Package 'mifa'

May 10, 2025

Title Multiple Imputation for Exploratory Factor Analysis

Version 0.2.1

URL https://github.com/teebusch/mifa

BugReports https://github.com/teebusch/mifa/issues

Imports stats, mice, dplyr, checkmate

Suggests psych, testthat, knitr, rmarkdown, ggplot2, tidyr, covr

Description Impute the covariance matrix of incomplete data so that factor analysis can be performed. Imputations are made using multiple imputation by Multivariate Imputation with Chained Equations (MICE) and combined with Rubin's rules. Parametric Fieller confidence intervals and nonparametric bootstrap confidence intervals can be obtained for the variance explained by different numbers of principal components. The method is described in Nassiri et al. (2018) <doi:10.3758/s13428-017-1013-4>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Vahid Nassiri [aut],
Anikó Lovik [aut],
Geert Molenberghs [aut],
Geert Verbeke [aut],
Tobias Busch [aut, cre] (ORCID:
<https://orcid.org/0000-0002-8390-7892>)

Maintainer Tobias Busch <teebusch@gmail.com>

Repository CRAN

Date/Publication 2025-05-10 20:30:02 UTC

Contents

`mifa` *Get covariance matrix of incomplete data using multiple imputation*

Description

Compute covariance matrix of incomplete data using multiple imputation. For multiple imputation, *Multivariate Imputation by Chained Equations* (MICE) from the mice package is used. The covariance matrices of the imputed data sets are combined using Rubin's rules.

Usage

```
mifa(
  data,
  cov_vars = dplyr::everything(),
  n_pc,
  ci = FALSE,
  conf = 0.95,
  n_boot = 1000,
  ...
)
```

Arguments

`data`	A data frame with missing values coded as `NA`.
`cov_vars`	Variables in `data` for which to calculate the covariance matrix. Supports (tidy selection)`dplyr::select()`. This allows to select variables that are used for the imputations of missing values, but not the calculations of the covariance matrix. This is especially useful when there are categorical predictors that can improve the imputation of the response variables, but for which covariance cannot be calculated. By default, all variables in `data` are used for both, the imputation and the covariance matrix. Note: Variables and rows used for the imputation, as well as the method for imputation can be configured using the `...`. See also `mice::mice()`.
`n_pc`	Integer or integer vector indicating number of principal components (eigenvectors) for which explained variance (eigenvalues) should be obtained and for which confidence intervals should be computed. Defaults to all principal components, i.e., the number of variables in the data.
`ci`	A character string indicating which types of confidence intervals should be constructed for the variance explained by the principal components. If `"boot"`, `"fieller"`, or `"both"`, the corresponding intervals are computed. If `FALSE` (the default) no confidence intervals will be computed. The components for which confidence intervals should be computed can be set with `n_pc`. See `mifa_ci_boot()` and `mifa_ci_fieller()` for details about the two methods.
`conf`	Confidence level for constructing confidence intervals. The default is `.95` that is, 95% confidence intervals.

n_boot Number of bootstrap samples to use for bootstrapped confidence intervals. The default is 1000.

... Arguments passed on to mice::mice

m Number of multiple imputations. The default is m=5.

method Can be either a single string, or a vector of strings with length length(blocks), specifying the imputation method to be used for each column in data. If specified as a single string, the same method will be used for all blocks. The default imputation method (when no argument is specified) depends on the measurement level of the target column, as regulated by the defaultMethod argument. Columns that need not be imputed have the empty method "". See details.

predictorMatrix A numeric matrix of length(blocks) rows and ncol(data) columns, containing 0/1 data specifying the set of predictors to be used for each target column. Each row corresponds to a variable block, i.e., a set of variables to be imputed. A value of 1 means that the column variable is used as a predictor for the target block (in the rows). By default, the predictorMatrix is a square matrix of ncol(data) rows and columns with all 1's, except for the diagonal. Note: For two-level imputation models (which have "2l" in their names) other codes (e.g, 2 or -2) are also allowed.

ignore A logical vector of nrow(data) elements indicating which rows are ignored when creating the imputation model. The default NULL includes all rows that have an observed value of the variable to imputed. Rows with ignore set to TRUE do not influence the parameters of the imputation model, but are still imputed. We may use the ignore argument to split data into a training set (on which the imputation model is built) and a test set (that does not influence the imputation model estimates). Note: Multivariate imputation methods, like mice.impute.jomoImpute() or mice.impute.panImpute(), do not honour the ignore argument.

