41 rows and 13 columns. The first twelve columns are continuous variables: the first ten columns correspond to the performance of the athletes for the 10 events of the decathlon and the columns 11 and 12 correspond respectively to the rank and the points obtained. The last column is a categorical variable corresponding to the athletic meeting (2004 Olympic Game or 2004 Decastar).

As in [Lê et al. \(2008\)](#), a principal component analysis (PCA) is performed with the first ten variables as “active” variables, the two following ones as “quantitative supplementary” variables, finally the last one as “qualitative supplementary” variable. In other words, the ten first variables are the ones that are used directly to build the main dimensions of variability whereas the three last ones are the ones that are used indirectly to interpret the data: as we are in a PCA framework, active variables have to be quantitative whereas illustrative variables can be either quantitative either qualitative.

```
> data(decathlon)
> res.pca <- PCA(decathlon, quanti.sup = 11:12, quali.sup=13)
> res.pca
```

```
**Results for the Principal Component Analysis (PCA)**
```

The analysis was done on 41 individuals, described by 13 variables

\*The results are available in the following objects:

	nom	description
1	"\$eig"	"eigenvalues"
2	"\$var"	"results for the variables"
3	"\$var\$coord"	"coordinates of the variables"
4	"\$var\$cor"	"correlations variables - dimensions"
5	"\$var\$cos2"	"cos2 for the variables"
6	"\$var\$contrib"	"contributions of the variables"
7	"\$ind"	"results for the individuals"
8	"\$ind\$coord"	"coord. for the individuals"

```

9 "$ind$cos2"           "cos2 for the individuals"
10 "$ind$contrib"      "contributions of the individuals"
11 "$quanti.sup"       "results for the supplementary quantitative variables"
12 "$quanti.sup$coord" "coord. of the supplementary quantitative variables"
13 "$quanti.sup$cor"   "correlations supp. quantitative variables - dimensions"
14 "$quali.sup"        "results for the supplementary qualitative variables"
15 "$quali.sup$coord"  "coord. of the supplementary categories"
16 "$quali.sup$vttest" "v-test of the supplementary categories"
17 "$call"             "summary statistics"
18 "$call$centre"      "mean for the variables"
19 "$call$ecart.type"  "standard error for the variables"
20 "$call$row.w"       "weights for the individuals"
21 "$call$col.w"       "weights for the variables"

```

The `dynGraph()` function is directly applied to the list of results provided by the `PCA()` function of the **FactoMineR** package which contains for each principal component:

- its eigenvalue,
- the coordinate, quality of representation, contribution of each variable,
- the coordinate, quality of representation, contribution of each individual,
- the coordinate and the quality of representation of each supplementary variable (quantitative and/or qualitative).

```
> dynGraph(res.pca)
```

Those results, as well as the ones provided by the `dimdesc` function are then transferred to the `dynGraph` Java application by mean of the **rjava** package; let's recall that `dimdesc` (that stands for *dimension description*) is a **FactoMineR** package function designed to point out the variables and the categories that are the most characteristic of each component.

The `dynGraph` Java application then provides an interactive graphical output divided into three frames (cf. Figure 1): the scatterplot of the individuals on the first two components, the scatterplot of the variables on the first two components, a plot with the first two quantitative variables as x-axis and y-axis.

## 2.1. Customizing labels

In a lot of applications, knowing the “identity” of your statistical units or individuals is priceless when interpreting the results. The **dynGraph** package allows you an easy handling of the labels. The main features related to the labels allow you to:

- move the labels without any loss of information since each object of the plot (individuals, variables, categories) is connected to its label by a line that appears when the label is selected (cf. Figure 2);

