Package: Iscores (via r-universe)

June 8, 2026

Type Package

Title Proper Scoring Rules for Missing Value Imputation

Version 1.2.0

Description Provides tools for evaluating and ranking missing value imputation methods using proper scoring rules. Implements the Energy-I-Score and the DR-I-Score for the assessment of deterministic, stochastic and multiple imputation methods for numerical and mixed datasets, following Näf et al. (2022) <doi:10.48550/arXiv.2106.03742> and Näf et al. (2025) <doi:10.48550/arXiv.2507.11297>.

Encoding UTF-8

RoxygenNote 7.3.3

Imports energy, kernlab, pbapply, pbmcapply, ranger, scoringRules, stats

Suggests knitr, mice, rmarkdown, spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

URL https://krystynagrzesiak.github.io/Iscores/

License GPL-3

Language en-US

NeedsCompilation no

Author Krystyna Grzesiak [aut, cre] (ORCID: <https://orcid.org/0000-0003-2581-7722>), Loris Michel [aut, ctb], Meta-Lina Spohn [aut, ctb], Jeffrey Näf [aut, ctb] (ORCID: <https://orcid.org/0000-0003-0920-1899>)

Maintainer Krystyna Grzesiak <krygrz11@gmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-06-08 18:20:25 UTC

RemoteUrl https://github.com/cran/Iscores

RemoteRef HEAD

RemoteSha 861592d82ac3f1888168e271461b605725b5af8f

Contents

`compare_Iscores` *Calculates IScores for multiple imputation functions*

Description

Calculates IScores for multiple imputation functions

Usage

```
compare_Iscores(X, methods_list, score = c("energy_IScore", "DR_IScore"), ...)
```

Arguments

`X`	data containing missing values denoted with NA's.
`methods_list`	a named list of imputing functions.
`score`	a vector of names of scores to calculate. It can be `"energy_IScore"` and `"DR_IScore"`.
`...`	other arguments to be passed to energy_IScore or DR_IScore

Value

a vector of IScores for provided methods

Examples

```
set.seed(111)
X <- random_mcar_data(100, 3, 0.2)
methods_list <- list(exp = exp_imputation,
                     norm = norm_imputation)
compare_Iscores(X, methods_list = methods_list, m = 2,
                n_proj = 10, n_trees_per_proj = 2 )
```

DR_IScore *Compute the imputation KL-based scoring rules*

Description

Compute the imputation KL-based scoring rules

Usage

```
DR_IScore(
  X,
  imputation_func = NULL,
  X_imp = NULL,
  m = 5,
  n_proj = 100,
  n_trees_per_proj = 5,
  min_node_size = 10,
  n_cores = 1,
  projection_function = NULL,
  ...
)
```

Arguments

X	data containing missing values denoted with NA's.
imputation_func	an imputing function. If NULL, please provide imputed datasets X_imp and m.
X_imp	a list of imputed datasets. If NULL it will be obtained using imputation_func.
m	the number of multiple imputations to consider, default to 5.
n_proj	an integer specifying the number of projections to consider for the score.
n_trees_per_proj	an integer, the number of trees per projection.
min_node_size	the minimum number of nodes in a tree.
n_cores	an integer, the number of cores to use.
projection_function	a function providing the user-specific projections.
...	used for compatibility

Value

numeric value of the score obtained for provided imputation method.

References

This method is described in detail in:

Näf, Jeffrey, Meta-Lina Spohn, Loris Michel, and Nicolai Meinshausen. 2022. “Imputation Scores.” https://arxiv.org/abs/2106.03742.

Examples

```
set.seed(111)
X <- random_mcar_data(100, 3, 0.2)
imputation_func <- exp_imputation
DR_IScore(X, imputation_func, m = 2, n_proj = 10, n_trees_per_proj = 2 )
```

---

edistance *Energy distance*

Description

Calculating energy distance/statistic.

Usage

```
edistance(X, X_imp, rescale = FALSE)
```

Arguments

X	a complete original dataset (without missing values).
X_imp	an imputed dataset
rescale	a logical, indicating whether the returned value should be rescaled. Default to FALSE. See "details" section for more information.

Details

This function uses the eqdist.e function. According to this implementation, by default, the function returns the energy statistic which is given by

$$E(X,Y) = \frac{nm}{n+m}\hat{\varepsilon}(X,Y),$$

where $\hat{\varepsilon}(X,Y)$ is the raw energy distance. To obtain raw energy distance use rescale = TRUE.

Value

A numeric value giving the energy distance between the original dataset and the imputed dataset.

