Package: midoc (via r-universe)

October 13, 2026

Title A Decision-Making System for Multiple Imputation

Version 1.0.0.9000

Description A guidance system for analysis with missing data. It incorporates expert, up-to-date methodology to help researchers choose the most appropriate analysis approach when some data are missing. You provide the available data and the assumed causal structure, including the likely causes of missing data. 'midoc' will advise which analysis approaches can be used, and how best to perform them. 'midoc' follows the framework for the treatment and reporting of missing data in observational studies (TARMOS). Lee et al (2021). <doi:10.1016/j.jclinepi.2021.01.008>.

License MIT + file LICENSE

Encoding UTF-8

Suggests knitr, testthat

Imports arm, blorr, dagitty, ggplot2, grDevices, gridExtra, mfp2, mice (>= 3.16.0), RefBasedMI, rlang, rmarkdown, shiny, stats, table1, utils

VignetteBuilder knitr

Depends R (>= 3.6)

LazyData true

URL https://elliecurnow.github.io/midoc/, https://github.com/elliecurnow/midoc/, https://mrcieu.r-universe.dev/midoc

BugReports https://github.com/elliecurnow/midoc/issues/

Config/roxygen2/markdown TRUE

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev libnode-dev libx11-dev zlib1g-dev

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

Date/Publication 2026-10-13 16:21:58 UTC

RemoteUrl https://github.com/elliecurnow/midoc

RemoteRef HEAD

RemoteSha af23d4f4965d935abcb4e64a6d60cdbfcf472e0e

Contents

adr 2
asrds 3
bmi 3
checkCRA 4
checkMI 5
checkModSpec 6
descMissData 8
doMImice 9
doMNARMImice 10
doRefBasedMI 11
exploreDAG 13
midocShinyApp 14
midocVignette 15
proposeMI 15
qol 17
summMissData 17

Index **19**

`adr` *Administrative data*

Description

A simulated dataset

Usage

```
adr
```

Format

`adr`**:**

A data frame with 1000 rows and 5 columns:

gcse_score Child's GCSE score (at approximately age 16 years), as a percentage, calculated from the total of an individual's top eight GCSE qualifications, ranked in terms of points

log_income Logarithm of the family income per annum, in UK pounds, when the child is aged 5 years

mated Mother's educational level: post-16 years qualification or not

ks2_score Child's Key Stage 2 score (at approximately age 11 years), as a percentage, calculated from the total of an individual's English, maths and science scores

r_cra Missingness indicator: complete record (case) or not

`asrds` *Adapted Self-Report Delinquency Scale data*

Description

A simulated dataset

Usage

```
asrds
```

Format

`asrds`**:**

A data frame with 1000 rows and 5 columns:

asrds17 Child's Adapted Self-Report Delinquency Scale (ASRDS) value at age 17 years, as a percentage of the maximum possible value of the square-root of ASRDS

log_income5 Logarithm of the family income per annum, in UK pounds, when the child is aged 5 years

mated Mother's educational level at time of pregnancy: post-16 years qualification or not

asrds14 Child's ASRDS value at age 14 years, as a percentage of the maximum possible value of the square-root of ASRDS

r_cra Missingness indicator: complete record (case) or not

`bmi` *Child body mass index data*

Description

A simulated dataset

Usage

```
bmi
```

Format

`bmi` **A data frame with 1000 rows and 6 columns::**

bmi7 Child's body mass index at age 7 years

matage Mother's age at first pregnancy, standardised relative to a mean age of 30 years

mated Mother's educational level: post-16 years qualification or not

pregsize Mother's pregnancy size: singleton or twins

bwt Child's birth weight in kilograms

r Missingness indicator: complete record (case) or not

---

checkCRA *Inspect complete records analysis model*

Description

Check complete records analysis is valid under the proposed analysis model and directed acyclic graph (DAG). Validity means that the proposed approach will allow unbiased estimation of the estimand(s) of interest. In typical use this includes regression parameters, covariate associations, and causal effects.

