Package 'manMCMedMiss'

<div align="center">January 6, 2023</div>

Title Monte Carlo Confidence Intervals for the Indirect Effect with
Missing Data

Version 0.9.2

Description Research compendium for the manuscript
Monte Carlo Confidence Intervals for the Indirect Effect with Missing Data
containing scripts used in the simulation study and other supplementary materials.

URL https://github.com/jeksterslab/manMCMedMiss,
https://jeksterslab.github.io/manMCMedMiss/

BugReports https://github.com/jeksterslab/manMCMedMiss/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

Depends R (>= 3.5.0)

Imports stats, ggplot2, MASS, mice, party

Suggests knitr, rmarkdown, testthat, DT, microbenchmark, semmcci

RoxygenNote 7.2.3

NeedsCompilation no

Author Ivan Jacob Agaloos Pesigan [aut, cre, cph]
(<https://orcid.org/0000-0003-4818-8420>),
Shu Fai Cheung [ctb] (<https://orcid.org/0000-0002-9871-9448>)

Maintainer Ivan Jacob Agaloos Pesigan <r.jeksterslab@gmail.com>

R topics documented:

AmputeData   *Ampute Data*

Description

The function generates a data set with missing values for data generated by `GenData` using the multivariate amputation approach. See `mice::ampute()` for more details.

Usage

```
AmputeData(data_complete, mech = "MAR", prop = 0.3, patterns = NULL)
```

Arguments

data_complete	Numeric matrix. Output of the `GenData` function or a three-column data set with complete data.
mech	Missing data mechanism.
prop	Proportion of missing data.
patterns	Numeric matrix consisting of zeroes and ones. Each row in the matrix represents a missing data pattern where 0 indicates a missing observation. If `patterns = NULL`, the default value is all possible missing data patterns for a data set with 3 columns.

Value

Returns a dataframe.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Schouten, R. M., Lugtig, P. and Vink, G. (2018). Generating missing values for simulation purposes: A multivariate amputation procedure. *Journal of Statistical Computation and Simulation*, *88*(15), 1909–1930. doi:10.1080/00949655.2018.1491577

See Also

Other Data Generation Functions: GenData(), ImputeData()

Check   *Check Replication*

Description

Check Replication

Usage

```
Check(taskids, repids, output_folder, nbmi)
```

Arguments

taskids	Vector of taskids.
repids	Vector of repids.
output_folder	Character string. Output folder.
nbmi	Logical. If nbmi = TRUE, include MINB, NBMI, and StackedMINB.

Value

Returns file names of replications that need to be run.

Author(s)

Ivan Jacob Agaloos Pesigan

CheckAmputeData   *Check Replication - SimAmputeData*

Description

Check Replication - SimAmputeData

Usage

```
CheckAmputeData(taskid, repid, output_folder, suffix)
```

Arguments

taskid	Positive integer. Task ID.
repid	Positive integer. Replication ID.
output_folder	Character string. Output folder.
suffix	Character string. Output of manMCMedMiss:::.SimSuffix().

Details

This function is executed via the Check function.

Value

Returns file names of replications that need to be run.

Author(s)

Ivan Jacob Agaloos Pesigan

CheckFitModelMI   *Check Replication - SimFitModelMI*

Description

Check Replication - SimFitModelMI

Usage

```
CheckFitModelMI(taskid, repid, output_folder, suffix)
```

Arguments

`taskid`	Positive integer. Task ID.
`repid`	Positive integer. Replication ID.
`output_folder`	Character string. Output folder.
`suffix`	Character string. Output of `manMCMedMiss:::.SimSuffix()`.

Details

This function is executed via the `Check` function.

Value

Returns file names of replications that need to be run.

Author(s)

Ivan Jacob Agaloos Pesigan

CheckFitModelML   *Check Replication - SimFitModelML*

Description

Check Replication - SimFitModelML

Usage

```
CheckFitModelML(taskid, repid, output_folder, suffix)
```

Arguments

`taskid`	Positive integer. Task ID.
`repid`	Positive integer. Replication ID.
`output_folder`	Character string. Output folder.
`suffix`	Character string. Output of `manMCMedMiss:::.SimSuffix()`.

Details

This function is executed via the `Check` function.

Value

Returns file names of replications that need to be run.

Author(s)

Ivan Jacob Agaloos Pesigan

CheckGenData *Check Replication - SimGenData*

Description

Check Replication - SimGenData

Usage

```
CheckGenData(taskid, repid, output_folder, suffix)
```

Arguments

taskid	Positive integer. Task ID.
repid	Positive integer. Replication ID.
output_folder	Character string. Output folder.
suffix	Character string. Output of manMCMedMiss:::.SimSuffix().

Details

This function is executed via the Check function.

Value

Returns file names of replications that need to be run.

Author(s)

Ivan Jacob Agaloos Pesigan

CheckImputeData *Check Replication - SimImputeData*

Description

Check Replication - SimImputeData

Usage

```
CheckImputeData(taskid, repid, output_folder, suffix)
```

Arguments

taskid	Positive integer. Task ID.
repid	Positive integer. Replication ID.
output_folder	Character string. Output folder.
suffix	Character string. Output of `manMCMedMiss:::.SimSuffix()`.

Details

This function is executed via the `Check` function.

Value

Returns file names of replications that need to be run.

Author(s)

Ivan Jacob Agaloos Pesigan

CheckJointSigMI   *Check Replication - SimJointSigMI*

Description

Check Replication - SimJointSigMI

Usage

```
CheckJointSigMI(taskid, repid, output_folder, suffix)
```

Arguments

taskid	Positive integer. Task ID.
repid	Positive integer. Replication ID.
output_folder	Character string. Output folder.
suffix	Character string. Output of `manMCMedMiss:::.SimSuffix()`.

Details

This function is executed via the `Check` function.

Value

Returns file names of replications that need to be run.

Author(s)

Ivan Jacob Agaloos Pesigan

CheckJointSigML *Check Replication - SimJointSigML*

Description

Check Replication - SimJointSigML

Usage

```
CheckJointSigML(taskid, repid, output_folder, suffix)
```

Arguments

taskid	Positive integer. Task ID.
repid	Positive integer. Replication ID.
output_folder	Character string. Output folder.
suffix	Character string. Output of manMCMedMiss:::.SimSuffix().

Details

This function is executed via the `Check` function.

Value

Returns file names of replications that need to be run.

Author(s)

Ivan Jacob Agaloos Pesigan

CheckMCMI *Check Replication - SimMCMI*

Description

Check Replication - SimMCMI

Usage

```
CheckMCMI(taskid, repid, output_folder, suffix)
```

Arguments

`taskid`	Positive integer. Task ID.
`repid`	Positive integer. Replication ID.
`output_folder`	Character string. Output folder.
`suffix`	Character string. Output of `manMCMedMiss:::.SimSuffix()`.

