Journal of Statistical Software

April 2016, Volume 70, Issue 1. doi: 10.18637/jss.v070.i01

missMDA: A Package for Handling Missing Values in Multivariate Data Analysis

Julie Josse
Agrocampus Ouest Rennes

François Husson
Agrocampus Ouest Rennes

Abstract

We present the R package **missMDA** which performs principal component methods on incomplete data sets, aiming to obtain scores, loadings and graphical representations despite missing values. Package methods include principal component analysis for continuous variables, multiple correspondence analysis for categorical variables, factorial analysis on mixed data for both continuous and categorical variables, and multiple factor analysis for multi-table data. Furthermore, **missMDA** can be used to perform single imputation to complete data involving continuous, categorical and mixed variables. A multiple imputation method is also available. In the principal component analysis framework, variability across different imputations is represented by confidence areas around the row and column positions on the graphical outputs. This allows assessment of the credibility of results obtained from incomplete data sets.

Keywords: missing values, principal component analysis, single imputation, multiple imputation, multi-table data, mixed data, multiple correspondence analysis, multiple factor analysis.

1. Introduction

When starting a new project involving statistical analysis, it is important to first describe, explore and visualize the given data. Principal component methods can be useful in such cases, and several methods are available depending on the nature of the data: Principal component analysis (PCA) can be used for continuous data, multiple correspondence analysis (MCA) for categorical data (Gifi 1990; Greenacre and Blasius 2006; Husson and Josse 2014), factorial analysis for mixed data (FAMD) for both continuous and categorical data (Escofier 1979; Kiers 1991), and multiple factor analysis (MFA) for data structured in groups of variables (Escofier and Pagès 2008; Pagès 2015). These methods involve reducing data dimensionality in order to provide a subspace that best represents the data in the sense of maximizing the

variability of the projected points. From a technical point of view, the core of all these methods is the singular value decomposition (SVD) of certain matrices with specific metrics.

Unfortunately, data sets often have missing values, and most principal component methods available in software packages can only be applied to complete data sets. To this end, the **missMDA** package (Husson and Josse 2016) for the R system (R Core Team 2016) performs principal component methods on incomplete data, aiming at estimating parameters (left and right singular vectors and singular values) and obtaining graphical representations despite missing values. The package is based on the methodology presented in Josse and Husson (2012).

As we will see throughout the paper, missing value imputation is automatically associated with parameter estimation. Consequently, the **missMDA** package has a broad range of applications since it can be used to impute incomplete data sets with continuous, categorical or mixed variables (i.e., data with both types of variables). Then, it is possible to apply statistical methods. As it is based on a principal component method, imputation takes into account both similarities between individuals and relationships between variables. However, one has to be careful when applying statistical methods to imputed data sets since a unique value is predicted for a missing entry and therefore cannot reflect uncertainty in the prediction. Indeed, a model (explicit or not) is used to predict the value for the missing entry based on the observed data, so there is uncertainty associated with the prediction. This implies that when applying statistical methods to an imputed data set, the variance of the estimators is underestimated since variability due the imputation of missing values is not taken into account. This issue can be minimized by using multiple imputation methods (Rubin 1987) whereby several plausible values are predicted for each missing entry, leading to a set of imputed data tables. The variability between imputations reflects variance in predictions. Multiple imputation methods then perform the desired analysis on each imputed data set and combine the results.

In this paper, we first give a brief overview (Section 2) of other known software packages where the implemented methods might be considered to run principal component methods with missing values. Then, we present in detail our method to perform PCA with missing values in Section 3. Since the core of all principal component methods is SVD, the methodology we present serves as a basis for MCA (Section 4), FAMD (Section 5) and MFA (Section 6) with missing values. We briefly present each method as well as the algorithm to deal with missing values, and show how to perform analyses with **missMDA** on real data sets. In Section 7, we highlight the ability of the package to be used for single and multiple imputation. Note that the proposed methods were designed under the missing at random (MAR) assumption (Rubin 1976), meaning that we do not address the case missing non at random (MNAR) where the probability that a value is missing is related to the value itself. Such informative missing values require to model the mechanism that generates the missing entries.

2. Possible competitors

One of the pioneering works on missing values in PCA is that of Wold and Lyttkens (1969) in the field of chemometrics. They present the "NIPALS" algorithm which obtains the first principal component (also known as the scores) and first principal axis (also known as the loadings) from an incomplete data set using an alternating weighted least squares algorithm,

where two weighted simple linear regressions are alternated. Then, the following dimensions are obtained by applying the same method to the residuals matrix. This method is implemented in (commercial) software products dedicated to chemometrics such as **SIMCA** (Umetrics 2013) and **Unscrambler** (CAMO 2013), and a MATLAB toolbox (Eigenvector Research 2011). It is also implemented in R with the function `nipals` in the **ade4** package (Dray 2007) and the function `nipalsPca` in the **pcaMethods** package (Stacklies, Redestig, Scholz, Walther, and Selbig 2007). However, this algorithm has many drawbacks. It is known to be very unstable (i.e., it provides estimates of the parameters with large variability), does not minimize some explicit criterion and does not allow one to perform a standardized PCA with missing values (Josse, Pagès, and Husson 2009).

Theoretical advances have been made on the topic of missing values and PCA, leading to other methods being proposed. For instance, another alternating least squares algorithm (Gabriel and Zamir 1979), which can be seen as an extension of "NIPALS", directly estimates a subspace and does not work sequentially. Also, there is the iterative PCA algorithm (Kiers 1997) which estimates parameters and missing values simultaneously. This algorithm is more popular that the other since it can be used as a single imputation method to produce a complete data set, not only to perform PCA with missing values. However, Josse *et al.* (2009) and Ilin and Raiko (2010) highlighted this algorithm's overfitting problems, and suggested regularized versions to tackle these issues.

A regularized version of the iterative PCA algorithm is available in MATLAB toolboxes (Porta, Verbeek, and Kröse 2005; Ilin 2010). In R an implementation was provided by Stacklies *et al.* (2007) in the **pcaMethods** package. In the **missMDA** package we also implemented the iterative PCA algorithm and a regularized iterative PCA algorithm where the properties are given in the associated theoretical papers (Josse *et al.* 2009; Josse and Husson 2012).

The main difference between the two R packages **missMDA** and **pcaMethods** is that the primary aim of **missMDA** is to estimate PCA parameters and obtain the associated graphical representations in spite of missing values, whereas **pcaMethods** focuses more on imputation aspects. Indeed, as we will explain in Section 3, we implemented in package **missMDA** a method to perform standardized PCA with missing values in which scaling is considered as part of the analysis and not as a pre-processing step. Another point of difference is that the **pcaMethods** package and MATLAB toolboxes only provide point estimates of parameters from incomplete data, whereas an idea of variability is given in **missMDA**. In particular, a multiple imputation method is included with the possibility to draw confidence areas on graphical outputs in order to know how much credibility can be given to outputs obtained from incomplete data.

Next, let us consider MCA for categorical variables. From a technical point of view, MCA consists of coding categorical variables with an indicator matrix of dummy variables and then performing a SVD on this matrix, weighted with weights that depend on the margins. MCA is also known as homogeneity analysis (Gifi 1990). In this framework, a popular approach to deal with missing value is the "missing passive" approach (Meulman 1982), based on the following assumption: An individual who has a missing entry for a variable is considered to have not chosen any category for that variable. Consequently, in the indicator matrix, the entries in the row and columns corresponding to this individual and variable are marked 0.

This method is implemented in the function `homals` in the R package **homals** (De Leeuw and Mair 2009a) as well as in the function `CATPCA` in the software package SPSS (Meulman,

Heiser, and SPSS 2003). Greenacre and Pardo (2006) showed that it is also closely related to their "subset MCA" approach, in which one adds a new category for the missing values and then treats it as an additional element. Their strategy is implemented in the R package **ca** (Nenadic and Greenacre 2007) using the function `mjca` with argument `subsetcol`.

Josse, Chavent, Liquet, and Husson (2012) provided a review of existing methods for handling missing values in MCA and suggested a new approach named regularized iterative MCA. They distinguish between different kinds of missing values and show which method is best-suited to each kind. In questionnaires for instance, it often happens that a missing value for a categorical variable corresponds to "a new category in itself", e.g., when the respondent cannot find an appropriate category to select among the available categories and consequently does not answer the question. The missing value is not really a "true" missing value (which could be imputed using the available categories for instance) but corresponds to a new category like "no opinion" or "do not know". In this case, the "missing passive" approach for estimating MCA parameters from incomplete data is appropriate. In other cases, Josse *et al.* (2012) suggest using their approach for estimating MCA parameters from incomplete data. This method is implemented in package **missMDA**. Note that unlike other approaches, this method allows to impute an incomplete data set when variables are categorical; see Section 7 for further details.

Regarding the FAMD and MFA methods, no solution was previously available to perform them with missing values. Package **missMDA** therefore provides the first implementation of the regularized iterative FAMD algorithm (Audigier, Husson, and Josse 2016a) and the regularized iterative MFA algorithm (Husson and Josse 2013). However, we note that related algorithms have been suggested for other multi-table and multi-way methods, including PARAFAC (Tomasi and Bron 2005; Kroonenberg 2008; Acar, Dunlavy, Kolda, and Mrup 2011; Smilde, Bro, and Geladi 2004). Some are available as MATLAB toolboxes (Andersson and Bro 2000) or as standalone software products (Kroonenberg 2011). More information can be found at `http://www.leidenuniv.nl/fsw/three-mode`. Note that in these implementations, however, no attempt is made to use regularized versions of the algorithms.

Note that in the current version of package **missMDA**, we do not address the issue of performing correspondence analysis (CA) with missing values. CA (Greenacre 1984, 2007) is dedicated to analyze multivariate count data. Nora-Chouteau (1974) suggested an algorithm to estimate CA parameters with missing values which is implemented in the package **anacor** (De Leeuw and Mair 2009b) and which can be seen as underlying the algorithms dedicated to perform principal component methods with missing values.

In summary, the **missMDA** package provides a complete framework for performing principal component methods for different data types in the presence of missing values. In addition, it provides solutions for selecting the number of underlying dimensions from incomplete data sets.

In this section, we did not detail software packages that are available for exploring and visualizing data with missing values outside the principal component framework such as the **VIM** package (Templ, Alfons, Kowarik, and Prantner 2015) as well as software packages that are available for imputing data; these implementations will be discussed in Section 7. Such packages aim at completing a data set and do not to perform principal component methods with missing values. It is still possible to perform a principal component method on a data set that has been completed by such imputation methods, e.g., using random forests (Stekhoven

and Bühlmann 2012). However, the properties of the results obtained in terms of quality of parameter estimation are unknown and, in addition, it is not straightforward to combine the results of different principal component methods for multiple imputation methods. Thus, we do not advise this strategy.

3. Principal component analysis

3.1. Point estimates

PCA in the complete case is often presented from a geometrical point of view as providing a subspace that maximizes the variance of the projected points, and therefore represents the diversity of the individuals. Equivalently, it can be presented as providing a subspace that minimizes the Euclidean distance between individuals and their projection onto the subspace. It boils down to finding a matrix of low rank S that gives the best approximation of the matrix $\mathbf{X}_{n\times p}$ with n individuals and p variables in the least squares sense:

$$\|\mathbf{X}_{n\times p} - \hat{\mathbf{X}}_{n\times p}\|^2. \tag{1}$$

The PCA solution is given by the first S terms of the singular value decomposition of the matrix $\mathbf{X}$: $\hat{\mathbf{X}} = \mathbf{U}_{n\times S}\mathbf{\Lambda}^{\frac{1}{2}}_{S\times S}\mathbf{V}^{\top}_{p\times S}$, with $\mathbf{U}$ and $\mathbf{V}$ being the left and right singular vectors and $\mathbf{\Lambda}$ the diagonal matrix containing the eigenvalues. The matrix $\mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}$ is also known as the scores matrix, principal components matrix, or matrix of the coordinates of the individuals on the axes, and the matrix $\mathbf{V}$ as the loadings matrix, principal axes matrix or coefficients matrix.

