Technical Report – Agrocampus
Applied Mathematics Department, September 2010.
http://www.agrocampus-ouest.fr/math/

Principal component methods - hierarchical clustering - partitional clustering: why would we need to choose for visualizing data?

François Husson
Agrocampus

Julie Josse
Agrocampus

Jérôme Pagès
Agrocampus

Abstract

This paper combines three exploratory data analysis methods, principal component methods, hierarchical clustering and partitioning, to enrich the description of the data. Principal component methods are used as preprocessing step for the clustering in order to denoise the data, transform categorical data in continuous ones or balanced groups of variables. The principal component representation is also used to visualize the hierarchical tree and/or the partition in a 3D-map which allows to better understand the data. The proposed methodology is available in the `HCPC` (Hierarchical Clustering on Principal Components) function of the **FactoMineR** package.

Keywords: Exploratory Data Analysis, Principal Component Methods, PCA, Hierarchical Clustering, Partitioning, Graphical Representation.

1. Introduction

Exploratory Data Analysis (EDA) refers to all descriptive methods for multivariate data set which allow to describe and visualize the data set. One of the central issue of these methods is to study the resemblances and differences between individuals from a multidimensional point of view. EDA is crucial in all statistical analyses and can be used as a main objective or as a preliminary study before modelling for example. Three kinds of methods are distinguished in this paper. The first kind is principal component methods such as Principal Component Analysis (PCA) for continuous variables, Multiple Correspondence Analysis (MCA) for categorical variables (Greenacre 2006), Multiple Factor Analysis (MFA) in the sense of Escofier and Pagès (1998) for variables structured by groups, etc. Individuals are considered in a high dimensional Euclidean space and studying the similarities between individuals means studying the shape of the cloud of points. Principal component methods then approximate

this cloud of points into an Euclidean subspace of lower dimensions while preserving as much as possible the distances between individuals. Another way to study the similarities between individuals with respect to all the variables is to perform a hierarchical clustering. Hierarchical clustering requires to define a distance and an agglomeration criterion. Many distances are available (Manhattan, Euclidean, etc.) as well as several agglomeration methods (Ward, single, centroid, etc.). The indexed hierarchy is represented by a tree named a dendrogramm. A third kind of method is partitional clustering. Many algorithms of partitional clustering are available and the most famous one is the K-means algorithm. This latter is based on the Euclidean distance. Clusters of individuals are then described by the variables. The aim of this paper is to combine the three kinds of methods, principal component methods, hierarchical clustering and partitional clustering to better highlight and better describe the resemblances between individuals. The three methods can be combined if the same distance (the Euclidean one) between individuals is used. Moreover, the Ward criterion has to be used in the hierarchical clustering because it is based on the multidimensional variance (*i.e.* inertia) as well as principal component methods. Section 2 describes how principal component methods can be used as a pre-processing step before hierarchical clustering and partitional clustering. As usual in clustering, it is necessary to define the number of clusters. Section 3 describes an empirical criterion to choose the number of clusters from a hierarchical tree. Section 4 then focuses on graphical representations and how the three methods complement each other. Finally section 5 gives an example on a real data set and a second example which consists in converting continuous variable(s) in categorical one(s) in a straightforward way.

2. Principal component methods as a pre-process for clustering

The core idea common to all principal component methods is to describe a data set (X with I individuals and K variables) using a small number ($S < K$) of uncorrelated variables while retaining as much information as possible. The reduction is achieved by transforming the data into a new set of continuous variables called the principal components.

2.1. Case of continuous variables

Hierarchical clustering as well as partitional clustering can be performed on the principal components of the PCA (*i.e.* the scores scaled to the associated eigenvalues). If all the components are used, the distances between individuals are the same than the ones obtained from the raw data set, and consequently the subsequent analysis remains the same. It is then more interesting to perform the clustering onto the first S principal components. Indeed, PCA can be viewed as a denoising method which separates signal and noise: the first dimensions extract the essential of the information while the last ones are restricted to noise. Then without the noise in the data, the clustering is more stable than the one obtained from the original distances. Consequently, if a hierarchical tree is built from another subsample of individuals, the shape of the top of the hierarchical tree remains approximately the same. PCA is thus considered as a preprocessing step before performing clustering methods. The number of dimensions kept for the clustering can be chosen with several methods (Jolliffe 2002). If this number is too small, it leads suppression of information. It is less problematic to specify an excessive number of clusters than a too small number that leads to loss of information.

2.2. Case of categorical variables and mixed variables

Clustering on categorical variables is an own research domain. A lot of resemblance measures exist such as Jaccard index, Dice's coefficient, Sørensen's quotient of similarity, simple match, etc. However, these indices are well-fitted for presence/absence data. When categorical variables have more than two categories, it is usual to use the χ^2-distance. Performing a clustering with the χ^2-distance is equivalent to perform a clustering onto all the principal components issue from Multiple Correspondence Analysis. MCA can then be viewed as a way to code categorical variables into a set of continuous variables (the principal components). As for PCA only the first dimensions can be retained to stabilize the clustering by deleting the noise from the data. Performing a clustering onto the first principal components of MCA is a very usual practice especially for questionnaires.

In the same way, it is possible to take into account both categorical and continuous variables in a clustering. Indeed, principal components can be obtained for mixed data with methods such as the Hill-Smith method (Hill and Smith 1976; Pagès 2004a). From the (first) principal components, distances between individuals are derived and a clustering can then be performed.