Usage

```
checkCRA(y, covs = NULL, r_cra, mdag)
```

Arguments

`y`	The analysis model outcome variable(s), specified as a string (space delimited) or a list
`covs`	Optional analysis model covariate(s), specified as a string (space delimited) or a list
`r_cra`	The complete record indicator, specified as a string
`mdag`	The DAG, specified as a string using dagitty syntax, or as a dagitty graph object

Details

The DAG should include all observed and unobserved variables related to the analysis model variables and their missingness, as well as the complete record ("missingness") indicator.

In general, complete records analysis of the conditional relationship between the analysis model outcome and the specified covariates is valid if the outcome and complete record indicator are unrelated, given the specified covariates. This is determined using the proposed DAG by checking whether the analysis model outcome(s) and complete record indicator are 'd-separated', conditional on the covariates.

Value

A message indicating whether complete records analysis is valid under the proposed DAG and analysis model outcome and covariate(s)

References

Hughes R, Heron J, Sterne J, Tilling K. 2019. Accounting for missing data in statistical analyses: multiple imputation is not always the answer. Int J Epidemiol. doi:10.1093/ije/dyz032

Bartlett JW, Harel O, Carpenter JR. 2015. Asymptotically Unbiased Estimation of Exposure Odds Ratios in Complete Records Logistic Regression. Am J Epidemiol. doi:10.1093/aje/kwv114

Examples

```
# Example DAG for which complete records analysis is not valid
checkCRA(y="bmi7", covs="matage", r_cra="r",
         mdag="matage -> bmi7 mated -> matage mated -> bmi7
               sep_unmeas -> mated sep_unmeas -> r")
# For the DAG in the example above, complete records analysis is valid
## if a different set of covariates is used
checkCRA(y="bmi7", covs="matage mated", r_cra="r",
         mdag="matage -> bmi7 mated -> matage mated -> bmi7
               sep_unmeas -> mated sep_unmeas -> r")
# Example where complete records analysis is never valid
checkCRA(y="bmi7", covs="matage mated", r_cra="r",
         mdag="matage -> bmi7 mated -> matage mated -> bmi7
               sep_unmeas -> mated sep_unmeas -> r bmi7 -> r")
```

---

checkMI — *Inspect multiple imputation model*

Description

Check multiple imputation is valid under the proposed imputation model(s) and directed acyclic graph (DAG). Validity means that the proposed approach will allow unbiased estimation of the estimand(s) of interest. In typical use this includes regression parameters, covariate associations, and causal effects.

Usage

```
checkMI(dep, preds = NULL, r_cra, mdag)
```

Arguments

dep	The partially observed variable(s) to be imputed, specified as a string (space delimited) or a list
preds	Optional fully observed imputation model predictor(s), specified as a string (space delimited) or a list
r_cra	The complete record indicator, specified as a string
mdag	The DAG, specified as a string using dagitty syntax, or as a dagitty graph object

Details

Imputation model(s) should include all other analysis model variables as predictors, as well as any auxiliary variables. The DAG should include all observed and unobserved variables related to the analysis model variables and their missingness, as well as the complete record ("missingness") indicator.

In principle, multiple imputation is valid if all partially observed variables are unrelated to missingness, given the (fully observed) imputation model predictors. This is determined using the proposed DAG by checking whether all the partially observed variables are 'd-separated' from the complete record indicator, conditional on the imputation model predictors. It is assumed that all the specified imputation model predictors are fully observed and will be used to impute all the specified partially observed variables.

Value

A message indicating whether multiple imputation is valid under the proposed DAG and imputation model predictor(s)

References

Curnow E, Tilling K, Heron JE, Cornish RP, Carpenter JR. 2023. Multiple imputation of missing data under missing at random: including a collider as an auxiliary variable in the imputation model can induce bias. Frontiers in Epidemiology. doi:10.3389/fepid.2023.1237447

Examples

```
# Example DAG for which multiple imputation is valid, because collider
## variable 'bwt' is not included as a predictor
checkMI(dep="bmi7", preds="matage mated pregsize", r_cra="r",
        mdag="matage -> bmi7 mated -> matage mated -> bmi7
              sep_unmeas -> mated sep_unmeas -> r pregsize -> bmi7
              pregsize -> bwt sep_unmeas -> bwt")

# Example DAG for which multiple imputation is not valid
checkMI(dep="bmi7", preds="matage", r_cra="r",
        mdag="matage -> bmi7 mated -> matage mated -> bmi7
               sep_unmeas -> mated sep_unmeas -> r")
```

---

checkModSpec *Specify and inspect parametric model specification*

Description

Specify a parametric model (which may represent the analysis or imputation model). Optionally, if a dataset is supplied, explore whether the observed relationships in the specified dataset are consistent with the proposed parametric model.