A common approach to deal with missing values in PCA involves ignoring the missing values by minimizing the least squares criterion (1) over all non-missing entries. This can be achieved by the introduction of a weighted matrix $\mathbf{W}$ in the criterion, with $w_{ij} = 0$ if x_{ij} is missing and $w_{ij} = 1$ otherwise:

$$\|\mathbf{W}_{n\times p} * (\mathbf{X}_{n\times p} - \mathbf{U}_{n\times S}\mathbf{\Lambda}^{\frac{1}{2}}_{S\times S}\mathbf{V}^{\top}_{p\times S})\|^2, \tag{2}$$

where $*$ is the Hadamard product. The aim is to estimate the PCA parameters despite the missing entries which have been skipped. In contrast to the complete case, there is no explicit solution to minimize criterion (2) and it is necessary to resort to iterative algorithms. Many algorithms were proposed and re-discovered in the literature under different names and in different fields (see Josse and Husson 2012 for references), including the iterative PCA algorithm suggested by Kiers (1997) and detailed in Josse and Husson (2012).

To illustrate this algorithm, let us consider a small data set with five individuals and two variables and a missing value for individual 4 and variable 2 identified by a small blue segment in Figure 1. The first step of the algorithm consists in imputing the missing entry with an initial value such as the mean of the variable over the observed values (red point in Figure 1, top middle panel), then PCA is performed on the imputed data set. The PCA red line is the best approximation of the 2-dimensional data set in one dimension in the least squares sense. Then, the value fitted by PCA is used to predict a new value for the missing one. The observed values are the same but the missing entry is replaced by the fitted one (green

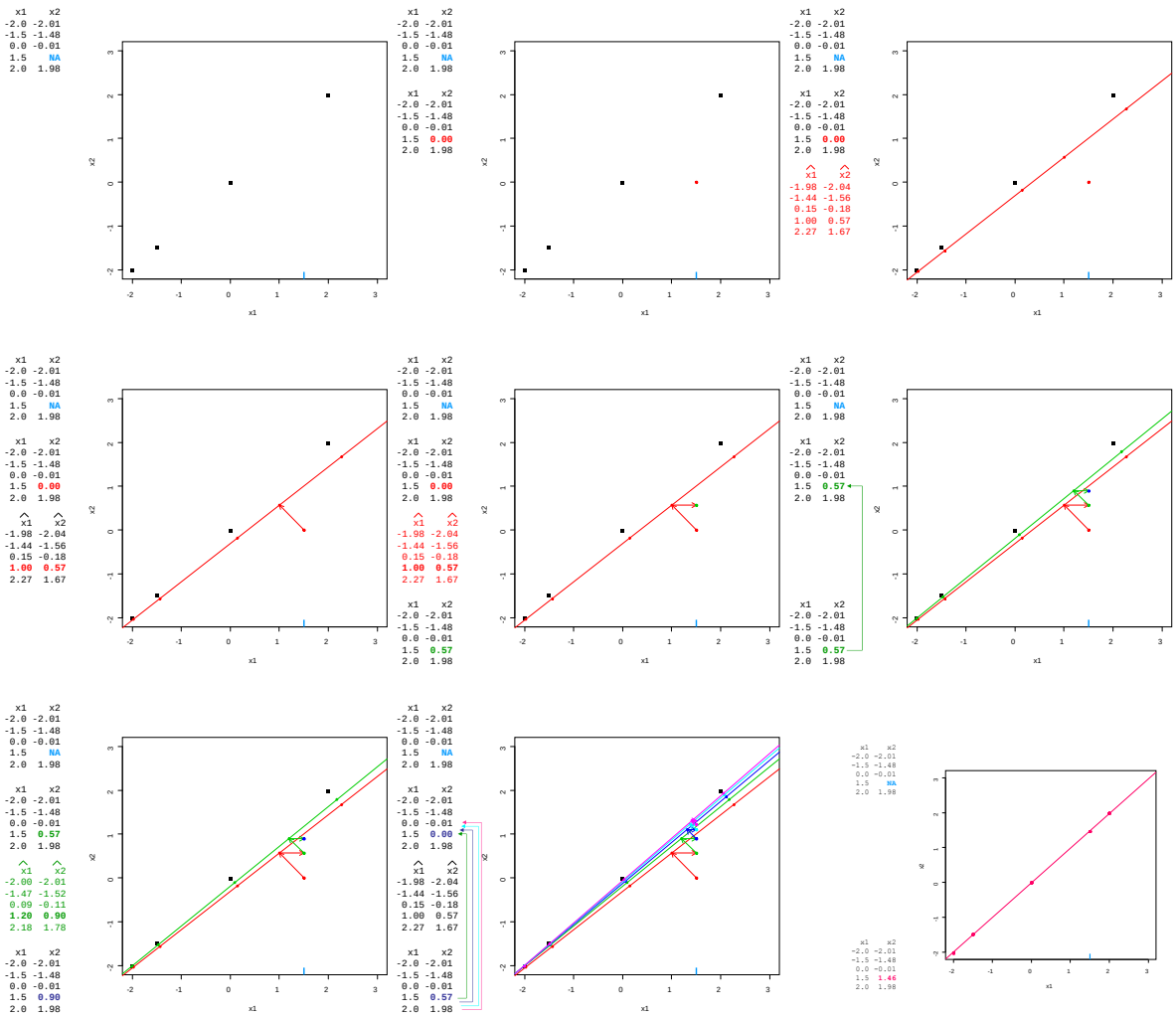

Figure 1: Illustration of the iterative PCA algorithm.

point in Figure 1). On the new completed data set, the same procedure is applied. These two steps of parameter estimation with PCA (estimation of the PCA line) and imputation of the missing entry using the values predicted by PCA are repeated until convergence. At the end of the algorithm, we obtained both an estimation of the PCA parameters from an incomplete data set as well as an imputed data set. This explains why this algorithm is so popular, since it can be used either to perform PCA despite missing values or to complete data sets. These kinds of algorithms have become popular for matrix completion problems, especially in the machine learning community for collaborative filtering problems, e.g., the Netflix prize (Netflix 2009).

We note that the iterative PCA algorithm is also known as the EM-PCA algorithm (expectation-maximization PCA; Dempster, Laird, and Rubin 1977). This makes sense because it corresponds to an EM algorithm for a PCA fixed-effects model (Caussinus 1986), where data are generated as a fixed structure having a low rank representation in S dimensions corrupted by

noise:

$$x_{ij} = \sum_{s=1}^{S} \sqrt{\lambda_s} u_{is} v_{js} + \varepsilon_{ij}, \text{ with } \varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2). \tag{3}$$

Such algorithms converge to a possible local maximum.

The iterative PCA algorithm provides a good estimation of the PCA parameters when there are very strong correlations between variables and the number of missing values is very small. However, it very rapidly suffers from overfitting problems when data are noisy and/or there are many missing values. This means that the observed values are well-fitted but prediction quality is poor. Such problems are illustrated in Ilin and Raiko (2010) and Josse and Husson (2012). To tackle this issue, a common strategy is to use regularized methods. Josse *et al.* (2009) suggested the following regularized iterative PCA algorithm:

1. Initialization $\ell = 0$: Substitute missing values with initial values such as the mean of the variables with non-missing entries, the imputed matrix is denoted $\mathbf{X}^0$. Calculate $\mathbf{M}^0$, the matrix of the vector containing the mean of the variables of $\mathbf{X}^0$, repeated in each row of $\mathbf{M}^0$.

2. Step $\ell \geq 1$:

   (a) Perform the PCA, i.e., the SVD of $\left(\mathbf{X}^{\ell-1} - \mathbf{M}^{\ell-1}\right)$ to estimate parameters $\mathbf{U}^\ell$, $\mathbf{V}^\ell$ and $\left(\mathbf{\Lambda}^\ell\right)^{1/2}$.

   (b) Keep the first S dimensions and build the fitted matrix with

   $$\hat{x}_{ij}^\ell = \sum_{s=1}^{S} \left(\sqrt{\lambda_s^\ell} - \frac{(\hat{\sigma}^2)^\ell}{\sqrt{\lambda_s^\ell}} \right) u_{is}^\ell v_{js}^\ell,$$

   with the noise variance estimated as $(\hat{\sigma}^2)^\ell = \frac{\| \mathbf{X}^{\ell-1} - \mathbf{U}^\ell \left(\mathbf{\Lambda}^\ell\right)^{1/2} (\mathbf{V}^\ell)^\top \|^2}{np - nS - pS + S^2}$, and define the new imputed data set as $\mathbf{X}^\ell = \mathbf{W} * \mathbf{X} + (\mathbf{1} - \mathbf{W}) * \hat{\mathbf{X}}^\ell$, where $\mathbf{1}$ is a matrix of size $n \times p$ with only ones. The observed values are the same but the missing ones are replaced by the (regularized) fitted values.

   (c) From the new completed matrix, $\mathbf{M}^\ell$ is updated.

3. Steps (2.a), (2.b) and (2.c) are repeated until the change in the imputed matrix falls below a predefined threshold $\sum_{ij} (\hat{x}_{ij}^{\ell-1} - \hat{x}_{ij}^\ell)^2 \leq \varepsilon$, with ε equal to 10^{-6} for example.

Additional justification for the specific regularization shown here is given in Verbanck, Josse, and Husson (2015). Much can be said about the regularized iterative PCA algorithm. First, note that the mean matrix $\mathbf{M}$ is updated during the algorithm. Indeed, after each imputation step, the means of the variables change. Consequently, it is necessary to re-center the data after each imputation step. In the same vein, if one wishes to perform a standardized PCA (to give the same weight to each variable in the analysis) with missing values, a re-scaling step must be incorporated after each imputation step. In the complete case, scaling is often carried out prior to analysis and thus often regarded as a pre-processing step. In the incomplete case, it is necessary to consider the scaling process as a part of the analysis. As far as we know,

many algorithms performing PCA with missing values do not include these re-centering or re-scaling steps and thus variables do not have the same weights in the analysis.

Next, the algorithm requires as tuning parameter the number of dimensions S, chosen *a priori*. Many strategies are available in the literature to select the number of dimensions from a complete data set (Jolliffe 2002). Cross-validation (Bro, Kjeldahl, Smilde, and Kiers 2008; Josse and Husson 2011b) is one such method which shows good performance and can be easily extended to the incomplete case. We implemented three cross-validation techniques in **missMDA**: leave-one-out, k-fold and generalized cross-validation.

Leave-one-out cross-validation consists of removing each observed value x_{ij} of the data matrix $\mathbf{X}$ one at a time. Then, for a fixed number of dimensions S, we predict its value using the PCA model obtained from the data set that excludes this cell (using the iterative PCA algorithm on the incomplete data set). The predicted value is denoted $\left(\hat{x}_{ij}^{-ij}\right)^S$. Lastly, the prediction error is computed and the operation repeated for all observed cells in $\mathbf{X}$ and for a number of dimensions varying from 0 to $\min(n-2, p-1)$. The number S that minimizes the mean square error of prediction (MSEP) is kept:

$$\text{MSEP}(S) = \frac{1}{np}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(x_{ij} - \left(\hat{x}_{ij}^{-ij}\right)^S\right)^2.$$

This method is computationally costly, especially when the number of cells is large, since it requires S times the number of observed cells to perform the iterative PCA algorithm. To reduce the computational cost it is possible to use a k-fold approach which consists in removing more than one value in the data set, for instance 10% of the cells and predict them simultaneously. Lastly, inspired by generalized cross-validation (GCV; Craven and Wahba 1979) in the framework of regression, Josse and Husson (2012) defined a GCV value in the framework of PCA to approximate the MSEP. It is defined as follows:

$$\text{GCV}(S) = \frac{np\sum_{i=1}^{n}\sum_{j=1}^{p}\left(x_{ij} - (\hat{x}_{ij})^S\right)^2}{np - p - nS - pS + S^2 + S}.$$

The GCV value can be interpreted as a classical model selection criterion where the residuals sum of squares is penalized by the number of degrees of freedom. This strategy is faster since it only requires running the iterative PCA algorithm once to estimate the $\hat{x}_{ij}$ and no additional cells are removed.