Examples

```
X <- matrix(rnorm(100), nrow = 25)
X_imp <- matrix(rnorm(100), nrow = 25)
edistance(X, X_imp)
```

`energy_IScore` *Calculates Imputation Score for imputation function*

Description

Calculates Imputation Score for imputation function

Usage

```
energy_IScore(
  X,
  imputation_func,
  X_imp = NULL,
  multiple = TRUE,
  N = 50,
  max_length = NULL,
  skip_if_needed = TRUE,
  scale = FALSE,
  n_cores = 1,
  silent = TRUE
)
```

Arguments

`X`	data containing missing values denoted with NA's.
`imputation_func`	a function that imputes data.
`X_imp`	imputed dataset of the same size as `X`. It's `NULL` by default meaning that it will be obtained by imputation of `X` using the `imputation_func`.
`multiple`	a logical indicating whether provided imputation method is a multiple imputation approach (i.e. it generates different values to impute for each call). Default to TRUE. Note that if multiple equals to FALSE, N is automatically set to 1.
`N`	a numeric value. Number of samples from imputation distribution H. Default to 50.
`max_length`	Maximum number of variables X_j to consider, can speed up the code. Default to `NULL` meaning that all the columns will be taken under consideration.
`skip_if_needed`	logical, indicating whether some observations should be skipped to obtain complete columns for scoring. If FALSE, NA will be returned for column with no observed variable for training.

scale	a logical value. If TRUE, each variable is scaled in the score.
n_cores	a number of cores for parallelization.
silent	logical indicating whether warnings and messages should be printed.

Details

This function relies on functions energy_Iscore_num and energy_Iscore_cat. Depending on the presence of factor-type data, these functions compute a score either for purely numerical data or for mixed data types.

If you want to compute the score for numerical data, make sure that the dataset does not contain any factor-type variables.

If you want to compute the score for categorical data, ensure that all categorical variables are preserved as factors.

If your imputation method does not support categorical variables represented as factors, implement a wrapper function that handles the appropriate data type conversions before and after imputation.

Value

a numerical value denoting weighted Imputation Score obtained for provided imputation function and a table with scores and weights calculated for particular columns.

References

Näf, J., Grzesiak, K., and Scornet, E. (2025). How to rank imputation methods? arXiv preprint. doi:10.48550/arXiv.2507.11297.

Examples

```
set.seed(111)
X <- random_mcar_data(100, 4)
imputation_func <- exp_imputation
energy_IScore(X, imputation_func)

X <-  random_mcar_mixed_data(100, 4, 2)
imputation_func <- median_mode_imputation
energy_IScore(X, imputation_func)
```

---

exp_imputation *Standard exponential imputation*

Description

Imputes all missing values by independent draws from an exponential distribution with rate 1.

Usage

```
exp_imputation(X_miss)
```

Arguments

X_miss	A data set containing missing values.

Value

A completed data set with all missing values replaced by draws from an Exp(1) distribution.

Examples

```
X <- random_mcar_data(100, 3)
X_imp <- exp_imputation(X)
```

---

median_mode_imputation

Median/mode imputation

Description

Imputes numerical variables using their median and categorical variables using their most frequent observed category.

Usage

```
median_mode_imputation(X_miss)
```

Arguments

X_miss	A data set containing missing values.

Value

A completed data set with all missing values imputed.

Examples

```
X <- random_mcar_mixed_data(100, 3, n_fac = 1)
X_imp <- median_mode_imputation(X)
```

norm_imputation *Standard normal imputation*

Description

Imputes all missing values by independent draws from a standard normal distribution.

Usage

```
norm_imputation(X_miss)
```

Arguments

X_miss	A data set containing missing values.

Value

A completed data set with all missing values replaced by draws from a $N(0, 1)$ distribution.

Examples

```
X <- random_mcar_data(100, 3)
X_imp <- norm_imputation(X)
```

random_mcar_data *Generate random data with MCAR missing values*

Description

Generates a numerical dataset consisting of independent standard normal variables and introduces missing values according to a Missing Completely at Random (MCAR) mechanism.

Usage

```
random_mcar_data(n, p, ratio = 0.2)
```

Arguments

n	Number of observations.
p	Number of numerical variables.
ratio	Proportion of entries to replace with missing values.

Value

A data frame with n rows and p numerical variables containing missing values.

Examples

```
X <- random_mcar_data(100, 3, ratio = 0.2)
head(X)
```

`random_mcar_mixed_data` *Generate random mixed data with MCAR missing values*

Description

Generates a mixed dataset containing independent standard normal variables and categorical variables, then introduces missing values according to a Missing Completely at Random (MCAR) mechanism.

Usage

```
random_mcar_mixed_data(n, p, n_fac = 1, ratio = 0.2)
```

Arguments

`n`	Number of observations.
`p`	Number of numerical variables.
`n_fac`	Number of categorical variables.
`ratio`	Proportion of entries to replace with missing values.

Value

A data frame containing `p` numerical variables and `n_fac` factor variables with missing values.

Examples

```
X <- random_mcar_mixed_data(100, 3, n_fac = 2, ratio = 0.2)
str(X)
```

Index