#### 4 *dynGraph*: interactive visualization of “factorial planes” integrating numerical indicators

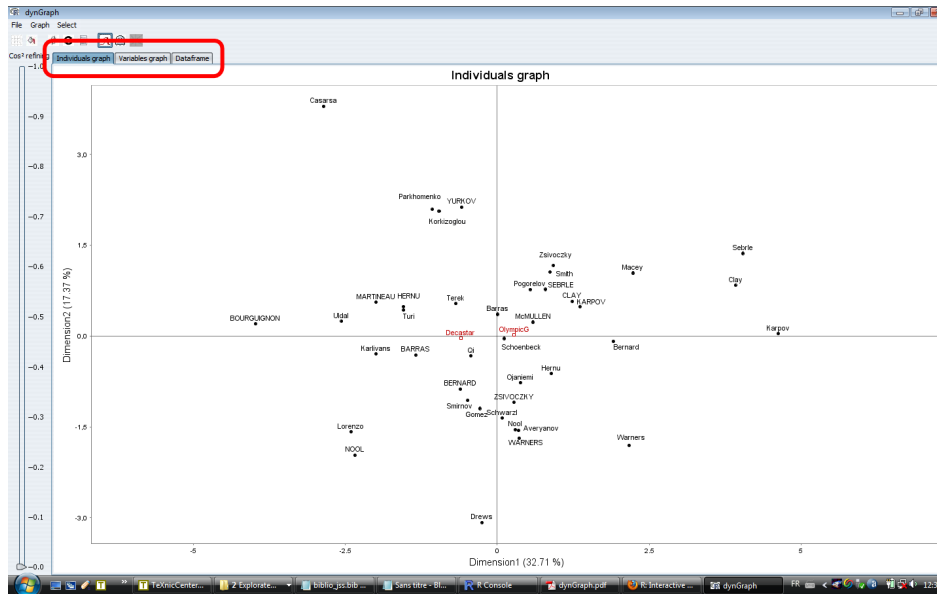


Figure 1: The *dynGraph* Java application/environment: three different points of view on the data

- hide very easily all the labels (to get an idea of the shape of the only scatterplot), the labels that are selected, the labels that are not selected (by mean of the “*reverse selection*” icon) (cf. Figure 3);
- have an access to the objects by mean of their labels through a list, in order to select them (cf. Figure 3).

### 2.2. Hiding objects (individuals, variables, categories)

When the graphical output is too crowded, the *dynGraph* package allows you to hide some of the objects that are represented:

- by selecting them the usual “*click and drag*” way combined with the “*reverse selection*” icon if needed;
- by selecting them through a list;
- by using the cursor on the left and by representing the objects that are well represented (according to the notion of quality of representation defined in *Lê et al. (2008)*) (cf. Figure 4).

### 2.3. Adding external information

When objects are represented on a two dimensional graphical output, it can be interesting to add another third or fourth dimension by mean of external information that may come from a quantitative variable and/or a qualitative variable. With *dynGraph* you can:

