Package 'missDiag'

July 22, 2025

Title Comparing Observed and Imputed Values under MAR and MCAR

Version 1.0.1

Author Moritz Marbach <moritz.marbach@tamu.edu> [aut, cre]

Maintainer Moritz Marbach <moritz.marbach@tamu.edu>

Description Implements the computation of discrepancy statistics summarizing differences between the density of imputed and observed values and the construction of weights to balance covariates that are part of the missing data mechanism as described in Marbach (2021) <doi:10.48550/arXiv.2107.05427>.

License GPL-3

URL https://github.com/sumtxt/missDiag/

BugReports https://github.com/sumtxt/missDiag/issues

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports Formula (>= 1.2-3), cobalt (>= 4.1.0)

Suggests mice (>= 3.13.0), sbw (>= 1.1.3), ebal (>= 0.1-6)

Depends R (>= 2.10)

NeedsCompilation no

Repository CRAN

Date/Publication 2021-08-06 18:00:02 UTC

Contents

anes08 *11 Variables from the 2008 American National Election Study*

Description

The survey dataset includes 11 variables from the 2008 edition of the American National Election Study.

Usage

```
anes08
```

Format

A data frame with 2265 rows and 11 variables:

age Age of the respondent

female Sex of the respondent

white Non-white vs. white

education No high school, some high school, high school diploma, college

income Low, medium, high.

religion Protestant, Catholic/Orthodox, Atheist/other

married Single, married, no longer married

jobs_r 7 response categories ranging from "Govt should let each person get ahead on own" (1) to "Govt should see to jobs and standard of living" (7)

imp_enviro Whether the respondent sees the environment as an important issue (important/not important)

vote Vote choice in the 2008 presidential election: Obama, McCain, No vote/Other

time Time for the respondent to complete the survey

Details

The dataset `anes` was prepared by Kropko et al. (2014). The nine variables contain different levels of missingness.

The dataset was imputed using random value imputation and predictive mean matching via `mice::mice()` (Version: 3.13.0, seed=42).

The imputed datasets are available in `anes_rng` (random value imputation) and `anes_pmm` (predictive mean matching).

Source

https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/24672

References

J. Kropko, B. Goodrich, A. Gelman, and J. Hill. 2014. Multiple Imputation for Continuous and Categorical Data: Comparing Joint Multivariate Normal and Conditional Approaches, Political Analysis 22(4):497–519.

`missDiag` *Comparing Observed and Imputed Values under MAR and MCAR*

Description

Function to compute discrepancy statistics comparing the (reweighted) density of imputed and observed values. To compute the weights balancing covariates that are part of the missing data mechanism, the function relies on either entropy balancing or stable balancing weights.

Usage

```
missDiag(
  original,
  imputed,
  formula = NULL,
  skip_n = 25,
  scale = FALSE,
  adjust = "ebal",
  use_imputed_cov = TRUE,
  convert = c("numeric"),
  categories = 3,
  verbose = 0,
  output_diag = FALSE,
  ebal_param = param_ebal(),
  sbw_param = param_sbw(),
  cobalt_param = param_cobalt()
)
```

Arguments

`original`	data frame with missing values.
`imputed`	an imputed data frame, a list of imputed data frames or a `mids` object from the `mice::mice()` package.
`formula`	two-sided formula with the variables to compute discrepancy statistics (right-hand side) and variables to balance (left-hand side).
`skip_n`	skip variables with fewer than `skip_n` missing values.
`scale`	scale the design matrix?
`adjust`	either 'none', 'sbw' or 'ebal'.
`use_imputed_cov`	use imputed covariates?

convert	vector of variable types to bin before coercion (ignored if use_imputed_cov=TRUE).
categories	how many bins (ignored if use_imputed_cov=TRUE).
verbose	one of three levels (0,1 or 2).
output_diag	add covariate balance statistics to output?
ebal_param	list of parameters passed to ebal.
sbw_param	list of parameters passed to sbw.
cobalt_param	list of parameters passed to cobalt.

Details

Let y be a variable with observed and imputed values and let m be a corresponding indicator if a value in y is observed (0) or imputed (1). We use X to denote all K covariates and x_k as the k^th covariate.

By default, missDiag computes discrepancy statistics (balance statistics) comparing the distribution of observed and imputed values, ie. comparing f(y|m=1) vs. f(y|m=0). The distribution are expected to be equal under MCAR.

If the data are missing at random (MAR), set adjust="ebal" or adjust="sbw", to construct weights to balance the covariate distribution X before computing the discrepancy statistics.

The left-hand side (lhs) of the formula gives the name of y while the right-hand side (rhs) defines the covariates X to balance under MAR. Rhs variables are transformed into a numeric matrix via the stats::model.matrix() function. Therefore factor and string variables are dummy encoded and functional transformations such as log() are applied.

To run missDiag on many y variables sequentially, supply a list of formulas or name all y variables on the lhs separating each variable with a "|", e.g. y1 | y2 ~ x1 + ... + xK.

If formula=NULL, missDiag runs on all variables with at least n_skip missing values in original. Weights are constructed such that all X covariates are balanced (if adjust!='none').

If use_imputed_cov=FALSE, all imputed values in the covariates X are deleted, variables are coerced to factor variables with missing values encoded as one category. Numerical variables are binned by quantiles before coercion. Use the parameter categories to define the number of bins. By default only variables of class "numeric" are binned. Use convert = c("numeric", "integer") to also bin integer variables before coercion to factor variables.

To balance the covariates, missDiag computes weights using either the ebal::ebalance() function or the sbw::sbw() function. These two packages have to be installed separately by the user.

To pass parameters to the sbw::sbw() or ebal::ebalance() use the functions param_sbw() and param_sbw() to generate valid parameter lists and pass these lists via the parameters sbw_param or ebal_param.

