Package 'missMDA'

June 25, 2018

Type Package

Title Handling Missing Values with Multivariate Data Analysis

Version 1.13

Date 2018-06-25

Author Francois Husson, Julie Josse

Maintainer Francois Husson <husson@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 or MCA.

Depends R (>= 3.3.0)

Imports FactoMineR,graphics,grDevices,mice,mvtnorm,stats,utils

Suggests knitr

License GPL (>= 2)

URL http://math.agrocampus-ouest.fr/infoglueDeliverLive/membres/Francois.Husson, http://juliejosse.com/

Encoding latin1

LazyLoad yes

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2018-06-25 14:28:03 UTC

R topics documented:

missMDA-package 2
estim_ncpFAMD 3
estim_ncpMCA 5
estim_ncpPCA 6

gene . . . 8
geno . . . 8
imputeCA . . . 9
imputeFAMD . . . 10
imputeMCA . . . 12
imputeMFA . . . 15
imputeMultilevel . . . 17
imputePCA . . . 19
MIMCA . . . 21
MIPCA . . . 23
orange . . . 25
Overimpute . . . 26
ozone . . . 27
plot.MIMCA . . . 28
plot.MIPCA . . . 30
prelim . . . 31
snorena . . . 32
TitanicNA . . . 33
vnf . . . 34

Index **35**

`missMDA-package` *Handling missing values with/in multivariate data analysis (principal component methods)*

Description

handle missing values in exploratory multivariate analysis such as principal component analysis (PCA), multiple correspondence analysis (MCA), factor analysis for mixed data (FAMD) and multiple factor analysis (MFA)
impute missing values in continuous data sets using the PCA model, categorical data sets using MCA, mixed data using FAMD
generate multiple imputed data sets for continuous data using the PCA model and for categorical data using MCA
visualize multiple imputation in PCA and MCA

Details

Package:	missMDA
Type:	Package
Version:	1.11
Date:	2017-03-16
License:	GPL
LazyLoad:	yes

Author(s)

Francois Husson, Julie Josse

Maintainer: <husson@agrocampus-ouest.fr>

References

Josse, J. & Husson, F. (2012). Handling missing values in exploratory multivariate data analysis methods. Journal de la SFdS, 153(2), pp. 79-99.

Julie Josse, Francois Husson (2016). missMDA: A Package for Handling Missing Values in Multivariate Data Analysis. Journal of Statistical Software, 70(1), 1-31. doi:10.18637/jss.v070.i01

Audigier, V., Husson, F., and Josse, J. (2016). Multiple imputation for continuous variables using a bayesian principal component analysis. Journal of Statistical Computation and Simulation, 86(11):2140-2156.

Audigier, V., Husson, F., and Josse, J. (2016). A principal component method to impute missing values for mixed data. Advances in Data Analysis and Classification, 10(1):5-26.

Audigier, V., Husson, F., and Josse, J. (2017). Mimca: multiple imputation for categorical variables with multiple correspondence analysis. Statistics and Computing, 27(2):501-518.

Some videos: https://www.youtube.com/playlist?list=PLnZgp6epRBbQzxFnQrcxg09kRt-PA66T_

`estim_ncpFAMD` *Estimate the number of dimensions for the Factorial Analysis of Mixed Data by cross-validation*

Description

Estimate the number of dimensions for the Factorial Analysis of Mixed Data by cross-validation

Usage

```
estim_ncpFAMD(don, ncp.min=0, ncp.max=5,  method = c("Regularized","EM"),
     method.cv = c("Kfold","loo"), nbsim=100, pNA=0.05, threshold=1e-4,
 verbose = TRUE)
```

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
`method`	"Regularized" by default or "EM"
`method.cv`	"Kfold" for cross-validation or "loo" for leave-one-out
`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
verbose	boolean. TRUE means that a progressbar is writtent

Details

For leave-one-out cross-validation (method.cv="loo"), each cell of the data matrix is alternatively removed and predicted with a FAMD model using ncp.min to ncp.max dimensions. The number of components which leads to the smallest mean square error of prediction (MSEP) is retained. For the Kfold cross-validation (method.cv="Kfold"), pNA percentage of missing values is inserted at random in the data matrix and predicted with a FAMD model using ncp.min to ncp.max dimensions. This process is repeated nbsim times. The number of components which leads to the smallest MSEP is retained. More precisely, for both cross-validation methods, the missing entries are predicted using the imputeFAMD function, it means using it means using the regularized iterative FAMD algorithm (method="Regularized") or the iterative FAMD algorithm (method="EM"). The regularized version is more appropriate to avoid overfitting issues.

Value

ncp	the number of components retained for the FAMD
criterion	the criterion (the MSEP) calculated for each number of components

Author(s)

Vincent Audigier <audigier@agrocampus-ouest.fr>, Francois Husson <husson@agrocampus-ouest.fr> and Julie Josse <Julie.Josse@agrocampus-ouest.fr>

References

Audigier, V., Husson, F. & Josse, J. (2014). A principal components method to impute mixed data. Advances in Data Analysis and Classification

See Also

imputeFAMD

Examples

```
## Not run:
data(ozone)
result <- estim_ncpFAMD(ozone)

## End(Not run)
```

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

Description

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

Usage

```
estim_ncpMCA(don, ncp.min=0, ncp.max=5,  method = c("Regularized","EM"),
     method.cv = c("Kfold","loo"), nbsim=100, pNA=0.05, threshold=1e-4,
 verbose = TRUE)
```

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
method	"Regularized" by default or "EM"
method.cv	"Kfold" for cross-validation or "loo" for leave-one-out
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
verbose	boolean. TRUE means that a progressbar is writtent

Details

For leave-one-out cross-validation (method.cv="loo"), each cell of the data matrix is alternatively removed and predicted with a MCA model using ncp.min to ncp.max dimensions. The number of components which leads to the smallest mean square error of prediction (MSEP) is retained. For the Kfold cross-validation (method.cv="Kfold"), pNA percentage of missing values is inserted at random in the data matrix 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 MSEP is retained. More precisely, for both cross-validation methods, the missing entries are predicted using the imputeMCA function, it means using it means using the regularized iterative MCA algorithm (method="Regularized") or the iterative MCA algorithm (method="EM"). The regularized version is more appropriate to avoid overfitting issues.