3.2. Confidence areas

The (regularized) iterative PCA algorithm provides estimates of the PCA parameters despite the absence of some data, as well as a single imputed data set. Josse and Husson (2011a) proposed a multiple imputation method called MIPCA which generates several imputed data sets. The observed values from one imputed data set to another are the same but the imputed values for missing data differ. Variability across the various imputations reflects variability in the prediction of missing values. The multiple imputation is "proper" in the sense of Little and Rubin (1987, 2002), meaning that the variance of predictions is composed of two parts: variability in the estimated values of the PCA parameters plus variability due to noise. Josse and Husson (2011a) used a residuals bootstrap procedure to obtain the variance of parameters. Indeed, it is in agreement with model (3) where the randomness part can be

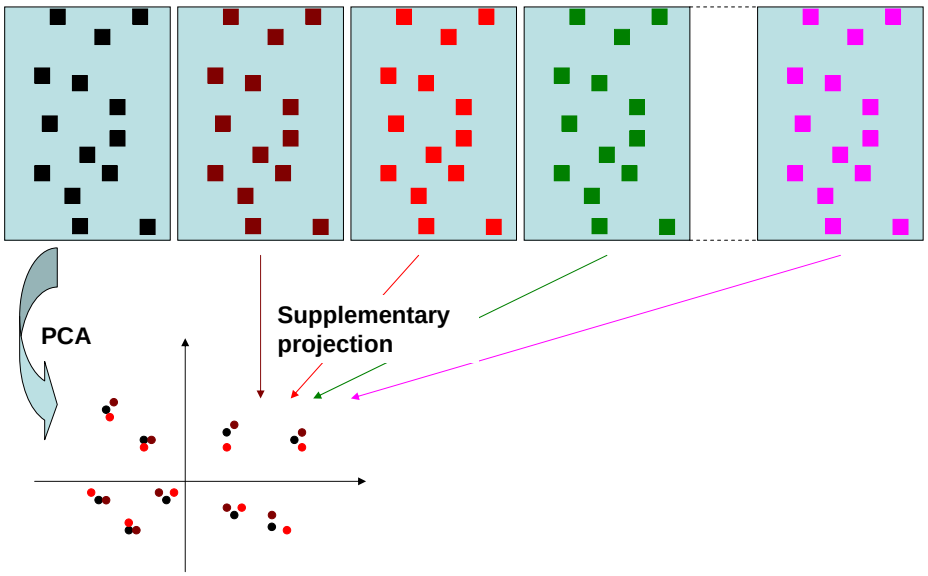

Figure 2: Supplementary projection of the multiple imputed data sets onto the reference configuration (in black).

seen as coming only from the measurement errors. Thus, it is possible to create B bootstrap replicates of the data $\mathbf{X}^b$, $b = 1, \ldots, B$, by adding to the estimator $\hat{\mathbf{X}}$, B new matrices of residuals obtained by bootstrapping the current residuals matrix $\hat{\varepsilon} = \mathbf{X} - \hat{\mathbf{X}}$. Then, the (regularized) iterative PCA algorithm is applied to each new matrix $\mathbf{X}^b$ which gives B new estimators $\hat{\mathbf{X}}^1, \ldots, \hat{\mathbf{X}}^B$ representing the variability of the PCA parameters. Finally, the B imputed values are obtained by drawing from the predictive distribution meaning that a Gaussian noise with variance equal to the residuals variance is added to each matrix $\mathbf{X}^b$. This strategy is implemented in package **missMDA**. An alternative is to use a Bayesian approach, as suggested in Audigier, Husson, and Josse (2016b).

The impact of different predicted values on the PCA results can be visualized using a strategy illustrated in Figure 2. In Figure 2, the blue color is used for the observed values and each square corresponds to an imputed value. The first data table with black squares corresponds to the one obtained after performing the (regularized) iterative PCA algorithm. Then, the B other imputed data sets are obtained with the multiple imputation method MIPCA. The observed values are the same from one table to another but the imputed values are different and the variability across the imputation represents the variance of prediction of the missing entries. Individuals in the B imputed data sets generated by the MIPCA algorithm are projected as supplementary individuals onto the reference configuration (the one obtained with the regularized iterative PCA algorithm). It means that we compute the inner-product between the individuals and the axes (loadings) of the reference configuration. An individual without any missing values is projected onto its corresponding point whereas an individual with missing entries is projected around its initial position. In this way, we can visualize the position of individuals with different missing value predictions. Next, confidence ellipses are drawn assuming a Gaussian distribution.

Then, the impact of the various imputed values on the PCA parameters is obtained with a PCA performed on each imputed data set as illustrated in Figure 3. This leads to scores and

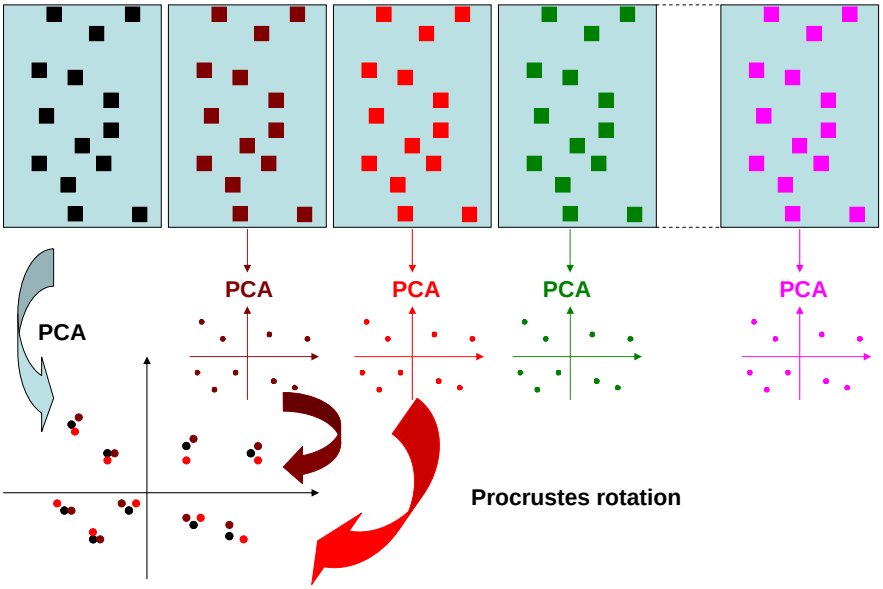

Figure 3: Procrustes rotations of the PCA configurations obtained from the multiple imputed data sets onto the reference configuration (in black).

loadings that are different from one table to the next. Individuals (using their scores) are represented in the same subspace with Procrustes rotations (Gower and Dijksterhuis 2004). The underlying idea is to perform geometric transformations on each PCA configuration (translation, rotation and dilatation) in order to obtain configurations comparable to the one obtained by the (regularized) iterative PCA algorithm. Then, confidence ellipses around the positions of individuals are drawn. These ellipses represent between-imputation variability.

Note that the ellipses should be interpreted as areas of variability rather than confidence ellipses. Indeed, we cannot ensure that the probability that the "true" values lie inside the ellipses equals 95%. The first ellipses represent only the position of individuals with different missing entries whereas the second ones represent the variability of the parameters due to missing values (the between-imputation variability) and not the total variability which should be composed of the between and within variability. More research has to be done in this area. Nevertheless, these ellipses are still very valuable to assess the impact of the missing values strategies on the analysis.

3.3. PCA on a genotype-environment data set

We illustrate the method on a genotype-environment data set coming from the field of crop science. After installing the **missMDA** package, the package as well as the incomplete data set `geno` are loaded.

```
R> set.seed(1234)
R> library("missMDA")
R> data("geno", package = "missMDA")
R> summary(geno)
R> head(round(geno, 2))
```

```
    ACOR AORE  ASAL  CALB  CBAD  CCOR  CLER  CSE1  CSE2  CTO1
C_1 0.64 0.43  0.37    NA -0.64 -0.14    NA    NA  0.16 -0.47
C_2 2.17 0.86    NA  0.01    NA  0.04 -1.32 -0.56 -0.77 -0.53
C_3 0.85 0.96 -0.26 -0.13 -0.39  0.27 -0.20 -0.49 -0.33 -0.27
C_4 1.41 1.50 -0.03 -0.10 -0.32 -0.16 -0.73 -0.54 -1.09  0.07
C_5 0.87 0.86  0.54 -0.10 -0.03  0.05 -0.88 -0.34 -0.66 -0.32
C_6 1.71 0.34  0.53  0.17    NA -0.34 -0.83 -0.51 -1.03  0.06
```

The data set `geno` has 16 rows corresponding to genotypes (triticale lines) and 10 columns corresponding to different environments where the genotypes were sown. Each cell of the data matrix corresponds to the grain yield (kilograms per hectare) for one genotype in a given environment. The first six genotypes correspond to the so-called "complete" type, while the next eight are of the "substituted" type; two reference genotypes are also included. More details about the data can be found in Royo, Rodriguez, and Romagosa (1993) and Josse, Eeuwijk, Piepho, and Denis (2014). Such data sets are often incomplete. Indeed, it frequently happens that all varieties are not assessed in all environments.

To perform a PCA on an incomplete data set, we proceed in three steps, and only require the three following lines of code:

```
R> ncomp <- estim_ncpPCA(geno)
R> res.imp <- imputePCA(geno, ncp = ncomp$ncp)
R> res.pca <- PCA(res.imp$completeObs)
```

We now detail these steps and the associated functions. First, we select the number of dimensions that will be used in the algorithm using the function `estim_ncpPCA`.

```
R> ncomp <- estim_ncpPCA(geno, ncp.min = 0, ncp.max = 6)
R> ncomp$ncp

[1] 2
```

This returns the MSEP for a number of dimensions varying from `ncp.min` to `ncp.max` in the `criterion` object, as well as the number of dimensions minimizing the MSEP in the object `ncp` (here, two dimensions). By default, the function `estim_ncpPCA` uses the GCV method. It is possible to use another cross-validation strategy by specifying the argument `method.cv` as follows:

```
R> ncomp$ncp <- estim_ncpPCA(geno, ncp.min = 0, ncp.max = 6,
+    method.cv = "Kfold", nbsim = 100, pNA = 0.05)
```

With the `Kfold` method two additional arguments are useful: `pNA` indicates the percentage of missing values inserted and predicted with PCA using `ncp.min` to `ncp.max` dimensions, and `nbsim` the number of times this process is repeated.

The second step consists of performing the (regularized) iterative PCA algorithm with the number of dimensions selected in the previous step, using the function `imputePCA`:

```
R> res.imp <- imputePCA(geno, ncp = 2, scale = TRUE,
+    method = "Regularized", row.w = NULL, coeff.ridge = 1,
+    threshold = 1e-06, seed = NULL, nb.init = 1, maxiter = 1000)
```

By default, the function `imputePCA` uses the regularized iterative PCA algorithm. However, it is possible to use the argument `method = "EM"` to perform the iterative PCA algorithm. The argument `scale = TRUE` is used by default to perform a standardized PCA (with the standard deviations updated during the algorithm as explained in Section 3); the alternative is `scale = FALSE`. Since the algorithm may converge to a local minimum, it is possible to run it `nb.init` times with different random initializations (different imputations) and keep the best solution in the sense of the least squares criterion. This can be performed by specifying a value for the argument `seed`. The algorithm stops either when the relative difference between two successive imputations is less than `threshold`, or when the number of iterations exceeds the fixed number `maxiter`. The other options are rarely used; it is possible to give weights to the individuals (`row.w`) and to apply a milder or a stronger penalization (`coeff.ridge`).