2.3. Taking into account a partition on the variables

Data sets are often organized into groups of variables. This situation may arise when data are provided from different sources. For example, in ecological data, soils can be described by both spectroscopy variables and physico-chemical measures. It frequently happens to have more spectrum variables than physico-chemical ones. Consequently, the Euclidean distances between individuals are almost due to the spectrum data. However, it may be interesting to take into account the group structure to compute the distances and to balance the influence of each data measurements. A solution is to perform a clustering onto the principal components of multi-way methods such as Multiple Factor Analysis (Escofier and Pagès 1998; Pagès 2004b). The core of MFA is a weighted PCA which allows to balance the influence of each group of variables in the analysis. In other words, a particular metric is assigned to the space of the individuals. The complete data set X is the concatenation of J groups of variables: $X = [X_1, X_2, ...X_J]$. The first eigenvalue λ_1^j associated with each data set is computed. Then a global PCA is performed on $[X_1/\sqrt{\lambda_1^1}\ ,\ X_2/\sqrt{\lambda_1^2}\ , ...,\ X_J/\sqrt{\lambda_1^J}]$. Each variable within one group is scaled by the same value to preserve the structure of each group (*i.e.* the shape of each sub-cloud of points), whereas each group is scaled by a different value. The idea of the weighting in MFA is in the same vein than the standardization in PCA where a same weight is given to each variable to balance the influence of each variable. A clustering performed on the first principal components issues from MFA allows to create a clustering balancing the influence of each group of variables.

In some data sets, variables are structured according to a hierarchy leading to groups and subgroups of variables. This case is frequently encountered with questionnaires structured into topics and subtopics. As for groups of variables, it is interesting to take the group and sub-group structure when computing distances between individuals. The clustering can then be performed onto the principal components of methods such as hierarchical multiple factor analysis (Le Dien and Pagès 2003a,b) which is an extension of MFA to the case where variables are structured according to a hierarchy.

3. Hierarchical clustering and partitioning

3.1. Ward's method

Hierarchical trees considered in this paper use the Ward's criterion. This criterion is based on the Huygens theorem which allows to decompose the total inertia (total variance) in between and within-group variance. The total inertia can be decomposed:

$$\begin{aligned}\sum_{k=1}^{K}\sum_{q=1}^{Q}\sum_{i=1}^{I_q}(x_{iqk}-\bar{x}_k)^2 &= \sum_{k=1}^{K}\sum_{q=1}^{Q} I_q(\bar{x}_{qk}-\bar{x}_k)^2 &+ \sum_{k=1}^{K}\sum_{q=1}^{Q}\sum_{i=1}^{I_q}(x_{iqk}-\bar{x}_{qk})^2,\\ \text{Total inertia} &= \text{Between inertia} &+ \text{Within inertia},\end{aligned}$$

with x_{iqk} the value of the variable k for the individual i of the cluster q, $\bar{x}_{qk}$ the mean of the variable k for cluster q, $\bar{x}_k$ the overall mean of variable k and I_q the number of individuals in cluster q.

The Ward's method consists in aggregating two clusters such that the growth of within-inertia is minimum (in other words minimising the reduction of the between-inertia) at each step of the algorithm. The within inertia characterises the homogeneous of a cluster.

The hierarchy is represented by a dendrogram which is indexed by the gain of within-inertia. As previously mentioned, the hierarchical clustering is performed onto the principal components.

3.2. Choosing the number of clusters from a hierarchical tree

Choosing the number of clusters is a core issue and several approaches have been proposed. Some of them rest on the hierarchical tree. Indeed, a hierarchical tree can be considered as a sequence of nested partitions from the one in which each individual is a cluster to the one in which all the individuals belong in the same cluster. The number of clusters can then be chosen looking at the overall appearance (or the shape) of the tree, the bar plot of the gain in within inertia, etc. These rules are often based implicitly or not on the growth of inertia. They suggest a division into Q clusters when the increase of between-inertia between $Q-1$ and Q clusters is much greater than the one between Q and $Q+1$ clusters. An empirical criterion can formalize this idea. Let $\Delta(Q)$ the between-inertia increase when moving from $Q-1$ to Q clusters, the criterion proposed is:

$$\frac{\Delta(Q)}{\Delta(Q+1)}.$$

The number Q which minimised this criterion is kept.

The `HCPC` function (Hierarchical Clustering on Principal Components) presented below implements this calculation after having constructed the hierarchy and suggests an "optimal" level for division. When studying a tree, this level of division generally corresponds to the one expected merely from looking at it.

3.3. Partitioning

Different strategies are available to obtain clusters. The simplest one consists in keeping the Q clusters defined by the tree. A second strategy consists in performing a K-means algorithm

with the number of clusters fixed at Q. Another strategy combines the two previous ones. The partition obtained from the cut of the hierarchical tree is introduced as the initial partition of the K-means algorithm and several iterations of this algorithm are done. The partition resulting from this algorithm is finally retained. Usually, the initial partition is never entirely replaced, but rather improved (or "consolidated"). This improvement can be measured by inspecting the [(between inertia)/(total inertia)] ratio. However, the hierarchical tree is not in agreement with the chosen partition.

4. Complementarity of the three methods for the visualization

4.1. Visualization on the principal component representation

Principal component methods have only been used as a pre-processing step but they also give a framework to visualize data. The clustering methods can then be represented onto the map (often the two dimensional solution) provided by the principal component methods. The simultaneous use of the three methods enrich the descriptive analysis.

The simultaneous analysis of a principal component map and a hierarchical clustering mainly means representing the partition issue from the dendrogram on the map. It can be done by representing the centres of gravity of the partition (the highest nodes of the hierarchy). However, the whole hierarchical tree can be represented in three dimensions on the principal component map. When a partitional clustering is performed, the centres of gravity of this partition are represented onto the principal component map. For the two clustering methods, individuals can be coloured according to their belonging cluster.

