Package 'smdi'

July 17, 2023

Type Package

Title Perform Structural Missing Data Investigations

Version 0.2.2

Description An easy to use implementation of routine structural missing data diagnostics with functions to visualize the proportions of missing observations, investigate missing data patterns and conduct various empirical missing data diagnostic tests. Reference: Weberpals J, Raman SR, Shaw PA, Lee H, Russo M, Hammil BG, Toh D, Connolly JG, Dandreo KJ, Tian F, Liu W, Li Jie, Hernandez-Munos JJ, Glynn RJ, Desai RJ (2023, in submission). ``A Principled Approach to Characterize and Analyze Partially Observed Confounder Data From Electronic Health Records: A Plasmode Simulation Study."

License GPL (>= 3)

URL https://janickweberpals.gitlab-pages.partners.org/smdi

BugReports https://gitlab-scm.partners.org/janickweberpals/smdi/-/issues

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Imports broom, dplyr, fastDummies, forcats, ggplot2, glue, gt, Hotelling, lifecycle, magrittr, methods, mice, naniar, parallel, pROC, randomForest, stringr, survival, tableone, tibble, tidyr, tidyselect

Depends R (>= 2.10)

Suggests gridExtra, gtsummary, here, knitr, reactR, reactable, rmarkdown, simsurv, survminer, usethis, testthat (>= 3.0.0), vdiffr

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Janick Weberpals [aut, cre, cph] (<https://orcid.org/0000-0003-0404-7394>)

Maintainer Janick Weberpals <jweberpals@bwh.harvard.edu>

Repository CRAN

Date/Publication 2023-07-17 14:20:02 UTC

R topics documented:

`smdi_asmd` *Computes mean/median absolute standardized mean differences between observed and missing observations*

Description

This function takes a dataframe with covariates which are partially observed/missing and returns the median/average absolute standardized mean difference (asmd) and more details for every specified covariate in covar (if NULL all covariates with at least one NA are considered).

Important: don't include variables like ID variables, ZIP codes, dates, etc.

Usage

```
smdi_asmd(
  data = NULL,
  covar = NULL,
  median = TRUE,
  includeNA = FALSE,
  n_cores = 1
)
```

Arguments

`data`	dataframe or tibble object with partially observed/missing variables
`covar`	character covariate or covariate vector with partially observed variable/column name(s) to investigate. If NULL, the function automatically includes all columns with at least one missing observation and all remaining covariates will be used as predictors
`median`	logical if the median (= TRUE; recommended default) or mean of all absolute standardized mean differences (asmd) should be computed
`includeNA`	logical, should missingness of other partially observed covariates be explicitly modeled (default is FALSE)
`n_cores`	integer, if >1, computations will be parallelized across amount of cores specified in n_cores (only UNIX systems)

Details

The asmd may be one indicator as to how much patient characteristics differ between patients with and without an observed value for a partially observed covariate. If the median/average asmd is above a certain threshold this may indicate imbalance in patient covariate distributions which may be indicative of the partially observed covariate following a missing at random (MAR) mechanims, i.e. the missingness is explainable by other observed covariates. Similarly, no imbalance between observed covariates may be indicative that missingness cannot be explained with observed covariates and the underlying missingness mechanism may be completely at random (MCAR) or not at random (e.g. missingness is only associated with unobserved factors or through the partially observed covariate itself).

A clear cut-off is hard to determine and analogues to propensity scores, some researchers have proposed that a standardized difference of 0.1 (10 per cent) denotes meaningful imbalance in the baseline covariate.

The asmd is computed for every covariate one-by-one and not jointly. If there is multivariate missingness, i.e. more than just one missing covariate exists, you can decide what should happen with the other partially observed 'predictor' covariates using the includeNA parameter. That is, if includeNA is set to FALSE (default), only the asmd between observed cases will be computed, and if includeNA is set to TRUE, missingness is modeled as an explicit category (categorical covariates only).