As described in Section 3, at convergence the algorithm provides both an estimation of the scores and loadings as well as a completed data set. The `imputePCA` function outputs the imputed data set. Indeed, as will be discussed in Section 7, this allows the possibility of using the **missMDA** package to complete data before performing any statistical analyses (on the imputed data set). The completed data set is in the object `completeObs`:

```
R> head(round(res.imp$completeObs, 2))
```

```
    ACOR AORE  ASAL  CALB  CBAD  CCOR  CLER  CSE1  CSE2  CTO1
C_1 0.64 0.43  0.37 -0.07 -0.64 -0.14 -0.51 -0.29  0.16 -0.47
C_2 2.17 0.86  0.50  0.01 -0.56  0.04 -1.32 -0.56 -0.77 -0.53
C_3 0.85 0.96 -0.26 -0.13 -0.39  0.27 -0.20 -0.49 -0.33 -0.27
C_4 1.41 1.50 -0.03 -0.10 -0.32 -0.16 -0.73 -0.54 -1.09  0.07
C_5 0.87 0.86  0.54 -0.10 -0.03  0.05 -0.88 -0.34 -0.66 -0.32
C_6 1.71 0.34  0.53  0.17 -0.36 -0.34 -0.83 -0.51 -1.03  0.06
```

The `imputePCA` function also outputs the fitted matrix $\hat{\mathbf{X}}$ in the object `recon`.

The last step of the analysis is to perform PCA on the imputed data set. To this end, we propose to use the `PCA` function of the **FactoMineR** package (Husson, Josse, Lê, and Mazet 2016; Lê, Josse, and Husson 2008), but any other function performing PCA can be used such as `prcomp` or `dudi.pca` from the **ade4** package (Dray 2007):

```
R> res.pca <- PCA(res.imp$completeObs)
```

This provides the classical graphical outputs represented in Figure 4 for individuals and variables. Even if these outputs are obtained first by imputing the incomplete data set and then performing PCA on the completed data set, they correspond to the solution minimizing the criterion (2) which skips missing values. Therefore, they correspond to the estimates of the axes and components of a PCA performed on the incomplete data table. In this example, the first dimension of variability separates the complete and substituted genotypes. The substituted genotypes yield more in the environments that have high coordinates on the first dimension (CTO1, CBAD, ..., CES1) and less in the environments ACOR and AORE. The two reference genotypes (S_M and S_F) are located around the center of gravity. More details about how to interpret the results of a PCA can be found in many books such as in Husson, Lê, and Pagès (2010).

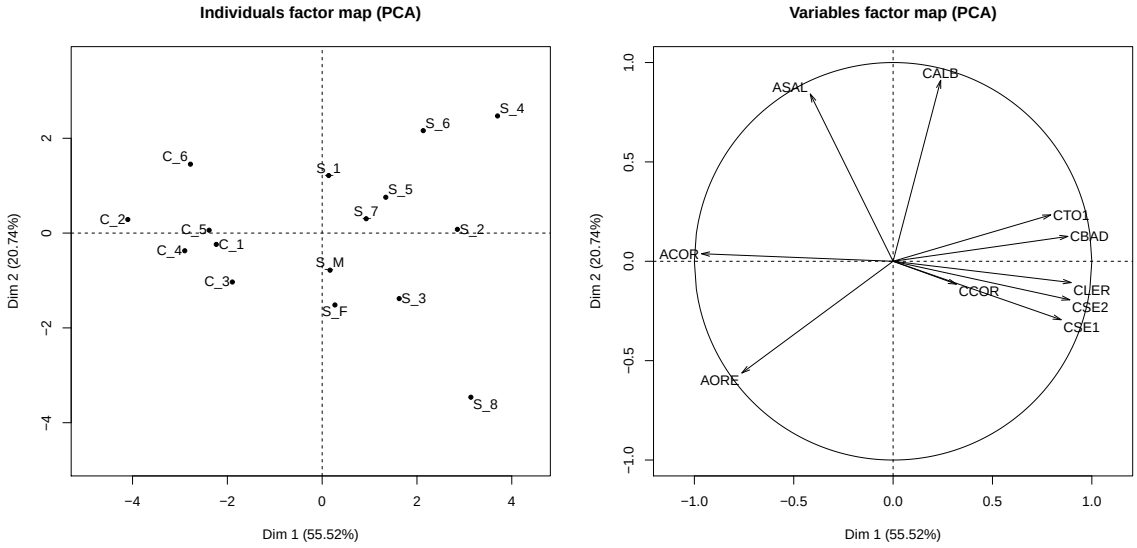

Figure 4: Outputs of the `PCA` function: graph of individuals (left) and correlation circle (right).

To perform multiple imputation, we use the `MIPCA` function which generates multiple imputed data sets. Then, the `plot` method for '`MIPCA`' objects is applied to visualize the impact of the different imputed values on the PCA results.

```
R> resMIPCA <- MIPCA(geno, ncp = 2, nboot = 200)
R> plot(resMIPCA)
```

The function `MIPCA` gives as output the data set imputed by the (regularized) iterative PCA algorithm (in `res.imputePCA`) and the other data sets generated by the MIPCA algorithm (in `res.MI`). The number of data sets generated by this algorithm is controlled by the `nboot` argument, equal to 100 by default. The other arguments of this function are the same as those for the `imputePCA` function. The `plot` method for '`MIPCA`' objects draws the graphs represented in Figure 5. Those on the left represent the projection of the individuals (top) and variables (bottom) of each imputed data set as supplementary elements onto the reference configuration obtained with the (regularized) iterative PCA algorithm, as described in Section 3.2. For the individuals, a confidence area is constructed for each, and if one has no missing entries, its confidence area is restricted to a point. The graphs on the right are obtained after applying PCA to each imputed data set. The top one corresponds to the representation of the individuals obtained after performing Procrustes rotations as described in Section 3.2. Even if an individual has no missing entries, its confidence area is not restricted to a point since the PCA components are not strictly identical from one PCA to another due to the different imputed values of the other individuals. The bottom-right graph corresponds to the projection of the two first principal components of each imputed data set onto the two first principal components of the reference configuration obtained with the (regularized) iterative PCA algorithm. In this example, we can observe that genotype S_8 has no missing values, genotype S_6 has one missing entry for a variable with a high value on the second dimension such as CALB. In this example, all the plots show that the variability across different

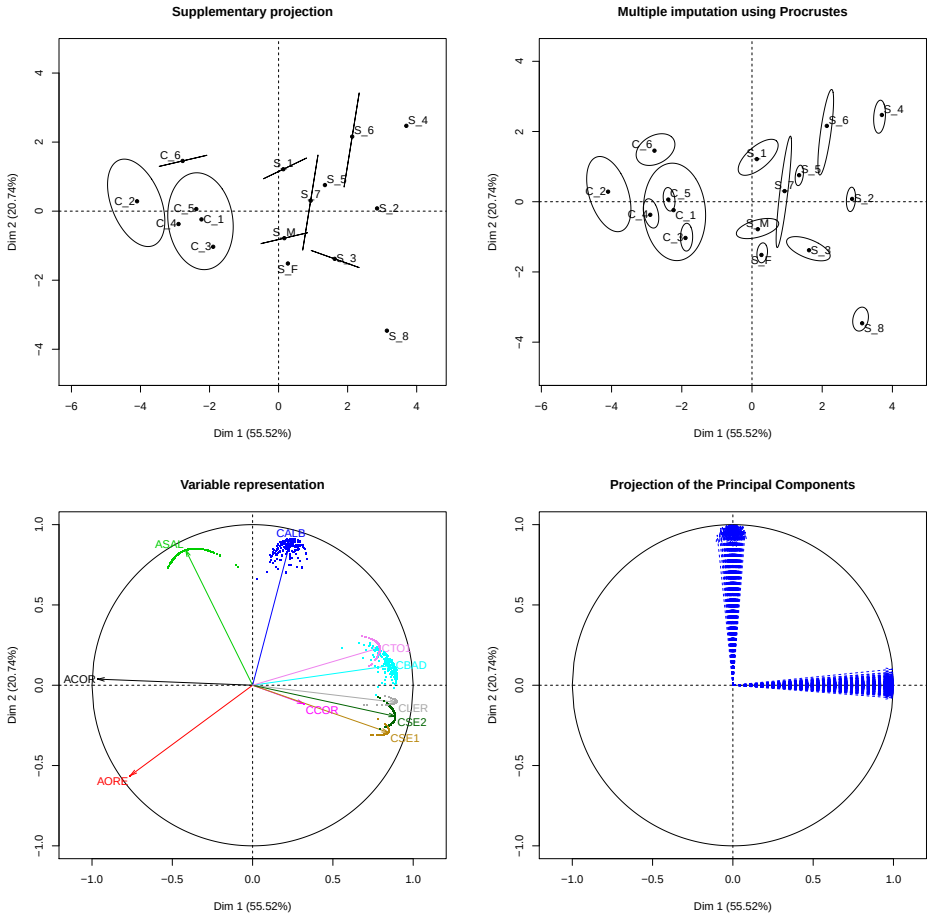

Figure 5: Visualization of multiple imputed data sets on PCA outputs. The graphs on the left represent the projection as supplementary elements of the individuals (top) and variables (bottom) of the imputed data sets. The graphs on the right are obtained after applying PCA to each imputed data set. At the top, Procrustes rotations are performed to obtain the individuals graph and on the bottom the first two principal components of each imputed data sets are projected onto the first two principal components of the reference configuration.

imputations is small and a user can interpret the PCA results with confidence.

The arguments of the `plot` method for '`MIPCA`' objects are as follows:

```
R> plot(resMIPCA , choice = "all", axes = c(1, 2), new.plot = TRUE,
+     main = NULL, level.conf = 0.95)
```

and can be modified using the following options:

- `choice`: the graphs to plot. By default `all` the graphs are plotted; `ind.supp` to draw a confidence ellipse around each individual (Figure 5, top left) by projecting the imputed data sets as supplementary elements; `var` for the projection of the variables of the imputed data sets as supplementary variables (Figure 5, bottom left); `ind.proc` to draw a confidence ellipse around the position of each individual using Procrustes rotations

(Figure 5, top right); `dim` for the projection of the first two principal components of each imputed data set (Figure 5, bottom right).

- `axes`: a length 2 vector specifying the dimensions to plot.
- `new.plot`: a Boolean. If `TRUE`, a new graphical device is created.
- `main`: a string corresponding to the graph's title, by default `NULL`.
- `level.conf`: a confidence level used to construct the ellipses; by default 0.95.
- `...`: other standard options of the `plot` function.

4. Multiple correspondence analysis

In the **missMDA** package, it is also possible to handle missing values in MCA. MCA can be seen as the equivalent of PCA for categorical data. It is often used to analyze survey data and aims to visualize similarities between respondents and relationships between categorical variables, and more precisely associations between categories. MCA can be seen as a PCA on a certain data matrix. More precisely, it means performing a PCA on the following triplet:

$$\left(I\mathbf{X}\mathbf{D}_{\Sigma}^{-1}, \frac{1}{IJ}\mathbf{D}_{\Sigma}, \frac{1}{I}\mathbb{I}_I\right),$$

with $\mathbf{X}$ the indicator matrix of dummy variables (represented in Figure 6) for the I individuals and J categorical variables; the row margins of $\mathbf{X}$ are equal to the number of variables J, the column margins of $\mathbf{X}$ to I_k, the number of individuals taking the category k. $\mathbf{D}_{\Sigma} = \text{diag}\left((I_k)_{k=1,\ldots,K}\right)$ is the diagonal matrix of the column margins of $\mathbf{X}$. The matrix $\mathbf{D} = \frac{1}{I}\mathbb{I}_I$ corresponds to the row masses and $\mathbf{M} = \frac{1}{IJ}\mathbf{D}_{\Sigma}$ is the metric (used to compute distances between rows).