In a representation with the principal component map, the hierarchical tree and the clusters, the approaches complement one another in two ways:

- firstly, a continuous view (the trend identified by the principal components) and a discontinuous view (the clusters) of the same data set are both represented in a unique framework;
- secondly, the two-dimensional map provides no information about the position of the individuals in the other dimensions; the tree and the clusters, defined from more dimensions, offer some information "outside of the map"; two individuals close together on the map can be in the same cluster (and therefore not too far from one another along the other dimensions) or in two different clusters (as they are far from one another along other dimensions).

4.2. Sorting individuals in a dendrogram

The construction of the hierarchical tree allows to sort the individuals according to different criteria. Let us consider the simple following example with eight elements that take the values 6, 7, 2, 0, 3, 15, 11, 12. Figure 1 gives two dendrograms that are exactly similar from the point of view of the clustering. The tree on the right takes into account additional information (the elements have been sorted according to their value) which can be useful to better highlight the similarities between individuals.

```
> X <- c(6,7,2,0,3,15,11,12)
> names(X) <- X
> library(cluster)
> par(mfrow=c(1,2))
> plot(as.dendrogram(agnes(X)))
> plot(as.dendrogram(agnes(sort(X))))
```

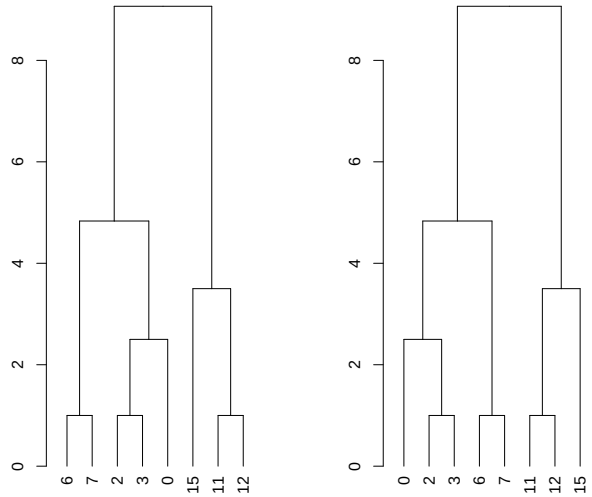

Figure 1: Hierarchical tree with original data and sorted data.

In the framework of multidimensional data, this idea can be extended by sorting the individuals according to their coordinates onto the first axis (*i.e.* the first principal component). It may improve the reading of the tree because individuals are sorted according to the main trend. In a certain sense, this strategy smoothes over the differences from one cluster to another.

5. Example

5.1. The temperature data

This paper presents the `HCPC` function (for Hierarchical Clustering on Principal Components) from the **FactoMineR** package (Lê, Josse, and Husson 2008; Husson, Josse, Lê, and Mazet 2009), a package dedicated to exploratory data analysis in R (R Development Core Team 2008). The aim of the `HCPC` function is to perform clustering and use the complementaries between clustering and principal component methods to better highlight the main feature of the data set. This function allows to perform hierarchical clustering and partitioning on the principal components of several methods, to choose the number of clusters, to visualize the tree, the partition and the principal components in a convenient way. Finally it provides description of the clusters.

The first example deals with the climates of several European countries. The dataset gathers

the temperatures (in Celsius) collected monthly for big European cities. We will focus on the first 23 rows of the data set which correspond to the main European capitals. A PCA is performed on the data where variables are standardized (even if the variables have the same unit) to give the same weight to each variable. The first 12 variables correspond to the temperatures and are used as active variables while the other variables are used as supplementary variables (four continuous and one categorical). Supplementary variables do not intervene in the construction of the principal components but are useful to enrich the interpretation of the PCA.

The first two dimensions of the PCA explained 97% of the total inertia. A hierarchical clustering is performed on the first two principal components. In this case, using PCA as a pre-processing step is not decisive since the distances between the capitals calculated from the first two dimensions are roughly similar to those calculated from all the dimensions. Note that as supplementary variables do not intervene in the distances calculus, they do not intervene for the clustering but they can be useful to describe the clustering.

The code to perform the PCA and the clustering is:

```
> library(FactoMineR)
> temperature <- read.table("http://factominer.free.fr/book/temperature.csv",
header=TRUE, sep=";", dec=".", row.names=1)
> res.pca <- PCA(temperature[1:23,], quanti.sup=13:16, quali.sup=17,
scale.unit=TRUE, ncp=2, graph = FALSE)
> res.hcpc <- HCPC(res.pca, nb.clust=0, conso=0, min=3, max=10)
```

The `PCA` function keeps the first two dimensions (`ncp=2`) and thus the hierarchical clustering only used these two dimensions. The hierarchical clustering is performed via the `HCPC` function on the outputs `res.pca` of the `PCA` function. The shape of the dendrogram (see Fig. 2) suggests partitioning the capitals into three clusters. The optimal level of division suggested by the `HCPC` function and represented with a solid black line also indicates three clusters. This number of clusters is chosen between `min=3` and `max=10` by default (if the minimum number of clusters equals 2, the procedure often defines an optimal number of 2 clusters, that is why we suggest to use 3 by default). The user has to click on the graph to specify the number of clusters (the one suggested or another) since the argument `nb.clust=0` is used by default. If `nb.clust=-1` the optimal number of clusters is used and if `nb.clust` is an integer it fixes the number of clusters. This latter option is useful for users who want to use another criterion to define the number of clusters. The argument `conso=0` means that no partitional clustering is used to consolidate the partition obtained by the hierarchical tree.