Details

This function is executed via the `Check` function.

Value

Returns file names of replications that need to be run.

Author(s)

Ivan Jacob Agaloos Pesigan

CheckMCML   *Check Replication - SimMCML*

Description

Check Replication - SimMCML

Usage

```
CheckMCML(taskid, repid, output_folder, suffix)
```

Arguments

`taskid`	Positive integer. Task ID.
`repid`	Positive integer. Replication ID.
`output_folder`	Character string. Output folder.
`suffix`	Character string. Output of `manMCMedMiss:::.SimSuffix()`.

Details

This function is executed via the `Check` function.

Value

Returns file names of replications that need to be run.

Author(s)

Ivan Jacob Agaloos Pesigan

CheckMINB   *Check Replication - SimMINB*

Description

Check Replication - SimMINB

Usage

```
CheckMINB(taskid, repid, output_folder, suffix)
```

Arguments

taskid	Positive integer. Task ID.
repid	Positive integer. Replication ID.
output_folder	Character string. Output folder.
suffix	Character string. Output of manMCMedMiss:::.SimSuffix().

Details

This function is executed via the Check function.

Value

Returns file names of replications that need to be run.

Author(s)

Ivan Jacob Agaloos Pesigan

CheckNBMI   *Check Replication - SimNBMI*

Description

Check Replication - SimNBMI

Usage

```
CheckNBMI(taskid, repid, output_folder, suffix)
```

Arguments

`taskid`	Positive integer. Task ID.
`repid`	Positive integer. Replication ID.
`output_folder`	Character string. Output folder.
`suffix`	Character string. Output of `manMCMedMiss:::.SimSuffix()`.

Details

This function is executed via the `Check` function.

Value

Returns file names of replications that need to be run.

Author(s)

Ivan Jacob Agaloos Pesigan

CheckNBML   *Check Replication - SimNBML*

Description

Check Replication - SimNBML

Usage

```
CheckNBML(taskid, repid, output_folder, suffix)
```

Arguments

`taskid`	Positive integer. Task ID.
`repid`	Positive integer. Replication ID.
`output_folder`	Character string. Output folder.
`suffix`	Character string. Output of `manMCMedMiss:::.SimSuffix()`.

Details

This function is executed via the `Check` function.

Value

Returns file names of replications that need to be run.

Author(s)

Ivan Jacob Agaloos Pesigan

CheckStackedMINB	*Check Replication - SimStackedMINB*

Description

Check Replication - SimStackedMINB

Usage

```
CheckStackedMINB(taskid, repid, output_folder, suffix)
```

Arguments

taskid	Positive integer. Task ID.
repid	Positive integer. Replication ID.
output_folder	Character string. Output folder.
suffix	Character string. Output of `manMCMedMiss:::.SimSuffix()`.

Details

This function is executed via the `Check` function.

Value

Returns file names of replications that need to be run.

Author(s)

Ivan Jacob Agaloos Pesigan

Compress	*Compress Replication*

Description

Compress Replication

Usage

```
Compress(taskid, repid, output_folder)
```

Arguments

taskid	Positive integer. Task ID.
repid	Positive integer. Replication ID.
output_folder	Character string. Output folder.

Value

The output is saved as an external file in `output_folder`.

Author(s)

Ivan Jacob Agaloos Pesigan

---

elliot2007	*Intervention on Healthy Dietary Behavior Via Knowledge of Healthy Dietary Behavior*

Description

A study by Elliot and others (2007) on the effects of an intervention on healthy dietary behavior mediated by knowledge of healthy dietary behavior. This data was used as an empirical example in Yuan and MacKinnon (2009) and Wu and Jia (2013).

Usage

```
elliot2007
```

Format

A data frame with 354 cases and 3 variables:

x Intervention group membership.

m Knowledge of healthy dietary behavior (knowledge post-intervention minus knowledge pre-intervention).

y Healthy dietary behavior (behavior post-intervention minus behavior pre-intervention).

Source

https://odin.mdacc.tmc.edu/~yyuan/Software/Mediation/Bayesian/CodeRelease/BayesianMediation/firefighters.txt

References

Elliot, D. L., Goldberg, L., Kuehl, K. S., Moe, E. L., Breger, R. K., & Pickering, M. A. (2007). The PHLAME (Promoting Healthy Lifestyles: Alternative Models' Effects) firefighter study: Outcomes of two models of behavior change. *Journal of occupational and environmental medicine*, *49*(2), 204–213. doi:10.1097/JOM.0b013e3180329a8d

Wu, W., & Jia, F. (2013). A new procedure to test mediation with missing data through nonparametric bootstrapping and multiple imputation, *Multivariate Behavioral Research*, *48*(5), 663-691. doi:10.1080/00273171.2013.816235

Yuan, Y., & MacKinnon, D. P. (2009). Bayesian mediation analysis. *Psychological methods*, *14*(4), 301–322. doi:10.1037/a0016972

FigBoxPlot   *Box Plot of Type I Error Rate, Statistical Power, and Miss Rate*

Description

Box Plot of Type I Error Rate, Statistical Power, and Miss Rate

Usage

```
FigBoxPlot(results, type, mech, prop, n = NULL)
```

Arguments

results	Data frame. The package data frame `results()`.
type	Character string. Type of results to plot. Valid values are `"type1"`, `"power"`, and `"miss"` for Type I error rate, statistical power, and miss rate, respectively.
mech	Character string. Missing data mechanism. Valid values are `"COMPLETE"`, `"MAR"`, and `"MCAR"`.
prop	Numeric. Proportion of missing data. Valid values are `.1`, `.2`, `.3`. Note that if `mech = "COMPLETE"`, prop will be set to `0`.
n	Vector of positive integers. Sample size. Valid values are `50`, `75`, `100`, `150`, `200`, `250`, `500`, and `1000`. If `n = NULL`, use all sample sizes.

Value

Returns a `ggplot2` plot object.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Figure Functions: `FigScatterPlot()`

FigScatterPlot	*Scatter Plot of Type I Error Rate, Statistical Power, and Miss Rate*

Description

Scatter Plot of Type I Error Rate, Statistical Power, and Miss Rate

Usage

```
FigScatterPlot(results, type, mech, prop, n = NULL)
```

Arguments

results	Data frame. The package data frame `results()`.
type	Character string. Type of results to plot. Valid values are `"type1"`, `"power"`, and `"miss"` for Type I error rate, statistical power, and miss rate, respectively.
mech	Character string. Missing data mechanism. Valid values are `"COMPLETE"`, `"MAR"`, and `"MCAR"`.
prop	Numeric. Proportion of missing data. Valid values are `.1`, `.2`, `.3`. Note that if `mech = "COMPLETE"`, prop will be set to `0`.
n	Vector of positive integers. Sample size. Valid values are `50`, `75`, `100`, `150`, `200`, `250`, `500`, and `1000`. If `n = NULL`, use all sample sizes.