Usage

```
checkModSpec(
  formula,
  family,
  by = NULL,
  data = NULL,
  plot = TRUE,
  message = TRUE
)
```

Arguments

formula	A symbolic description of the model to be fitted, with the dependent variable on the left of a ~ operator, and the covariates, separated by + operators, on the right, specified as a string
family	A description of the error distribution and link function to be used in the model, specified as a string; family functions that are supported are "gaussian(identity)" and "binomial(logit)"
by	Optional stratification variable(s), specified as a string (space delimited) or a list of factors; if specified, the parametric model will be fit for each subset of the data determined by the values of the factor(s)
data	Optionally, a data frame containing all the variables stated in the formula and if specified, stratification variable(s)
plot	If TRUE (the default), and a dataset is supplied, displays a plot which can be used to explore the form of the specified model; note that stratification variables are ignored in the plot; use plot = FALSE to disable the plot
message	If TRUE (the default), and a dataset is supplied, displays a message indicating whether the relationships between the dependent variable and covariates are likely to be correctly specified or not; use message = FALSE to suppress the message

Value

An object of type 'mimod' (a list containing the specified formula, family, and, if specified, dataset name). Optionally, if required and a dataset is supplied, a message indicating whether the relationships between the dependent variable and covariates are likely to be correctly specified or not.

References

Curnow E, Carpenter JR, Heron JE, et al. 2023. Multiple imputation of missing data under missing at random: compatible imputation models are not sufficient to avoid bias if they are mis-specified. J Clin Epidemiol. doi:10.1016/j.jclinepi.2023.06.011

Examples

```
# Example (incorrectly) assuming a linear relationship
checkModSpec(formula="bmi7~matage+mated+pregsize",
             family="gaussian(identity)", data=bmi)
```

```
  ## For the example above, (correctly) assuming a quadratic relationship
checkModSpec(formula="bmi7~matage+I(matage^2)+mated+pregsize",
             family="gaussian(identity)", data=bmi)
```

`descMissData` *Lists missing data patterns in the specified dataset*

Description

Summarises the missing data patterns in the specified dataset. Each row in the output corresponds to a missing data pattern (1=observed, 0=missing). The number and percentage of observations is also displayed for each missing data pattern. The first column indicates the number of missing data patterns. The second column refers to the analysis model outcome ('y'), with all other variables ('covs') displayed in subsequent columns. Alternatively, 'y' can indicate the primary variable of interest, e.g. 'y' could refer to an exposure or intervention, with all other variables listed in 'covs'.

Usage

```
descMissData(y, data, covs = NULL, by = NULL, plot = FALSE)
```

Arguments

`y`	The analysis model outcome variable(s), specified as a string (space delimited) or a list
`data`	A data frame containing the specified analysis model outcome, covariate(s), and if specified, stratification variable(s)
`covs`	Optional analysis model covariate(s), specified as a string (space delimited) or a list
`by`	Optional stratification variable(s), specified as a string (space delimited) or a list of factors; if specified, the data are subsetted by the values of the factor(s) and missing data patterns are displayed for each subset in turn; can only be used when the total number of variables listed in 'y' and 'covs' is greater than one
`plot`	If TRUE, displays a plot using md.pattern to visualise the missing data patterns; if stratification variable(s) are specified, a separate plot will be displayed for each subset; use plot = FALSE (the default) to disable the plot

Value

A summary of the missing data patterns

Examples

```
descMissData(y="bmi7", covs="matage mated", data=bmi)
descMissData(y="bmi7", covs="matage mated bwt", by="pregsize", data=bmi)
```

doMImice *Performs multiple imputation*

Description

Creates multiple imputations using mice, based on the options and dataset specified by a call to proposeMI. Further arguments can be passed to this function - see the help file for mice for more information. If stratification variable(s) are included in the 'miprop' object, multiple imputation will be performed for each subset of the data determined by the values of the stratification variable(s) and the resulting imputed datasets will be combined. If a substantive model is specified, the pooled estimates are calculated using pool.

