

# Package ‘missMDA’

February 14, 2012

**Type** Package

**Title** Handling missing values with/in multivariate data analysis (principal component methods)

**Version** 1.2

**Date** 2010-10-13

**Author** Francois Husson, Julie Josse

**Maintainer** Francois Husson <husson@agrocampus-ouest.fr>, Julie Josse  
<josse@agrocampus-ouest.fr>

**Description** Imputation of incomplete continuous or categorical datasets; Missing values are imputed with a principal component analysis (PCA), a multiple correspondence analysis (MCA) model or a multiple factor analysis (MFA) model; Perform multiple imputation with and in PCA

**Depends** FactoMineR

**License** GPL (>= 2)

**URL** <http://www.agrocampus-ouest.fr/math/husson>,  
<http://www.agrocampus-ouest.fr/math/josse>

**Encoding** latin1

**LazyLoad** yes

**Repository** CRAN

**Date/Publication** 2010-10-13 11:29:44

## R topics documented:

estim_ncpMCA . . . . .	2
estim_ncpPCA . . . . .	3
imputeMCA . . . . .	4
imputeMFA . . . . .	5
imputePCA . . . . .	7

MIPCA . . . . .	8
orange . . . . .	10
plot.MIPCA . . . . .	10
vnf . . . . .	12

<b>Index</b>	<b>13</b>
--------------	-----------

---

estim_ncpMCA	<i>Estimate the number of dimensions for the Multiple Correspondence Analysis by cross-validation</i>
--------------	---

---

### Description

Estimate the number of dimensions for the Multiple Correspondence Analysis by cross-validation

### Usage

```
estim_ncpMCA(don, ncp.min=0, ncp.max=5, nbsim=100, pNA=0.05, threshold=1e-4)
```

### Arguments

don	a data.frame with categorical variables; with missing entries or not
ncp.min	integer corresponding to the minimum number of components to test
ncp.max	integer corresponding to the maximum number of components to test
nbsim	number of simulations
pNA	percentage of missing values added in the data set
threshold	the threshold for assessing convergence

### Details

For the cross-validation, pNA percentage of missing values are removed at random and predicted with a MCA model using ncp.min to ncp.max dimensions. This process is repeated nbsim times. The number of components which leads to the smallest MSEF is retained. Each cell is predicted using the imputeMCA function, it means using the regularized iterative MCA algorithm.

### Value

ncp	the number of components retained for the MCA
criterion	the criterion (the MSEF) calculated for each number of components

### Author(s)

Francois Husson <husson@agrocampus-ouest.fr> and Julie Josse <Julie.Josse@agrocampus-ouest.fr>

### References

Josse, J., Chavent, M., Liquet, B. and Husson, F. (2010). Handling missing values with Regularized Iterative Multiple Correspondence Analysis.

**Examples**

```
## Not run:
data(vnf)
result <- estim_ncpMCA(vnf,ncp.min=0, ncp.max=3, nbsim=100)

## End(Not run)
```

---

estim_ncpPCA	<i>Estimate the number of dimensions for the Principal Component Analysis by cross-validation</i>
--------------	---

---

**Description**

Estimate the number of dimensions for the Principal Component Analysis by cross-validation

**Usage**

```
estim_ncpPCA(X, ncp.min = 0, ncp.max = 5, method = "Regularized", scale = TRUE, method.cv = "loo", nbsim
```

**Arguments**

X	a data.frame with continuous variables; with missing entries or not
ncp.min	integer corresponding to the minimum number of components to test
ncp.max	integer corresponding to the maximum number of components to test
method	"Regularized" by default or "EM"
scale	boolean. By default TRUE leading to a same weight for each variable
method.cv	string with the values "loo" for leave-one-out or "Kfold" cross-validation
nbsim	number of simulations, useful only if method.cv="Kfold"
pNA	percentage of missing values added in the data set, useful only if method.cv="Kfold"
threshold	the threshold for assessing convergence

**Details**

For leave-one-out (loo) cross-validation, each value is alternatively removed and predicted with a PCA model using ncp.min to ncp.max dimensions. The number of components which leads to the smallest MSE is retained. Each cell is predicted using the imputePCA function, it means using the regularized iterative PCA algorithm or the iterative PCA (EM cross-validation).

For the Kfold cross-validation, pNA percentage of missing values is removed and predicted with a PCA model using ncp.min to ncp.max dimensions. This process is repeated nbsim times. The leave-one-out method is time-consuming (method.cv="loo") when the number of cells is important in the data.frame.

The regularized version is more appropriate when there are many missing values in the dataset (to avoid overfitting).