Value

Returns a `ggplot2` plot object.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Figure Functions: FigBoxPlot()

FitModelIndirect | *Fit the Simple Mediation Model using Normal Theory Maximum Likelihood (Indirect Effect)*

Description

Fit the Simple Mediation Model using Normal Theory Maximum Likelihood (Indirect Effect)

Usage

```
FitModelIndirect(data_complete, consistent = TRUE)
```

Arguments

data_complete   Numeric matrix. Output of the `GenData` function or a three-column data set with complete data.

consistent   Logical. If `consistent = TRUE`, use the consistent estimate of the covariance matrix (maximum likelihood estimator). If `consistent = FALSE`, use the unbiased estimate of the covariance matrix.

Value

Returns the indirect effect.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Model Fitting Functions: `FitModelMI()`, `FitModelML()`

FitModelMI | *Fit the Simple Mediation Model using Multiple Imputation*

Description

Fit the Simple Mediation Model using Multiple Imputation

Usage

```
FitModelMI(data_mi, mplus_bin)
```

Arguments

`data_mi`	List of numeric matrices. Output of the `ImputeData` function or a list of three-column data sets with imputed data.
`mplus_bin`	Character string. Path of Mplus binary.

Details

Let

$$M = \text{Number of imputations,} \quad \text{and}$$

$$m = \{1, 2, \cdots, M\}.$$

The vector of pooled coefficients/parameter estimates is given by

$$\bar{\boldsymbol{\theta}} = M^{-1} \sum_{m=1}^{M} \hat{\boldsymbol{\theta}}_m.$$

The covariance within imputations is given by

$$\mathbf{V}_{\text{within}} = M^{-1} \sum_{m=1}^{M} \text{Var}\left(\hat{\boldsymbol{\theta}}_m\right)$$

where $\text{Var}\left(\hat{\boldsymbol{\theta}}_m\right)$ is the parameter covariance matrix for the mth imputation. The covariance between imputations is given by

$$\mathbf{V}_{\text{between}} = (M-1)^{-1} \sum_{m=1}^{M} \left(\hat{\boldsymbol{\theta}}_m - \bar{\boldsymbol{\theta}}\right) \left(\hat{\boldsymbol{\theta}}_m - \bar{\boldsymbol{\theta}}\right)'.$$

The total covariance matrix is given by

$$\mathbf{V}_{\text{total}} = \mathbf{V}_{\text{within}} + \mathbf{V}_{\text{between}} + M^{-1}\mathbf{V}_{\text{between}}.$$

The adjusted total covariance matrix is given by

$$\tilde{\mathbf{V}}_{\text{total}} = (1 + \text{ARIV})\,\mathbf{V}_{\text{within}}$$

where

$$\text{ARIV} = k^{-1}\left[\left(1 + M^{-1}\right)\text{tr}\left(\mathbf{V}_{\text{between}}\mathbf{V}_{\text{within}}^{-1}\right)\right]$$

and k is the number of parameters.

Value

Returns a list with the following elements:

coef Vector of pooled coefficients/parameter estimates $\bar{\boldsymbol{\theta}}$.

vcov Total covariance matrix $\mathbf{V}_{\text{total}}$.

vcov_tilde Adjusted total covariance matrix $\tilde{\mathbf{V}}_{\text{total}}$.

vcov_between Covariance between imputations $\mathbf{V}_{\text{between}}$.

vcov_within Covariance within imputations $\mathbf{V}_{\text{within}}$.

ariv Average relative increase in variance ARIV.

m Number of imputations M.

k Number of parameters k.

nu1 Numerator degrees of freedom ν_1 for D_1.

nu2 Denominator degrees of freedom ν_2 for D_1.

d1 D_1 test statistic.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Li, K. H., Raghunathan, T. E., & Rubin, D. B. (1991). Large-sample significance levels from multiply imputed data using moment-based statistics and an F reference distribution. *Journal of the American Statistical Association*, 86 (416), 1065–1073. doi:10.1080/01621459.1991.10475152

Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys*. John Wiley & Sons, Inc. doi:10.1002/9780470316696

See Also

Other Model Fitting Functions: `FitModelIndirect`(), `FitModelML`()

FitModelML	*Fit the Simple Mediation Model using Normal Theory Maximum Likelihood*

Description

Fit the Simple Mediation Model using Normal Theory Maximum Likelihood

Usage

```
FitModelML(data, mplus_bin)
```

Arguments

data	Numeric matrix. Output of the `GenData` or `AmputeData` functions.
mplus_bin	Character string. Path of Mplus binary.

Value

Returns a list with the following elements:

fit Model fit.

coef Coefficients/parameter estimates

vcov Sampling variance-covariance matrix.

output Mplus output.

Values in `fit`.

free_parameters Number of free parameters

h0_loglikelihood H0 loglikelihood.

h1_loglikelihood H1 loglikelihood.

aic Akaike information criterion (AIC).

bic Bayesian information criterion (BIC).

sabic Sample-size adjusted BIC (SABIC).

chisq Chi-square value.

chisq_df Chi-square degrees of freedom.

chisq_p Chi-square p-value.

cfi Comparative fit index (CFI).

tli Tucker–Lewis index (TLI).

rmsea Root mean square error of approximation (RMSEA) estimate.

rmsea_low Root mean square error of approximation (RMSEA) lower limit confidence interval.

rmsea_up Root mean square error of approximation (RMSEA) upper limit confidence interval.

rmsea_p Root mean square error of approximation (RMSEA) probability.

srmr Standardized root mean square residual (SRMR).

condition_number Condition number for the information matrix (ratio of smallest to largest eigen-value).

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Model Fitting Functions: FitModelIndirect(), FitModelMI()

GenData   *Generate Data*

Description

Let the simple mediation model be defined by the equations

$$Y = \delta_Y + \tau'X + \beta M + \varepsilon_Y,$$

and

$$M = \delta_M + \alpha X + \varepsilon_M.$$

The function generates data from the multivariate normal distribution using the model-implied mean vector and covariance matrix of the simple mediation model. See `MASS::mvrnorm()` for more details.