Usage

```
doMImice(mipropobj, seed, substmod = NULL, message = TRUE, ...)
```

Arguments

mipropobj	An object of type 'miprop', created by a call to 'proposeMI'
seed	An integer that is used to set the seed of the 'mice' call
substmod	Optionally, a symbolic description of the substantive model to be fitted, specified as a string; if supplied, the model will be fitted to each imputed dataset and the results pooled
message	If TRUE (the default), displays a message summarising the analysis that has been performed; use message = FALSE to suppress the message
...	Further arguments passed to mice

Value

A 'mice' object of class 'mids' (the multiply imputed datasets). Optionally, a message summarising the analysis that has been performed.

Examples

```
# First specify the imputation model as a 'mimod' object
## (suppressing the message)
mimod_bmi7 <- checkModSpec(formula="bmi7~matage+I(matage^2)+mated+pregsize",
                           family="gaussian(identity)",
                           data=bmi,
                           message=FALSE)
# Save the proposed 'mice' options as a 'miprop' object
## (suppressing the message)
miprop <- proposeMI(mimodobj=mimod_bmi7,
                    data=bmi,
                    message=FALSE,
                    plot = FALSE)
# Create the set of imputed datasets using the proposed 'mice' options
```

```
imp <- doMImice(miprop,123)

# Additionally, fit the substantive model to each imputed dataset and display
## the pooled results
doMImice(miprop, 123, substmod="lm(bmi7 ~ matage + I(matage^2) + mated)")
```

---

doMNARMImice *Performs not-at-random multiple imputation*

Description

Creates multiple imputations using mice. Imputations are based on the options and dataset specified by a call to proposeMI, and additionally on the specified missing not at random (MNAR) mechanism. Further arguments can be passed to this function - see the help file for mice for more information. If stratification variable(s) are included in the 'miprop' object, multiple imputation will be performed for each subset of the data determined by the values of the stratification variable(s) and the resulting imputed datasets will be combined. If a substantive model is specified, the pooled estimates are calculated using pool.

Usage

```
doMNARMImice(
  mipropobj,
  mnardep,
  mnardelta,
  seed,
  substmod = NULL,
  message = TRUE,
  ...
)
```

Arguments

mipropobj	An object of type 'miprop', created by a call to 'proposeMI'
mnardep	The partially observed variable to be imputed under MNAR, specified as a string
mnardelta	The desired sensitivity (delta) parameter as a function of other variables and values, specified as a string
seed	An integer that is used to set the seed of the 'mice' call
substmod	Optionally, a symbolic description of the substantive model to be fitted, specified as a string; if supplied, the model will be fitted to each imputed dataset and the results pooled
message	If TRUE (the default), displays a message summarising the analysis that has been performed; use message = FALSE to suppress the message
...	Further arguments passed to mice

Details

Imputation is performed using the NARFCS procedure (Tompsett et al, 2018) for the specified variable. See mice.impute.mnar.logreg for further details. All other partially observed variables are assumed to be missing at random (MAR) and imputed using the method(s) specified as per the 'miprop' object.

Value

A 'mice' object of class 'mids' (the multiply imputed datasets). Optionally, a message summarising the analysis that has been performed.