If any other behavior is desired, data transformations for example with the `smdi_na_indicator` function, may make sense before calling the function.

The dataframe should generally consist of the exposure variable, the outcome variable(s), the partially observed covariates and all other fully observed covariates which are deemed important for the final modeling and (optionally) which could be considered as auxiliary variables. If no partially observed covariates are provided, the function automatically looks for all variables/columns with NA (powered by the `smdi_summarize` function)

Value

returns an asmd object with average/median absolute standardized mean differences. That is, for each covar, the following outputs are provided:

- asmd_covar: name of covariate investigated
- asmd_table1: detailed "table 1" illustrating distributions and differences of patient characteristics between those without (1) and with (0) observed covariate
- asmd_plot: plot of absolute standardized mean differences (asmd) between patients without (1) and with (0) observed covariate (sorted by asmd)
- asmd_aggregate: average/median absolute standardized mean difference (and min, max) of patient characteristics between those without (1) and with (0) observed covariate

References

Austin PC. Balance diagnostics for comparing the distribution of baseline covariates between treatment groups in propensity-score matched samples. Stat Med. 2009 Nov 10;28(25):3083-107.

Normand SLT, Landrum MB, Guadagnoli E, Ayanian JZ, Ryan TJ, Cleary PD, McNeil BJ. Validating recommendations for coronary angiography following an acute myocardial infarction in the elderly: a matched analysis using propensity scores. Journal of Clinical Epidemiology. 2001;54:387–398.

See Also

CreateTableOne

Examples

```
library(smdi)
library(dplyr)

# S3 print method
asmd <- smdi_asmd(data = smdi_data)
asmd

# let's look at the first variable
# we can check the complete covariate distribution
asmd$pdl1_num$asmd_table1
```

`smdi_check_covar` *This is a utility function to help check input data and covariates provided*

Description

This is a utility function to help check input data and covariates provided

Usage

```
smdi_check_covar(data = NULL, covar = NULL)
```

Arguments

data	dataframe or tibble object with partially observed/missing variables
covar	character covariate or covariate vector with partially observed variable/column name(s) to investigate. If NULL, the function automatically looks for and includes all columns with at least one missing observation

Value

returns the covariate vector for subsequent tasks or warnings/errors

---

smdi_data *smdi exemplary lung cancer dataset*

Description

Example dataset with partially observed covariates.

Usage

```
smdi_data
```

Format

smdi_data:

A data frame with 2,500 rows and 14 columns:

exposure Treatment assignment variable (binary). Indicates initiation of the exposure of interest (1) versus a comparator regimen (0)

age_num Age at baseline in years

female_cat Is gender female (0 = no, 1 = yes)

ecog_cat ECOG performance score at baseline (0 versus 1). Shows 30% missingness following an MCAR mechanism.

smoking_cat Smoking status at baseline (0 = non-smoker, 1 = smoker)

physical_cat Physical activity at baseline (not active versus active)

egfr_cat EGFR mutation status (0 = wild-type, 1 = alteration). Shows 20% missingness following an MAR mechanism.

alk_cat ALK transolcation mutation status (0 = wild-type, 1 = alteration)

pdl1_num PD-L1 cell staining biomarker in %. Shows 40% missingness following an MNAR(value) mechanism

histology_cat Tumor histology (0 = nonsquamous, 1 = squamous)

ses_cat Socio-economic status (multi-categorical: 1-low, 2-middle, 3-high)

copd_cat COPD comorbidity at baseline

eventtime time to censoring event

status event indicator at time t; 0 = censored, 1 = deceased

Source

https://janickweberpals.gitlab-pages.partners.org/smdi/articles/data_generation.html

---

`smdi_data_complete` *smdi exemplary lung cancer dataset (with complete data)*

Description

Example dataset with complete cases.