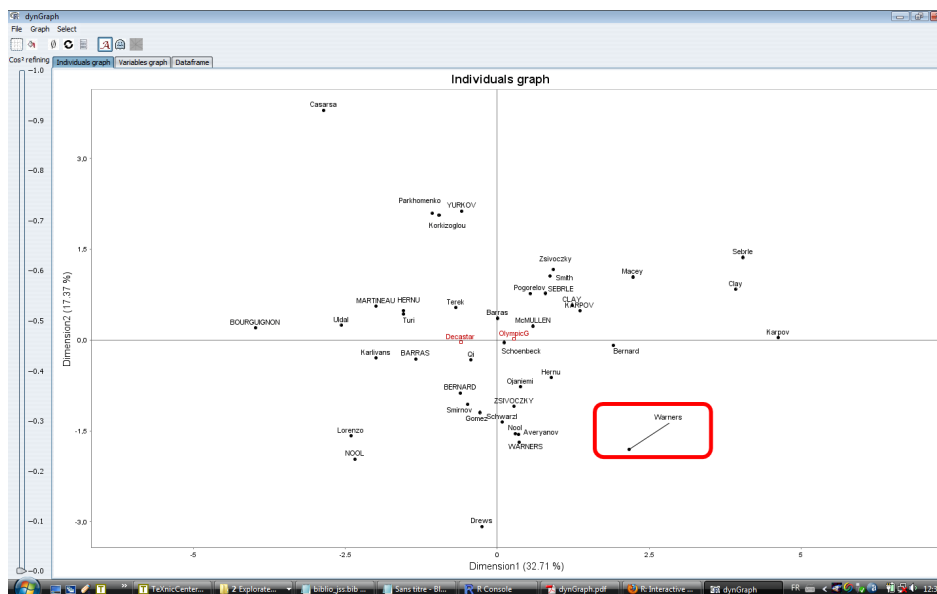


Figure 2: The dynGraph Java application: connecting labels and objects

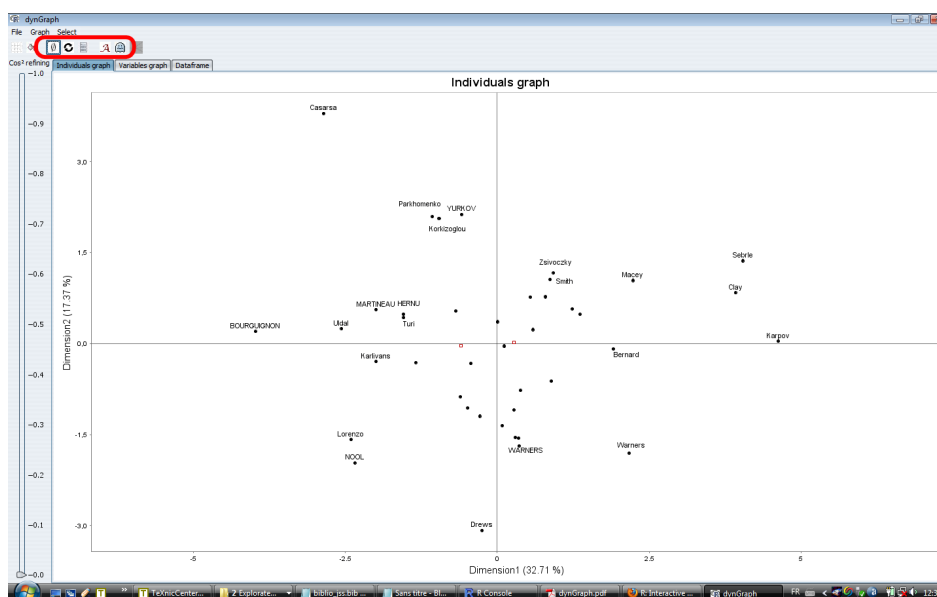


Figure 3: The dynGraph Java application: displaying labels

- colour individuals according to the group they belong to (where groups are induced by a qualitative variable);
- represent labels associated to objects using a font size proportional to the absolute value of any quantitative variable of the dataset, including quality of representation of objects or contribution to the construction of the axes (cf. Figure 5);

## 6 dynGraph: interactive visualization of “factorial planes” integrating numerical indicators

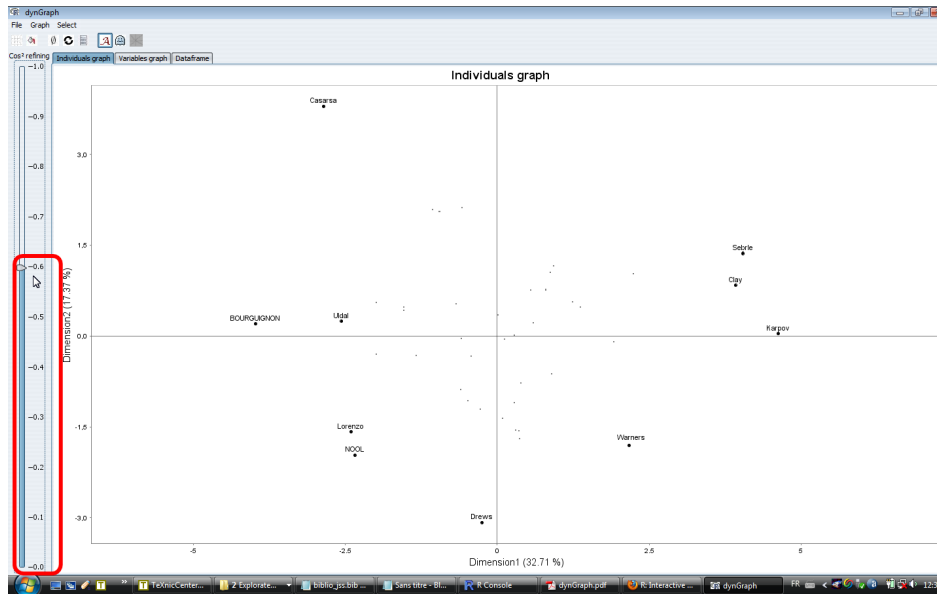


Figure 4: The dynGraph Java application: hiding objects

- represent individuals using a disc proportional to the absolute value of any quantitative variable of the dataset, including quality of representation of individuals or contribution to the construction of the axes;
- combine the three previous possibilities.