where A data frame or matrix with logicals of the same dimensions as data indicating where in the data the imputations should be created. The default, where = is.na(data), specifies that the missing data should be imputed. The where argument may be used to overimpute observed data, or to skip imputations for selected missing values. Note: Imputation methods that generate imptutations outside of mice, like mice.impute.panImpute() may depend on a complete predictor space. In that case, a custom where matrix can not be specified.

blocks List of vectors with variable names per block. List elements may be named to identify blocks. Variables within a block are imputed by a multivariate imputation method (see method argument). By default each variable is placed into its own block, which is effectively fully conditional specification (FCS) by univariate models (variable-by-variable imputation). Only variables whose names appear in blocks are imputed. The relevant columns in the where matrix are set to FALSE of variables that are not block members. A variable may appear in multiple blocks. In that case, it is effectively re-imputed each time that it is visited.

visitSequence A vector of block names of arbitrary length, specifying the

sequence of blocks that are imputed during one iteration of the Gibbs sampler. A block is a collection of variables. All variables that are members of the same block are imputed when the block is visited. A variable that is a member of multiple blocks is re-imputed within the same iteration. The default `visitSequence = "roman"` visits the blocks (left to right) in the order in which they appear in `blocks`. One may also use one of the following keywords: `"arabic"` (right to left), `"monotone"` (ordered low to high proportion of missing data) and `"revmonotone"` (reverse of monotone). *Special case*: If you specify both `visitSequence = "monotone"` and `maxit = 1`, then the procedure will edit the `predictorMatrix` to conform to the monotone pattern. Realize that convergence in one iteration is only guaranteed if the missing data pattern is actually monotone. The procedure does not check this.

`formulas` A named list of formula's, or expressions that can be converted into formula's by `as.formula`. List elements correspond to blocks. The block to which the list element applies is identified by its name, so list names must correspond to block names. The `formulas` argument is an alternative to the `predictorMatrix` argument that allows for more flexibility in specifying imputation models, e.g., for specifying interaction terms.

`blots` A named `list` of `alist`'s that can be used to pass down arguments to lower level imputation function. The entries of element `blots[[blockname]]` are passed down to the function called for block `blockname`.

`post` A vector of strings with length `ncol(data)` specifying expressions as strings. Each string is parsed and executed within the `sampler()` function to post-process imputed values during the iterations. The default is a vector of empty strings, indicating no post-processing. Multivariate (block) imputation methods ignore the `post` parameter.

`defaultMethod` A vector of length 4 containing the default imputation methods for 1) numeric data, 2) factor data with 2 levels, 3) factor data with > 2 unordered levels, and 4) factor data with > 2 ordered levels. By default, the method uses `pmm`, predictive mean matching (numeric data) `logreg`, logistic regression imputation (binary data, factor with 2 levels) `polyreg`, polytomous regression imputation for unordered categorical data (factor > 2 levels) `polr`, proportional odds model for (ordered, > 2 levels).

`maxit` A scalar giving the number of iterations. The default is 5.

`printFlag` If `TRUE`, `mice` will print history on console. Use `print=FALSE` for silent computation.

`seed` An integer that is used as argument by the `set.seed()` for offsetting the random number generator. Default is to leave the random number generator alone.

`data.init` A data frame of the same size and type as `data`, without missing data, used to initialize imputations before the start of the iterative process. The default `NULL` implies that starting imputation are created by a simple random draw from the data. Note that specification of `data.init` will start all `m` Gibbs sampling streams from the same imputation.

Details

The function also computes the variance explained by different numbers of principal components and the corresponding Fieller (parametric) or bootstrap (nonparametric) confidence intervals.

Value

A list:

cov_combined The estimated covariance matrix of the incomplete data, based on the combined covariance matrices of imputed data sets.

cov_imputations A list containing the estimated covariance matrixes for all imputed data sets.

var_explained A data frame containing the estimated proportions of explained variance for each of specified `n_pc` components. Depending o `n ci`, it will also contain the estimated Fieller's (parametric) and/or bootstrap (nonparametric) confidence interval for the proportion of variance explained by the different numbers of principal components defined by `n_pc`.

mids Object of type mice::mids. This is the results of the multiple imputation step for the covariance matrix. Can be useful for diagnosing the multiple imputations.