Usage

```
smdi_data_complete
```

Format

`smdi_data_complete`**:**

A data frame with 2,500 rows and 14 columns:

exposure Treatment assignment variable (binary). Indicates initiation of the exposure of interest (1) versus a comparator regimen (0)

age_num Age at baseline in years

female_cat Is gender female (0 = no, 1 = yes)

ecog_cat ECOG performance score at baseline (0 versus 1)

smoking_cat Smoking status at baseline (0 = non-smoker, 1 = smoker)

physical_cat Physical activity at baseline (not active versus active)

egfr_cat EGFR mutation status (0 = wild-type, 1 = alteration)

alk_cat ALK transolcation mutation status (0 = wild-type, 1 = alteration)

pdl1_num PD-L1 cell staining biomarker in %

histology_cat Tumor histology (0 = nonsquamous, 1 = squamous)

ses_cat Socio-economic status (multi-categorical: 1-low, 2-middle, 3-high)

copd_cat COPD comorbidity at baseline

eventtime time to censoring event

status event indicator at time t; 0 = censored, 1 = deceased ...

Source

https://janickweberpals.gitlab-pages.partners.org/smdi/articles/data_generation.html

`smdi_diagnose` *Computes three group missing data summary diagnostics*

Description

This function bundles and calls all three group diagnostics and returns the most important summary metrics. For more information and details, please refer to the individual functions.

Important: don't include variables like ID variables, ZIP codes, dates, etc.

Usage

```
smdi_diagnose(
  data = NULL,
  covar = NULL,
  median = TRUE,
  includeNA = FALSE,
  train_test_ratio = c(0.7, 0.3),
  set_seed = 42,
  ntree = 1000,
  n_cores = 1,
  model = c("logistic", "linear", "cox"),
  form_lhs = NULL,
  exponentiated = FALSE
)
```

Arguments

`data`	dataframe or tibble object with partially observed/missing variables
`covar`	character covariate or covariate vector with partially observed variable/column name(s) to investigate. If NULL, the function automatically includes all columns with at least one missing observation and all remaining covariates will be used as predictors
`median`	logical if the median (= TRUE; recommended default) or mean of all absolute standardized mean differences (asmd) should be computed (smdi_asmd())
`includeNA`	logical, should missingness of other partially observed covariates be explicitly modeled for computation of absolute standardized mean differences (default is FALSE)
`train_test_ratio`	numeric vector to indicate the test/train split ratio for random forest missingness prediction model, e.g. c(.7, .3) is the default
`set_seed`	seed for reproducibility of random forest missingness prediction model, defaults to 42
`ntree`	integer, number of trees for random forest missingness prediction model (defaults to 1000 trees)

n_cores	integer, if >1, computations will be parallelized across amount of cores specified in n_cores (only UNIX systems)
model	character describing which outcome model to fit to assess the association between covar missingness indicator and outcome. Currently supported are models of type logistic, linear and cox (see smdi_outcome)
form_lhs	string specifying the left-hand side of the outcome formula (see smdi_outcome)
exponentiated	logical, should results of outcome regression to assess association between missingness and outcome be exponentiated (default is FALSE)

Details

Wrapper for individual diagnostics function.

Value

smdi object including a summary table of all three smdi group diagnostics:

Group 1 diagnostic:

- asmd_mean/median: average/median absolute standardized mean difference (and min, max) of patient characteristics between those without (1) and with (0) observed covariate
- hotteling_p: p-value of hotelling test. Rejecting the H0 means that Hotelling's test detects a significant difference in the distribution between patients without (1) and with (0) the observed covariate

Group 2 diagnostic:

- rf_auc: The area under the receiver operating curve (AUC) as a measure of the ability to predict the missingness of the partially observed covariate

Group 3 diagnostic:

- estimate_univariate: univariate association between missingness indicator of covar and outcome
- estimate_adjusted: association between missingness indicator of covar and outcome conditional on other fully observed covariates and missing indicator variables of other partially observed covariates

References

TBD

See Also

smdi_asmd smdi_hotelling smdi_little smdi_rf smdi_outcome

Examples

```
library(smdi)

smdi_diagnose(
 data = smdi_data,
 covar = "egfr_cat",
 model = "cox",
 form_lhs = "Surv(eventtime, status)"
 )
```

`smdi_hotelling` *Computes hotelling's multivariate t-test*

Description

Hotelling's multivariate t-test, which examines variable differences conditional on having an observed covariate value or not. As the power of statistical hypothesis tests can be influenced by sample size, the combined investigation along with smdi_asmd() is highly recommended.