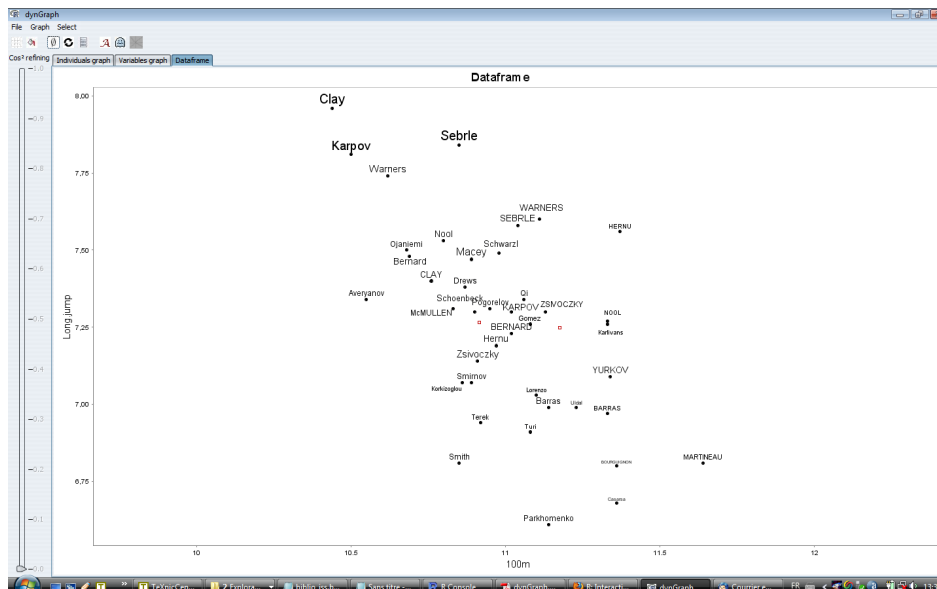


Figure 5: The dynGraph Java application: adding a third dimension (in this example, the total number of points, which is an illustrative variable)

## 2.4. Interpreting results

Multivariate exploratory techniques are somehow subjective when interpreting results. Hence a clear need for integrating useful information when interpreting dimensions provided by PCA for instance. Since a dimension is a linear combination of the “active” variables used when performing the analysis, it makes sense to highlight variables that are significantly correlated to this particular variable. When clicking on a dimension (cf. Figure 6) (or any variable of the “correlation circle” (cf. Figure 7)) **dynGraph** provides a list of variables that are significantly linked to the dimension or the variable of interest:

- when names of variables appear in grey, it means that those variables are not significantly linked;
- when names of variables appear in blue, it means that those variables are significantly linked in a negative way (in other words the correlation coefficient between the two variables is significantly negative);
- when names of variables appear in red, it means that those variables are significantly linked in a positive way (in other words the correlation coefficient between the two variables is significantly positive).

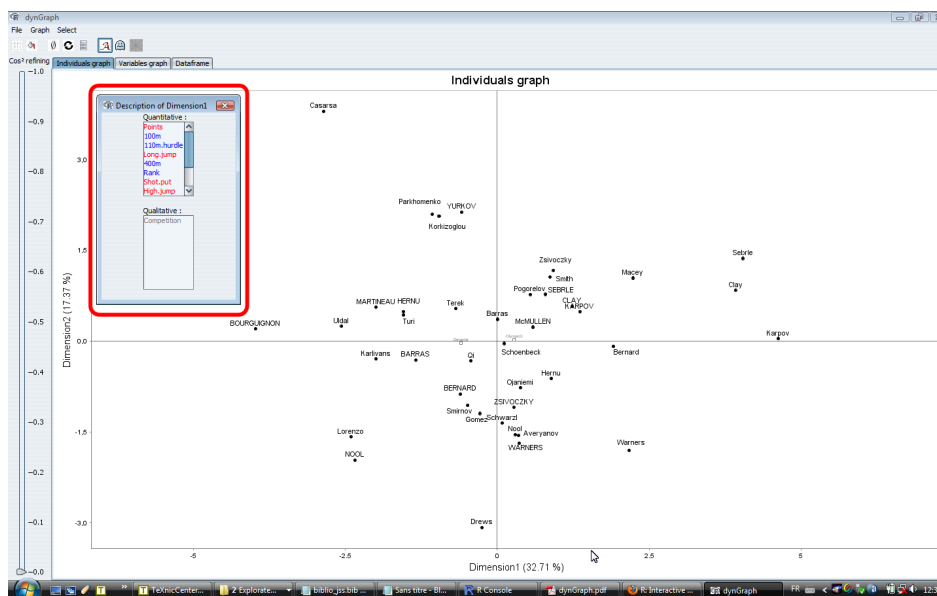


Figure 6: The dynGraph Java application: interpreting components

## 2.5. Saving your work

Once the interactive exploration of the graphical output done, you may save your work in many different kind of formats:

- a special format that allows you to come back to your interactive exploration as you left it;

## 8 *dynGraph*: interactive visualization of “factorial planes” integrating numerical indicators

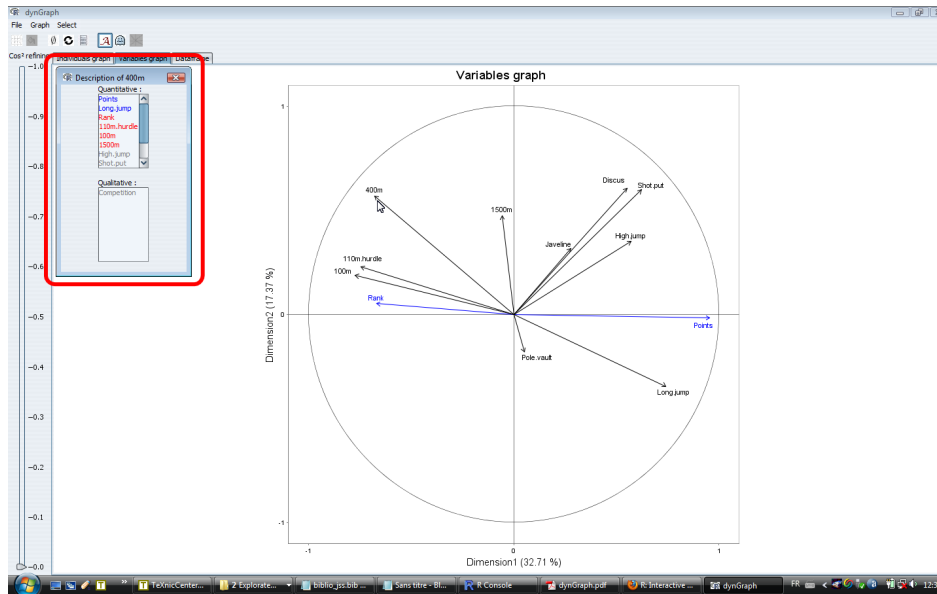


Figure 7: The *dynGraph* Java application: interpreting variables

- the usual formats (*.png*, *.svg*, *.pdf*, *.emf*) that allow you to integrate your work in most applications.

### 3. Concluding remarks

This paper presented the R package **dynGraph** designed for the interactive exploration of factorial planes provided by the **FactoMineR**. Beyond the connecting line that links an object to its label, the different features that allow an easy handling of the labels and the various scatterplots, **dynGraph** reflects a geometrical point of view mainly used in France when looking at multivariate data. This point of view allows adding external information such as quality of representation or contribution to the construction of a component. **dynGraph** was extended to dataframes and therefore can be used directly on a dataset, without any pre-processing, when at least two quantitative variables describe individuals. You will find a demonstration of the main features of the package at the following address, <http://dyngraph.free.fr>.

### References

- Husson F, Josse J, Lê S, Mazet J (2008). *FactoMineR: Factor Analysis and Data Mining with R*. R package version 1.09, URL <http://factominer.free.fr>, <http://www.agrocampus-rennes.fr/math/>.
- Lê S, Durand J (2010). *dynGraph: Interactive visualization of dataframes and factorial planes*. R package version 0.99100403, URL <http://dyngraph.free.fr>.



Lê S, Josse J, Husson F (2008). “FactoMineR: an R package for multivariate analysis.” *Journal of Statistical Software*, **25** (1), 1–18.

**Affiliation:**

Sébastien Lê  
Agrocampus Rennes  
UMR CNRS 6625  
65 rue de Saint-Brieuc  
35042 Rennes Cedex, France  
E-mail: [sebastien.le@agrocampus-ouest.fr](mailto:sebastien.le@agrocampus-ouest.fr)  
URL: <http://www.agrocampus-ouest.fr/math/le/>