Value

ncp	the number of components retained for the MCA
criterion	the criterion (the MSEP) 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, Journal of Clcassification, 29 (1), pp. 91-116.

See Also

imputeMCA

Examples

```
## Not run:
data(vnf)
result <- estim_ncpMCA(vnf,ncp.min=0, ncp.max=5)

## End(Not run)
```

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

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 = c("Regularized","EM"),
       scale = TRUE, method.cv = c("gcv","loo","Kfold"), nbsim = 100,
   pNA = 0.05, threshold=1e-4, verbose = TRUE)
```

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. TRUE implies a same weight for each variable
method.cv	string with the values "gcv" for generalised cross-validation, "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
verbose	boolean. TRUE means that a progressbar is writtent

Details

For leave-one-out (loo) cross-validation, each cell of the data matrix 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 mean square error of prediction (MSEP) is retained. For the Kfold cross-validation, pNA percentage of missing values is inserted and predicted with a PCA model using ncp.min to ncp.max dimensions. This process is repeated nbsim times. The number of components which leads to the smallest MSEP is retained.
For both cross-validation methods, missing entries are predicted using the imputePCA function, it means using the regularized iterative PCA algorithm (method="Regularized") or the iterative PCA algorithm (method="EM"). The regularized version is more appropriate when there are already many missing values in the dataset to avoid overfitting issues.
Cross-validation (especially method.cv="loo") is time-consuming. The generalised cross-validation criterion (method.cv="gcv") can be seen as an approximation of the loo cross-validation criterion which provides a straightforward way to estimate the number of dimensions without resorting to a computationally intensive method.

This argument scale has to be chosen in agreement with the PCA that will be performed. If one wants to perform a normed PCA (where the variables are centered and scaled, i.e. divided by their standard deviation), then the argument scale has to be set to the value TRUE.

Value

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

Author(s)

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

Josse, J. and Husson, F. (2011). Selecting the number of components in PCA using cross-validation approximations. Computational Statistics and Data Analysis. 56 (6), pp. 1869-1879.

See Also

`imputePCA`

Examples

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

## End(Not run)
```

gene *Gene expression*

Description

A data frame with 53 brain tumors of 4 different types defined by the standard World Health Organization (WHO) 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 at the genome level (76 continuous variables for CGH data) as illustrated. 10 rows are missing for the expression data.

Format

A data-frame with 53 rows and 432 continuous variables

Source

de 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.

Examples

```
## Not run:
data(gene)
res.impute <- imputeMFA(gene[,-1], group = c(76,356),
    type = rep("s",2), ncp = 2)
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)
plot(res.mfa, habillage = 1, lab.ind = FALSE)
plot(res.mfa, habillage = "group", invisible = "ind", partial = "all")
plot(res.mfa, habillage = "group", lab.ind = FALSE, partial = "all")
plot(res.mfa, choix = "var", habillage = "group", lab.var = FALSE)
plot(res.mfa, choix = "group", habillage = "group")

## End(Not run)
```

geno *Genotype-environment data set with missing values*

Description

A data-frame with 16 rows corresponding to genotypes (triticale lines) and 10 columns corresponding to different environments where the genotypes were sown. Each cell of the data-frame corresponds to the grain yield (kilograms per hectare) for one genotype in an environment. The first six genotypes correspond to the so-called "complete" type, while the next eight were of the "substituted" type and two check genotypes were included. Such data sets are often incomplete. Indeed, it frequently happens that all the varieties are not assessed on all the environments.

Format

A data-frame with 16 rows and 10 columns

Source

Royo C, Rodriguez A, Romagosa I (1993). Differential adaptation of complete and substitute triticale. Plant Breeding, 111, 113- 119.

Examples

```
## Not run:
data(geno)
ncomp <- estim_ncpPCA(geno)
res.imp <- imputePCA(geno, ncp= ncomp$ncp)
res.pca <- PCA(res.imp$completeObs)

## End(Not run)
```

---

imputeCA *Impute contingency table*

Description

Impute the missing entries of a contingency table using Correspondence Analysis (CA). Can be used as a preliminary step before performing CA on an incomplete dataset.

Usage

```
imputeCA(X, ncp = 2, threshold = 1e-08, maxiter = 1000)
```

Arguments

X	a data.frame that is a contingency table containing missing values
ncp	integer corresponding to the number of dimensions used to predict the missing entries
threshold	the threshold for assessing convergence
maxiter	integer, maximum number of iterations for the regularized iterative CA algorithm

Details

Impute the missing entries of a contingency table using a regularized CA algorithm. The (regularized) iterative CA algorithm first consists in initializing missing values with random initial values. The second step of the (regularized) iterative CA algorithm consists in performing CA on the completed dataset. Then, it imputes the missing values with the (regularized) reconstruction formulae of order ncp (the fitted matrix computed with ncp components for the (regularized) scores and loadings). These steps of estimation of the parameters via CA and imputation of the missing values using the (regularized) fitted matrix are iterate until convergence.
In this regularized algorithm, the singular values of the CA are shrinked.
The number of components ncp used in the algorithm should be small. A small number of components can also be seen as a way to regularize more and consequently may be advices to get more stable predictions.
The output of the algorithm can be used as an input of the CA function of the FactoMineR package in order to perform CA on an incomplete dataset.