Important: don't include variables like ID variables, ZIP codes, dates, etc.

Usage

```
smdi_hotelling(data = NULL, covar = NULL, n_cores = 1)
```

Arguments

`data`	dataframe or tibble object with partially observed/missing variables
`covar`	character covariate or covariate vector with partially observed variable/column name(s) to investigate. If NULL, the function automatically includes all columns with at least one missing observation and all remaining covariates will be used as predictors
`n_cores`	integer, if >1, computations will be parallelized across amount of cores specified in n_cores (only UNIX systems)

Details

CAVE: Hotelling's and Little's show high susceptibility with large sample sizes and it is recommended to always interpret the results along with the other diagnostics.

Value

returns a hotelling object with statistics on hotellings test by covariate. That is, for each covar, the following outputs are provided:

- stats: hotelling test statistics (for more information see `hotelling.test`)
- pval: p-value of hotelling test

References

Hotelling H. The Generalization of Student's Ratio. Ann Math Stat. 1931;2(3):360-378.

See Also

hotelling.test

Examples

```
library(smdi)

smdi_hotelling(data = smdi_data)
```

`smdi_little` *Computes Little's test*

Description

Little's chi-squared test takes into account possible patterns of missingness across all variables in the dataset. Rejection of the null hypothesis of this test would provide sufficient evidence to indicate that the data are (globally) not MCAR. Please note that compared to smdi_hotelling, this function tests for MCAR globally across all missing covariates.

#' #' Important: don't include variables like ID variables, ZIP codes, dates, etc.

Usage

```
smdi_little(data = NULL)
```

Arguments

`data`	dataframe or tibble object with partially observed/missing variables

Details

CAVE: Hotelling's and Little's show high susceptibility with large sample sizes and it is recommended to always interpret the results along with the other diagnostics.

Value

returns a little object with statistics on little's test globally.

References

Little RJA. A Test of Missing Completely at Random for Multivariate Data with Missing Values. J Am Stat Assoc. 1988;83(404):1198-1202.

See Also

mcar_test

Examples

```
library(smdi)
library(dplyr)

smdi_data %>%
 smdi_little()
```

`smdi_na_indicator` *Create binary missing indicator variables by two different strategies*

Description

This function takes a dataframe and creates binary missing indicator variable. This can be realized with two different approaches:

Approach 1 (drop_NA_col = FALSE): creates a binary missing indicator variable for partially observed variables and retains both original and indicator variables.

Approach 2 (drop_NA_col = TRUE): creates a binary missing indicator variable for partially observed variables and only retains indicator variables (and drops the original variables).

Important: Make sure you have your variables format correct and avoid to include variables like ID variables, ZIP codes, dates, etc.

Usage

```
smdi_na_indicator(data = NULL, covar = NULL, drop_NA_col = TRUE)
```

Arguments

`data`	dataframe or tibble object with partially observed/missing variables
`covar`	character covariate or covariate vector with partially observed variable/column name(s) to investigate. If NULL, the function automatically includes all columns with at least one missing observation.
`drop_NA_col`	logical, drop specified columns with NA (default) or retain those columns

Value

returns the dataframe with missing indicator variables (column names are ending on "_NA")

Examples

```
library(smdi)
library(dplyr)

smdi_data %>%
  smdi_na_indicator(drop_NA_col = FALSE) %>%
  glimpse()

smdi_data %>%
  smdi_na_indicator(drop_NA_col = TRUE) %>%
  glimpse()
```

`smdi_outcome` *Computes association between missingness and outcome*

Description

This function fits outcome models with a covariate missingness indicator(s) of the covariates specified with *covar*. The estimates are computed by univariate and adjusted models on all other prognostic covariates in the dataset. Based on the underlying missingness mechanism, the estimate for the covariate missingness indicator may indicate a meaningful difference in the outcome between patients with vs w/o the observed confounder conditional on other covariates that could explain that difference.

