

Package ‘manMCMedMiss’

January 10, 2023

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	3
Check	4
CheckAmputeData	4
CheckFitModelMI	5
CheckFitModelML	6

CheckGenData	6
CheckImputeData	7
CheckJointSigMI	8
CheckJointSigML	8
CheckMCMI	9
CheckMCML	10
CheckMINB	10
CheckNBMI	11
CheckNBML	12
CheckStackedMINB	12
Compress	13
elliott2007	14
FigBoxPlot	15
FigScatterPlot	16
FitModelIndirect	17
FitModelMI	17
FitModelML	19
GenData	21
ImputeData	22
JointSigMI	22
JointSigML	23
MCMI	24
MCML	25
MICombine	25
MINB	26
NBMI	27
NBML	28
NBStackedMI	29
params	30
results	30
Sim	32
SimAmputeData	33
SimFitModelMI	34
SimFitModelML	35
SimFN	36
SimGenData	36
SimImputeData	37
SimJointSigMI	38
SimJointSigML	39
SimMCMI	40
SimMCML	41
SimMINB	42
SimNBMI	43
SimNBML	44
SimNBStackedMI	45
SimProj	46
Sum	47
SumBind	47

SumHit	48
SumJointSigMI	48
SumJointSigML	49
SumMCMI	50
SumMCML	50
SumMINB	51
SumNBMI	52
SumNBML	52
SumNBStackedMI	53
TabDescribe	54
Tree	54

Index	56
--------------	-----------

AmputeData	<i>Ampute Data</i>
------------	--------------------

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

<code>data_complete</code>	Numeric matrix. Output of the GenData function or a three-column data set with complete data.
<code>mech</code>	Missing data mechanism.
<code>prop</code>	Proportion of missing data.
<code>patterns</code>	Numeric matrix consisting of zeroes and ones. Each row in the matrix represents a missing data pattern where 0 indicates a missing observation. If <code>patterns = NULL</code> , 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](https://doi.org/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	<i>Check Replication - SimAmputeData</i>
-----------------	--

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	<i>Check Replication - SimFitModelMI</i>
-----------------	--

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 <code>manMCMedMiss:::SimSuffix()</code> .

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 <code>manMCMedMiss:::SimSuffix()</code> .

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	<i>Check Replication - SimGenData</i>
--------------	---------------------------------------

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	<i>Check Replication - SimImputeData</i>
-----------------	--

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	<i>Check Replication - SimJointSigML</i>
-----------------	--

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

CheckMCMCI	<i>Check Replication - SimMCMCI</i>
------------	-------------------------------------

Description

Check Replication - SimMCMCI

Usage

CheckMCMCI(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 <code>manMCMedMiss:::SimSuffix()</code> .

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 <code>manMCMedMiss:::SimSuffix()</code> .

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 <code>manMCMedMiss:::SimSuffix()</code> .

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 <code>manMCMedMiss:::SimSuffix()</code> .

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	<i>Check Replication - SimStackedMINB</i>
------------------	---

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	<i>Compress Replication</i>
----------	-----------------------------

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

elliott2007

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

elliott2007

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 <code>results()</code> .
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 <code>mech = "COMPLETE"</code> , <code>prop</code> 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 <code>n = NULL</code> , use all sample sizes.

Value

Returns a `ggplot2` plot object.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Figure Functions: [FigScatterPlot\(\)](#)

FigScatterPlot	<i>Scatter Plot of Type I Error Rate, Statistical Power, and Miss Rate</i>
----------------	--

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 <code>results()</code> .
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 <code>mech = "COMPLETE"</code> , <code>prop</code> 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 <code>n = NULL</code> , use all sample sizes.

Value

Returns a `ggplot2` plot object.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Figure Functions: [FigBoxPlot\(\)](#)

FitModelIndirect	<i>Fit the Simple Mediation Model using Normal Theory Maximum Likelihood (Indirect Effect)</i>
------------------	--

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	<i>Fit the Simple Mediation Model using Multiple Imputation</i>
------------	---

Description

Fit the Simple Mediation Model using Multiple Imputation

Usage

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

Arguments

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

Details

Let

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

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

The vector of pooled coefficients/parameter estimates is given by

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

The covariance within imputations is given by

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

where $\text{Var}(\hat{\theta}_m)$ is the parameter covariance matrix for the m^{th} imputation. The covariance between imputations is given by

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

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

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

and k is the number of parameters.

Value

Returns a list with the following elements:

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

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](https://doi.org/10.1080/01621459.1991.10475152)

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

See Also

Other Model Fitting Functions: [FitModelIndirect\(\)](#), [FitModelML\(\)](#)

FitModelML	<i>Fit the Simple Mediation Model using Normal Theory Maximum Likelihood</i>
------------	--

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).

sabik 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 eigenvalue).