Value

The imputed contingency table; the observed values are kept for the non-missing entries and the missing values are replaced by the predicted ones.

Author(s)

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

Examples

```
## Not run:
data(children)

## Impute the indicator matrix and perform a CA
res.impute <- imputeCA(children, ncp=2)
res.ca <- CA(res.impute)

## End(Not run)
```

---

imputeFAMD *Impute mixed dataset*

Description

Impute the missing values of a mixed dataset (with continuous and categorical variables) using the principal component method "factorial analysis for mixed data" (FAMD). Can be used as a preliminary step before performing FAMD on an incomplete dataset.

Usage

```
imputeFAMD(X, ncp = 2, method=c("Regularized","EM"), row.w = NULL,
      coeff.ridge=1,threshold = 1e-06, seed = NULL, maxiter = 1000,...)
```

Arguments

X	a data.frame with continuous and categorical variables containing missing values
ncp	integer corresponding to the number of components used to predict the missing entries
method	"Regularized" by default or "EM"
row.w	row weights (by default, uniform row weights)
coeff.ridge	1 by default to perform the regularized imputeFAMD algorithm; useful only if method="Regularized". Other regularization terms can be implemented by setting the value to less than 1 in order to regularized less (to get closer to the results of the EM method) or more than 1 to regularized more
threshold	the threshold for assessing convergence
seed	integer, by default seed = NULL implies that missing values are initially imputed by the mean of each variable for the continuous variables and by the proportion of the category for the categorical variables coded with indicator matrices of dummy variables. Other values leads to a random initialization
maxiter	integer, maximum number of iteration for the algorithm
...	further arguments passed to or from other methods

Details

Impute the missing entries of a mixed data using the iterative FAMD algorithm (method="EM") or the regularised iterative FAMD algorithm (method="Regularized"). The (regularized) iterative FAMD algorithm first consists in coding the categorical variables using the indicator matrix of dummy variables. Then, in the initialization step, missing values are imputed with initial values such as the mean of the variable for the continuous variables and the proportion of the category for each category using the non-missing entries. If the argument seed is set to a specific value, a random initialization is performed: the initial values are drawn from a gaussian distribution with mean and standard deviation calculated from the observed values for each continuous variable. The second step of the (regularized) iterative FAMD algorithm is to perform FAMD on the completed dataset. Then, it imputes the missing values with the (regularized) reconstruction formulae of order ncp (the fitted matrix computed with ncp components for the (regularized) scores and loadings). These steps of estimation of the parameters via FAMD and imputation of the missing values using the (regularized) fitted matrix are iterate until convergence.

We advice to use the regularized version of the algorithm to avoid the overfitting problems which are very frequent when there are many missing values. In the regularized algorithm, the singular values of the FAMD are shrinked.

The output of the algorithm can be used as an input of the FAMD function of the FactoMineR package in order to perform FAMD on an incomplete dataset.

Value

tab.disj	the imputed matrix; the observed values are kept for the non-missing entries and the missing values are replaced by the predicted ones. The categorical variables

are coded with the indicator matrix of dummy variables. In this indicator matrix, the imputed values are real numbers but they met the constraint that the sum of the entries corresponding to one individual and one variable is equal to one. Consequently they can be seen as degree of membership to the corresponding category

`completeObs` the mixed imputed dataset; the observed values are kept for the non-missing entries and the missing values are replaced by the predicted ones. For the continuous variables, the values are the same as in the tab.disj output; for the categorical variables missing values are imputed with the most plausible categories according to the values in the tab.disj output

`call` the matched call

Author(s)

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

References

Audigier, V., Husson, F. & Josse, J. (2013). A principal components method to impute mixed data. Advances in Data Analysis and Classification, 10(1), 5-26. https://arxiv.org/abs/1301.4797

See Also

imputePCA

Examples

```
## Not run:
data(ozone)
res.impute <- imputeFAMD(ozone, ncp=3)
## The output can be used as an input of the FAMD function of the FactoMineR package
##to perform the FAMD on the incomplete data ozone
require(FactoMineR)
res.afdm <- FAMD(ozone,tab.comp=res.impute$tab.disj)

## End(Not run)
```

---

imputeMCA *Impute categorical dataset*

Description

Impute the missing values of a categorical dataset using Multiple Correspondence Analysis (MCA). Can be used as a preliminary step before performing MCA on an incomplete dataset.