Usage

```
GenData(n = 100L, tauprime = 0, beta = 0.5, alpha = 0.5, mu = 0, sigmasq = 1)
```

Arguments

n	Positive integer. Sample size.
tauprime	Numeric. τ', that is, the slope for Y regressed on X, adjusting for M.
beta	Numeric. β, that is, the slope for Y regressed on M, adjusting for X.
alpha	Numeric vector. Significance level.
mu	Numeric. Common mean for X, M, and Y, that is, $\mu_X = \mu_M = \mu_Y$.
sigmasq	Numeric. Common variance for X, M, and Y, that is, $\sigma_X^2 = \sigma_M^2 = \sigma_Y^2$.

Value

Returns a matrix.

Author(s)

Ivan Jacob Agaloos Pesigan

References

MacKinnon, D. P. (2008). *Introduction to statistical mediation analysis.* Lawrence Erlbaum Associates.

See Also

Other Data Generation Functions: AmputeData(), ImputeData()

ImputeData	*Impute Data*

Description

Generates `m` complete data sets using multiple imputation.

Usage

```
ImputeData(data_missing, m, mplus_bin)
```

Arguments

data_missing	Numeric matrix. Output of the `AmputeData` function or a three-column data set with missing data.
m	Positive integer. Number of imputations.
mplus_bin	Character string. Path of Mplus binary.

Value

Returns a list of complete data sets.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Asparouhov, T., & Muthen, B. (2022). *Multiple imputation with Mplus*. Retrieved from http://www.statmodel.com/download/Imputations7.pdf

See Also

Other Data Generation Functions: AmputeData(), GenData()

JointSigMI	*Joint Significance Test for the Indirect Effect using Multiple Imputation*

Description

Joint Significance Test for the Indirect Effect using Multiple Imputation

Usage

```
JointSigMI(fit_mi, alpha = c(0.05, 0.01, 0.001))
```

Arguments

fit_mi	Object. Output of the FitModelMI function.
alpha	Numeric vector. Significance level.

Value

Returns a list with the following elements:

vcov Joint significance test based on the total covariance matrix.

vcov_tilde Joint significance test based on the adjusted total covariance matrix.

Note the 1 corresponds to a significant result and 0 otherwise. The output vector's name corresponds to the alpha level.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Significance Test Functions: [JointSigML](')

JointSigML	*Joint Significance Test for the Indirect Effect using Normal Theory Maximum Likelihood*

Description

Joint Significance Test for the Indirect Effect using Normal Theory Maximum Likelihood

Usage

```
JointSigML(fit_ml, alpha = c(0.05, 0.01, 0.001))
```

Arguments

fit_ml	Object. Output of the FitModelML function.
alpha	Numeric vector. Significance level.

Value

Note the 1 corresponds to a significant result and 0 otherwise. The output vector's name corresponds to the alpha level.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Significance Test Functions: `JointSigMI()`

MCMI   *Monte Carlo Confidence Intervals for the Indirect Effect using Multiple Imputation*

Description

Monte Carlo Confidence Intervals for the Indirect Effect using Multiple Imputation

Usage

```
MCMI(fit_mi, R = 20000L, alpha = c(0.05, 0.01, 0.001))
```

Arguments

fit_mi	Object. Output of the `FitModelMI` function.
R	Positive integer. Number of Monte Carlo replications.
alpha	Numeric vector. Significance level.

Value

Returns a list with the following elements:

vcov Monte Carlo confidence intervals based on the total covariance matrix.

vcov_tilde Monte Carlo confidence intervals based on the adjusted total covariance matrix.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Confidence Interval Functions: `MCML()`, `MINB()`, `NBMI()`, `NBML()`, `NBStackedMI()`

MCML	*Monte Carlo Confidence Intervals for the Indirect Effect using Normal Theory Maximum Likelihood*

Description

Monte Carlo Confidence Intervals for the Indirect Effect using Normal Theory Maximum Likelihood

Usage

```
MCML(fit_ml, R = 20000L, alpha = c(0.05, 0.01, 0.001))
```

Arguments

fit_ml	Object. Output of the `FitModelML` function.
R	Positive integer. Number of Monte Carlo replications.
alpha	Numeric vector. Significance level.

Value

Monte Carlo confidence intervals.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Confidence Interval Functions: `MCMI()`, `MINB()`, `NBMI()`, `NBML()`, `NBStackedMI()`

MICombine	*Combine Parameter Estimates Vector and Sampling Variance-Covariance Matrix Estimated from Multiple Imputations*

Description

Combine Parameter Estimates Vector and Sampling Variance-Covariance Matrix Estimated from Multiple Imputations

Usage

```
MICombine(param_vec, var_mat)
```

Arguments

param_vec   List of vectors of paramater estimates.

var_mat   List of matrices of sampling variances and covariances.

Value

Returns a list with the following elements:

m Number of imputations M.

k Number of parameters k.

estimates Vector of pooled coefficients/parameter estimates $\bar{\boldsymbol{\theta}}$.

within Covariance within imputations $\mathbf{V}_{\mathrm{within}}$.

between Covariance between imputations $\mathbf{V}_{\mathrm{between}}$.

total Total covariance matrix $\mathbf{V}_{\mathrm{total}}$.

total_adjusted Adjusted total covariance matrix $\tilde{\mathbf{V}}_{\mathrm{total}}$.

ariv Average relative increase in variance ARIV.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Li, K. H., Raghunathan, T. E., & Rubin, D. B. (1991). Large-sample significance levels from multiply imputed data using moment-based statistics and an F reference distribution. *Journal of the American Statistical Association*, 86 (416), 1065–1073. doi:10.1080/01621459.1991.10475152

Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys*. John Wiley & Sons, Inc. doi:10.1002/9780470316696

MINB	*Nonparametric Bootstrap Confidence Intervals for the Indirect Effect using Multiple Imputation (NB nested within MI or MI(NB))*

Description

Nonparametric Bootstrap Confidence Intervals for the Indirect Effect using Multiple Imputation (NB nested within MI or MI(NB))

Usage

```
MINB(data_mi, B = 5000L, alpha = c(0.05, 0.01, 0.001))
```

Arguments

`data_mi`	List of numeric matrices. Output of the `ImputeData` function or a list of three-column data sets with imputed data.
`B`	Positive integer. Number of bootstrap samples.
`alpha`	Numeric vector. Significance level.