Important: don't include variables like ID variables, ZIP codes, dates, etc.

Usage

```
smdi_outcome(
  data = NULL,
  covar = NULL,
  model = c("logistic", "linear", "cox"),
  form_lhs = NULL,
  exponentiated = FALSE,
  n_cores = 1
)
```

Arguments

`data`	dataframe or tibble object with partially observed/missing variables
`covar`	character covariate or covariate vector with partially observed variable/column name(s) to investigate. If NULL, the function automatically includes all columns with at least one missing observation and all remaining covariates will be used as predictors
`model`	character describing which outcome model to fit to assess the association between covar missingness indicator and outcome. Currently supported are models of type logistic, linear and cox

form_lhs	string specifying the left-hand side of the outcome formula (see details)
exponentiated	logical, should results be exponentiated (default is FALSE)
n_cores	integer, if >1, computations will be parallelized across amount of cores specified in n_cores (only UNIX systems)

Details

The function automatically fits a univariate and adjusted outcome model. The currently supported models are logistic (glm), linear (lm) and cox (survival). For adjusted models, the function uses all available covariates found in the dataset specified with the data parameter. If covariates should not be include in the outcome model, these covariates should be dropped beforehand (as with all other functions in the smdi package).

The left-hand side of the formula (form_lhs) needs to specify the outcome in one of the following ways:

- glm (binary): character of column name with binary outcome, e.g. "MACE"
- lm (continuous): character of column name with binary outcome, e.g. "WEIGHT_LOSS"
- cox (time-to-event): LHS specifying time-to-event outcome, e.g. "Surv(TIME, STATUS)"

Value

returns a tibble with univariate and adjusted estimates for each partially observed covar:

- estimate_univariate: univariate association between missingness indicator of covar and outcome
- estimate_adjusted: association between missingness indicator of covar and outcome conditional on other fully observed covariates and missing indicator variables of other partially observed covariates

References

...

Examples

```
library(smdi)

smdi_outcome(
 data = smdi_data,
 model = "cox",
 form_lhs = "Surv(eventtime, status)"
)
```

smdi_rf *Computes random forest-based AUC*

Description

The function trains and fits a random forest model to assess the ability to predict missingness for the specified covariate(s). If missing indicator can be predicted as a function of observed covariates, MAR may be a likely scenario and would imply that imputation may be feasible.

Important: don't include variables like ID variables, ZIP codes, dates, etc.

Usage

```
smdi_rf(
  data = NULL,
  covar = NULL,
  train_test_ratio = c(0.7, 0.3),
  set_seed = 42,
  ntree = 1000,
  n_cores = 1
)
```

Arguments

data	dataframe or tibble object with partially observed/missing variables
covar	character covariate or covariate vector with partially observed variable/column name(s) to investigate. If NULL, the function automatically includes all columns with at least one missing observation and all remaining covariates will be used as predictors
train_test_ratio	numeric vector to indicate the test/train split ratio, e.g. c(.7, .3) which is the default
set_seed	seed for reproducibility, defaults to 42
ntree	integer, number of trees (defaults to 1000 trees)
n_cores	integer, if >1, computations will be parallelized across amount of cores specified in n_cores (only UNIX systems)

Details

The random forest utilizes the randomForest engine.

CAVE: If the missingness indicator variables of other partially observed covariates (indicated by suffix _NA) have an extremely high variable importance (combined with an unusually high AUC), this might be an indicator of a monotone missing data pattern. In this case it is advisable to exclude other partially observed covariates and run missingness diagnostics separately.