Usage

```
imputeMCA(don, ncp=2, method = c("Regularized","EM"), row.w=NULL, coeff.ridge=1,
    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 predict the missing entries
method	"Regularized" by default or "EM"
row.w	row weights (by default, a vector of 1 for uniform row weights)
coeff.ridge	1 by default to perform the regularized imputeMCA algorithm; useful only if method="Regularized". Other regularization terms can be implemented by setting the value to less than 1 in order to regularized less (to get closer to the results of the EM method) or more than 1 to regularized more (to get closer to the results of the proportion imputation)
threshold	the threshold for assessing convergence
seed	integer, by default seed = NULL implies that missing values are initially imputed by the proportion of the category for the categorical variables coded with indicator matrices of dummy variables. Other values leads to a random initialization
maxiter	integer, maximum number of iterations for the regularized iterative MCA algorithm

Details

Impute the missing entries of a categorical data using the iterative MCA algorithm (method="EM") or the regularised iterative MCA algorithm (method="Regularized"). The (regularized) iterative MCA algorithm first consists in coding the categorical variables using the indicator matrix of dummy variables. Then, in the initialization step, missing values are imputed with initial values such as the proportion of the category for each category using the non-missing entries. This imputation corresponds also to using the algorithm with ncp=0 and is sometimes called in the literature the "missing fuzzy average method". If the argument seed is set to a specific value, a random initialization is performed: random values are drawn in such a way that the constraint that the sum of the entries corresponding to one individual and one variable is equal to one in the indicator matrix of dummy variables. The second step of the (regularized) iterative MCA algorithm consists in performing MCA on the completed dataset. Then, it imputes the missing values with the (regularized) reconstruction formulae of order ncp (the fitted matrix computed with ncp components for the (regularized) scores and loadings). These steps of estimation of the parameters via MCA and imputation of the missing values using the (regularized) fitted matrix are iterate until convergence.
We advice to use the regularized version of the algorithm to avoid the overfitting problems which are very frequent when there are many missing values. In the regularized algorithm, the singular values of the MCA are shrinked.
The number of components ncp used in the algorithm can be selected using the function ncpMCA. A small number of components can also be seen as a way to regularize more and consequently may be advices to get more stable predictions.
The output of the algorithm can be used as an input of the MCA function of the FactoMineR package in order to perform MCA on an incomplete dataset.

Value

tab.disj	The imputed indicator matrix; the observed values are kept for the non-missing entries and the missing values are replaced by the predicted ones. The imputed values are real numbers but they but they met the constraint that the sum of the entries corresponding to one individual and one variable is equal to one. Consequently they can be seen as degree of membership to the corresponding category.
completeObs	The categorical imputed dataset; the observed values are kept for the non-missing entries and the missing values are replaced by the predicted ones. Missing values are imputed with the most plausible categories according to the values in the tab.disj output

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, Journal of Clcassification, 29 (1), pp. 91-116.

See Also

estim_ncpMCA,
Video showing how to perform MCA on an incomplete dataset

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 the indicator matrix and perform a MCA
res.impute <- imputeMCA(vnf, ncp=4)

## The imputed indicator matrix can be used as an input of the MCA function of the
## FactoMineR package to perform the MCA on the incomplete data vnf
res.mca <- MCA(vnf,tab.disj=res.impute$tab.disj)

## With supplementary variables (var 11 to 14), impute the active ones
res.impute <- imputeMCA(vnf[,1:10], ncp=4)
res.mca <- MCA(vnf,tab.disj=res.impute$tab.disj,quali.sup=11:14)

## End(Not run)
```

imputeMFA *Impute dataset with variables structured into groups of variables (groups of continuous or categorical variables)*

Description

Impute the missing values of a dataset with Multiple Factor Analysis (MFA). The variables are structured a priori into groups of variables. The variables can be continuous or categorical but within a group the nature of the variables is the same. Can be used as a preliminary step before performing MFA on an incomplete dataset.

Usage

```
imputeMFA(X, group, ncp = 2, type=rep("s",length(group)), method = c("Regularized","EM"),
      row.w = NULL, coeff.ridge = 1,threshold = 1e-06, seed = NULL, maxiter = 1000, ...)
```

Arguments

X	a data.frame with groups of continuous or categorical variables containing missing values
group	a vector indicating the number of variables in each group
ncp	integer corresponding to the number of components used to predict the missing entries
type	the type of variables in each group; three possibilities: "c" or "s" for continuous variables (for "c" the variables are centered and for "s" variables are scaled to unit variance), "n" for categorical variables
method	"Regularized" by default or "EM"
row.w	row weights (by default, a vector of 1 for uniform row weights)
coeff.ridge	1 by default to perform the regularized imputeMFA algorithm; useful only if method="Regularized". Other regularization terms can be implemented by setting the value to less than 1 in order to regularized less (to get closer to the results of the EM method) or more than 1 to regularized more
threshold	the threshold for assessing convergence
seed	integer, by default seed = NULL implies that missing values are initially imputed by the mean of each variable for the continuous variables and by the proportion of the category for the categorical variables coded with indicator matrices of dummy variables. Other values leads to a random initialization
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 with groups of variables using the iterative MFA algorithm (method="EM") or the regularised iterative MFA algorithm (method="Regularized"). The (regularized) iterative MFA algorithm first consists in coding the categorical variables using the indicator matrix of dummy variables. Then, in the initialization step, missing values are imputed with initial values such as the mean of the variable for the continuous variables and the proportion of the category for each category using the non-missing entries. If the argument seed is set to a specific value, a random initialization is performed: the initial values are drawn from a gaussian distribution with mean and standard deviation calculated from the observed values for each continuous variable. The second step of the (regularized) iterative MFA algorithm is to perform MFA on the completed dataset. Then, it imputes the missing values with the (regularized) reconstruction formulae of order ncp (the fitted matrix computed with ncp components for the (regularized) scores and loadings). These steps of estimation of the parameters via MFA and imputation of the missing values using the (regularized) fitted matrix are iterate until convergence.

We advice to use the regularized version of the algorithm to avoid the overfitting problems which are very frequent when there are many missing values. In the regularized algorithm, the singular values of the MFA are shrinked.

The output of the algorithm can be used as an input of the MFA function of the FactoMineR package in order to perform the MFA on an incomplete dataset.

Value

`tab.disj`	the imputed matrix; the observed values are kept for the non-missing entries and the missing values are replaced by the predicted ones. The categorical variables are coded with the indicator matrix of dummy variables. In this indicator matrix, the imputed values are real numbers but they met the constraint that the sum of the entries corresponding to one individual and one variable is equal to one. Consequently they can be seen as degree of membership to the corresponding category
`completeObs`	the imputed dataset; the observed values are kept for the non-missing entries and the missing values are replaced by the predicted ones. For the continuous variables, the values are the same as in the tab.disj output; for the categorical variables missing values are imputed with the most plausible categories according to the values in the tab.disj output
`call`	the matched call

Author(s)

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

References

F. Husson, J. Josse (2013) Handling missing values in multiple factor analysis. Food Quality and Preferences, 30 (2), 77-85.