Value

Nonparametric bootstrap percentile confidence intervals.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Wu, Wei., & Jia, F. (2012). A new procedure to test mediation with missing data through nonparametric bootstrapping and multiple imputation. *Multivariate Behavioral Research*, *48*(5), 663–691. doi:10.1080/00273171.2013.816235

See Also

Other Confidence Interval Functions: `MCMI()`, `MCML()`, `NBMI()`, `NBML()`, `NBStackedMI()`

NBMI	*Nonparametric Bootstrap Confidence Intervals for the Indirect Effect using Multiple Imputation (MI nested within NB or NB(MI))*

Description

Nonparametric Bootstrap Confidence Intervals for the Indirect Effect using Multiple Imputation (MI nested within NB or NB(MI))

Usage

```
NBMI(
  data_missing,
  data_mi,
  B = 5000L,
  m = 100L,
  alpha = c(0.05, 0.01, 0.001),
  mplus_bin
)
```

Arguments

data_missing	Numeric matrix. Output of the AmputeData function or a three-column data set with missing data.
data_mi	List of numeric matrices. Output of the ImputeData function or a list of three-column data sets with imputed data.
B	Positive integer. Number of bootstrap samples.
m	Positive integer. Number of imputations.
alpha	Numeric vector. Significance level.
mplus_bin	Character string. Path of Mplus binary.

Value

Nonparametric bootstrap percentile confidence intervals.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Zhang, Z., & Wang, L. (2012). Methods for mediation analysis with missing data. *Psychometrika*, *78*(1), 154–184. doi:10.1007/s1133601293015

Zhang, Z., Wang, L., & Tong, X. (2015). Mediation analysis with missing data through multiple imputation and bootstrap. *Quantitative psychology research* (pp. 341–355). Springer International Publishing. doi:10.1007/9783319199771_24

See Also

Other Confidence Interval Functions: MCMI(), MCML(), MINB(), NBML(), NBStackedMI()

NBML	*Nonparametric Bootstrap Confidence Intervals for the Indirect Effect using Normal Theory Maximum Likelihood (ML nested within NB or NB(ML))*

Description

Nonparametric Bootstrap Confidence Intervals for the Indirect Effect using Normal Theory Maximum Likelihood (ML nested within NB or NB(ML))

Usage

```
NBML(data, B = 5000L, mplus_bin)
```

Arguments

data	Numeric matrix. Output of the GenData or AmputeData functions.
B	Positive integer. Number of bootstrap samples.
mplus_bin	Character string. Path of Mplus binary.

Value

Nonparametric bootstrap confidence intervals. bc corresponds to bias-corrected and pc corresponds to percentile confidence intervals.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Confidence Interval Functions: MCMI(), MCML(), MINB(), NBMI(), NBStackedMI()

NBStackedMI	*Nonparametric Bootstrap Confidence Intervals for the Indirect Effect using Multiple Imputation (NB using Stacked MI)*

Description

Nonparametric Bootstrap Confidence Intervals for the Indirect Effect using Multiple Imputation (NB using Stacked MI)

Usage

```
NBStackedMI(data_mi, B = 5000L, alpha = c(0.05, 0.01, 0.001))
```

Arguments

data_mi	List of numeric matrices. Output of the ImputeData function or a list of three-column data sets with imputed data.
B	Positive integer. Number of bootstrap samples.
alpha	Numeric vector. Significance level.

Value

Nonparametric bootstrap percentile confidence intervals.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Confidence Interval Functions: `MCMI()`, `MCML()`, `MINB()`, `NBMI()`, `NBML()`

---

params   *Simulation Parameters*

Description

Simulation Parameters

Usage

```
params
```

Format

A dataframe with 472 rows and 7 columns:

taskid Simulation Task ID.

tauprime τ', that is, the path from X to Y, adjusting for M.

beta β, that is, the path from M to Y.

alpha α, that is, the path from X to M.

n Sample size.

sigmasqepsilonm Error variance $\sigma^2_{\varepsilon_M}$.

sigmasqepsilony Error variance $\sigma^2_{\varepsilon_Y}$.

alphabeta $\alpha\beta$, that is, the indirect effect of X on Y via M.

Author(s)

Ivan Jacob Agaloos Pesigan

---

results   *Simulation Results*

Description

Simulation Results

Usage

```
results
```

Format

A dataframe with 24,544 rows and 13 columns:

zero_hit The proportion of replications where the confidence intervals contained zero.

theta_hit The proportion of replications where the confidence intervals contained the population $\alpha\beta$.

replications Simulation replications.

taskid Simulation Task ID.

tauprime τ', that is, the path from X to Y, adjusting for M.

beta β, that is, the path from M to Y.

alpha α, that is, the path from X to M.

n Sample size.

sigmasqepsilonm Error variance $\sigma_{\varepsilon_M}^2$.

sigmasqepsilony Error variance $\sigma_{\varepsilon_Y}^2$.

alphabeta $\alpha\beta$, that is, the indirect effect of X on Y via M.

mechanism Missing data mechanism. "COMPLETE" for complete data, "MCAR" for missing completely at random, and "MAR" for missing at random.

proportion Proportion of missing data (.0, .1, .2, .3).

method Method used.

type1 Type I error rate.

power Statistical power.

miss Miss rate.

The methods are as follows:

MC.COMPLETE for Monte Carlo method with maximum likelihood estimates for complete data.

MC.FIML for Monte Carlo method with full information maximum likelihood estimates.

MC.MI for Monte Carlo method with multiple imputation estimates.

MC.MI.ADJ for Monte Carlo method with adjusted multiple imputation estimates.

NBBC.COMPLETE for bias-corrected nonparametric bootstrap with maximum likelihood estimates for complete data.

NBBC.FIML for full maximum likelihood nested within bias-corrected nonparametric bootstrap.

NBPC.COMPLETE for percentile nonparametric bootstrap with full maximum likelihood estimates for complete.

NBPC.FIML for full maximum likelihood nested within percentile nonparametric bootstrap.

SIG.COMPLETE for joint-significant test for complete data.

SIG.FIML for the joint-significant test with full maximum likelihood estimates.

SIG.MI for joint-significant test with multiple imputation estimates.

SIG.MI.ADJ for the joint-significant test with adjusted multiple imputation estimates.

Author(s)

Ivan Jacob Agaloos Pesigan

Sim   *Simulation Replication*

Description

Simulation Replication

Usage

```
Sim(
  taskid,
  repid,
  output_folder,
  overwrite,
  integrity,
  params_taskid,
  alpha,
  m,
  R,
  B,
  mplus_bin,
  nbmi
)
```

Arguments

taskid	Positive integer. Task ID.
repid	Positive integer. Replication ID.
output_folder	Character string. Output folder.
overwrite	Logical. Overwrite existing output in output_folder.
integrity	Logical. If integrity = TRUE, check for the output file integrity when overwrite = FALSE.
params_taskid	Data frame with a single row. Simulation parameters for a specific taskid.
alpha	Numeric vector. Significance level.
m	Positive integer. Number of imputations.
R	Positive integer. Number of Monte Carlo replications.
B	Positive integer. Number of bootstrap samples.
mplus_bin	Character string. Path of Mplus binary.
nbmi	Logical. If nbmi = TRUE, include MINB, NBMI, and StackedMINB.

