Package ‘miclust’
March 10, 2014

Type Package

Title Multiple imputation in cluster analysis

Version 1.2.5

Date 2014-03-10

Author Jose Barrera-Gómez and Xavier Basagaña

Maintainer Jose Barrera-Gómez <jbarrera@creal.cat>

Depends R (>= 2.14.0), flexclust, combinat, irr, doBy, miscTools

Description This package implements cluster analysis with selection of the final number of clusters and an optional variable selection procedure. The package is designed to integrate the results of multiple imputed datasets while accounting for the uncertainty that the imputations introduce in the final results. In addition, the package can also be used for a cluster analysis of the complete cases of a single database. The package also includes specific functions to summarize and plot the results.

License GPL (>= 2)

R topics documented:

  miclust-package ........................................ 1
gedata .................................................... 2
getVariablesFrequency .................................. 3
miclust .................................................... 5
minhanes ................................................... 8
summary.miclust ......................................... 9

---

miclust-package

Multiple imputation in cluster analysis.

Description

This package implements cluster analysis with selection of the final number of clusters and an optional variable selection procedure. The package is designed to integrate the results of multiple imputed datasets while accounting for the uncertainty that the imputations introduce in the final results. In addition, the package can also be used for a cluster analysis of the complete cases of a single database. The package also includes specific functions to summarize and plot the results.
Details

Package: miclust
Type: Package
Version: 1.2.5
Date: 2014-03-10
License: GPL (>=2)

Author(s)

Jose Barrera-Gómez and Xavier Basagaña.
Maintainer: Jose Barrera-Gómez <jbarrera@creal.cat>

References


gedata                          Function for create a miData object.

Description

This function creates an object of class miData to be clustered by the function miclust.

Usage

gedata(data)

Arguments

<table>
<thead>
<tr>
<th>data</th>
</tr>
</thead>
<tbody>
<tr>
<td>a list or data.frame object. If it is a data frame, it is assumed to contain the raw data, with or without missing data, and without imputations. If it is a list of data frames, it is assumed that the first element contains the raw data and the remaining ones correspond to multiple imputed datasets. Since all variables are considered in the clustering procedure, no identifier variables must be present in the data.</td>
</tr>
</tbody>
</table>

Value

<table>
<thead>
<tr>
<th>rawData</th>
</tr>
</thead>
<tbody>
<tr>
<td>a data frame containing the raw data.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>impData</th>
</tr>
</thead>
<tbody>
<tr>
<td>if data is an object of class list, impData is a list containing the standardized imputed datasets. This standardization is performed by recoding binary variables to 0/1, centering all variables at their mean and standardizing continuous variables by their standard deviation.</td>
</tr>
</tbody>
</table>
getVariablesFrequency

Function to get the ranked selection frequency of the variables.

Description

This function creates the ranked selection frequencies of all the variables that have been selected at least once along the analyzed imputed datasets. The function can be useful for customizing the plot of these frequencies as it is shown in ‘Examples’.

Author(s)

Jose Barrera-Gómez and Xavier Basaganya.

See Also

miclust

Examples

# data minhanes:
data(minhanes)
class(minhanes)

# number of imputed datasets:
length(minhanes) - 1

# raw data with missing values:
summary(minhanes[[1]])

# first imputed dataset:
minhanes[[2]]
summary(minhanes[[2]])

# data preparation for a complete case cluster analysis:
data1 <- getData(minhanes[[1]])
class(data1)
names(data1)

# no imputed datasets:
data1$impData

# data preparation for a multiple imputation cluster analysis:
data2 <- getData(minhanes)
class(data2)
names(data2)

# number of imputed datasets:
length(data2$impData)

# imputed datasets are standardized:
summary(data2$rawData)
summary(data2$impData[[1]])
getVariablesFrequency

Usage

getVariablesFrequency(x, k = NULL)

Arguments

x an object of class miclust obtained with the function miclust.

k the number of clusters. The default value is the optimal number of clusters obtained by the function miclust.

Value

percFreq vector of the selection frequencies (percentage of times) of the variables in decreasing order.

varNames names of the variables.

Author(s)

Jose Barrera-Gómez and Xavier Basagaña.