References

Tompsett D, Leacy F, Moreno-Betancur M, Heron J, & White IR. 2018. On the use of the not-at-random fully conditional specification (NARFCS) procedure in practice. Statistics in Medicine. doi:10.1002/sim.7643

Examples

```
# First specify the imputation model as a 'mimod' object
## (suppressing the message)
mimod_bmi7 <- checkModSpec(formula="bmi7~matage+I(matage^2)+mated+pregsize",
                           family="gaussian(identity)",
                           data=bmi,
                           message=FALSE)
# Save the proposed 'mice' options as a 'miprop' object
## (suppressing the message)
miprop <- proposeMI(mimodobj=mimod_bmi7,
                    data=bmi,
                    message=FALSE,
                    plot = FALSE)
# Create the set of imputed datasets using the proposed 'mice' options and
# desired MNAR mechanism (decreasing imputed values of bmi7 by 2 units)
imp <- doMNARMImice(mipropobj=miprop, mnardep="bmi7", mnardelta="-2", seed=123)

# Additionally, fit the substantive model to each imputed dataset and display
## the pooled results
doMNARMImice(mipropobj=miprop, mnardep="bmi7", mnardelta="-2", seed=123,
             substmod="lm(bmi7 ~ matage + I(matage^2) + mated)")
```

---

doRefBasedMI *Performs reference-based multiple imputation*

Description

Creates multiple imputations under reference-based multiple imputation using RefBasedMI. Imputations are based on the dataset and relevant options specified by a call to proposeMI. If a substantive model is specified, the pooled estimates are also calculated using pool. The dataset is assumed to be

in 'wide' format, with one row per subject. It is assumed that the outcome is measured repeatedly over time, with at least one measurement at an intermediate time-point (i.e. between baseline and study end-point). Data are assumed to be multivariate normal within each category of the grouping variable (in typical use, this denotes the treatment allocation).

Usage

```
doRefBasedMI(
  mipropobj,
  y,
  groupvar,
  covs = NULL,
  idvar = NULL,
  method,
  reference,
  seed,
  substmod = NULL,
  message = TRUE
)
```

Arguments

`mipropobj`	An object of type 'miprop', created by a call to 'proposeMI'
`y`	The analysis model outcome variables (at least two are required), specified as a string (space delimited) or a list
`groupvar`	Group variable; can be numeric or string
`covs`	Optional analysis model covariate(s), specified as a string (space delimited) or a list; a maximum of five covariates can be specified
`idvar`	Optional participant identifier variable; if not provided, an identifier variable, named 'id', will be automatically created
`method`	Reference-based imputation method; methods that are supported are "J2R", "CR", and "CIR"
`reference`	Reference group for the specified method; can be numeric or string
`seed`	An integer that is used to set the seed of the 'mice' call
`substmod`	Optionally, a symbolic description of the substantive model to be fitted, specified as a string; if supplied, the model will be fitted to each imputed dataset and the results pooled
`message`	If TRUE (the default), displays a message summarising the analysis that has been performed; use message = FALSE to suppress the message

Details

Reference-based multiple imputation uses observed data from one category of the grouping variable - the 'reference' group - to impute missing values in other categories. Available reference-based methods are 'jump-to-reference' (J2R), 'copy reference' (CR), and 'copy increments in reference' (CIR). J2R assumes that the distribution of outcomes for individuals who drop out 'jumps to' the

distribution observed in the reference group following their last observed time point. CR assumes individuals who drop out behave as if they are in the specified reference group for the full duration of the trial. CIR assumes that the distribution of outcomes for individuals who drop out follows the mean increments observed in the reference group, following their last observed time point.

Value

A 'mice' object of class 'mids' (the multiply imputed datasets). Optionally, a message summarising the analysis that has been performed.

Examples

```
# First specify the imputation model as a 'mimod'object
## (suppressing the message)
mimod_qol12 <- checkModSpec(formula="qol12 ~ factor(group) + age0 + qol0 +
  qol3", family="gaussian(identity)", data=qol, message=FALSE)

# Save the proposed 'mice' options as a 'miprop' object
## (suppressing the message)
miprop_qol12 <- proposeMI(mimodobj=mimod_qol12, data=qol, message=FALSE,
  plot = FALSE)

# Create the set of imputed datasets using the proposed mice' options and
## specified reference-based imputation method; then fit the substantive
## model to each imputed dataset and display the pooled results
doRefBasedMI(mipropobj=miprop_qol12, y="qol3 qol12", groupvar="group",
 covs="age0 qol0", idvar="id", method="J2R", reference=1, seed=123,
 substmod = "lm(qol12 ~ factor(group) + age0 + qol0)")
```

`exploreDAG` *Compares data with proposed DAG*

Description

Explore the relationships implied by the proposed directed acyclic graph (DAG). Optionally, if a dataset is supplied, explore whether relationships between fully observed variables in the specified dataset are consistent with the proposed DAG.