To display information about the computations set verbose to the value 1 (some information) or 2 (more information when constructing weights).

To compute the discrepancy statistics, missDiag relies on the function cobalt::bal.tab() from the cobalt package. To pass parameters to this function use param_cobalt() to generate a valid parameter list and pass this list via the parameter cobalt_param.

Value

A single `data.frame` with the results for each imputed dataset and all lhs variables.

If `output_diag=FALSE`, the dataset only includes the discrepancy statistics comparing y's observed values and imputed values.

If y is continuous, there will be exactly one row per imputed dataset. The discrepancy statistics are reported in the respective columns. For a categorical y cobalt computes discrepancy statistics for all K categories which means that there will be K rows for each imputed dataset in the output. The label for each category is listed in `vname`.

If the default settings are adopted, the dataset includes the following discrepancy statistics for continuous variables:

- `diff_adj`: Standardized mean difference
- `v_ratio_adj`: Variance ratio
- `ks_adj`: Kolmogorov–Smirnov statistic
- `ovl_adj`: 1-Overlap coefficient

No variance ratio is reported for categorical variables.

If `output_diag=TRUE`, the dataset also includes the balance statistics for all K covariates.

These balance statistics are useful to diagnose if the fitted weights are successful in balancing the covariate distribution.

For these diagnostics the column `vname` lists the covariate name (for continuous variables) or the covariate category (for categorical variables).

References

Moritz Marbach. 2021. Choosing Imputation Models.

José R Zubizarreta. 2015. Stable Weights that Balance Covariates for Estimation with Incomplete Outcome Data, Journal of the American Statistical Association, 110(511): 910-922.

Jens Hainmueller. 2012. Entropy Balancing for Causal Effects: A Multivariate Reweighting Method to Produce Balanced Samples in Observational Studies, Political Analysis 20(1): 25–46.

Examples

```
# Compare random value imputation
# with predictive mean imputation
# under MCAR

diag_rng <- missDiag(
 original=anes08,
 imputed=anes08_rng,
 verbose = 1,
 adjust = 'none',
 formula = time ~ .)

diag_pmm <- missDiag(
 original=anes08,
```

```
 imputed=anes08_pmm,
 verbose = 1,
 adjust = 'none',
 formula = time ~ .)

# SMD:

 mean(diag_pmm$diff_adj)
 mean(diag_rng$diff_adj)

# log(Variance ratio)
 mean(log(diag_pmm$v_ratio_adj))
 mean(log(diag_rng$v_ratio_adj))

# KS
 mean(diag_pmm$ks_adj)
 mean(diag_rng$ks_adj)

# 1-OVL
 mean(diag_pmm$ovl_adj)
 mean(diag_rng$ovl_adj)
```

`param_cobalt` *Construct parameter list for cobalt*

Description

Construct parameter list for cobalt

Usage

```
param_cobalt(
  stats = c("m", "v", "ks", "ovl"),
  continuous = "std",
  binary = "std",
  abs = TRUE,
  s.d.denom = "pooled"
)
```

Arguments

`stats`	character; which statistic(s) should be reported.
`continuous`	whether mean differences for continuous variables should be standardized ("std") or raw ("raw").
`binary`	whether mean differences for binary variables (i.e., difference in proportion) should be standardized ("std") or raw ("raw").

abs	logical; whether displayed balance statistics should be in absolute value or not.
s.d.denom	character; how the denominator for standardized mean differences should be calculated, if requested.

Details

For more information about these parameters, see `cobalt::bal.tab()`.

Value

A `list` of parameters that can be passed to `missDiag()`.

`param_ebal` *Construct parameter list for ebalance*

Description

Construct parameter list for ebalance

Usage

```
param_ebal(
  coefs = NULL,
  max.iterations = 200,
  base.weight = NULL,
  constraint.tolerance = 1,
  norm.constant = NULL,
  trim = FALSE,
  max.weight = NULL,
  min.weight = 0,
  max.trim.iterations = 200,
  max.weight.increment = 0.92,
  min.weight.increment = 1.08
)
```

Arguments

coefs	starting values for model coefficients.
max.iterations	maximum number of iterations.
base.weight	vector of base weights.
constraint.tolerance	tolerance level.
norm.constant	An optional normalizing constant.
trim	trim weights via `ebalance.trim`
max.weight	Target for the ratio of the maximum to mean weight.

`min.weight` Target for the ratio of the minimum to mean weight.

`max.trim.iterations`
Maximum number of trimming iterations.

`max.weight.increment`
Increment for iterative trimming of the ratio of the maximum to mean weight.

`min.weight.increment`
Increment for iterative trimming of the ratio of the minimum to mean weight.

Details

For more information about these parameters, see `ebal::ebalance()` and `ebal::ebalance.trim()`.

Value

A `list` of parameters that can be passed to `missDiag()`.

`param_sbw` *Construct parameter list for sbw*

Description

Construct parameter list for sbw

Usage

```
param_sbw(
  bal_alg = TRUE,
  bal_tol = 0,
  bal_std = "group",
  bal_gri = c(0.0001, 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1),
  bal_sam = 1000,
  sol_nam = "quadprog"
)
```

Arguments

`bal_alg`	use tuning algorithm as in Wang and Zubizarreta (2020)?
`bal_tol`	tolerance level.
`bal_std`	tolerances adjustment.
`bal_gri`	grid of values for the tuning algorithm.
`bal_sam`	number of replicates to be used by the tuning algorithm.
`sol_nam`	solver name. Either "cplex", "gurobi", "mosek", "pogs", or "quadprog".

Details

For more information about these parameters, see `sbw::sbw()`.

Value

A `list` of parameters that can be passed to `missDiag()`.

Index