Value

returns an rf object which comes as a list that contains the ROC AUC value and corresponding variable importance in training dataset (latter as ggplot object). That is, for each covar, the following outputs are provided:

- rf_table: The area under the receiver operating curve (AUC) as a measure of the ability to predict the missingness of the partially observed covariate
- rf_plot: ggplot object illustrating the variable importance for the prediction made expressed by the mean decrease in accuracy per predictor. That is how much would the accuracy of the prediction (# of correct predictions/Total # of predictions made) decrease, had we left out this specific predictor.

References

Sondhi A, Weberpals J, Yerram P, Jiang C, Taylor M, Samant M, Cherng S. A systematic approach towards missing lab data in electronic health records: A case study in non-small cell lung cancer and multiple myeloma. CPT Pharmacometrics Syst Pharmacol. 2023 Jun 15. <doi: 10.1002/psp4.12998.> Epub ahead of print. PMID: 37322818.

See Also

randomForest

Examples

```
library(smdi)

smdi_rf(data = smdi_data, covar = "ecog_cat")
```

---

smdi_style_gt *Takes an object of class smdi and styles it to a publication-ready gt table*

Description

This function takes either an object of class smdi or data.frame or tibble as input and styles it to a publication-ready table based on the gt package. The output is of class gt and can take further gt-based arguments for customization.

Usage

```
smdi_style_gt(
  smdi_object = NULL,
  include_little = TRUE,
  font_size = 13,
  tbl_width = 800
)
```

Arguments

smdi_object	object of class "smdi" or data.frame/tibble
include_little	can be logical (TRUE/FALSE) for displaying Little's p-value that is part of an "smdi" object or a separate object of class "little"
font_size	integer to determine table font size
tbl_width	integer to determine table width

Details

[Experimental]

Value

returns a formatted gt table object

See Also

gt

Examples

```
library(smdi)
library(dplyr)

smdi_diagnose(
  data = smdi_data,
  covar = "egfr_cat",
  model = "cox",
  form_lhs = "Surv(eventtime, status)"
  ) %>%
smdi_style_gt()
```

---

smdi_summarize *Utility helper to give a light summary of partially observed covariates*

Description

This function takes a dataframe and automatically returns the amount and proportion of missing for partially observed covariates assuming a one-row-per-patient dataframe. This is an important utility function for other functions in this package. Results can also be stratified by another variable in which case the proportion missing refers to the amount of patients in the respective stratum.

Usage

```
smdi_summarize(data = NULL, covar = NULL, strata = NULL)
```

Arguments

data	dataframe or tibble object with partially observed/missing variables. Assumes a a one-row-per-patient format.
covar	character covariate or covariate vector with partially observed variable/column name(s) to investigate. If NULL, the function automatically includes all columns with at least one missing observation.
strata	character name of variable/column by which results should be stratified

Value

returns count and proportion of missing values. If strata is specified, the returned proportion refers to the amount of patients in the respective stratum.

Examples

```
library(smdi)

smdi_vis(data = smdi_data)
```

---

smdi_vis *Quick ggplot2 barchart visualization of partially observed/missing variables*

Description

This function takes a dataframe and outputs a nicely formatted ggplot2 vertical barchart plot that visualizes the proportion missing for a given variable (vector) or all existent missing variables. Results can also be stratified by another variable in which case the proportion missing refers to the amount of patients in the respective stratum.

Important: Function assumes the data is in a one-row-per-patient format.

Usage

```
smdi_vis(data = NULL, covar = NULL, strata = NULL)
```

Arguments

data	dataframe or tibble object with partially observed/missing variables. Assumes a a one-row-per-patient format
covar	character covariate or covariate vector with partially observed variable/column name(s) to investigate. If NULL, the function automatically includes all columns with at least one missing observation
strata	character name of variable/column by which results should be stratified

Value

returns ggplot2 graph displaying selected or automatically identified variables by percent missing

Examples

```
library(smdi)

smdi_vis(data = smdi_data)
```

Index