The individuals are sorted according to the first principal component as far as possible. This is done using the argument `order=TRUE` (used by default). The individuals can be arranged according to another criterion; they have first to be arranged according to the chosen criterion, then PCA is performed and the argument `order=FALSE` is used in the `HCPC` function.

Remark. A hierarchical clustering can be performed on a raw data set with the `HCPC` function (the input is the data table considered as a `data.frame`). In this case, a non-standardised PCA is performed and all the components are kept for the clustering.

The outputs of the `HCPC` function contain many objects. The object `res.hcpc$call$t` (table 1) contains the results of the ascending hierarchical clustering. It gathers the following results:

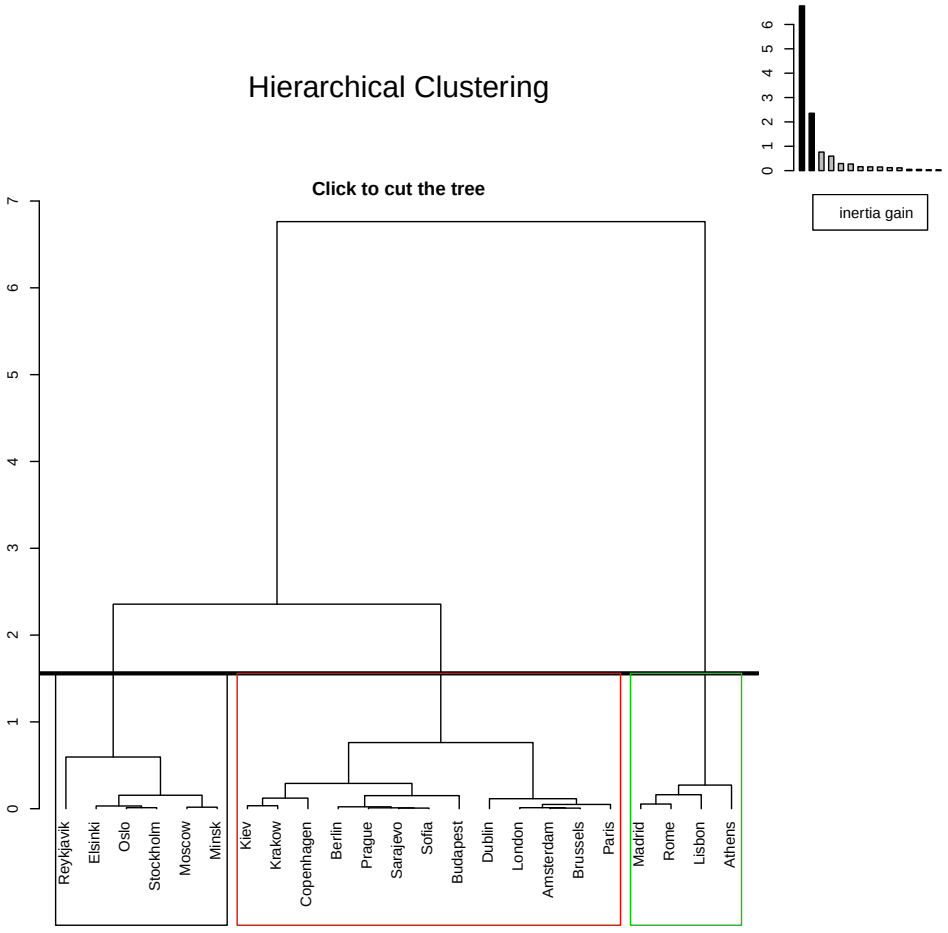

Figure 2: Hierarchical clustering; the individuals are sorted according to their coordinate on the first principal component.

- the outputs of the principal component method in `call$t$res`;
- the outputs of the `agnes` function (from the **cluster** package) in `call$t$tree`;
- the number of "optimal" clusters estimated (`$call$t$nb.clust`): this number is determined between the minimum and maximum number of clusters defined by the user and such that the criterion `$call$t$quot` might be as small as possible;
- the within-group inertia of the partitioning into Q clusters (`$call$t$intra`); for $Q = 1$ cluster (partition into one cluster) within-group inertia is equal to 12 (because there are 12 standardized variables), for 2 clusters 5.237, etc.
- the increase in between-group inertia (or equivalently the decrease in within-group inertia) when moving from Q to $Q + 1$ clusters is given by `$call$t$inter`; for 2 clusters (*i.e.* moving from 1 to 2 clusters) the increase in between-group inertia is equal to 6.763, for 3 clusters (*i.e.* moving from 2 to 3 clusters) 2.356, etc.

```
$call$t$nb.clust
[1] 3

$call$t$intra
 [1] 11.999  5.237  2.881  2.119  1.523  1.232  0.959  0.798  0.643  0.492
[11]  0.370  0.255  0.201  0.152  0.118  0.087  0.065  0.047  0.036  0.024
[21]  0.014  0.007  0.000

$call$t$inert.gain
 [1] 6.763 2.356 0.762 0.596 0.291 0.272 0.161 0.155 0.151 0.122 0.115 0.054
[13] 0.049 0.034 0.031 0.022 0.017 0.012 0.012 0.010 0.007 0.007

$call$t$quot
[1] 0.550 0.736 0.719 0.809 0.779 0.832 0.806 0.766

$call$t$i
[1] 11.999
```

Table 1: Hierarchical clustering outputs.

- the ratio between two successive within-group inertias is given in `$call$t$quot` (for example $0.550 = 2.881/5.237$).