**Value**

ncp                    the number of components retained for the PCA  
 criterion            the criterion (the MSE) calculated for each number of components

**Author(s)**

Francois Husson <husson@agroparcampus-ouest.fr> and Julie Josse <Julie.Josse@agroparcampus-ouest.fr>

**References**

Bro, R., Kjeldahl, K. Smilde, A. K. and Kiers, H. A. L. (2008) Cross-validation of component models: A critical look at current methods. *Analytical and Bioanalytical Chemistry*, 5, 1241-1251.  
 J. Josse, F. Husson et J. Pagès (2009) Gestion des données manquantes en Analyse en Composantes Principales. *Journal de la SFdS*. 150 (2), pp. 28-51.

**Examples**

```
## Not run:
data(orange)
nb <- estim_ncpPCA(orange,ncp.min=0,ncp.max=4) ## Time consuming, nb = 2

## End(Not run)
```

---

imputeMCA	<i>Impute missing values in categorical variables with Multiple Correspondence Analysis</i>
-----------	---

---

**Description**

Impute the missing values of a categorical dataset (in the indicator matrix) with Multiple Correspondence Analysis

**Usage**

```
imputeMCA(don, ncp=2, threshold=1e-06, seed=NULL, maxiter=1000)
```

**Arguments**

don                    a data.frame with categorical variables containing missing values  
 ncp                    integer corresponding to the number of dimensions used to reconstruct data with the reconstruction formulae  
 threshold            the threshold for assessing convergence  
 seed                    an integer to specify the seed for the initialization for the regularized iterative MCA algorithm (if seed = NULL the initialization step corresponds to the imputation of the proportion of each category)  
 maxiter                integer, maximum number of iterations for the regularized iterative MCA algorithm

**Details**

Use a Regularized Iterative Multiple Correspondence Analysis to impute missing values. The regularized iterative MCA algorithm first imputes the missing values in the indicator matrix with initial values (the proportion of each category), then performs MCA on the completed dataset, imputes the missing values with the reconstruction formulae of order `ncp` and iterates until convergence.

If `ncp=0`, the Average method (imputation with the proportion) is performed.

**Value**

Return the imputed indicator matrix. The imputed values are real numbers and may be seen as degree of membership to the corresponding category.

**Author(s)**

Francois Husson <husson@agrocampus-ouest.fr> and Julie Josse <Julie.Josse@agrocampus-ouest.fr>

**References**

Josse, J., Chavent, M., Liquet, B. and Husson, F. (2010). Handling missing values with Regularized Iterative Multiple Correspondence Analysis.

**See Also**

[estim\\_ncpMCA](#)

**Examples**

```
## Not run:
data(vnf)
## First the number of components has to be chosen
## (for the reconstruction step)
## nb <- estim_ncpMCA(vnf,ncp.max=5) ## Time-consuming, nb = 4

## Impute indicator matrix
tab.disj <- imputeMCA(vnf, ncp=4)

## A MCA can be performed
res.mca <- MCA(vnf, tab.disj=tab.disj)

## End(Not run)
```

---

imputeMFA

*Impute dataset with MFA*

---

**Description**

Impute the missing values of a dataset with the Multiple Factor Analysis model. Can be used as a preliminary step before performing a MFA on an incomplete dataset with continuous variables.

**Usage**

```
imputeMFA(X, group, ncp = 2, scale = TRUE, method = "Regularized",
          threshold = 1e-06, seed = NULL, nb.init = 1, maxiter = 1000, ...)
```

**Arguments**

X	a data.frame with continuous variables containing missing values
group	a list indicating the number of variables in each group
ncp	integer corresponding to the number of components used to reconstruct data with the PCA reconstruction formulae
scale	boolean. By default TRUE leading to a same weight for each variable
method	"Regularized" by default or "EM"
threshold	the threshold for assessing convergence
seed	a single value, interpreted as an integer for the set.seed function (if seed = NULL, missing values are initially imputed by the mean of each variable)
nb.init	integer corresponding to the number of random initializations; the first initialization is the mean of each variable
maxiter	integer, maximum number of iteration for the algorithm
...	further arguments passed to or from other methods

**Details**

Impute the missing entries of a data frame using the iterative MFA algorithm (EM) or a regularized iterative MFA algorithm. The iterative MFA algorithm first imputes the missing values with initial values (the means of each variable), then performs MFA on the completed dataset, imputes the missing values with the reconstruction formulae of order ncp and iterates until convergence. The regularized version allows to avoid overfitting problems, especially important when there are many missing values.