See Also

miclust

Examples

require(graphics)
data(minhanes)

# Data preparation:
minhanes1 <- getData(data = minhanes)

# Using only the imputations 1 to 10 for the clustering process and exploring 2 vs. 3 clusters:
minhanes1clust <- miclust(data = minhanes1, search = "backward", ks = 2:3, usedImp = 1:10, seed = 4321)

minhanes1clust

# Obtaining the selection frequency of the variables for the optimal number of clusters:
y <- getVariablesFrequency(minhanes1clust)
y

# Plot:
plot(y$percFreq, type = "h", main = "", xlab = "Variable", ylab = "Percentage of times selected", xlim = 0.5 + c(0, length(y$varNames)), lwd = 15, col = "blue", xaxt = "n")
axis(1, at = 1:length(y$varNames), labels = y$varNames)
miclust

*Multiple imputation in cluster analysis.*

**Description**

This function implements cluster analysis with selection of the final number of clusters and an optional variable selection procedure. The function is designed to integrate the results of multiple imputed datasets while accounting for the uncertainty that the imputations introduce in the final results. In addition, the function can also be used for a cluster analysis of the complete cases of a single database. See ‘References’ for further details about the clustering algorithm.

**Usage**

```r
miclust(data, method = "kmeans", search = "none", ks = 2:3, maxVars = NULL, usedImp = "all", distance = "manhattan", centPos = "colMeans", seed = 4321, initCl = "hc")
```

## S3 method for class 'miclust'

- `print(x, ...)`
- `plot(x, k = NULL, ...)`

**Arguments**

- **data**: object of class miData obtained with the function `getData`.
- **method**: clustering method. Currently only "kmeans" is accepted.
- **search**: search algorithm for the selection variable procedure: "backward", "forward" or "none". If "none" (default), no variable selection is performed.
- **ks**: the values of the explored number of clusters. Default is exploring 2 and 3 clusters.
- **maxVars**: if method is "forward", the maximum number of variables to be selected.
- **usedImp**: imputed datasets included in the analysis. Default is "all".
- **distance**: two metrics are allowed to compute distances: "manhattan" (default) and "euclidean".
- **centPos**: position computation of the cluster centroid. If "colMeans" (default) the position of the centroid is computed by the mean. If "colMedians", by the median.
- **seed**: the random number seed. The same seed is used in all the imputations in order to avoid an increase of uncertainty due to the clustering algorithm.
- **initCl**: starting values for the clustering algorithm. If "rand", they are randomly selected; if "hc", they are computed via hierarchical clustering. See ‘Details’.
- **x**: object of class miclust.
- **k**: number of clusters for the final within-cluster summary. Default value is the optimal number of clusters.
- **...**: further arguments for print or plot functions.
Details

The optimal number of clusters and the final set of variables are selected according to CritCF. CritCF is defined as

\[
\left(1 + \frac{1}{2m}\right) \left(1 + \frac{W}{B}\right)^{\frac{1+\log_2(k+1)}{1+\log_2(m+1)}},
\]

where \(m\) is the number of variables, \(k\) is the number of clusters, and \(W\) and \(B\) are the within- and between-cluster inertias. Higher values of CritCF are preferred (Breaban, 2011). See ‘References’ for further details about the clustering algorithm.

For computational reasons, option "rand" is suggested instead "hc" for high dimensional data.

Value

- clustering: a list of lists containing the results of the clustering algorithm for each analyzed dataset and for each analyzed number of clusters. Includes information about selected variables and the cluster vector.
- completeCasesPerc: if data contains a single data frame, percentage of complete cases in data.
- data: input data.
- ks: the values of the explored number of clusters.
- M: number of imputed datasets provided in data.
- usedImp: if data is a list, number of imputed datasets used in the clustering procedure. When summarizing results with the summary function, all available imputations are used.
- kfin: optimal number of clusters.
- CritCF: if data contains a single data frame, CritCF contains the optimal (maximum) value of CritCF (see Details’) and the number of selected variables in the reduction procedure for each explored number of clusters. If data is a list, CritCF contains the optimal value of CritCF for each imputed dataset and for each explored value of the number of clusters.
- NumberOfSelectedVars: number of selected variables.
- selectedkdistribution: if data is a list, frequency of selection of each analyzed number of clusters.
- method: input method.
- search: input search.
- maxVars: input maxVars.
- distance: input distance.
- centPos: input centPos.
- selMetricCent: an object of class kccaFamily needed by the summary function.
- initCl: input initCl.

Author(s)

Jose Barrera-Gómez and Xavier Basagaña.
miclust

References


See Also
gedata for data preparation before using miclust.