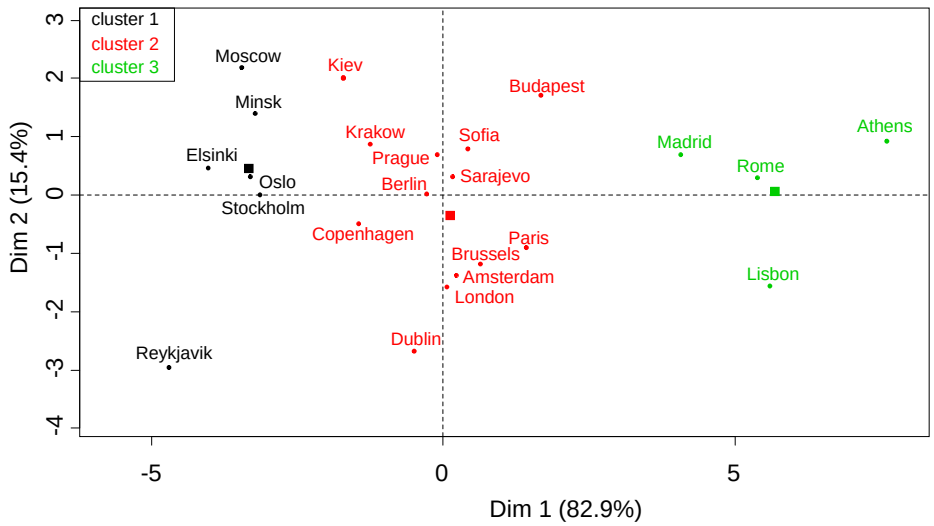

Figure 3: Representation of the clusters on the map induced by the first two principal components

The partitioning in three clusters is represented on the map produced by the first two principal components and the individuals are coloured according to their cluster (Fig. 3). The barycentre of each cluster is also represented by a square. The graph shows that the three clusters are well-separated on the first two principal components.

Figure 4 shows 3-dimensional representation of the hierarchical tree on the map produced by the first two principal components. In this graph, the principal components map, the

hierarchical tree and the partition issue from this tree bring different information that are superimposed to better visualize the data set.

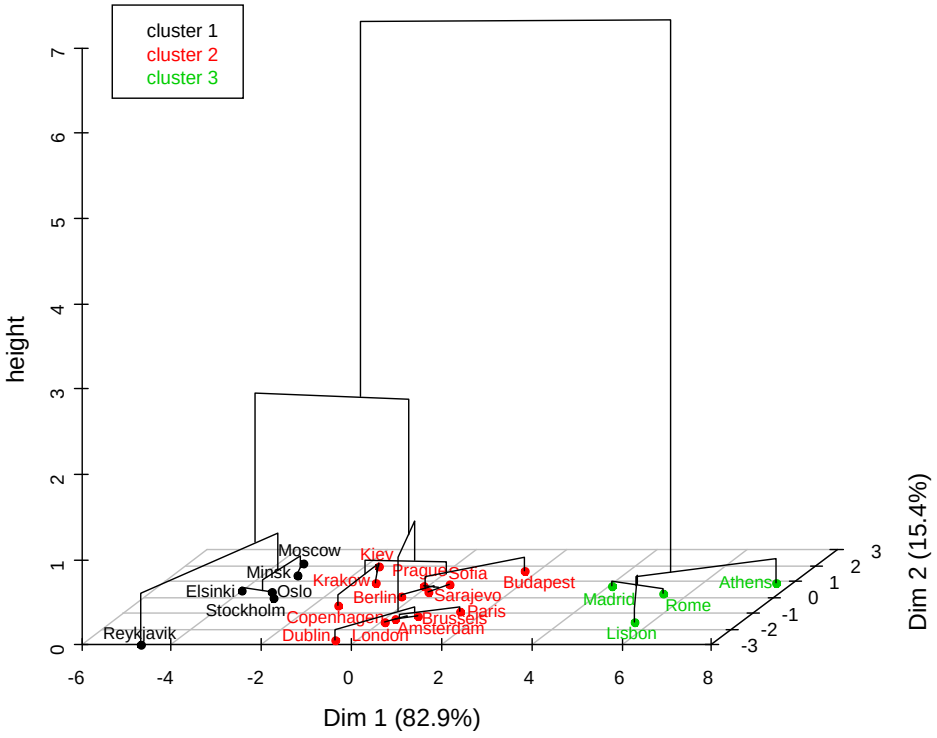

Figure 4: Hierarchical tree represented on the map induced by the first two principal components.

In the object `res.hcpc$data.clust`, the data set is available with a supplement column named `clust`. This column gives in label of the cluster to which each individual belongs.

After performing clustering, one of the major issue is the description of the clusters which can be achieved with the help of the outputs `desc.var`. All of the variables from the original data set are used for the description, whether the variables are continuous, categorical, active or supplementary. The methodology is described in the section 3.3 of Lê *et al.* (2008) and in Lebart, Morineau, and Warwick (1984). The results are gathered in table 2 for the continuous variables and in table 3 for the categorical ones. For continuous variables, it gives the average of a variable in the cluster (Mean in category), the average of the variable for the whole data set (Overall mean), the associated standard deviations and the p-value corresponding to the test of the following hypothesis: "the mean of the category is equal to the overall mean". A value of the v.test greater than 1.96 corresponds to a p-value less than 0.05; the sign of the v.test indicates if the mean of the cluster is lower or greater than the overall mean. The capitals from cluster 1 are characterised by below average temperatures throughout the year, and particularly in April (4.22 degrees on average for the capitals in this cluster compared with 8.38 degrees for all of the capitals), March, October, etc. None of the variables characterise the cities in cluster 2. The capitals in cluster 3 have a hot-climate since the average annual temperature of this cluster (15.7 degrees) is much higher than the average for all the capitals (9.37 degrees). The clusters are also characterised by categorical variables. First, a χ^2-test

is performed between the categorical variable(s) and the cluster variable. The p-value is less than 0.05 showing that the categorical variable *Area* is linked to the cluster variable (see table 3). Then, the categories of the cluster variable are characterised by the categories of the categorical variable(s). For example, the cluster 3 is characterised by the category *south*: there are more southerly cities in this cluster than in the others. Indeed, 80 % of southerly cities belong to cluster 3, and 100 % of the cities in cluster 3 are southerly cities. These percentages are high because 21.7 % of the cities are southerly. The hypergeometric test is significant since the p-value is less than 0.05.