See Also

imputePCA

Examples

```
## Not run:
data(orange)
## Impute the data and perform a MFA
## with groups of continuous variables only
res.impute <- imputeMFA(orange, group=c(5,3), type=rep("s",2),ncp=2)
res.mfa <- MFA(res.impute$completeObs,group=c(5,3),type=rep("s",2))

## End(Not run)
## Not run:
data(vnf)
## Impute the indicator matrix and perform a MFA
## with groups of categorical variables only
res.comp <- imputeMFA(vnf,group=c(6,5,3),type=c("n","n","n"),ncp=2)
res.mfa <- MFA(vnf,group=c(6,5,3),type=c("n","n","n"),tab.comp=res.comp)

## End(Not run)
```

---

imputeMultilevel *Impute a multilevel mixed dataset*

Description

Impute the missing values of a multilevel mixed dataset (with a variable that groups the individuals, and with continuous and categorical variables) using the principal component method "multilevel factorial analysis for mixed data".

Usage

```
imputeMultilevel(X, ifac = 1, ncpB = 2, ncpW=2, method=c("Regularized","EM"),
    scale=TRUE, row.w = NULL, threshold = 1e-04, maxiter = 1000,...)
```

Arguments

X	a data.frame with continuous and categorical variables containing missing values
ifac	integer corresponding to the index of the group variable
ncpB	integer corresponding to the number of components used for the between group
ncpW	integer corresponding to the number of components used for the within group
method	"Regularized" by default or "EM"
scale	boolean. By default TRUE leading to a same weight for each variable. This is useful only when all the variables are continuous.

row.w	row weights (by default, uniform row weights)
threshold	the threshold for assessing convergence
maxiter	integer, maximum number of iteration for the algorithm
...	further arguments passed to or from other methods

Details

Impute the missing entries of a multilevel mixed data using the iterative multilevel FAMD algorithm (method="EM") or the regularised iterative multilevel FAMD algorithm (method="Regularized").

We advice to use the regularized version of the algorithm to avoid the overfitting problems which are very frequent when there are many missing values. In the regularized algorithm, the singular values of the FAMD are shrinked.

Value

completeObs	the mixed imputed dataset; the observed values are kept for the non-missing entries and the missing values are replaced by the predicted ones. For the continuous variables, the values are the same as in the tab.disj output; for the categorical variables missing values are imputed with the most plausible categories according to the values in the tab.disj output

Author(s)

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

References

Imputation of mixed data with multilevel singular value decomposition. F. Husson, J. Josse, B. Narasimhan, G. Robin

See Also

imputePCA,imputeFAMD

Examples

```
## Not run:
## Example on artificial data
data(ozone)
res <- imputeMultilevel(ozone, ifac=12, ncpB=2, ncpW=2)

## End(Not run)
```

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 completed dataset.

Usage

```
imputePCA(X, ncp = 2, scale = TRUE, method = c("Regularized","EM"),
      row.w = NULL, coeff.ridge = 1, 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 to predict the missing entries
scale	boolean. By default TRUE leading to a same weight for each variable
method	"Regularized" by default or "EM"
row.w	row weights (by default, a vector of 1 for uniform row weights)
coeff.ridge	1 by default to perform the regularized imputePCA algorithm; useful only if method="Regularized". Other regularization terms can be implemented by setting the value to less than 1 in order to regularized less (to get closer to the results of the EM method) or more than 1 to regularized more (to get closer to the results of the mean imputation)
threshold	the threshold for assessing convergence
seed	integer, by default seed = NULL implies that missing values are initially imputed by the mean of each variable. Other values leads to a random initialization
nb.init	integer corresponding to the number of random initializations; the first initialization is the initialization with the mean imputation
maxiter	integer, maximum number of iteration for the algorithm
...	further arguments passed to or from other methods

Details

Impute the missing entries of a mixed data using the iterative PCA algorithm (method="EM") or the regularised iterative PCA algorithm (method="Regularized"). The (regularized) iterative PCA algorithm first consists imputing missing values with initial values such as the mean of the variable. If the argument seed is set to a specific value, a random initialization is performed: the initial values are drawn from a gaussian distribution with mean and standard deviation calculated from the observed values. nb.init different random initialization can be drawn. In such a situation, the solution

giving the smallest objective function (the mean square error between the fitted matrix and the observed one) is kept. The second step of the (regularized) iterative PCA algorithm is to perform PCA on the completed dataset. Then, it imputes the missing values with the (regularized) reconstruction formulae of order ncp (the fitted matrix computed with ncp components for the (regularized) scores and loadings). These steps of estimation of the parameters via PCA and imputation of the missing values using the (regularized) fitted matrix are iterate until convergence. The iterative PCA algorithm is also known as the EM-PCA algorithm since it corresponds to an EM algorithm of the fixed effect model where the data are generated as a fixed structure (with a low rank representation) corrupted by noise. The number of components used in the algorithm can be found using cross-validation criteria implemented in the function estim_ncpPCA.

We advice to use the regularized version of the algorithm to avoid the overfitting problems which are very frequent when there are many missing values. In the regularized algorithm, the singular values of the PCA are shrinked.

The output of the algorithm can be used as an input of the PCA function of the FactoMineR package in order to perform PCA on an incomplete dataset.

Value

`completeObs`	the imputed dataset; the observed values are kept for the non-missing entries and the missing values are replaced by the predicted ones.
`fittedX`	the reconstructed data

Author(s)

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

References

Josse, J & Husson, F. (2013). Handling missing values in exploratory multivariate data analysis methods. Journal de la SFdS. 153 (2), pp. 79-99.

See Also

`estim_ncpPCA`, `MIPCA`,
Video showing how to perform PCA on an incomplete dataset

Examples

```
## Not run:
data(orange)
## First the number of components has to be chosen
##   (for the imputation step)
## nb <- estim_ncpPCA(orange,ncp.max=5) ## Time consuming, nb = 2