References

Nassiri, V., Lovik, A., Molenberghs, G., & Verbeke, G. (2018). On using multiple imputation for exploratory factor analysis of incomplete data. Behavioral Research Methods 50, 501–517. doi:10.3758/s1342801710134

See Also

`mifa_ci_boot()`, `mifa_ci_fieller()`, `mice::mice()`

Examples

```
if(requireNamespace("psych")) {
  data <- psych::bfi
  mifa(data, cov_vars = -c(age, education, gender), ci = "fieller", print = FALSE)
}
```

`mifa_ci_boot` *Bootstrap confidence intervals for explained variance*

Description

Compute bootstrap confidence intervals for the proportion of explained variance for the covariance of an incomplete data imputed using multiple imputation. For multiple imputation, Multivariate Imputation by Chained Equations (MICE) from the mice package is used.

Usage

```
mifa_ci_boot(
  data,
  cov_vars = dplyr::everything(),
  n_pc,
  conf = 0.95,
  n_boot = 1000,
  progress = FALSE,
  ...
)
```

Arguments

data	A data frame with missing values coded as NA.
cov_vars	Variables in data for which to calculate the covariance matrix. Supports (tidy selection)dplyr::select(). This allows to select variables that are used for the imputations of missing values, but not the calculations of the covariance matrix. This is especially useful when there are categorical predictors that can improve the imputation of the response variables, but for which covariance cannot be calculated. By default, all variables in data are used for both, the imputation and the covariance matrix. Note: Variables and rows used for the imputation, as well as the method for imputation can be configured using the .... See also mice::mice().
n_pc	Integer or integer vector indicating number of principal components (eigenvectors) for which explained variance (eigenvalues) should be obtained and for which confidence intervals should be computed. Defaults to all principal components, i.e., the number of variables in the data.
conf	Confidence level for constructing confidence intervals. The default is .95 that is, 95% confidence intervals.
n_boot	Number of bootstrap samples to use for bootstrapped confidence intervals. The default is 1000.
progress	Logical. Whether to show progress bars for computation of bootstrap confidence intervals. Default is FALSE.
...	Arguments passed on to mice::mice m Number of multiple imputations. The default is m=5. method Can be either a single string, or a vector of strings with length length(blocks), specifying the imputation method to be used for each column in data. If specified as a single string, the same method will be used for all blocks. The default imputation method (when no argument is specified) depends on the measurement level of the target column, as regulated by the defaultMethod argument. Columns that need not be imputed have the empty method "". See details. predictorMatrix A numeric matrix of length(blocks) rows and ncol(data) columns, containing 0/1 data specifying the set of predictors to be used for each target column. Each row corresponds to a variable block, i.e., a set of variables to be imputed. A value of 1 means that the column variable

is used as a predictor for the target block (in the rows). By default, the predictorMatrix is a square matrix of ncol(data) rows and columns with all 1's, except for the diagonal. Note: For two-level imputation models (which have "2l" in their names) other codes (e.g, 2 or -2) are also allowed.

ignore A logical vector of nrow(data) elements indicating which rows are ignored when creating the imputation model. The default NULL includes all rows that have an observed value of the variable to imputed. Rows with ignore set to TRUE do not influence the parameters of the imputation model, but are still imputed. We may use the ignore argument to split data into a training set (on which the imputation model is built) and a test set (that does not influence the imputation model estimates). Note: Multivariate imputation methods, like mice.impute.jomoImpute() or mice.impute.panImpute(), do not honour the ignore argument.

where A data frame or matrix with logicals of the same dimensions as data indicating where in the data the imputations should be created. The default, where = is.na(data), specifies that the missing data should be imputed. The where argument may be used to overimpute observed data, or to skip imputations for selected missing values. Note: Imputation methods that generate imptutations outside of mice, like mice.impute.panImpute() may depend on a complete predictor space. In that case, a custom where matrix can not be specified.

blocks List of vectors with variable names per block. List elements may be named to identify blocks. Variables within a block are imputed by a multivariate imputation method (see method argument). By default each variable is placed into its own block, which is effectively fully conditional specification (FCS) by univariate models (variable-by-variable imputation). Only variables whose names appear in blocks are imputed. The relevant columns in the where matrix are set to FALSE of variables that are not block members. A variable may appear in multiple blocks. In that case, it is effectively re-imputed each time that it is visited.

visitSequence A vector of block names of arbitrary length, specifying the sequence of blocks that are imputed during one iteration of the Gibbs sampler. A block is a collection of variables. All variables that are members of the same block are imputed when the block is visited. A variable that is a member of multiple blocks is re-imputed within the same iteration. The default visitSequence = "roman" visits the blocks (left to right) in the order in which they appear in blocks. One may also use one of the following keywords: "arabic" (right to left), "monotone" (ordered low to high proportion of missing data) and "revmonotone" (reverse of monotone). *Special case*: If you specify both visitSequence = "monotone" and maxit = 1, then the procedure will edit the predictorMatrix to conform to the monotone pattern. Realize that convergence in one iteration is only guaranteed if the missing data pattern is actually monotone. The procedure does not check this.