**Value**

completeObs	the imputed dataset; the observed values for non-missing entries and the imputed values for missing values
objective	the criterion, the mean square error between the reconstructed data and the observed data
recon	the reconstructed data

**Author(s)**

Francois Husson <husson@agrocampus-ouest.fr> and Julie Josse <Julie.Josse@agrocampus-ouest.fr>

**References**

PhD thesis of J. Josse or HDR of F. Husson

**See Also**[imputePCA](#)**Examples**

```
data(orange)
res.comp <- imputePCA(orange,group=c(5,3),scale=TRUE,ncp=2)
res.pca <- MFA(res.comp$completeObs,group=c(5,3),type=rep("s",2))
```

imputePCA

*Impute dataset with PCA***Description**

Impute the missing values of a dataset with the Principal Components Analysis model. Can be used as a preliminary step before performing a PCA on an incomplete dataset.

**Usage**

```
imputePCA(X, ncp = 2, scale = TRUE, method = "Regularized",
          threshold = 1e-06, seed = NULL, nb.init = 1, maxiter = 1000, ...)
```

**Arguments**

X	a data.frame with continuous variables containing missing values
ncp	integer corresponding to the number of components used to reconstruct data with the PCA reconstruction formulae
scale	boolean. By default TRUE leading to a same weight for each variable
method	"Regularized" by default or "EM"
threshold	the threshold for assessing convergence
seed	a single value, interpreted as an integer for the set.seed function (if seed = NULL, missing values are initially imputed by the mean of each variable)
nb.init	integer corresponding to the number of random initializations; the first initialization is the mean of each variable
maxiter	integer, maximum number of iteration for the algorithm
...	further arguments passed to or from other methods

**Details**

Impute the missing entries of a data frame using the iterative PCA algorithm (EM) or a regularized iterative PCA algorithm. The iterative PCA algorithm first imputes the missing values with initial values (the means of each variable), then performs PCA on the completed dataset, imputes the missing values with the reconstruction formulae of order ncp and iterates until convergence. The regularized version allows to avoid overfitting problems, especially important when there are many missing values.

**Value**

completeObs	the imputed dataset; the observed values for non-missing entries and the imputed values for missing values
objective	the criterion, the mean square error between the reconstructed data and the observed data
recon	the reconstructed data

**Author(s)**

Francois Husson <husson@agrocampus-ouest.fr> and Julie Josse <Julie.Josse@agrocampus-ouest.fr>

**References**

J. Josse, F. Husson et J. Pagès (2009) Gestion des données manquantes en Analyse en Composantes Principales. Journal de la SFdS. 150 (2), pp. 28-51.  
 Josse, J., Husson, F. (2010). Multiple Imputation in PCA.

**See Also**

[estim\\_ncpPCA,MIPCA](#)

**Examples**

```
data(orange)
## First the number of components has to be chosen
## (for the reconstruction step)
## nb <- estim_ncpPCA(orange,ncp.max=5) ## Time consuming, nb = 2

## Imputation
res.comp <- imputePCA(orange,ncp=2)

## A PCA can be performed
res.pca <- PCA(res.comp$completeObs)
```

---

MIPCA

---

*Multiple Imputation with PCA*


---

**Description**

MIPCA performs Multiple Imputation with a PCA model. Can be used as a preliminary step to perform Multiple Imputation in PCA

**Usage**

```
MIPCA(X, ncp = 2, scale = TRUE, method = "Regularized",
      threshold = 1e-04, nboot = 100)
```

**Arguments**

X	a data.frame with continuous variables containing missing values
ncp	integer corresponding to the number of components used to reconstruct data with the PCA reconstruction formulae
scale	boolean. By default TRUE leading to a same weight for each variable
method	"Regularized" by default or "EM"
threshold	the threshold for the criterion convergence
nboot	the number of imputed datasets

**Details**

MIPCA generates nboot imputed datasets from a PCA model. The observed values are the same from one dataset to the others whereas the imputed values change. The variation among the imputed values reflects the variability with which missing values can be predicted. The multiple imputation is proper in the sense of Little and Rubin (2002) since it takes into account the variability of the parameters.

**Value**

res.imputePCA	A matrix corresponding to the imputed dataset obtained with the function imputePCA (the completed dataset)
res.MI	An array corresponding to nboot imputed dataset. The dimensions of the array are: the number of row of X, the number of column of X and nboot

**Author(s)**

Francois Husson <husson@agrocampus-ouest.fr> and Julie Josse <Julie.Josse@agrocampus-ouest.fr>

**References**

Josse, J., Husson, F. (2010). Multiple Imputation in PCA.

**See Also**

[imputePCA](#), [plot.MIPCA](#)

**Examples**

```
data(orange)
## First the number of components has to be chosen
## (for the reconstruction step)
## nb <- estim_ncpPCA(orange,ncp.max=5) ## Time consuming, nb = 2

## Multiple Imputation
resMI <- MIPCA(orange,ncp=2)

## Visualization on the PCA map
plot(resMI)
```

---

orange

*Sensory description of 12 orange juices by 8 attributes.*

---

### **Description**

Sensory description of 12 orange juices by 8 attributes. Some values are missing.