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

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

## End(Not run)
```

MIMCA *Multiple Imputation with MCA*

Description

MIMCA performs multiple imputations for categorical data using Multiple Correspondence Analysis.

Usage

```
MIMCA(X, nboot=100, ncp, coeff.ridge=1, threshold = 1e-06, maxiter = 1000, verbose=FALSE)
```

Arguments

X	a data.frame with categorical variables containing missing values
nboot	the number of imputed datasets
ncp	integer corresponding to the number of components used to reconstruct data with the MCA reconstruction formulae
coeff.ridge	1 by default to perform the regularized imputeMCA algorithm. Other regularization terms can be implemented by setting the value to less than 1 in order to regularized less (to get closer to the results of an EM method) or more than 1 to regularized more (to get closer to the results of the proportion imputation)
threshold	the threshold for assessing convergence for the (regularized) iterative MCA algorithm
maxiter	integer, maximum number of iterations for the (regularized) iterative MCA algorithm
verbose	use verbose=TRUE for screen printing of iteration numbers

Details

MIMCA generates nboot imputed data sets from MCA. The observed values are the same from one dataset to the others whereas the imputed values change. First, nboot weightings are defined for the individuals. Then, the iterative regularized MCA algorithm (Josse, 2012) is applied according to each weighting, leading to nboot imputed tables. These imputed tables are scaled to verify the constraint that the sum is equal to one per variable and per individual. Lastly, missing categories are drawn from the probabilities given by the imputed tables. Thus, nboot imputed categorical data sets are obtained. 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 using a non-parametric bootstrap approach.

Value

res.MI	A list of data frames corresponding to the nboot imputed categorical data sets
res.imputeMCA	A matrix corresponding to the single imputed disjunctive table obtained with the function imputeMCA
call	The matched call

Author(s)

Vincent Audigier <audigier@agrocampus-ouest.fr>, Francois Husson <husson@agrocampus-ouest.fr> and Julie Josse <josse@agrocampus-ouest.fr>

References

Audigier, V., Husson, F., Josse, J. (2015). MIMCA: Multiple imputation for categorical variables with multiple correspondence analysis.

Josse, J., Chavent, M., Liquet, B. and Husson, F. (2010). Handling missing values with Regularized Iterative Multiple Correspondence Analysis, Journal of Classification, 29 (1), pp. 91-116.

Little R.J.A., Rubin D.B. (2002) Statistical Analysis with Missing Data. Wiley series in probability and statistics, New-York.

See Also

imputeMCA,MIPCA,estim_ncpMCA,with.mids,pool,summary.mira

Examples

```
## Not run:
data(TitanicNA)

## First the number of components has to be chosen
##   (for the reconstruction step)
## nb <- estim_ncpMCA(TitanicNA) ## Time-consuming, nb = 5

## Multiple Imputation
res.mi <- MIMCA(TitanicNA, ncp=5, verbose=TRUE)

## First completed data matrix
res.mi$res.MI[[1]]

## Analysis and pooling with mice
require(mice)
imp<-prelim(res.mi,TitanicNA)
fit <- with(data=imp,exp=glm(SURV~CLASS+AGE+SEX,family = "binomial"))
res.pool<-pool(fit)
summary(res.pool)


## End(Not run)
```

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=c("Regularized","EM"), threshold = 1e-04,
    nboot = 100,  method.mi="Boot", Lstart=1000, L=100, verbose=FALSE)
```

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
method.mi	a string. If "Bayes", the uncertainty on the parameters of the imputation model is taken into account using a Bayesian treatment of PCA. By default "Boot" leading to a MI which reflect uncertainty a bootstrap procedure. See details.
Lstart	number of iterations for the burn-in period (only used if method.mi="Bayes")
L	number of skipped iterations to keep one imputed data set after the burn-in period (only used if method.mi="Bayes")
verbose	use verbose=TRUE for screen printing of iteration numbers

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. Two versions are available: multiple imputation using a parametric bootstrap (Josse, J., Husson, F. (2010)) and multiple imputation using a Bayesian treatment of the PCA model (Audigier et al 2015). The methods differ by the way in which the variability due to missing values is reflected. The method used is controlled by the `method.mi` argument. By default, MIPCA uses the parametric bootstrap `method.mi="Boot"`. This bootstrap method is more recommended to evaluate uncertainty in PCA (through confidence ellipses). Otherwise, the Bayesian method can be used by specifying the argument `method.mi="Bayes"`. It is based on an iterative algorithm which alternates imputation of the data set and draw of the PCA parameters in a posterior distribution. These steps are repeated

Lstart times to reach a convergence. Then, one imputed data set is kept each L iterations to ensure independence between imputed values from a data set to another. The Bayesian method is more recommanded to apply a statistical method on an incomplete data set.

Value

res.imputePCA	A matrix corresponding to the imputed dataset obtained with the function imputePCA (the completed dataset)
res.MI	A list of data frames corresponding to the nboot imputed data sets
call	the matched call

Author(s)

Francois Husson <husson@agrocampus-ouest.fr>, Julie Josse <Julie.Josse@agrocampus-ouest.fr> and Vincent Audigier <audigier@agrocampus-ouest.fr>

References

Josse, J., Husson, F. (2011). Multiple Imputation in PCA. Advances in Data Analysis and Classification.

Audigier, V. Josse, J., Husson, F. (2015). Multiple imputation for continuous variables using a Bayesian principal component analysis. Journal of Statistical Computation and Simulation.

Little R.J.A., Rubin D.B. (2002) Statistical Analysis with Missing Data. Wiley series in probability and statistics, New-York.