The clusters are also described by the principal components. In order to do so, a description identical to that carried out by the continuous variables is conducted from the individuals' coordinates (principal components). The table 4 shows that the capitals in cluster 1 (and

```
$quanti
$quanti$`1`
          v.test   Mean in  Overall     sd in  Overall    p.value
                  category     mean  category       sd
Latitude   2.882    57.100 49.8800    5.7730    6.978 0.0039520
July      -2.156    16.350 18.9300    2.3880    3.329 0.0310500
June      -2.313    14.220 16.7700    2.3580    3.070 0.0207000
August    -2.624    14.980 18.3000    2.0460    3.526 0.0086790
May       -2.832    10.270 13.2700    2.1350    2.959 0.0046230
January   -2.855    -5.017  0.1739    2.8160    5.066 0.0043060
December  -2.866    -2.800  1.8430    1.9540    4.515 0.0041600
February  -3.055    -4.533  0.9565    2.5180    5.008 0.0022540
November  -3.085     0.500  5.0780    0.9798    4.136 0.0020380
September -3.160    10.530 14.7100    1.3470    3.682 0.0015770
Average   -3.239     5.233  9.3740    0.4346    3.563 0.0012010
October   -3.311     5.467 10.0700    0.6289    3.870 0.0009287
March     -3.390    -1.283  4.0610    1.1330    4.393 0.0006989
April     -3.415     4.217  8.3780    1.1620    3.395 0.0006367

$quanti$`2`
NULL

$quanti$`3`
          v.test   Mean in  Overall     sd in  Overall    p.value
                  category     mean  category       sd
Average    3.852    15.750  9.3740     1.394    3.563 0.0001173
September  3.809    21.230 14.7100     1.537    3.682 0.0001396
October    3.718    16.750 10.0700     1.911    3.870 0.0002011
August     3.705    24.380 18.3000     1.883    3.526 0.0002113
November   3.693    12.170  5.0780     2.264    4.136 0.0002218
July       3.604    24.500 18.9300     2.089    3.329 0.0003139
April      3.532    13.950  8.3780     1.176    3.395 0.0004129
March      3.449    11.100  4.0610     1.275    4.393 0.0005636
February   3.435     8.950  0.9565     1.744    5.008 0.0005926
June       3.389    21.600 16.7700     1.864    3.070 0.0007004
December   3.387     8.950  1.8430     2.337    4.515 0.0007058
January    3.292     7.925  0.1739     2.077    5.066 0.0009931
May        3.183    17.650 13.2700     1.553    2.959 0.0014570
Latitude  -3.225    39.420 49.8800     1.524    6.978 0.0012590
```

Table 2: Cluster description by the continuous variables.

```
$desc.var
$test.chi2
         p.value df
Area 0.001034572  6

$category
$category$`1`
NULL

$category$`2`
NULL

$category$`3`
           Cla/Mod Mod/Cla   Global     p.value   v.test
Area=South      80     100 21.73913 0.001129305 3.256159
```

Table 3: Cluster description by the categorical variables.

3, respectively) have a significantly weaker (or stronger, respectively) coordinate than the others on the first dimension. Let us recall that the principal components are used to define the clusters and consequently the cluster variable is not independent of the principal components. Then, the tests must only be used as descriptive tools to sort and select the principal components for the clusters description.

```
$desc.axes
$quanti
$quanti$`1`
      v.test   Mean in    Overall     sd in  Overall   p.value
               category       mean  category       sd
Dim.1 -3.224    -3.649  1.692e-16     0.553    3.154  0.001264

$quanti$`2`
      v.test   Mean in    Overall     sd in  Overall   p.value
               category       mean  category       sd
NULL

$quanti$`3`
      v.test   Mean in    Overall     sd in  Overall   p.value
               category       mean  category       sd
Dim.1  3.863     5.662  1.692e-16     1.264    3.154  0.000112
```

Table 4: Cluster description by the principal components.

It may be interesting to illustrate the cluster using individuals specific to that cluster. In order to do so, two different kinds of specific individuals are suggested: paragons, that is to say, the individuals which are closest to the centre of the cluster; and the specific individuals, that is to say those furthest from the centres of other clusters. The object `desc.ind$para` (see top of the table 5) contains, for each cluster, the individuals sorted by the distance between each individual and the centre of its cluster. Thus, Oslo is the capital which best represents the cities in cluster 1, whereas Berlin and Rome are the paragons of clusters 2 and 3 respectively. The object `desc.ind$dist` gives, for each cluster, the individuals sorted according to their distance (from the highest to the smallest) to the closest cluster centre. Formally, the first

individual in each cluster correspond to the one for which:

$$\max_{i \in q} \min_{q' \neq q} d(i, C_{q'}).$$

with $C_{q'}$ the barycentre of the cluster q'. Reykjavik is specific to cluster 1 as it is the city furthest from the centres of clusters 2 and 3 (see bottom of the table 5). Brussels and Athens are specific to clusters 2 and 3.

```
$desc.ind
$desc.ind$para
cluster: 1
     Oslo   Elsinki Stockholm     Minsk    Moscow
0.4314473 0.7040156 0.8928363 1.2830030 2.0302288
----------------------------------------------------
cluster: 2
   Berlin  Sarajevo    Prague     Sofia  Brussels
0.4038069 0.6043247 0.8653273 1.0777958 1.2604097
----------------------------------------------------
cluster: 3
     Rome    Lisbon    Madrid    Athens
0.3596129 1.7368270 1.8352325 2.1668535

$desc.ind$dist
cluster: 1
Reykjavik    Moscow   Elsinki     Minsk      Oslo
 5.444421  4.125370  4.118042  3.535233  3.322268
----------------------------------------------------
cluster: 2
 Brussels     Paris  Budapest    Dublin Amsterdam
 4.526557  4.381074  4.374254  4.303686  4.225883
----------------------------------------------------
cluster: 3
  Athens   Lisbon     Rome   Madrid
7.773633 5.835851 5.470176 4.309856
```

Table 5: Cluster description by the individuals.

5.2. Cut continuous variables into intervals

In this example, hierarchical clustering is used to convert a continuous variable into a categorical one. For example, with a data set containing continuous and categorical variables, it is possible to perform a mixed data analysis to explore and sum-up the data set or as pre-processing (see section 2.2). However, it may be interesting to convert the continuous variables into categorical ones and perform a MCA. This strategy allows to take into account non linear relationships between the variables.

A first strategy to cut a continuous variable in intervals consists in using "natural" clusters defined *a priori* (for example less than 18 years old, 18-30 years old, etc.). A second strategy consists in cutting in equal-count or equal-width clusters. Of course, the number of clusters needs to be chosen *a priori*.

Let us use the well-know Fisher's example and focus on the sepal length variable. With three clusters, the following lines of code allow to cut in equal-count:

```
> data(iris)
> vari <- iris$Sepal.Length
> nb.clusters <- 3
> breaks <- quantile(vari, seq(0,1,1/nb.clusters))
> Xqual <- cut(vari,breaks, include.lowest=TRUE)
> summary(Xqual)
[4.3,5.4] (5.4,6.3] (6.3,7.9]
       52        47        51
```

A third strategy determines the number of clusters and the cut-points from the data, for example from the histogram which represents the distribution of the variable (Fig. 5). However, this choice is not easy. We propose to use the dendrogram to choose the number of clusters

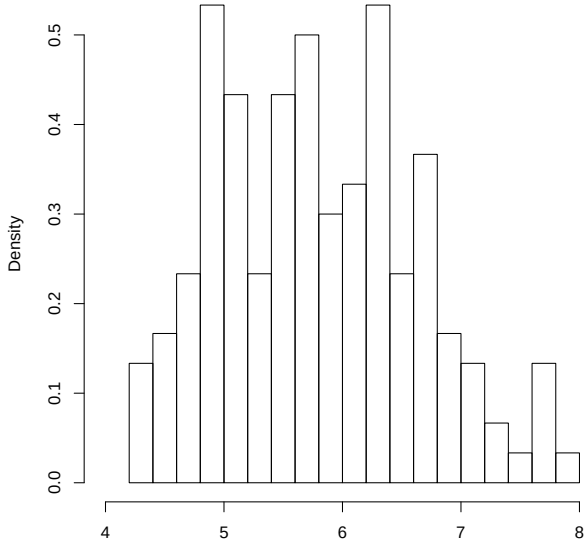

Figure 5: Histogram of sepal length.

and the partitioning (issue from the tree or the K-means algorithm) to define the clusters. The following lines of code allow to construct the partition from the hierarchical tree (using the empirical criterion based on inertia defined section 3.2) and the results are consolidated using the K-means algorithm (in practice, the K-means method converges very quickly when it is performed on one variable alone):

```
> res.hcpc <- HCPC(vari, min=2, max=10, iter.max=10)
```

By default, the `HCPC` function draws the hierarchical tree, the argument `iter.max=10` implies that the K-means method is performed. The hierarchical tree (Fig. 6 on the left) suggests to define three clusters. The x-axis of the tree (Fig. 6 on the right) corresponds to the individuals' values for the sepal length variable. It allows to better see the distances between individuals and between clusters.

A new categorical variable `new.fact` can then be defined in the following way:

```
> max.cla = unlist(by(res.hcpc$data.clust[,1],res.hcpc$data.clust[,2],max))
> breaks=c(min(vari),max.cla)
> new.fact = cut(vari, breaks, include.lowest=TRUE)
```