Value

The output is saved as an external file in output_folder.

Author(s)

Ivan Jacob Agaloos Pesigan

SimAmputeData	*Simulation Replication - AmputeData*

Description

Simulation Replication - AmputeData

Usage

```
SimAmputeData(taskid, repid, output_folder, seed, suffix, overwrite, integrity)
```

Arguments

taskid	Positive integer. Task ID.
repid	Positive integer. Replication ID.
output_folder	Character string. Output folder.
seed	Integer. Random seed.
suffix	Character string. Output of manMCMedMiss:::.SimSuffix().
overwrite	Logical. Overwrite existing output in output_folder.
integrity	Logical. If integrity = TRUE, check for the output file integrity when overwrite = FALSE.

Details

This function is executed via the Sim function.

Value

The output is saved as an external file in output_folder.

Author(s)

Ivan Jacob Agaloos Pesigan

SimFitModelMI   *Simulation Replication - FitModelMI*

Description

Simulation Replication - FitModelMI

Usage

```
SimFitModelMI(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  mplus_bin
)
```

Arguments

taskid	Positive integer. Task ID.
repid	Positive integer. Replication ID.
output_folder	Character string. Output folder.
seed	Integer. Random seed.
suffix	Character string. Output of manMCMedMiss:::.SimSuffix().
overwrite	Logical. Overwrite existing output in output_folder.
integrity	Logical. If integrity = TRUE, check for the output file integrity when overwrite = FALSE.
mplus_bin	Character string. Path of Mplus binary.

Details

This function is executed via the Sim function.

Value

The output is saved as an external file in output_folder.

Author(s)

Ivan Jacob Agaloos Pesigan

SimFitModelML   *Simulation Replication - FitModelML*

Description

Simulation Replication - FitModelML

Usage

```
SimFitModelML(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  mplus_bin
)
```

Arguments

taskid	Positive integer. Task ID.
repid	Positive integer. Replication ID.
output_folder	Character string. Output folder.
seed	Integer. Random seed.
suffix	Character string. Output of manMCMedMiss:::.SimSuffix().
overwrite	Logical. Overwrite existing output in output_folder.
integrity	Logical. If integrity = TRUE, check for the output file integrity when overwrite = FALSE.
mplus_bin	Character string. Path of Mplus binary.

Details

This function is executed via the Sim function.

Value

The output is saved as an external file in output_folder.

Author(s)

Ivan Jacob Agaloos Pesigan

SimFN   *Simulation File Name*

Description

Simulation File Name

Usage

```
SimFN(data_type, output_type, output_folder, suffix)
```

Arguments

data_type	Character string. Input data type. Valid values include "complete-00", "mar-30", "mar-20", "mar-10", "mcar-30", "mcar-20", and "mcar-10"
output_type	Character string. Output type. Valid values include "data", "fit-mi", "fit-ml", "mc-mi", "mc-ml", "mi-nb", "nb-mi", "nb-ml", "sig-mi", "sig-ml", and "stackedmi-nb"
output_folder	Character string. Output folder.
suffix	Character string. Output of manMCMedMiss:::.SimSuffix().

Value

Returns a character string file name with the output_folder in the OS-specific format.

SimGenData   *Simulation Replication - GenData*

Description

Simulation Replication - GenData

Usage

```
SimGenData(
  taskid,
  repid,
  output_folder,
  params_taskid,
  seed,
  suffix,
  overwrite,
  integrity
)
```

Arguments

`taskid`	Positive integer. Task ID.
`repid`	Positive integer. Replication ID.
`output_folder`	Character string. Output folder.
`params_taskid`	Data frame with a single row. Simulation parameters for a specific `taskid`.
`seed`	Integer. Random seed.
`suffix`	Character string. Output of `manMCMedMiss:::.SimSuffix()`.
`overwrite`	Logical. Overwrite existing output in `output_folder`.
`integrity`	Logical. If `integrity = TRUE`, check for the output file integrity when `overwrite = FALSE`.

Details

This function is executed via the `Sim` function.

Value

The output is saved as an external file in `output_folder`.

Author(s)

Ivan Jacob Agaloos Pesigan

SimImputeData	*Simulation Replication - ImputeData*

Description

Simulation Replication - ImputeData

Usage

```
SimImputeData(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  m,
  mplus_bin
)
```

Arguments

taskid	Positive integer. Task ID.
repid	Positive integer. Replication ID.
output_folder	Character string. Output folder.
seed	Integer. Random seed.
suffix	Character string. Output of manMCMedMiss:::.SimSuffix().
overwrite	Logical. Overwrite existing output in output_folder.
integrity	Logical. If integrity = TRUE, check for the output file integrity when overwrite = FALSE.
m	Positive integer. Number of imputations.
mplus_bin	Character string. Path of Mplus binary.

Details

This function is executed via the Sim function.

Value

The output is saved as an external file in output_folder.

Author(s)

Ivan Jacob Agaloos Pesigan

SimJointSigMI   *Simulation Replication - JointSigMI*

Description

Simulation Replication - JointSigMI

Usage

```
SimJointSigMI(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  alpha
)
```

Arguments

`taskid`	Positive integer. Task ID.
`repid`	Positive integer. Replication ID.
`output_folder`	Character string. Output folder.
`seed`	Integer. Random seed.
`suffix`	Character string. Output of `manMCMedMiss:::.SimSuffix()`.
`overwrite`	Logical. Overwrite existing output in `output_folder`.
`integrity`	Logical. If `integrity = TRUE`, check for the output file integrity when `overwrite = FALSE`.
`alpha`	Numeric vector. Significance level.

Details

This function is executed via the `Sim` function.

Value

The output is saved as an external file in `output_folder`.

Author(s)

Ivan Jacob Agaloos Pesigan

SimJointSigML	*Simulation Replication - JointSigML*

Description

Simulation Replication - JointSigML

Usage

```
SimJointSigML(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  alpha
)
```

Arguments

`taskid`	Positive integer. Task ID.
`repid`	Positive integer. Replication ID.
`output_folder`	Character string. Output folder.
`seed`	Integer. Random seed.
`suffix`	Character string. Output of `manMCMedMiss:::.SimSuffix()`.
`overwrite`	Logical. Overwrite existing output in `output_folder`.
`integrity`	Logical. If `integrity = TRUE`, check for the output file integrity when `overwrite = FALSE`.
`alpha`	Numeric vector. Significance level.

Details

This function is executed via the `Sim` function.

Value

The output is saved as an external file in `output_folder`.