See Also

imputePCA,plot.MIPCA,Overimpute,MIMCA,with.mids,pool,summary.mira

Examples

```
## Not run:
##########################################################
## Multiple Imputation for visualization on the PCA map
##########################################################

data(orange)
## First the number of components has to be chosen
##   (for the reconstruction step)
nb <- estim_ncpPCA(orange,ncp.max=4)

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

## Visualization on the PCA map
plot(resMI)

##########################################################
## Multiple Imputation for applying statistical methods
(Bayesian method)
```

```
#########################################################
data(ozone)

## First the number of components has to be chosen
nb <- estim_ncpPCA(ozone[,1:11])

## Multiple Imputation with Bayesian method
res.BayesMIPCA<-MIPCA(ozone[,1:11],ncp=2,method.mi="Bayes",verbose=TRUE)

## Regression on the multiply imputed data set and pooling with mice
require(mice)
imp<-prelim(res.mi=res.BayesMIPCA,X=ozone[,1:11])#creating a mids object
fit <- with(data=imp,exp=lm(maxO3~T9+T12+T15+Ne9+Ne12+Ne15+Vx9+Vx12+Vx15+maxO3v))#analysis
res.pool<-pool(fit);summary(res.pool)#pooling

## Diagnostics
res.over<-Overimpute(res.BayesMIPCA)

## End(Not run)
```

---

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

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

## End(Not run)
```

`Overimpute` *Overimputation diagnostic plot*

Description

Assess the fit of the predictive distribution after performing multiple imputation with the function `MIPCA`.

Usage

```
Overimpute(output, plotvars)
```

Arguments

`output`	output from the function `MIPCA`.
`plotvars`	column number of the variable to overimpute.

Details

This function imputes each observed values from each of the parameters of the imputation model obtained from the `MIPCA` procedure. The comparison between the "overimputed" values and the observed values is made by constructing a confidence interval for each observed value using the quantiles of the overimputed values (Blackwell et al. (2015)). Note that confidence intervals constructed with quantiles require a large number of imputations. If the model fits well the data, then the 90% confidence interval should contain the observed value in 90% of the cases. The function `Overimpute` takes as an input the output of the MIPCA function (`output`) and the indices of the variables that are plotted (`plotvars`).

Value

A list of 6-column matrix that contains (1) the row in the original data, (2) the observed value of that observation, (3) the mean of the overimputations, (4) the lower bound of the 90% confidence interval of the overimputations, (5) the upper bound of the 90% confidence interval of the overimputations, and (6) the proportion of the other variables that were missing for that observation in the original data.

References

Blackwell, M., Honaker, J. and King. G. 2015. A Unified Approach to Measurement Error and Missing Data: Overview and Applications. Sociological Methods and Research, 1-39.

See Also

MIPCA

Examples

```
## Not run:
require(Zelig)
data(ozone)

# First the number of components has to be chosen
nb <- estim_ncpPCA(ozone[,1:11])

# Multiple Imputation with Bayesian method
res.BayesMIPCA<-MIPCA(ozone[,1:11],ncp=2,method.mi="Bayes",verbose=T)

# Regression on the multiply imputed data set and pooling
z.out <- zelig(maxO3~., model = "ls", data = res.BayesMIPCA$res.MI,cite=F)
summary(z.out,digits=5)

# Diagnostics
res.over<-Overimpute(res.BayesMIPCA)


## End(Not run)
```

---

ozone — *Daily measurements of meteorological variables and ozone concentration*

Description

This dataset contains 112 daily measurements of meteorological variables (wind speed, temperature, rainfall, etc.) and ozone concentration recorded in Rennes (France) during the summer 2001. There are 11 continuous variables and 2 categorical variables with 2 or 4 levels. Some values are missing.

Usage

```
data(ozone)
```

Format

A data frame with 112 observations on 13 variables.

Source

Cornillon, P.-A., Guyader, A., Husson, F., Jégou, N., Josse, J., Kloareg, M., Matzner-Lober, E., Rouvière, L., (2012). R for Statistics. Chapman & Hall/CRC Computer Science & Data Analysis, Rennes.

Examples

```
## Not run:
data(ozone)
res.comp <- imputeFAMD(ozone, ncp=3)
require(FactoMineR)
res.afdm <- FAMD(ozone,tab.comp=res.comp$tab.disj)

## End(Not run)
```

---

plot.MIMCA *Plot the graphs for the Multiple Imputation in MCA*

Description

From the multiple imputed datasets, the function plots graphs for the individuals, categories and dimensions for the Multiple Correspondance Analysis (MCA)

Usage

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

Arguments

x	an object of class MIMCA
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 MCA, "ind.supp" the projection of the individuals as supplementary individuals, "mod.supp" the projection of the categories
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 MIMCA. The idea is to represent the multiple imputed dataset on a reference configuration (the map obtained from the MCA 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 MCA map; then confidence ellipses are drawn
`mod.supp`	The individuals of the imputed datasets are projected as supplementary individuals onto the reference MCA map, but only categories are plotted; then confidence ellipses are drawn
`ind.proc`	A PCA is performed on each imputed dataset and each configuration of scores is rotated onto the reference MCA 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 MCA dimensions

Author(s)

Audigier Vincent <vincent.audigier@univ-paris-diderot.fr>, Francois Husson <husson@agrocampus-ouest.fr> and Julie Josse <Julie.Josse@agrocampus-ouest.fr>

References

Audigier, V., Husson, F., Josse, J. (2016). MIMCA: Multiple imputation for categorical variables with multiple correspondence analysis

See Also

MIMCA,imputeMCA

Examples

```
## Not run:
data(TitanicNA)

## First the number of components has to be chosen
##   (for the reconstruction step)
## nb <- estim_ncpMCA(TitanicNA) ## Time-consuming, nb = 5

## Multiple Imputation
res.mi <- MIMCA(TitanicNA, ncp=5, verbose=TRUE)

## Plot the graphs
plot(res.mi)
```

```
## 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,imputePCA

Examples

```
## Not run:
data(orange)
## nb <- estim_ncpPCA(orange,ncp.max=5) ## Time consuming, nb = 2
resMI <- MIPCA(orange,ncp=2)
plot(resMI)

## End(Not run)
```

---

prelim *Converts a dataset imputed by MIMCA or MIPCA into a mids object*

Description

This function performs grouping and sorting operations on a multiply imputed dataset. It creates a mids object that is needed for input to with.mids, which allows analyse of the multiply imputed data set. The original incomplete data set needs to be available so that we know where the missing data are.