Examples
data(minhanes)
help(minhanes)

# Example 1: Multiple imputation clustering process with backward variable selection:
# Data preparation:
minhanes1 <- getData(data = minhanes)

# Using only the imputations 1 to 10 for the clustering process and exploring 2 vs. 3 clusters:
minhanes1clust <- miclust(data = minhanes1, search = "backward", ks = 2:3, usedImp = 1:10, seed = 4321)

minhanes1clust
minhanes1clust$kfin # Optimal number of clusters
plot(minhanes1clust)

# Default summary for the optimal number of clusters:
summary(minhanes1clust)

# Summary forcing 3 clusters:
summary(minhanes1clust, k = 3)

# Example 2: The same analysis without variable selection:
minhanes2clust <- miclust(data = minhanes1, ks = 2:3, usedImp = 1:10, seed = 4321)

minhanes2clust
plot(minhanes2clust)
summary(minhanes2clust)

# Example 3: Complete case clustering process with backward variable selection:
nhanes0 <- getData(data = minhanes[[1]])
nhanes2clust <- miclust(data = nhanes0, search = "backward", ks = 2:3, seed = 4321)
nhanes2clust

# nothing to plot for a single dataset analysis:
plot(nhanes2clust)

summary(nhanes2clust)
minhanes

Multiple imputation for nhanes data

Description

A list with 101 data frames. The first data frame contains nhanes data from mice package. The remaining data frames are datasets obtained by applying the multiple imputation function mice from package mice to the nhanes data.

Usage

data(minhanes)

Format

List of 101 data frames each of them with 25 observations of the following 4 variables:

- age  Age group (1=20-39, 2=40-59, 3=60+)
- bmi  Body mass index (kg/m^2)
- hyp  Hypertensive (1=no, 2=yes)
- chl  Total serum cholesterol (mg/dL)

Source

http://cran.r-project.org/web/packages/mice/index.html

Examples

data(minhanes)
# raw data:
minhanes[[1]]
summary(minhanes[[1]])
# number of imputed datasets:
length(minhanes) - 1
# first imputed dataset:
minhanes[[2]]
summary(minhanes[[2]])
### Description

This function performs a within-cluster descriptive analysis of the variables after the clustering process performed by the function `miclust`.

### Usage

```r
## S3 method for class 'miclust'
summary(object, k = NULL, quantileVars = NULL, ...)
## S3 method for class 'summary.miclust'
print(x, ..., digits = 2)
```

### Arguments

- `object`: an object of class `miclust` obtained by the function `miclust`.
- `x`: an object of class `summary.miclust` obtained by the function `summary.miclust`.
- `k`: the number of clusters. The default value is the optimal number of clusters obtained by `miclust`.
- `quantileVars`: numeric. If a variable selection procedure was used, the cut-off percentile in order to decide the number of selected variables in the variable reduction procedure by decreasing order of presence along the imputations results. The default value is `quantileVars = 0.5`, i.e., the number of selected variables is the median number of selected variables along the imputations.
- `digits`: minimal number of significant digits.
- `...`: further arguments for the summary function.

### Value

- `M`: number of imputations used in the descriptive analysis which is the total number of imputations provided.
- `k`: number of clusters.
- `summaryByCluster`: within-cluster descriptive analysis of the selected variables.
- `search`: search algorithm for the selection variable procedure.
- `cluster`: vector containing the individuals cluster assignments.
- `completeCasesPerc`: if a single dataset have been clustered, the percentage of complete cases in the dataset.
- `selectedVariables`: if a single dataset have been clustered, the selected variables obtained considering `quantileVars`.
- `classMatrix`: if imputations were analyzed, the individual probabilities of cluster assignment.
- `clusterVectors`: if imputations were analyzed, the individual cluster assignment in each imputation.
kappas if imputations were analyzed, the Cohen’s kappa values after comparing the
cluster vector in the first imputation with the cluster vector in each of the re-
maining imputations.

kappaDistribution a summary of kappas.

clustersSize if imputations were analyzed, size of the imputed cluster and between-imputations
summary of the cluster size.

allocationProbabilities if imputations were analyzed, descriptive summary of the probability of cluster
assignment.

summaryByCluster if imputations were analyzed, within-cluster descriptive analysis of the selected
variables.

usedImp imputations used in the clustering procedure.

quantileVars if imputations were analyzed and variable selection was performed, the input
value of quantileVars.

selectedVarsPresence if imputations were analyzed and variable selection was performed, the presence
of the selected variables along imputations.

selectedVariables if imputations were analyzed and variable selection was performed, the names
of the selected variables.

Author(s)

Jose Barrera-Gómez and Xavier Basagaña.

References

Basagaña X, Barrera-Gómez J, Benet M, Antó JM, Garcia-Aymerich J. A framework for multiple

See Also

miclust.