Author(s)

Ivan Jacob Agaloos Pesigan

SimMCMI   *Simulation Replication - MCMI*

Description

Simulation Replication - MCMI

Usage

```
SimMCMI(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  R,
  alpha
)
```

Arguments

`taskid`	Positive integer. Task ID.
`repid`	Positive integer. Replication ID.
`output_folder`	Character string. Output folder.
`seed`	Integer. Random seed.
`suffix`	Character string. Output of `manMCMedMiss:::.SimSuffix()`.
`overwrite`	Logical. Overwrite existing output in `output_folder`.
`integrity`	Logical. If `integrity = TRUE`, check for the output file integrity when `overwrite = FALSE`.
`R`	Positive integer. Number of Monte Carlo replications.
`alpha`	Numeric vector. Significance level.

Details

This function is executed via the `Sim` function.

Value

The output is saved as an external file in `output_folder`.

Author(s)

Ivan Jacob Agaloos Pesigan

SimMCML   *Simulation Replication - MCML*

Description

Simulation Replication - MCML

Usage

```
SimMCML(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  R,
  alpha
)
```

Arguments

taskid	Positive integer. Task ID.
repid	Positive integer. Replication ID.
output_folder	Character string. Output folder.
seed	Integer. Random seed.
suffix	Character string. Output of manMCMedMiss:::.SimSuffix().
overwrite	Logical. Overwrite existing output in output_folder.
integrity	Logical. If integrity = TRUE, check for the output file integrity when overwrite = FALSE.
R	Positive integer. Number of Monte Carlo replications.
alpha	Numeric vector. Significance level.

Details

This function is executed via the Sim function.

Value

The output is saved as an external file in output_folder.

Author(s)

Ivan Jacob Agaloos Pesigan

SimMINB   *Simulation Replication - MINB*

Description

Simulation Replication - MINB

Usage

```
SimMINB(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  B,
  alpha
)
```

Arguments

`taskid`	Positive integer. Task ID.
`repid`	Positive integer. Replication ID.
`output_folder`	Character string. Output folder.
`seed`	Integer. Random seed.
`suffix`	Character string. Output of `manMCMedMiss:::.SimSuffix()`.
`overwrite`	Logical. Overwrite existing output in `output_folder`.
`integrity`	Logical. If `integrity = TRUE`, check for the output file integrity when `overwrite = FALSE`.
`B`	Positive integer. Number of bootstrap samples.
`alpha`	Numeric vector. Significance level.

Details

This function is executed via the `Sim` function.

Value

The output is saved as an external file in `output_folder`.

Author(s)

Ivan Jacob Agaloos Pesigan

SimNBMI *Simulation Replication - NBMI*

Description

Simulation Replication - NBMI

Usage

```
SimNBMI(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  B,
  m,
  mplus_bin
)
```

Arguments

`taskid`	Positive integer. Task ID.
`repid`	Positive integer. Replication ID.
`output_folder`	Character string. Output folder.
`seed`	Integer. Random seed.
`suffix`	Character string. Output of `manMCMedMiss:::.SimSuffix()`.
`overwrite`	Logical. Overwrite existing output in `output_folder`.
`integrity`	Logical. If `integrity = TRUE`, check for the output file integrity when `overwrite = FALSE`.
`B`	Positive integer. Number of bootstrap samples.
`m`	Positive integer. Number of imputations.
`mplus_bin`	Character string. Path of Mplus binary.

Details

This function is executed via the `Sim` function.

Value

The output is saved as an external file in `output_folder`.

Author(s)

Ivan Jacob Agaloos Pesigan

SimNBML   *Simulation Replication - NBML*

Description

Simulation Replication - NBML

Usage

```
SimNBML(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  B,
  mplus_bin
)
```

Arguments

taskid	Positive integer. Task ID.
repid	Positive integer. Replication ID.
output_folder	Character string. Output folder.
seed	Integer. Random seed.
suffix	Character string. Output of manMCMedMiss:::.SimSuffix().
overwrite	Logical. Overwrite existing output in output_folder.
integrity	Logical. If integrity = TRUE, check for the output file integrity when overwrite = FALSE.
B	Positive integer. Number of bootstrap samples.
mplus_bin	Character string. Path of Mplus binary.

Details

This function is executed via the Sim function.

Value

The output is saved as an external file in output_folder.

Author(s)

Ivan Jacob Agaloos Pesigan

SimNBStackedMI	*Simulation Replication - NBStackedMI*

Description

Simulation Replication - NBStackedMI

Usage

```
SimNBStackedMI(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  B,
  alpha
)
```

Arguments

`taskid`	Positive integer. Task ID.
`repid`	Positive integer. Replication ID.
`output_folder`	Character string. Output folder.
`seed`	Integer. Random seed.
`suffix`	Character string. Output of `manMCMedMiss:::.SimSuffix()`.
`overwrite`	Logical. Overwrite existing output in `output_folder`.
`integrity`	Logical. If `integrity = TRUE`, check for the output file integrity when `overwrite = FALSE`.
`B`	Positive integer. Number of bootstrap samples.
`alpha`	Numeric vector. Significance level.

Details

This function is executed via the `Sim` function.

Value

The output is saved as an external file in `output_folder`.

Author(s)

Ivan Jacob Agaloos Pesigan

SimProj	*Simulation Project Name*

Description

Simulation Project Name

Usage

```
SimProj()
```

Value

Returns the project name as a character string.

Author(s)

Ivan Jacob Agaloos Pesigan

Sum *Summarize Simulations*

Description

Summarize Simulations

Usage

```
Sum(taskid, reps, output_folder, params_taskid, nbmi)
```

Arguments

taskid	Positive integer. Task ID.
reps	Positive integer. Number of replications.
output_folder	Character string. Output folder.
params_taskid	Data frame with a single row. Simulation parameters for a specific taskid.
nbmi	Logical. If nbmi = TRUE, include MINB, NBMI, and StackedMINB.

Value

The output is saved as an external file in output_folder.

Author(s)

Ivan Jacob Agaloos Pesigan

SumBind *Bind Results*

Description

Bind Results

Usage

```
SumBind(tasks, output_folder, data_raw_folder, params)
```

Arguments

tasks	Positive integer. Number of simulations tasks or cases.
output_folder	Character string. Output folder.
data_raw_folder	
	Character string. data-raw folder in the project directory.
params	Data frame. Simulation parameters for all simulation tasks or cases.

Value

Returns a dataframe of results and parameters.

Author(s)

Ivan Jacob Agaloos Pesigan

SumHit   *Extract Zero Hit and Theta Hit from Summary Output*

Description

Extract Zero Hit and Theta Hit from Summary Output

Usage

```
SumHit(summary_output, params_taskid)
```

Arguments

summary_output   Numeric matrix. Output of `Sum*` functions.

params_taskid   Data frame with a single row. Simulation parameters for a specific `taskid`.