Usage

```
prelim(res.mi,X)
```

Arguments

res.mi	an output of the functions MIPCA or MIMCA
X	the original incomplete data set corresponding to the res.mi argument

Value

imp.mids An object of type mids

Author(s)

Vincent Audigier <audigier@agrocampus-ouest.fr>, Francois Husson <husson@agrocampus-ouest.fr> and Julie Josse <josse@agrocampus-ouest.fr>

See Also

MIPCA,MIMCA,with.mids,pool,summary.mira

Examples

```
## Not run:
data(TitanicNA)

## First the number of components has to be chosen
##   (for the reconstruction step)
## nb <- estim_ncpMCA(TitanicNA,ncp.max=5) ## Time-consuming, nb = 5

## Multiple Imputation
res.mi <- MIMCA(TitanicNA, ncp=5, verbose=T)

#Analysis
imp<-prelim(res.mi,TitanicNA)
fit <- with(data=imp,exp=glm(SURV~CLASS+AGE+SEX,family = "binomial"))

#Pooling
res.pool<-pool(fit)
summary(res.pool)

## End(Not run)
```

---

snorena *Characterization of people who snore*

Description

This dataset contains 100 individuals and 7 variables (age, weight, size, alcohol, sex, snore, tobacco). There are 4 continuous variables and 3 categorical variables with 2 levels. Some values are missing.

Usage

```
data(snorena)
```

Format

A data frame with 100 observations on 7 variables.

Source

Cornillon, P.-A., Guyader, A., Husson, F., Jegou, N., Josse, J., Kloareg, M., Matzner-Lober, E., Rouviere, L., (2012). R for Statistics. Chapman & Hall/CRC Computer Science & Data Analysis, Rennes.

Examples

```
## Not run:
data(snorena)
res.comp <- imputeFAMD(snorena, ncp=3)
require(FactoMineR)
res.afdm <- FAMD(snorena, tab.comp = res.comp$tab.disj)

## End(Not run)
```

TitanicNA — *Categorical data set with missing values: Survival of passengers on the Titanic*

Description

This data set provides information on the fate of passengers on the fatal maiden voyage of the ocean liner Titanic, summarized according to economic status (class), sex, age and survival. Twenty percent of values are missing completely at random on each variable.

Usage

```
data(TitanicNA)
```

Format

A data frame with 2201 observations on the following 4 variables:

CLASS 0 = crew, 1 = first, 2 = second, 3 = third, denoting the economic status of the subject

AGE 1 = adult, 0 = child, denoting if the subject is an adult or a child

SEX 1 = male, 0 = female, denoting the sex of the subject

SURV 1 = yes, 0 = no, denoting if the passenger lived through the fatal maiden voyage of the ocean liner Titanic

Source

British Board of Trade (1990), Report on the Loss of the Titanic (S.S.). British Board of Trade Inquiry Report (reprint). Gloucester, UK: Allan Sutton Publishing.

Examples

```
data(TitanicNA)
```

---

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. Some values are missing.

Usage

```
data(vnf)
```

Format

A data frame with 1232 observations on 14 categorical variables.

References

Josse, J., Chavent, M., Liquet, B. and Husson, F. (2010). Handling missing values with Regularized Iterative Multiple Correspondence Analysis, Journal of Classification, 29 (1), pp. 91-116.

Examples

```
## Not run:
data(vnf)
tab.disj.impute <- imputeMCA(vnf, ncp=4)$tab.disj
require(FactoMineR)
res.mca <- MCA(vnf,tab.disj=tab.disj.impute)

## End(Not run)
```

Index

∗Topic **Factor analysis**
missMDA-package, 2
∗Topic **datasets**
gene, 8
geno, 8
orange, 25
ozone, 27
snorena, 32
TitanicNA, 33
vnf, 34
∗Topic **dplot**
plot.MIMCA, 28
plot.MIPCA, 30
∗Topic **models**
imputeCA, 9
imputeFAMD, 10
imputeMCA, 12
imputeMFA, 15
imputeMultilevel, 17
imputePCA, 19
∗Topic **multivariate,imputation,categorical,nominal**
prelim, 31
∗Topic **multivariate**
estim_ncpFAMD, 3
estim_ncpMCA, 5
estim_ncpPCA, 6
imputeCA, 9
imputeFAMD, 10
imputeMCA, 12
imputeMFA, 15
imputeMultilevel, 17
imputePCA, 19
MIMCA, 21
MIPCA, 23

estim_ncpFAMD, 3
estim_ncpMCA, 5, *14*, *22*
estim_ncpPCA, 6, *20*
gene, 8
geno, 8

imputeCA, 9
imputeFAMD, *4*, 10, *18*
imputeMCA, *6*, 12, *22*, *29*
imputeMFA, 15
imputeMultilevel, 17
imputePCA, *7*, *12*, *17, 18*, 19, *24*, *31*

MIMCA, 21, *24*, *29*, *32*
MIPCA, *20*, *22*, 23, *27*, *31, 32*
missMDA *(*missMDA-package*)*, 2
missMDA-package, 2

orange, 25
Overimpute, *24*, 26
ozone, 27

plot.MIMCA, 28
plot.MIPCA, *24*, 30
pool, *22*, *24*, *32*
prelim, 31

snorena, 32
summary.mira, *22*, *24*, *32*

TitanicNA, 33

vnf, 34

with.mids, *22*, *24*, *32*