Because MCA is a particular example of PCA, the method to deal with missing values in PCA can be extended to deal with missing values in MCA, taking into account details specific to

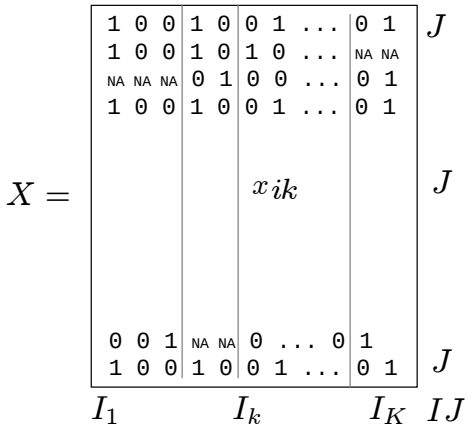

Figure 6: Coding of categorical variables with an indicator matrix of dummy variables $\mathbf{X}$. The first categorical variable has three categories and a missing value for individual 3.

MCA, namely the precise weighting. Josse *et al.* (2012) proposed the iterative MCA algorithm, which consists of an initialization step in which missing values in the indicator matrix are imputed by initial values such as the proportion of the category. Values can be non-integer as long as the sum for each individual and each variable is equal to 1. This initialization for categorical variables is equivalent to mean imputation for continuous variables. Then, MCA is performed on the imputed indicator matrix to obtain an estimation of parameters, and missing values are imputed using the fitted values. After the imputation step, the margins $\mathbf{D}_\Sigma$ change and thus it is necessary to incorporate a step for updating the margins like for means and standard deviations in PCA (Section 3). A regularized version of the algorithm has also been suggested, since overfitting problems are exacerbated in MCA due to high-dimensionality of the space induced by the categorical variables.

We illustrate the method on the `vnf` data set which concerns a user satisfaction survey of pleasure craft operators. These were asked numerous questions with categorical answers, each with two or three categories. 1232 individuals answered 14 questions involving a total of 35 categories. The data set has 9% of values missing, involving 42% of respondents. As in PCA (Section 3.3), we proceed in three steps to perform MCA with missing values:

```
R> data("vnf", package = "missMDA")
R> ncomp <- estim_ncpMCA(vnf, method.cv = "Kfold")
R> tab.disj.impute <- imputeMCA(vnf, ncp = 4)$tab.disj
R> res.mca <- MCA(vnf, tab.disj = tab.disj.impute)
```

The first step consists of estimating the number of dimensions, using the `estim_ncpMCA` function. Then, the regularized iterative MCA algorithm is performed using the `imputeMCA` function. The arguments of these functions are the same as those used in the analogous functions in `estim_ncpPCA` and `imputePCA`.

If the `seed` argument of the `imputeMCA` function is not `NULL`, a random initialization is performed. More precisely, random numbers are entered in the indicator matrix that meet the constraint that the sum of the entries corresponding to each individual and each variable is 1.

The first output of the `imputeMCA` function is `tab.disj` and corresponds to the completed indicator matrix resulting from the last step of the (regularized) iterative MCA algorithm. Using this completed indicator matrix as input to the `MCA` function of the **FactoMineR** package leads to an estimation of the MCA parameters obtained from the incomplete data by skipping the missing values. Indeed, a criterion of the same form as in Equation 2 is minimized with zero weights for missing entries. More details about the analysis can be found in Josse *et al.* (2012). Thus, to perform MCA using an incomplete data set, it uses the **FactoMineR** package (i.e., package **missMDA** imports **FactoMineR** functions) and the `MCA` function needs to use the argument `tab.disj`. This then leads to the classical MCA outputs such as a representation of the categories (Figure 7). This shows that respondents who answered response 2 for question 8.2 often answered response 2 for question 8.1. We can also say that category 2 of question 8.3 is a rare choice since it lies far from the other categories. The rules of interpretation are the same as usual and can be found in books on MCA such as Greenacre and Blasius (2006).

Note that an imputed value in the indicator matrix can be seen as a degree of membership to the associated category. Consequently, each missing entry of the original data set can be imputed with the most plausible category (as we will see in Section 5). This completed data table is available in the output `completeObs` and thus, as we will discuss in Section 7, the **missMDA** package can be used to impute categorical data sets.

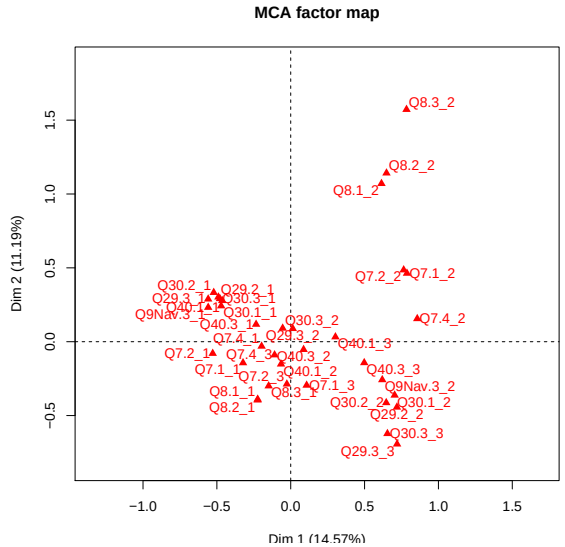

Figure 7: A representation of the categories of the incomplete `vnf` data set.

5. Factorial analysis for mixed data

When some variables are continuous and others categorical (known as mixed data), one way to explore the data with principal components methods is to transform the continuous variables into categorical variables and then perform MCA. Though this method is simple, it has drawbacks (Pagès 2015) since information is lost. Factorial analysis for mixed data (FAMD; Escofier 1979; Pagès 2015) is an alternative, also known as PCAMIX (Kiers 1991). The method consists of coding the data first as illustrated in Figure 8. Categorical variables are transformed into dummy variables and concatenated with the continuous variables. Then each continuous variable is standardized (centered and divided by its standard deviation) and each dummy variable is divided by the squared root of the proportion of individuals taking the associated category: $\sqrt{I/I_k}$ for category k. FAMD consists of performing a PCA on this weighted matrix. This specific weighting induces balance in the influence of both variable types. It is based exactly on the same rationale as the scaling in PCA which gives the same weight to each variable in the analysis; here the specific weights ensure that all continuous and categorical variables play the same role. In addition, the first principal component, denoted $\mathbf{F}_1$, maximizes the link between the continuous and categorical variables in the following sense:

$$\sum_{k=1}^{K_{cont}} r^2(\mathbf{F}_1, v_k) + \sum_{q=1}^{Q_{cat}} \eta^2(\mathbf{F}_1, v_q),$$

with v_q being the variable q, K_{cont} the number of continuous variables, Q_{cat} the number of categorical variables, r^2 the square of the correlation coefficient and η^2 the square of the correlation ratio.

FAMD is similar to PCA when there are only continuous variables and to MCA when there are only categorical variables. Consequently, the algorithm to perform FAMD with missing values (Audigier *et al.* 2016a) is close to the ones described in Sections 3 and 4 and can be summarized as follows:

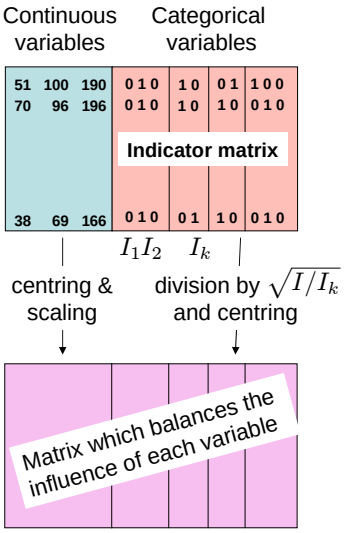

Figure 8: Coding the data in factorial analysis for mixed data.

1. Initialization: Imputation using for example the means (for continuous variables) and proportions (for the dummy variables). Compute the standard deviations and column margins and thus the weights for the concatenated matrix (of the continuous variables and indicator matrix of dummy variables).

2. Iterate until convergence:

   (a) Perform PCA on the completed weighted data matrix to estimate the parameters $\mathbf{U}, \mathbf{\Lambda}$ and $\mathbf{V}$.

   (b) Impute the missing values with the fitted values using S dimensions.

   (c) Update the mean, standard deviations (for the continuous variables) and column margins (for the categories).

We illustrate this method using the `imputeFAMD` function of the **missMDA** package and the `FAMD` function of the **FactoMineR** package on the incomplete mixed data set `snorena` represented in Figure 9 (left-hand side). The following lines of code perform FAMD on the incomplete data set:

```
R> data("snorena", package = "missMDA")
R> res.impute <- imputeFAMD(snorena, ncp = 3)
R> res.famd <- FAMD(snorena, tab.comp = res.impute)
```

The function `imputeFAMD` gives as output the object `tab.disj` which is the imputed matrix, i.e., the imputed continuous variables and imputed indicator matrix as illustrated in Figure 9 (bottom right-hand side). Note that in this imputed indicator matrix, imputed values are real numbers that satisfy the constraint that the sum of the entries corresponding to each individual and each variable is equal to 1. This property is inherited from the FAMD method without missing values and is due to the specific weighting (Pagès 2015). Consequently, they can be seen as the degree of membership in the corresponding category, and it is possible to

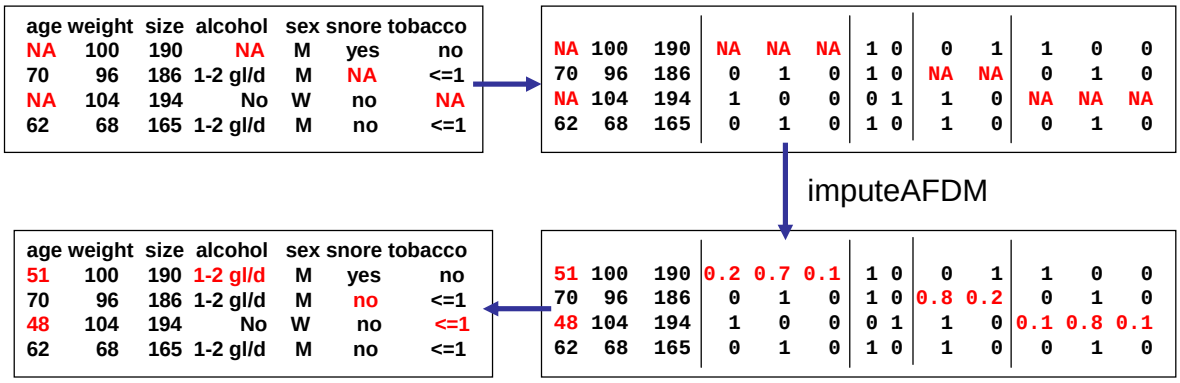

Figure 9: Illustration of the regularized iterative FAMD algorithm on the incomplete `snorena` data set. Top left: the incomplete data table; top right: coding the data with the categorical variables coded using the indicator matrix of dummy variables; bottom right: the imputed data matrix with the imputed continuous variables and the imputed indicator matrix of dummy variables; bottom left: the original data set completed; values imputed in the indicator matrix are used as degree of membership in the corresponding category, and imputation is performed using the most plausible categories.

impute the original categorical variables using the most plausible categories. The result of this operation is given in the object `completeObs`, as illustrated in Figure 9 (bottom left-had side). The value predicted for the individual 2 on variable snore is "no" since its corresponding imputed values are 0.8 and 0.2. As for MCA, using this completed matrix as input to the `FAMD` function of the **FactoMineR** package leads to an estimation of the FAMD parameters by skipping missing values. Note that for the function `FAMD`, the argument `tab.comp` takes as input the output of the function `imputeFAMD`.

6. Multiple factor analysis

Let us now extend the case of one data table to multi-tables. We consider data where the rows are described by several groups of variables. In many fields, it is more and more common to deal with heterogeneous data coming from different information sources. Let us consider an example from biology where 53 brain tumors of 4 different types defined by the standard World Health Organization classification (O, oligodendrogliomas; A, astrocytomas; OA, mixed oligo-astrocytomas and GBM, glioblastomas) are described by information at the transcriptome level with expression data (356 continuous variables for microarrays) and genome level (76 continuous variables for CGH data) as illustrated in Figure 10.