Usage

```
exploreDAG(mdag, data = NULL, message = TRUE)
```

Arguments

`mdag`	The DAG, specified as a string using dagitty syntax, or as a dagitty graph object
`data`	Optionally, a data frame containing all the variables stated in the DAG. All ordinal variables must be integer-coded and all categorical variables must be dummy-coded.

`message` If TRUE (the default), displays a message describing the exploratory analysis performed; use message = FALSE to suppress the message

Value

A message listing the pairs of variables that are implied to be independent (possibly conditional on other variables) by the proposed DAG. Optionally, if a dataset is supplied, a message indicating whether the relationships between fully observed variables in the specified dataset are consistent with the proposed DAG.

Examples

```
exploreDAG(mdag="matage -> bmi7 mated -> matage mated -> bmi7
                 sep_unmeas -> mated sep_unmeas -> r",
           data=bmi)
```

`midocShinyApp` *Run a browser-based version of the midoc package*

Description

Runs a browser-based version of the midoc package. In this version, you can explore midoc functions using exemplar datasets or apply midoc functions using your own DAG and data.

Usage

```
midocShinyApp()
```

Value

A browser-based version of the midoc package

Examples

```
# Run the browser-based version of the midoc package
midocShinyApp()
```

midocVignette — *Run an interactive vignette for the midoc package*

Description

Runs an interactive version of the midoc vignette: Multiple Imputation DOCtor (midoc). In the interactive version, you can apply midoc functions in shiny-package apps using your own DAG and data.

Usage

```
midocVignette()
```

Value

A browser-based, interactive version of the midoc vignette

Examples

```
# Run the interactive vignette
midocVignette()
```

proposeMI — *Suggests multiple imputation options*

Description

Suggests the mice options to perform multiple imputation, based on the set of imputation models (one for each partially observed variable) specified by calls to checkModSpec and the proportion of complete records. Optionally, if a dataset is supplied, diagnostic plots are created based on the proposed 'mice' options.

Usage

```
proposeMI(
  mimodobj,
  prop_complete = NA,
  data = NULL,
  plot = TRUE,
  plotprompt = TRUE,
  message = TRUE
)
```

Arguments

mimodobj	An object, or list of objects, of type 'mimod', which stands for 'multiple imputation model', created by a call to checkModSpec
prop_complete	Optionally, the proportion of complete records, specified as a number between 0 and 1. This is only required if a dataset is not specified. If a dataset is specified, the proportion of complete records will be calculated based on all columns included in the dataset.
data	Optionally, a data frame containing all the variables required for imputation and the substantive analysis; if stratification variable(s) are included in the 'mimod' object(s), these will be carried over to 'midoc' functions 'doMImice' and 'doMNARMImice' and multiple imputation will be performed for each subset of the data determined by the values of the stratification variable(s)
plot	If TRUE (the default), and a dataset is supplied, displays diagnostic plots for the proposed 'mice' call; use plot=FALSE to disable the plots
plotprompt	If TRUE (the default), and a dataset is supplied, the user is prompted before each subsequent plot is displayed; use plotprompt=FALSE to remove the prompt and display all plots at the same time
message	If TRUE (the default), displays a message describing the proposed 'mice' options; use message=FALSE to suppress the message

Value

An object of type 'miprop', which can be used to run 'mice' using the proposed options, plus optionally, a message and, if a dataset is supplied, diagnostic plots describing the proposed 'mice' options