Value

Returns a dataframe of results and parameters.

Author(s)

Ivan Jacob Agaloos Pesigan

SumJointSigMI   *Summarize Simulations - SimJointSigMI*

Description

Summarize Simulations - SimJointSigMI

Usage

```
SumJointSigMI(taskid, reps, output_folder, params_taskid)
```

Arguments

`taskid`	Positive integer. Task ID.
`reps`	Positive integer. Number of replications.
`output_folder`	Character string. Output folder.
`params_taskid`	Data frame with a single row. Simulation parameters for a specific `taskid`.

Details

This function is executed via the `Sum` function.

Value

Returns a numeric matrix.

Author(s)

Ivan Jacob Agaloos Pesigan

`SumJointSigML`	*Summarize Simulations - SimJointSigML*

Description

Summarize Simulations - SimJointSigML

Usage

```
SumJointSigML(taskid, reps, output_folder, params_taskid)
```

Arguments

`taskid`	Positive integer. Task ID.
`reps`	Positive integer. Number of replications.
`output_folder`	Character string. Output folder.
`params_taskid`	Data frame with a single row. Simulation parameters for a specific `taskid`.

Details

This function is executed via the `Sum` function.

Value

Returns a numeric matrix.

Author(s)

Ivan Jacob Agaloos Pesigan

SumMCMI   *Summarize Simulations - SimMCMI*

Description

Summarize Simulations - SimMCMI

Usage

```
SumMCMI(taskid, reps, output_folder, params_taskid)
```

Arguments

taskid	Positive integer. Task ID.
reps	Positive integer. Number of replications.
output_folder	Character string. Output folder.
params_taskid	Data frame with a single row. Simulation parameters for a specific taskid.

Details

This function is executed via the Sum function.

Value

Returns a numeric matrix.

Author(s)

Ivan Jacob Agaloos Pesigan

SumMCML   *Summarize Simulations - SimMCML*

Description

Summarize Simulations - SimMCML

Usage

```
SumMCML(taskid, reps, output_folder, params_taskid)
```

Arguments

taskid	Positive integer. Task ID.
reps	Positive integer. Number of replications.
output_folder	Character string. Output folder.
params_taskid	Data frame with a single row. Simulation parameters for a specific taskid.

Details

This function is executed via the `Sum` function.

Value

Returns a numeric matrix.

Author(s)

Ivan Jacob Agaloos Pesigan

SumMINB   *Summarize Simulations - SimMINB*

Description

Summarize Simulations - SimMINB

Usage

```
SumMINB(taskid, reps, output_folder, params_taskid)
```

Arguments

`taskid`	Positive integer. Task ID.
`reps`	Positive integer. Number of replications.
`output_folder`	Character string. Output folder.
`params_taskid`	Data frame with a single row. Simulation parameters for a specific `taskid`.

Details

This function is executed via the `Sum` function.

Value

Returns a numeric matrix.

Author(s)

Ivan Jacob Agaloos Pesigan

SumNBMI   *Summarize Simulations - SimNBMI*

Description

Summarize Simulations - SimNBMI

Usage

```
SumNBMI(taskid, reps, output_folder, params_taskid)
```

Arguments

taskid	Positive integer. Task ID.
reps	Positive integer. Number of replications.
output_folder	Character string. Output folder.
params_taskid	Data frame with a single row. Simulation parameters for a specific taskid.

Details

This function is executed via the Sum function.

Value

Returns a numeric matrix.

Author(s)

Ivan Jacob Agaloos Pesigan

SumNBML   *Summarize Simulations - SimNBML*

Description

Summarize Simulations - SimNBML

Usage

```
SumNBML(taskid, reps, output_folder, params_taskid)
```

Arguments

taskid	Positive integer. Task ID.
reps	Positive integer. Number of replications.
output_folder	Character string. Output folder.
params_taskid	Data frame with a single row. Simulation parameters for a specific taskid.

Details

This function is executed via the Sum function.

Value

Returns a numeric matrix.

Author(s)

Ivan Jacob Agaloos Pesigan

SumNBStackedMI   *Summarize Simulations - SimNBStackedMI*

Description

Summarize Simulations - SimNBStackedMI

Usage

```
SumNBStackedMI(taskid, reps, output_folder, params_taskid)
```

Arguments

taskid	Positive integer. Task ID.
reps	Positive integer. Number of replications.
output_folder	Character string. Output folder.
params_taskid	Data frame with a single row. Simulation parameters for a specific taskid.

Details

This function is executed via the Sum function.

Value

Returns a numeric matrix.

Author(s)

Ivan Jacob Agaloos Pesigan

TabDescribe *Descriptive Statistics of Type I Error Rate, Statistical Power, and Miss Rate*

Description

Descriptive Statistics of Type I Error Rate, Statistical Power, and Miss Rate

Usage

```
TabDescribe(results, type, mech, prop, n = NULL)
```

Arguments

results   Data frame. The package data frame `results()`.

type   Character string. Type of results to plot. Valid values are `"type1"`, `"power"`, and `"miss"` for Type I error rate, statistical power, and miss rate, respectively.

mech   Character string. Missing data mechanism. Valid values are `"COMPLETE"`, `"MAR"`, and `"MCAR"`.

prop   Numeric. Proportion of missing data. Valid values are `.1`, `.2`, `.3`. Note that if `mech = "COMPLETE"`, `prop` will be set to `0`.

n   Vector of positive integers. Sample size. Valid values are `50`, `75`, `100`, `150`, `200`, `250`, `500`, and `1000`. If `n = NULL`, use all sample sizes.

Value

Returns a table of means (`M`), standard deviations (`SD`), and row counts (`N`).

Author(s)

Ivan Jacob Agaloos Pesigan

Tree *Conditional Inference Tree*

Description

Conditional Inference Tree

Usage

```
Tree(results, type, alpha = 1e-04, maxdepth = 0, dichotomize = TRUE)
```

Arguments

`results`	Data frame. The package data frame `results()`.
`type`	Character string. Type of results to plot. Valid values are `"type1"`, `"power"`, and `"miss"` for Type I error rate, statistical power, and miss rate, respectively.
`alpha`	Significance level used for splitting.
`maxdepth`	maximum depth of the tree. The default `maxdepth = 0` means that no restrictions are applied to tree sizes.
`dichotomize`	Logical. Dichotomize the outcome variable into `robust = TRUE` and `robust = FALSE`.

Value

Returns the result of `party::ctree()`.

Author(s)

Ivan Jacob Agaloos Pesigan

Index