### **Usage**

```
data(orange)
```

### **Format**

A data frame with 12 rows and 8 columns. Rows represent the different orange juices, columns represent the attributes.

### **Details**

A sensory data frame.

### **Source**

Francois Husson, Agrocampus Rennes

### **Examples**

```
data(orange)
## Not run:
nb <- estim_ncpPCA(orange,ncp.min=0,ncp.max=5,method.cv="Kfold",nbsim=20,pNA=0.05)
res.comp <- imputePCA(orange,ncp=nb$ncp)
res.pca <- PCA(res.comp$completeObs)
resMI <- MIPCA(orange,ncp=nb$ncp)
plot(resMI)

## End(Not run)
```

---

plot.MIPCA

*Plot the graphs for the Multiple Imputation in PCA*

---

### **Description**

From the multiple imputed datasets, the function plots graphs for the individuals, variables and dimensions for the Principal Component Analysis (PCA)

**Usage**

```
## S3 method for class 'MIPCA'  
plot(x, choice = "all", axes = c(1, 2), new.plot = TRUE,  
      main = NULL, level.conf = 0.95, ...)
```

**Arguments**

x	an object of class MIPCA
choice	the graph(s) to plot. By default "all" the graphs are plotted. "ind.proc" the procrustean representation of the individuals, "dim" the representation of the dimensions of the PCA, "ind.supp" the projection of the individuals as supplementary individuals, "var" the projection of the variables as supplementary variables
axes	a length 2 vector specifying the components to plot
new.plot	boolean, if TRUE, a new graphical device is created
main	string corresponding to the title of the graph you draw (by default NULL and a title is chosen)
level.conf	confidence level used to construct the ellipses. By default, 0.95
...	further arguments passed to or from other methods

**Details**

Plots the multiple imputed datasets obtained by the function MIPCA. The idea is to represent the multiple imputed dataset on a reference configuration (the map obtained from the PCA on the incomplete dataset). Different ways are available to take into account and visualize the supplement variability due to missing values.

**Value**

Four graphs can be drawn:

ind.supp	The individuals of the imputed datasets are projected as supplementary individuals onto the reference PCA map; then confidence ellipses are drawn
var	The variables of the imputed datasets are projected as supplementary variables onto the reference PCA map
ind.proc	A PCA is performed on each imputed dataset and each configuration of scores is rotated onto the reference PCA map with procrustes rotation; then confidence ellipses are drawn
dim	The dimensions of each imputed dataset are projected as supplementary variables onto the dimensions of the reference PCA dimensions

**Author(s)**

Francois Husson <husson@agrocampus-ouest.fr> and Julie Josse <Julie.Josse@agrocampus-ouest.fr>

**References**

Josse, J., Husson, F. (2010). Multiple Imputation in PCA

**See Also**[MIPCA](#)**Examples**

```
data(orange)
## nb <- estim_ncpPCA(orange,ncp.max=5) ## Time consuming, nb = 2
resMI <- MIPCA(orange,ncp=2)
plot(resMI)
```

---

vnf

*Questionnaire done by 1232 individuals who answered 14 questions*

---

**Description**

A user satisfaction survey of pleasure craft operators on the “Canal des Deux Mers”, located in South of France, was carried out by the public corporation “Voies Navigables de France” responsible for managing and developing the largest network of navigable waterways in Europe

**Usage**

```
data(vnf)
```

**Format**

A data frame with 1232 observations on the following 14 categorical variables.

**Source**

Josse, J., Chavent, M., Liquet, B. and Husson, F. (2010). Handling missing values with Regularized Iterative Multiple Correspondence Analysis.

**Examples**

```
data(vnf)
tab.disj <- imputeMCA(vnf,ncp=2)

res.mca <- MCA(vnf,tab.disj=tab.disj)
```

# Index

\*Topic **datasets**

orange, [10](#)

vnf, [12](#)

\*Topic **dplot**

plot.MIPCA, [10](#)

\*Topic **models**

imputeMCA, [4](#)

imputeMFA, [5](#)

imputePCA, [7](#)

\*Topic **multivariate**

estim\_ncpMCA, [2](#)

estim\_ncpPCA, [3](#)

imputeMCA, [4](#)

imputeMFA, [5](#)

imputePCA, [7](#)

MIPCA, [8](#)

estim\_ncpMCA, [2](#), [5](#)

estim\_ncpPCA, [3](#), [8](#)

imputeMCA, [4](#)

imputeMFA, [5](#)

imputePCA, [7](#), [7](#), [9](#)

MIPCA, [8](#), [8](#), [12](#)

orange, [10](#)

plot.MIPCA, [9](#), [10](#)

vnf, [12](#)