Examples

```
# First specify each imputation model as a 'mimod' object
## (suppressing the message)
mimod_bmi7 <- checkModSpec(formula="bmi7~matage+I(matage^2)+mated+pregsize",
                           family="gaussian(identity)",
                           data=bmi,
                           message=FALSE)
mimod_pregsize <- checkModSpec(
                           formula="pregsize~bmi7+matage+I(matage^2)+mated",
                           family="binomial(logit)",
                           data=bmi,
                           message=FALSE)

# Display the proposed 'mice' options
## When specifying a single imputation model
proposeMI(mimodobj=mimod_bmi7,
                    data=bmi)
## When specifying more than one imputation model (suppressing the plots)
proposeMI(mimodobj=list(mimod_bmi7,mimod_pregsize),
                     data=bmi,
                     plot=FALSE)
```

qol — *Randomised controlled trial data*

Description

A simulated dataset

Usage

```
qol
```

Format

`qol`:

A data frame with 1000 rows and 6 columns:

group Randomisation group: 1 = Placebo, 2 = Active treatment

age0 Participant's age at randomisation (baseline), in years

qol0 Participant's quality of life at randomisation (baseline), measured using the EuroQol Visual Analogue Scale (EQ-VAS)

qol3 Participant's quality of life at 3 months post-randomisation, measured using EQ-VAS

qol12 Participant's quality of life at 12 months post-randomisation, measured using EQ-VAS

r Missingness indicator: complete record (case) or not

summMissData — *Summarise the data distribution for complete and incomplete records*

Description

Produces a table of descriptive statistics for the analysis model outcome ('y') and covariate ('covs') variables, stratified by the complete record indicator ('r_cra') and optionally, by other stratification variable(s). Alternatively, 'y' can indicate the primary variable of interest, e.g. 'y' could refer to an exposure or intervention, with all other variables listed in 'covs'. The table displays the percentage of missing values for each variable, plus the mean and standard deviation of numeric variables, and frequency and percentage of categorical variables. Optional plots display box plots for numeric variables and stacked bar charts for categorical variables.

Usage

```
summMissData(
  y,
  r_cra,
  data,
  covs = NULL,
  by = NULL,
```

```
  plot = TRUE,
  plotprompt = TRUE,
  message = TRUE,
  ...
)
```

Arguments

y	The analysis model outcome variable(s), specified as a string (space delimited) or a list
r_cra	The complete record indicator variable, which must be a factor
data	A data frame containing the specified analysis model outcome, covariate(s), and if specified, stratification variable(s)
covs	Optional analysis model covariate(s), specified as a string (space delimited) or a list
by	Optional additional stratification variable(s); if specified, the data are subsetted according to both the complete record indicator and the values of the stratification variable(s) and summary data are displayed for each subset in turn
plot	If TRUE (the default), summary plots are displayed; note that stratification variables are ignored in the plot; use plot = FALSE to disable the plots
plotprompt	If TRUE (the default), the user is prompted before each plot is displayed; use plotprompt=FALSE to remove the prompt
message	If TRUE (the default), displays a table of descriptive statistics comparing complete and incomplete records; use message = FALSE to suppress the message
...	Further arguments passed to table1

Details

The summary table is created using the table1 function. Further arguments can be passed to this function - see the help file for table1 for more information. Note that the 'transpose' feature is not available. The summary table and plots can be used to compare the characteristics of complete and incomplete records. Note that there may be differences in variable distributions between complete and incomplete records if data are missing at random or missing not at random. Therefore, any observed differences cannot be used as evidence for or against the data missing at random assumption.

Value

A summary of descriptive statistics comparing complete and incomplete records and, optionally, descriptive plots.

Examples

```
summMissData(y="bmi7", r_cra="r", data=bmi, covs="matage mated bwt pregsize",
              rowlabelhead="Complete")
```

Index

∗ **datasets**
    adr, 2
    asrds, 3
    bmi, 3
    qol, 17

adr, 2
asrds, 3

bmi, 3

checkCRA, 4
checkMI, 5
checkModSpec, 6, *15, 16*

dagitty, *4, 5, 13*
descMissData, 8
doMImice, 9
doMNARMImice, 10
doRefBasedMI, 11

exploreDAG, 13

md.pattern, *8*
mice, *9, 10, 15*
mice.impute.mnar.logreg, *11*
midocShinyApp, 14
midocVignette, 15

pool, *9–11*
proposeMI, *9–11*, 15

qol, 17

RefBasedMI, *11*

shiny-package, *15*
summMissData, 17

table1, *18*