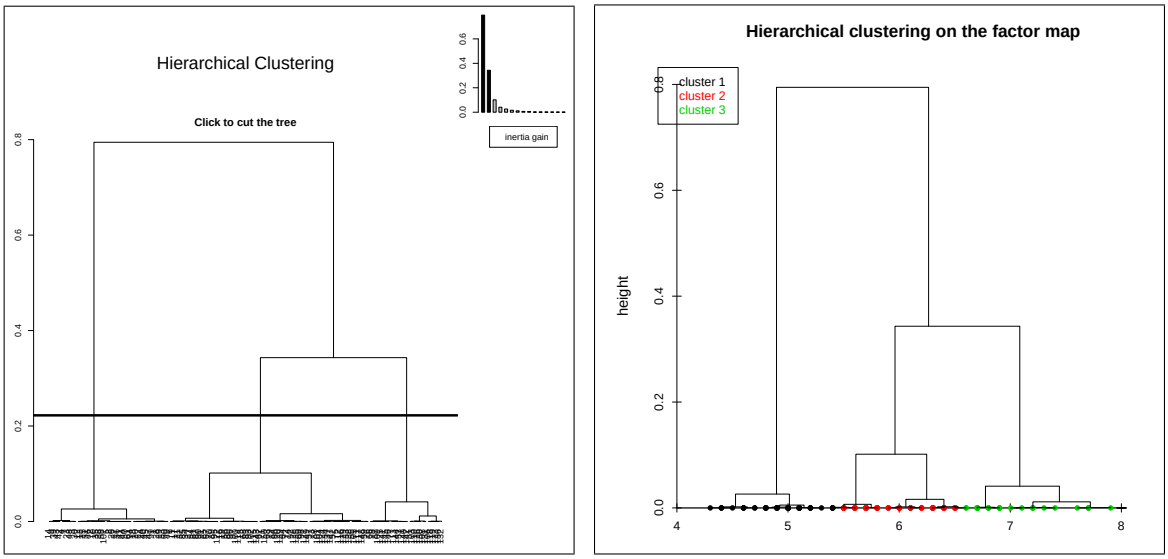

Figure 6: Dendrogram of the variable sepal length: the raw dendrogram with the "optimal level" to cut the graph (on the left) and the representation of the dendrogram with the individuals represented according to the sepal length variable on the x-axis (on the right).

```
> summary(new.fact)
[4.3,5.4] (5.4,6.5] (6.5,7.9]
       52        68        30
```

The breaks proposed allow to take into account the distribution of the variable.

If many continuous variables have to be cut into clusters, it can be tedious to determine the number of clusters and the cut-points variable by variable from the dendrogram (or an histogram). In such cases, the `HCPC` function is used variable by variable with the argument `nb.clust=-1` to detect and use the optimal number of clusters determined by the criterion (it is not necessary to click to define the number of clusters). The following lines of code are used to divide all of the continuous variables from the data set `iris` into clusters and to merge them in the new data set `iris.quali`:

```
> iris.quali <- iris
> for (i in 1:ncol(iris.quali)){
+   vari = iris.quali[,i]
+   if (is.numeric(vari)){
+     res=HCPC(vari, nb.clust=-1, min=2, graph=FALSE)
+     maxi = unlist(by(res$data.clust[,1], res$data.clust[,2],max))
+     breaks=c(min(vari),maxi)
+     new.fact = cut(vari, breaks, include.lowest=TRUE)
+     iris.quali[,i] = new.fact
+   } else {
+     iris.quali[,i] = iris[,i]
+   }
+ }
```

```
> summary(iris.quali)
   Sepal.Length    Sepal.Width  Petal.Length    Petal.Width        Species
 [4.3,5.4]:52    [2,3.1]  :94   [1,3]  :51    [0.1,0.6]: 50   setosa    :50
 (5.4,6.5]:68    (3.1,4.4]:56   (3,6.9]:99    (0.6,2.5]:100   versicolor:50
 (6.5,7.9]:30                                                 virginica :50
```

The resultant table `iris.quali` contains only categorical variables corresponding to the division into clusters of each of the continuous variables from the initial table `iris`.

6. Conclusion

Combining principal component methods, hierarchical clustering and partitional clustering, allows to better visualize data. Principal component methods can be used as preprocessing step for denoising the data, to transform categorical variables in continuous variables, to balance the influence of several groups of variables. It can also be useful to represent the partitional clustering and the hierarchical clustering on a map.

The visualization of the data proposed in this article can be used on data set where the number of individuals is small. When the number of individuals is very high, it is not possible to visualize the tree on the PCA map. Moreover, algorithms which construct hierarchical trees encounter many difficulties. However, a partition can be performed with an important number of clusters (for example 100) and then the hierarchical tree can be calculated from the centres of gravity of the partition weighted by the number of individuals of each cluster. Then the centres of gravity and the hierarchical tree can be represented on the factorial map. To do so, the `PCA` function can be performed using the argument `row.w` to affect weights of the centres of gravity considered as "individuals". From the principal components of the PCA, the hierarchical clustering can be performed as well as the partitional clustering and the results can be visualize on the principal component map.

The website http://factominer.free.fr/ gives other examples and use for the different methods.

References

Escofier B, Pagès J (1998). *Analyses factorielles simples et multiples*. Dunod.

Greenacre M (2006). *Multiple Correspondence Analysis and Related Methods*. Chapman & Hall.

Hill M, Smith J (1976). "Principal component analysis of taxonomic data with multi-state discrete characters." *Taxon*, **25**, 249–255.

Husson F, Josse J, Lê S, Mazet J (2009). *FactoMineR: Multivariate Exploratory Data Analysis and Data Mining with R*. R package version 1.12, URL `http://factominer.free.fr`.

Jolliffe IT (2002). *Principal Component Analysis*. Springer.

Lê S, Josse J, Husson F (2008). "FactoMineR: An R Package for Multivariate Analysis." *Journal of Statistical Software*, **25**(1), 1–18. ISSN 1548-7660. URL `http://www.jstatsoft.org/v25/i01`.

Le Dien S, Pagès J (2003a). "Analyse Factorielle Multiple Hiérarchique." *Revue de Statistique Appliquée*, **LI**, 83–93.

Le Dien S, Pagès J (2003b). "Hierarchical Multiple Factor Analysis: application to the comparison of sensory profiles." *Food Quality and Preference*, **14**, 397–403.

Lebart L, Morineau A, Warwick K (1984). *Multivariate descriptive statistical analysis*. Wiley.

Pagès J (2004a). "Analyse factorielle de données mixtes." *Revue Statistique Appliquée*, **LII**(4), 93–111.

Pagès J (2004b). "Multiple Factor Analysis: main features and application to sensory data." *Revista Colombiana de Estadística*, **4**, 7–29.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.