For this kind of data, we are interested in studying the similarities between rows from a multidimensional point of view, as well as the correlation between variables (same objectives as for PCA). In addition, the setting is more complex due to the data structure, and we are interested in highlighting similarities and differences between groups of variables, i.e., studying what is common to groups and what is specific. In other words, we would like to compare the information brought by each group and identify for instance if two tumors, similar from the point of view of the transcriptome, are also similar in terms of the genome.

Different multi-blocks methods are available in the literature (Kroonenberg 2008) to answer

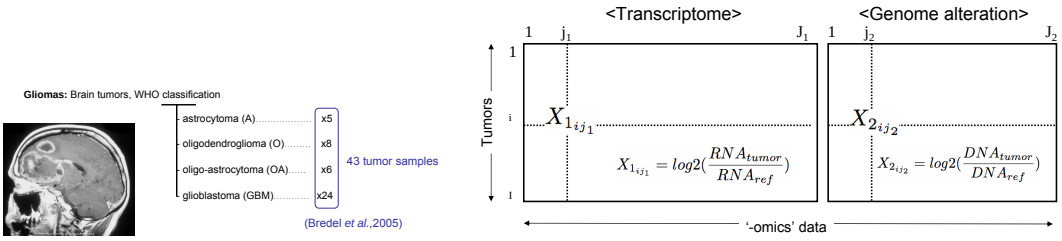

Figure 10: Four brain tumor types characterized by transcriptome and genome data.

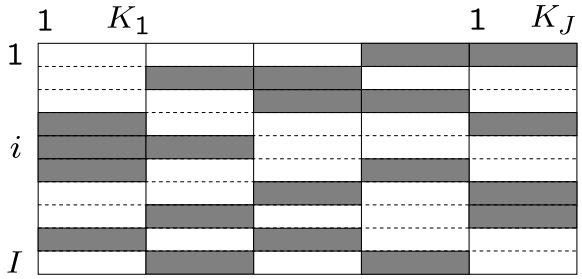

Figure 11: Pattern of missing values with missing rows in sub-tables.

such questions and visualize results; we focus here on multiple factor analysis (MFA; Pagès 2015). MFA handles tables where the groups of variables are continuous or categorical, or may be contingency tables. One of the aims of MFA is to balance the influence of the groups of variables in the analysis in such a way that no single group (with many correlated variables for instance) dominates the first dimension of variability. To do so, for each group of variables a principal component method is performed (PCA for a continuous group or MCA for a categorical one) and then each value in the group is divided by the square root of the first eigenvalue. Then, MFA consists in performing a global PCA on the data table that concatenates the weighted matrix of each group. More details about the method can be found in Pagès (2015). The rationale is also the same as in standardized PCA where variables are weighted to have the same influence in the analysis; here it can be seen as an extension to groups of variables where the first singular value plays the role of the standard deviation.

The risk of being confronted with missing values increases when there are many sources of information. In addition, with multi-table data, we can have specific patterns of missing values involving missing rows per sub-table, as illustrated in Figure 11 where $K_1, \ldots, K_J$ represent the number of variables in groups $1, \ldots, J$. In the previous example, ten samples were not available for the expression data. More details about the data and the results of MFA can be found in Tayrac, Lê, Aubry, Mosser, and Husson (2009), though note that they deleted samples with missing values to perform their analysis. Since the core of MFA is also a weighted PCA, it is possible to develop a method inspired by the one proposed for PCA to handle missing values. Husson and Josse (2013) developed a (regularized) iterative MFA algorithm to perform MFA using an incomplete data set. This algorithm also alternates steps of estimation of the parameters and imputation of the missing values as described for the other methods (PCA, MCA and FAMD) in Sections 3, 4, and 5, taking into account details specific to MFA, namely the precise weighting. They applied the method to examples coming from

the field of sensory analysis and also focused on missing values defined in an experimental design set-up. To apply the method to the tumor example, we run the following lines of code:

```
R> data("gene", package = "missMDA")
R> res.impute <- imputeMFA(gene[, -1], group = c(76, 356),
+    type = rep("s", 2), ncp = 2)
R> res.mfa <- MFA(cbind.data.frame(gene[, 1], res.impute$completeObs),
+    group = c(1, 76, 356), type = c("n", rep("s", 2)),
+    name.group = c("WHO", "CGH", "expr"), num.group.sup = 1)
```

The function `imputeMFA` takes as input the multi-table data `gene` (without its first column, corresponding to the categorical variable giving the tumor type), the argument `group` which specifies the number of variables per group, and the argument `type` which specifies the nature of the variables and the pre-processing step to apply to each group of variables. It takes as values `"n"` for "nominal" when the variables of the group are categorical, `"c"` when they are continuous and will be "centered", and `"s"` for continuous variables that will be "standardized" (each variable within a given group is divided by its standard deviation to give the same importance of each variable within a group). The other arguments are the same as those defined for the functions `imputePCA` and `imputeMCA`. Then, as before, the function gives as output a completed data set that is taken as input to the `MFA` function of the **FactoMineR** package. An MFA can then be performed on the completed data set. In this example, the first group `"WHO"`, which is composed of one categorical variable indicating tumor type, is added as supplementary information, i.e., it is not used when performing the global PCA; thus the principal components are obtained without information from this group. This group is therefore not part of the main analysis, and only used afterward to help enhance the interpretation of results (here for instance the individuals obtained from the global PCA are colored according to the variable in this group). Note that if at least one active group (not supplementary) has categorical variables, then the completed indicator matrix can be used in the `MFA` function like for the `MCA` and `FAMD` functions.

Let us present some of the graphical outputs obtained and interpret briefly the results. Figure 12 shows the representation of the tumors and variables. We interpret these graphs as those for PCA. They shows that the first dimension of variability separates the glioblastomas tumors from the lower grade tumors and that dimension 2 separates tumors O from tumors OA and A. The expression data is much more one-dimensional whereas the CGH data is represented in at least two dimensions (red arrows are hidden by green arrows).

```
R> plot(res.mfa, habillage = 1, lab.ind = FALSE)
R> plot(res.mfa, choix = "var", lab.var = FALSE, habillage = "group")
```

Figures 13 and 14 are specific to MFA. Figure 13 allows us to study the global similarities between groups using the following rule: Two groups are close if they induce the same structure, that is to say the relative positions of individuals in one group are similar to those of individuals in the other group. We can also see that the first dimension is common to the two groups (i.e., they have a high coordinate on dimension 1) whereas the second is mainly due to the group's CGH.

```
R> plot(res.mfa, choix = "group", habillage = "group")
```

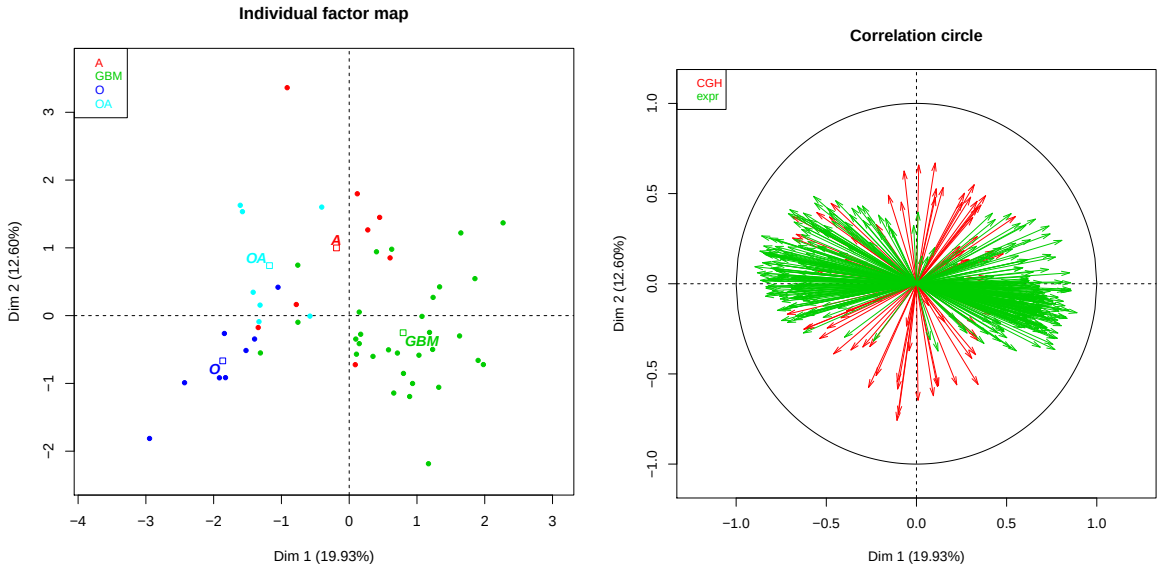

Figure 12: Representation of individuals and variables obtained from an MFA of the `gene` data set.

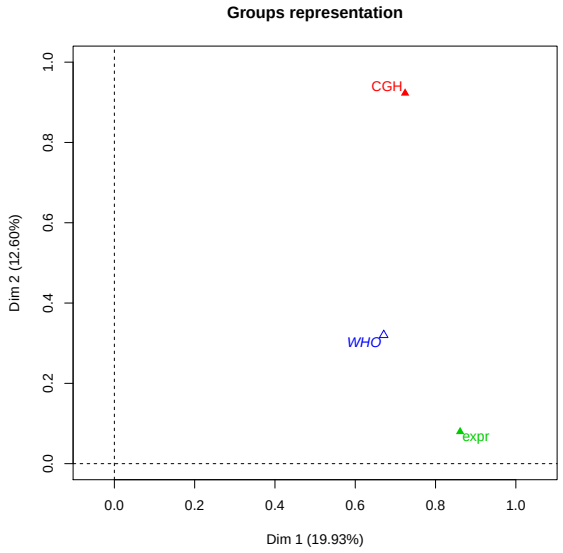

Figure 13: Group representations obtained from an MFA of the `gene` data set.

Figure 14 gives the "graph of partial points" that allows us to compare groups at the individual level. It represents an individual "seen" by each group of variables. Each individual is at the barycenter of its partial points and the more its partial points are close, the more "homogeneous" (i.e., seen in the same way by each group of variables) it is.

```
R> plot(res.mfa, invisible = "ind", partial = "all", habillage = "group")
R> plot(res.mfa, lab.ind = FALSE, partial = "GBM29", habillage = "group")
```

We can see on the left of Figure 14 that the partial coordinates of each tumor type are very

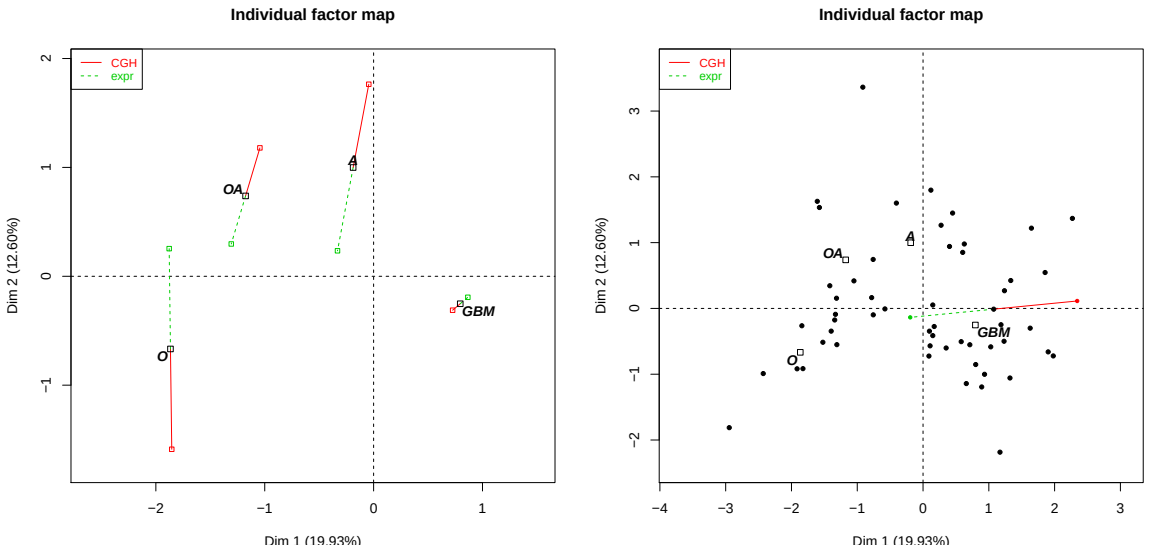

Figure 14: Representation of the partial points obtained from an MFA of the `gene` data set.

close in the first dimension. This means that both the expression and CGH data allow us to separate the glioblastomas from the other tumors. On the other hand, the coordinates of the expression data are nearly equal to zero for all points in the second dimension. This means that only the CGH data allows us to see differences between the O tumors and the OA and A. This representation is also useful to highlight specific individuals meriting closer inspection. Indeed, the individual with partial points in the right of Figure 14 does not have the same behavior as the others since its partial points have quite different coordinates in the first dimension. It means that when we look at its CGH data, this individual is on the side of the dangerous tumors (on the right of the plot) whereas it is on the side of the other tumors when considering its expression data. There is thus no consensus for this individual between the two information sources.