formulas A named list of formula's, or expressions that can be converted into formula's by as.formula. List elements correspond to blocks. The block to which the list element applies is identified by its name, so list names must

correspond to block names. The `formulas` argument is an alternative to the `predictorMatrix` argument that allows for more flexibility in specifying imputation models, e.g., for specifying interaction terms.

`blots` A named `list` of `alist`'s that can be used to pass down arguments to lower level imputation function. The entries of element `blots[[blockname]]` are passed down to the function called for block `blockname`.

`post` A vector of strings with length `ncol(data)` specifying expressions as strings. Each string is parsed and executed within the `sampler()` function to post-process imputed values during the iterations. The default is a vector of empty strings, indicating no post-processing. Multivariate (block) imputation methods ignore the `post` parameter.

`defaultMethod` A vector of length 4 containing the default imputation methods for 1) numeric data, 2) factor data with 2 levels, 3) factor data with > 2 unordered levels, and 4) factor data with > 2 ordered levels. By default, the method uses `pmm`, predictive mean matching (numeric data) `logreg`, logistic regression imputation (binary data, factor with 2 levels) `polyreg`, polytomous regression imputation for unordered categorical data (factor > 2 levels) `polr`, proportional odds model for (ordered, > 2 levels).

`maxit` A scalar giving the number of iterations. The default is 5.

`printFlag` If `TRUE`, `mice` will print history on console. Use `print=FALSE` for silent computation.

`seed` An integer that is used as argument by the `set.seed()` for offsetting the random number generator. Default is to leave the random number generator alone.

`data.init` A data frame of the same size and type as `data`, without missing data, used to initialize imputations before the start of the iterative process. The default `NULL` implies that starting imputation are created by a simple random draw from the data. Note that specification of `data.init` will start all `m` Gibbs sampling streams from the same imputation.

Details

This function uses the Shao and Sitter (1996) method to combine multiple imputation and bootstrapping. The imputations are done using `mice::mice()`.

Normally, this function does not need to be called directly. Instead, use `mifa(..., ci = "boot")`.

Value

A data frame containing bootstrapped confidence intervals for variance explained by different number of principal components.

References

Shao, J. & Sitter, R. R. (1996). Bootstrap for imputed survey data. Journal of the American Statistical Association 91.435 (1996): 1278-1288. doi:10.1080/01621459.1996.10476997

See Also

mifa(), mice::mice()

Other mifa confidence intervals: mifa_ci_fieller()

Examples

```
if(requireNamespace("psych")) {
  data <- psych::bfi[, 1:25]
  mifa_ci_boot(data, n_pc = 3:8, n_boot = 10, print = FALSE)
}
```

---

mifa_ci_fieller *Fieller confidence intervals for explained variance*

Description

Computes parametric confidence intervals for proportion of explained variance for given numbers of principal components using Fieller's method. Note that by setting ci = TRUE in mifa(), this confidence interval can be computed as well.

Usage

```
mifa_ci_fieller(cov_imps, n_pc, conf = 0.95, N)
```

Arguments

cov_imps	List containing the estimated covariance matrix within each imputed data. One can use cov_imputations returned by mifa().
n_pc	Integer or integer vector indicating number of principal components (eigenvectors) for which explained variance (eigenvalues) should be obtained and for which confidence intervals should be computed. Defaults to all principal components, i.e., the number of variables in the data.
conf	Confidence level for constructing confidence intervals. The default is .95 that is, 95% confidence intervals.
N	A scalar specifying sample size.

Details

Normally, this function does not need to be called directly. Instead, use mifa(..., ci = "fieller").

Value

A data frame containing confidence intervals for n_pc principal components.

References

Fieller, E. C. (1954). Some problems in interval estimation. Journal of the Royal Statistical Society. Series B (Methodological): 175-185.

See Also

mifa()

Other mifa confidence intervals: mifa_ci_boot()

Examples

```
if(requireNamespace("psych")) {
  data <- psych::bfi[, 1:25]
  mi <- mifa(data, print = FALSE)
  mifa_ci_fieller(mi$cov_imputations, n_pc = 3:8, N = nrow(data))
}
```

Index