7. Single and multiple imputation with missMDA

The primary aim of all algorithms presented in this paper is to perform principal component methods despite missing values, i.e., estimating parameters and obtaining graphical outputs from incomplete data sets. However, imputation is done during the running of the algorithm and consequently these methods can be used to impute data. Even if at first this "imputation" may be seen as an aside to these methods, it is in fact very valuable and indeed, the quality of imputation is usually high. This can be explained by the fact that imputation is based on the scores and loadings and thus takes into account similarities between individuals as well as relationships between variables. In the PCA community, this property explains the revival of interest (and publications) in the "PCA and missing values" question, especially in the machine learning community with matrix completion problems such as the Netflix prize (Netflix 2009). This property can also be used for other methods and consequently, the methodology implemented in the **missMDA** package can be seen as an alternative to all other methods dedicated to imputing various data types (continuous, categorical and mixed) that are available in other software packages.

Many methods are available to perform single imputation of continuous variables using explicit or implicit models, including imputation based on k-nearest neighbors (Troyanskaya, Cantor, Sherlock, Brown, Hastie, Tibshirani, Botstein, and Altman 2001), imputation based on the assumption of a joint Gaussian distribution for the variables (Schafer 1997), and imputation obtained by successively drawing from conditional distributions (Van Buuren, Boshuizen, and Knook 1999; Van Buuren 2007). This latter approach consists roughly in specifying for each variable a model to predict the missing values, then cycling through the variables. Recent propositions include the algorithm of Mazumder, Hastie, and Tibshirani (2009), which is similar to the regularized iterative PCA algorithm except that it applies a soft thresholding rule on the singular values.

For imputation of categorical variables, the k-nearest neighbors method is also a popular approach. Imputation assuming a joint distribution of the variables is possible via log-linear models (Schafer 1997). However, this approach encounters difficulties when there are many variables. An alternative can be the use of latent class models (Vermunt, Van Ginkel, Van der Ark, and Sijtsma 2008). Imputation can also be obtained by successively drawing from conditional distributions (Van Buuren *et al.* 1999; Van Buuren 2007). For more details on these methods, we refer the reader to Little and Rubin (1987, 2002), Van Buuren (2012) and Carpenter and Kenward (2013).

Lastly, for mixed data, there are fewer options. One consists in transforming the categorical variables into dummy variables and then using an imputation based on the assumption of a joint Gaussian distribution for the variables. Alternatively, one can specify a model for each variable (Van Buuren *et al.* 1999), requiring a significant modeling effort on the part of the user. Kropko, Goodrich, Gelman, and Hill (2014) compared and discussed both approaches. One of the most recent propositions for mixed data is that of Stekhoven and Bühlmann (2012) who suggest an approach based on random forests, which seems to outperform the other methods in terms of imputation quality. Audigier *et al.* (2016a) compared imputation quality of the regularized iterative FAMD algorithm to the one using the algorithm of Stekhoven and Bühlmann (2012) and highlighted the performance of the former approach when imputing continuous, categorical and mixed data. The results become more accurate whenever there are strong linear relationships between variables, as well as when there are categorical variables with small frequency categories.

Thus, package **missMDA** provides imputed data sets with good predictions for missing values. However, as mentioned in the introduction, if a statistical method is applied to the completed data set, the variance of the estimators is underestimated because uncertainty in the missing data is not taken into account. Multiple imputation can be a solution.

The PCA multiple imputation method (MIPCA) is also an alternative to the other multiple imputation methods suggested in the literature, including multiple imputation based on joint modeling (Schafer 1997) or conditional modeling (Van Buuren *et al.* 1999; Van Buuren 2007). The comparisons made in Audigier *et al.* (2016b) showed that MIPCA is competitive in terms of coverage for different estimators, with the advantage that it can be applied directly when the number of individuals is smaller than the number of variables.

As mentioned earlier, many software products and R packages are available to perform single and multiple imputation. This includes for single imputation the R package **SoftImpute** (Hastie and Mazumder 2015), where Mazumder *et al.* (2009)'s method is implemented, as well as the **missForest** package (Stekhoven 2013) for Stekhoven and Bühlmann (2012)'s method.

The main packages that implement multiple imputation methods are **Amelia** (Honaker, King, and Blackwell 2011), **mice** (Van Buuren and Groothuis-Oudshoorn 2011) and **mi** (Su, Gelman, Hill, and Yajima 2011). Yucel (2011) gives a good overview of multiple imputation software products and packages. Van Buuren's web page (http://www.stefvanbuuren.nl/mi/Software.html) is a source of information for R packages and software products performing multiple imputation, as well as the CRAN Task View on Official Statistics and Survey Methodology (Templ 2015). The web page http://missingdata.lshtm.ac.uk/ of the maintainers of the software **REALCOMP-IMPUTE** (Carpenter, Goldstein, and Kenward 2011), which provides imputation for multi-level data, also mentions a number of resources on the topic of missing values.

8. Conclusion

The **missMDA** package presented in this paper performs principal component methods with missing values (PCA, MCA, FAMD, MFA) and can be used to impute continuous, categorical and mixed data. In addition, a multiple imputation strategy is implemented to study the variability of results in PCA and can be used as an alternative to other multiple imputation methods. As a complement to the paper, several videos are available on the YouTube play list "Exploratory multivariate data analysis with R and **FactoMineR**" (https://www.youtube.com/watch?v=YDbx2pk9xNY&list=PLnZgp6epRBbTsZEFXi_p6W48HhNyqwxIu&index=9) that show how to use the **missMDA** package.

In this paper, we do not dwell on the specific kind of missing values (such as missing at random, missing non at random) exhibited. Of course, a first step in a typical analysis is to look for patterns in the missing values and the reasons for their occurrence; indeed, methods as well as the properties of the methods to deal with missing values depend on this evaluation. To study the patterns of missing values, packages such as **VIM** (Templ *et al.* 2015) may be used. We also suggest the approach consisting of coding with "o" the observed values and "m" the missing ones and performing a multiple correspondence analysis on this data set to study associations between missing entries. The following lines of code can be used to visualize the pattern of missing values in a data set called `MyData`:

```
R> mis.ind <- matrix("o", nrow = nrow(MyData), ncol = ncol(MyData))
R> mis.ind[is.na(MyData)] <- "m"
R> dimnames(mis.ind) <- dimnames(MyData)
R> library("FactoMineR")
R> resMCA <- MCA(mis.ind, graph=FALSE)
R> plot(resMCA, invis = "ind", title = "MCA graph of the categories")
```

Methods implemented in this package could be used on very large data sets. However, the implementation in package **missMDA** is not optimized for such purposes since it is based on the `MCA` and `PCA` functions which are as well not optimized for this. The computational time of the functions increases with the number of rows and columns and tends to increase with the number of missing values. However, when the variables are strongly related (i.e., there is a strong structure in the data), it is "easy" to impute the missing entries even if there are many missing values and the methods are fast. Of course the structure of the data is not known in advance and consequently it is difficult to know in advance the computational time for a given number of rows and columns.

Future research will be focused on the development of methods to take into account uncertainty for techniques other than PCA and develop multiple imputation methods for categorical and mixed data sets. We also note that combining the results of statistical analyses after multiple imputation is still a challenge. Indeed, theoretical results are only available for very simple models such as linear regression and only for specific parameters, e.g., coefficients. There is therefore huge room for improvement in this direction. Finally, the methods to select the number of dimensions from incomplete data have also to be developed for the methods FAMD and MFA.

References

Acar E, Dunlavy DM, Kolda TG, Mrup M (2011). "Scalable Tensor Factorizations for Incomplete Data." *Chemometrics and Intelligent Laboratory Systems*, **106**(1), 41–56. doi:10.1016/j.chemolab.2010.08.004.

Andersson C, Bro R (2000). "The *N*-Way Toolbox for MATLAB." *Chemometrics and Intelligent Laboratory Systems*, **52**(1), 1–4. doi:10.1016/s0169-7439(00)00071-x.

Audigier V, Husson F, Josse J (2016a). "A Principal Components Method to Impute Missing Values for Mixed Data." *Advances in Data Analysis and Classification*, **10**(1), 5–26. doi:10.1007/s11634-014-0195-1.

Audigier V, Husson F, Josse J (2016b). "Multiple Imputation for Continuous Variables Using a Bayesian Principal Component Analysis." *Journal of Statistical Computation and Simulation*. doi:10.1080/00949655.2015.1104683. Forthcoming.

Bro R, Kjeldahl K, Smilde A, Kiers H (2008). "Cross-Validation of Component Model: A Critical Look at Current Methods." *Analytical and Bioanalytical Chemistry*, **390**, 1241–1251. doi:10.1007/s00216-007-1790-1.

CAMO (2013). ***Unscrambler***. **Unscrambler** Software, URL http://www.camo.com/products/unscrambler-features.html.

Carpenter J, Kenward M (2013). *Multiple Imputation and Its Application*. John Wiley & Sons.

Carpenter JR, Goldstein H, Kenward MG (2011). "**REALCOM-IMPUTE** Software for Multilevel Multiple Imputation with Mixed Response Types." *Journal of Statistical Software*, **45**(5), 1–14. doi:10.18637/jss.v045.i05.

Caussinus H (1986). "Models and Uses of Principal Component Analysis." In J Leeuw, W Heiser, J Meulman, F Critchley (eds.), *Multidimensional Data Analysis*, pp. 149–178. D.S.W.O. Press.

Craven P, Wahba G (1979). "Smoothing Noisy Data with Spline Functions: Estimating the Correct Degree of Smoothing by the Method of Generalized Cross-Validation." *Numerische Mathematik*, **31**(4), 377–403. doi:10.1007/bf01404567.

De Leeuw J, Mair P (2009a). "Gifi Methods for Optimal Scaling in R: The Package **homals**." *Journal of Statistical Software*, **31**(4), 1–20. doi:10.18637/jss.v031.i04.

De Leeuw J, Mair P (2009b). "Simple and Canonical Correspondence Analysis Using the R Package **anacor**." *Journal of Statistical Software*, **31**(5), 1–18. doi:10.18637/jss.v031.i05.

Dempster A, Laird N, Rubin D (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society B*, **39**(1), 1–38.

Dray S (2007). "The **ade4** Package: Implementing the Duality Diagram for Ecologists." *Journal of Statistical Software*, **22**(4), 1–20. doi:10.18637/jss.v022.i04.

Eigenvector Research (2011). ***PLS** Toolbox: Advance Chemometrics Software for Use with MATLAB*. URL http://www.eigenvector.com/software/pls_toolbox.htm.

Escofier B (1979). "Traitement Simultané de Variables Quantitatives et Qualitatives en Analyse Factorielle." *Les Cahiers de l'Analyse des Données*, **4**(2), 137–146.

Escofier B, Pagès J (2008). *Analyses Factorielles Simples et Multiples*. Dunod.

Gabriel K, Zamir S (1979). "Lower Rank Approximation of Matrices by Least Squares with Any Choice of Weights." *Technometrics*, **21**(4), 236–246. doi:10.1080/00401706.1979.10489819.

Gifi A (1990). *Non-Linear Multivariate Analysis*. John Wiley & Sons, Chichester, England.

Gower J, Dijksterhuis G (2004). *Procrustes Problems*. Oxford University Press, New York.

Greenacre M (1984). *Theory and Applications of Correspondence Analysis*. Academic Press.

Greenacre M (2007). *Correspondence Analysis in Practice*. 2nd edition. Chapman & Hall/CRC. doi:10.1201/9781420011234.

Greenacre M, Blasius J (2006). *Multiple Correspondence Analysis and Related Methods*. Chapman & Hall/CRC. doi:10.1201/9781420011319.

Greenacre M, Pardo R (2006). "Subset Correspondence Analysis: Visualizing Relationships Among a Selected Set of Response Categories from a Questionnaire Survey." *Sociological Methods and Research*, **35**(2), 193–218. doi:10.1177/0049124106290316.

Hastie T, Mazumder R (2015). ***softImpute**: Matrix Completion via Iterative Soft-Thresholded SVD*. R package version 1.4, URL https://CRAN.R-project.org/package=softImpute.

Honaker J, King G, Blackwell M (2011). "**Amelia** II: A Program for Missing Data." *Journal of Statistical Software*, **45**(7), 1–47. doi:10.18637/jss.v045.i07.

Husson F, Josse J (2013). "Handling Missing Values in Multiple Factor Analysis." *Food Quality and Preferences*, **30**(2), 77–85. doi:10.1016/j.foodqual.2013.04.013.

Husson F, Josse J (2014). "Multiple Correspondence Analysis." In M Greenacre, J Blasius (eds.), *Visualization and Verbalization of Data*, pp. 163–181. Chapman & Hall/CRC, London.

Husson F, Josse J (2016). ***missMDA**: Handling Missing Values with/in Multivariate Data Analysis (Principal Component Methods)*. R package version 1.10, URL https://CRAN.R-project.org/package=missMDA.

Husson F, Josse J, Lê S, Mazet J (2016). ***FactoMineR**: Multivariate Exploratory Data Analysis and Data Mining with R*. R package version 1.32, URL https://CRAN.R-project.org/package=FactoMineR.

Husson F, Lê S, Pagès J (2010). *Exploratory Multivariate Analysis by Example Using R*. Chapman & Hall/CRC. doi:10.1201/b10345.

Ilin A (2010). *MATLAB Package for PCA for Datasets with Missing Values*. URL http://users.ics.aalto.fi/alexilin/software/.

Ilin A, Raiko T (2010). "Practical Approaches to Principal Component Analysis in the Presence of Missing Values." *Journal of Machine Learning Research*, **11**, 1957–2000. doi:10.1007/978-3-540-74958-5_69.

Jolliffe I (2002). *Principal Component Analysis*. Springer-Verlag.

Josse J, Chavent M, Liquet B, Husson F (2012). "Handling Missing Values with Regularized Iterative Multiple Correspondence Analysis." *Journal of Classification*, **29**(1), 91–116. doi:10.1007/s00357-012-9097-0.

Josse J, Eeuwijk F, Piepho HP, Denis JB (2014). "Another Look at Bayesian Analysis of AMMI Models for Genotype-Environment Data." *Journal of Agricultural, Biological, and Environmental Statistics*, **19**(2), 240–257. doi:10.1007/s13253-014-0168-z.

Josse J, Husson F (2011a). "Multiple Imputation in PCA." *Advances in Data Analysis and Classification*, **5**(3), 231–246. doi:10.1007/s11634-011-0086-7.

Josse J, Husson F (2011b). "Selecting the Number of Components in PCA Using Cross-Validation Approximations." *Computational Statistics & Data Analysis*, **56**(6), 1869–1879. doi:10.1016/j.csda.2011.11.012.

Josse J, Husson F (2012). "Handling Missing Values in Exploratory Multivariate Data Analysis Methods." *Journal de la Société Française de Statistique*, **153**(2), 79–99.

Josse J, Pagès J, Husson F (2009). "Gestion des Données Manquantes en Analyse en Composantes Principales." *Journal de la Société Française de Statistique*, **150**(2), 28–51.

Kiers H (1991). "Simple Structure in Component Analysis Techniques for Mixtures of Qualitative and Quantitative Variables." *Psychometrika*, **56**(2), 197–212. doi:10.1007/bf02294458.

Kiers H (1997). "Weighted Least Squares Fitting Using Ordinary Least Squares Algorithms." *Psychometrika*, **62**(2), 251–266. doi:10.1007/bf02295279.

Kroonenberg PM (2008). *Applied Multiway Data Analysis*. John Wiley & Sons.

Kroonenberg PM (2011). ***3WayPack** – Three-Mode Software*. URL http://www.leidenuniv.nl/fsw/three-mode.

Kropko J, Goodrich B, Gelman A, Hill J (2014). "Multiple Imputation for Continuous and Categorical Data: Comparing Joint Multivariate Normal and Conditional Approaches." *Political Analysis*, **22**(4), 497–519. doi:10.1093/pan/mpu007.

Lê S, Josse J, Husson F (2008). "**FactoMineR**: An R Package for Multivariate Analysis." *Journal of Statistical Software*, **25**(1), 1–18. doi:10.18637/jss.v025.i01.

Little R, Rubin D (1987, 2002). *Statistical Analysis with Missing Data*. John Wiley & Sons, New York.

Mazumder R, Hastie T, Tibshirani R (2009). "Spectral Regularization Algorithms for Learning Large Incomplete Matrices." *Journal of Machine Learning Research*, **11**, 2287–2322.

Meulman J (1982). *Homgeneity Analysis of Incomplete Data*. D.S.W.O. Press, Leiden.

Meulman J, Heiser W, SPSS (2003). ***CATPCA***. SPSS Categories 13.0. Chicago: SPSS.

Nenadic O, Greenacre M (2007). "Correspondence Analysis in R, with Two- And Three-Dimensional Graphics: The **ca** Package." *Journal of Statistical Software*, **20**(3), 1–13. doi:10.18637/jss.v020.i03.

Netflix (2009). "Netflix Challenge." URL http://www.netflixprize.com/.

Nora-Chouteau C (1974). *Une Méthode de Reconstitution et d'Analyse de Données Incomplètes*. Ph.D. thesis, Université Pierre et Marie Curie.

Pagès J (2015). *Multiple Factor Analysis by Example Using R*. Chapman & Hall/CRC. doi:10.1201/b17700.

Porta JM, Verbeek JJ, Kröse BJA (2005). "Active Appearance-Based Robot Localization Using Stereo Vision." *Autonomous Robots*, **18**(1), 59–80. doi:10.1023/b:auro.0000047287.00119.b6.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Royo C, Rodriguez A, Romagosa I (1993). "Differential Adaptation of Complete and Substitute Triticale." *Plant Breeding*, **111**, 113–119. doi:10.1111/j.1439-0523.1993.tb00616.x.

Rubin D (1976). "Inference and Missing Data." *Biometrika*, **63**(3), 581–592. doi:10.1093/biomet/63.3.581.

Rubin D (1987). *Multiple Imputation for Non-Response in Survey*. John Wiley & Sons.

Schafer J (1997). *Analysis of Incomplete Multivariate Data*. Chapman & Hall/CRC. doi:10.1201/9781439821862.

Smilde A, Bro R, Geladi P (2004). *Multi-Way Analysis: Applications in the Chemical Sciences*. John Wiley & Sons. doi:10.1002/0470012110.

Stacklies W, Redestig H, Scholz M, Walther D, Selbig J (2007). "**pcaMethods** – A Bioconductor Package Providing PCA Methods for Incomplete Data." *Bioinformatics*, **23**, 1164–1167. doi:10.1093/bioinformatics/btm069.

Stekhoven DJ (2013). ***missForest**: Nonparametric Missing Value Imputation Using Random Forest*. R package version 1.4, URL https://CRAN.R-project.org/package=missForest.

Stekhoven DJ, Bühlmann P (2012). "**missForest** – Nonparametric Missing Value Imputation for Mixed-Type Data." *Bioinformatics*, **28**, 113–118. doi:10.1093/bioinformatics/btr597.

Su YS, Gelman A, Hill J, Yajima M (2011). "Multiple Imputation with Diagnostics (**mi**) in R: Opening Windows into the Black Box." *Journal of Statistical Software*, **45**(2), 1–31. doi:10.18637/jss.v045.i02.

Tayrac M, Lê S, Aubry M, Mosser J, Husson F (2009). "Simultaneous Analysis of Distinct Omics Data Sets with Integration of Biological Knowledge: Multiple Factor Analysis Approach." *BMC Genomics*, **10**(32). doi:10.1186/1471-2164-10-32.

Templ M (2015). "CRAN Task View: Official Statistics & Survey Methodology." Version 2015-11-21, URL https://CRAN.R-project.org/view=OfficialStatistics.

Templ M, Alfons A, Kowarik A, Prantner B (2015). ***VIM**: Visualization and Imputation of Missing Values*. R package version 4.4.1, URL https://CRAN.R-project.org/package=VIM.

Tomasi G, Bron R (2005). "PARAFAC and Missing Values." *Chemometrics and Intelligent Laboratory Systems*, **75**, 163–180. doi:10.1016/j.chemolab.2004.07.003.

Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D, Altman R (2001). "Missing Value Estimation Methods for DNA Microarrays." *Bioinformatics*, **17**(6), 520–525. doi:10.1093/bioinformatics/17.6.520.

Umetrics (2013). ***SIMCA**: The Standard in Multivariate Data Analysis*. **SIMCA** Software, URL http://www.umetrics.com/products/simca/.

Van Buuren S (2007). "Multiple Imputation of Discrete and Continuous Data by Fully Conditional Specification." *Statistical Methods in Medical Research*, **16**(3), 219–242. doi:10.1177/0962280206074463.

Van Buuren S (2012). *Flexible Imputation of Missing Data*. Chapman & Hall/CRC, Boca Raton. doi:10.1201/b11826.

Van Buuren S, Boshuizen HC, Knook DL (1999). "Multiple Imputation of Missing Blood Pressure Covariates in Survival Analysis." *Statistics in Medicine*, **18**(6), 681–694. doi:10.1002/(sici)1097-0258(19990330)18:6<681::aid-sim71>3.0.co;2-r.

Van Buuren S, Groothuis-Oudshoorn K (2011). "**mice**: Multivariate Imputation by Chained Equations in R." *Journal of Statistical Software*, **45**(3), 1–67. doi:10.18637/jss.v045.i03.

Verbanck M, Josse J, Husson F (2015). "Regularized PCA to Denoise and Visualise Data." *Statistics and Computing*, **25**(2), 471–486. doi:10.1007/s11222-013-9444-y.

Vermunt J, Van Ginkel JR, Van der Ark LA, Sijtsma K (2008). "Multiple Imputation of Incomplete Categorical Data Using Latent Class Analysis." *Sociological Methodology*, **33**, 369–397.

Wold H, Lyttkens E (1969). "Nonlinear Iterative Partial Least Squares (NIPALS) Estimation Procedures." *Bulletin of the International Statistical Institute*, **43**, 29–51.

Yucel R (2011). "State of the Multiple Imputation Software." *Journal of Statistical Software*, **45**(1), 1–7. doi:10.18637/jss.v045.i01.

Affiliation:

Julie Josse
Department of Statistics
Agrocampus Ouest Rennes
35042 Rennes, France
E-mail: josse@agrocampus-ouest.fr
URL: http://math.agrocampus-ouest.fr/infoglueDeliverLive/membres/julie.josse

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
April 2016, Volume 70, Issue 1 *Submitted:* 2014-09-10
doi:10.18637/jss.v070.i01 *Accepted:* 2015-03-26