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	<i>Joint Significance Test for the Indirect Effect using Normal Theory Maximum Likelihood</i>
------------	---

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

<code>fit_mi</code>	Object. Output of the <code>FitModelMI</code> function.
<code>R</code>	Positive integer. Number of Monte Carlo replications.
<code>alpha</code>	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	<i>Monte Carlo Confidence Intervals for the Indirect Effect using Normal Theory Maximum Likelihood</i>
------	--

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: [MCMCI\(\)](#), [MINB\(\)](#), [NBMI\(\)](#), [NBML\(\)](#), [NBStackedMI\(\)](#)

MICombine	<i>Combine Parameter Estimates Vector and Sampling Variance-Covariance Matrix Estimated from Multiple Imputations</i>
-----------	---

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 parameter 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 $\bar{\theta}$.
within Covariance within imputations $\mathbf{V}_{\text{within}}$.
between Covariance between imputations $\mathbf{V}_{\text{between}}$.
total Total covariance matrix $\mathbf{V}_{\text{total}}$.
total_adjusted Adjusted total covariance matrix $\tilde{\mathbf{V}}_{\text{total}}$.
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](https://doi.org/10.1080/00273171.2013.816235)

See Also

Other Confidence Interval Functions: [MCMI\(\)](#), [MCML\(\)](#), [NBMI\(\)](#), [NBML\(\)](#), [NBStackedMI\(\)](#)

NBMI	<i>Nonparametric Bootstrap Confidence Intervals for the Indirect Effect using Multiple Imputation (MI nested within NB or NB(MI))</i>
------	---

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 <code>AmputeData</code> function or a three-column data set with missing data.
data_mi	List of numeric matrices. Output of the <code>ImputeData</code> 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	<i>Nonparametric Bootstrap Confidence Intervals for the Indirect Effect using Normal Theory Maximum Likelihood (ML nested within NB or NB(ML))</i>
------	--

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	<i>Nonparametric Bootstrap Confidence Intervals for the Indirect Effect using Multiple Imputation (NB using Stacked MI)</i>
-------------	---

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	<i>Simulation Parameters</i>
--------	------------------------------

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	<i>Simulation Results</i>
---------	---------------------------

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.

sigmasqepsilnm 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 <code>manMCMedMiss:::SimSuffix()</code> .
overwrite	Logical. Overwrite existing output in <code>output_folder</code> .
integrity	Logical. If <code>integrity = TRUE</code> , check for the output file integrity when <code>overwrite = FALSE</code> .

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 <code>manMCMedMiss:::SimSuffix()</code> .
overwrite	Logical. Overwrite existing output in <code>output_folder</code> .
integrity	Logical. If <code>integrity = TRUE</code> , check for the output file integrity when <code>overwrite = FALSE</code> .
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 <code>manMCMedMiss:::SimSuffix()</code> .
overwrite	Logical. Overwrite existing output in <code>output_folder</code> .
integrity	Logical. If <code>integrity = TRUE</code> , check for the output file integrity when <code>overwrite = FALSE</code> .
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	<i>Simulation File Name</i>
-------	-----------------------------

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 <code>manMCMedMiss:::SimSuffix()</code> .

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

SimGenData	<i>Simulation Replication - GenData</i>
------------	---

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 <code>manMCMedMiss:::SimSuffix()</code> .
overwrite	Logical. Overwrite existing output in <code>output_folder</code> .
integrity	Logical. If <code>integrity = TRUE</code> , check for the output file integrity when <code>overwrite = FALSE</code> .

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 <code>manMCMedMiss:::SimSuffix()</code> .
overwrite	Logical. Overwrite existing output in <code>output_folder</code> .
integrity	Logical. If <code>integrity = TRUE</code> , check for the output file integrity when <code>overwrite = FALSE</code> .
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 <code>manMCMedMiss:::SimSuffix()</code> .
overwrite	Logical. Overwrite existing output in <code>output_folder</code> .
integrity	Logical. If <code>integrity = TRUE</code> , check for the output file integrity when <code>overwrite = FALSE</code> .
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 <code>manMCMedMiss:::SimSuffix()</code> .
overwrite	Logical. Overwrite existing output in <code>output_folder</code> .
integrity	Logical. If <code>integrity = TRUE</code> , check for the output file integrity when <code>overwrite = FALSE</code> .
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 <code>manMCMedMiss:::SimSuffix()</code> .
overwrite	Logical. Overwrite existing output in <code>output_folder</code> .
integrity	Logical. If <code>integrity = TRUE</code> , check for the output file integrity when <code>overwrite = FALSE</code> .
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 <code>manMCMedMiss:::SimSuffix()</code> .
overwrite	Logical. Overwrite existing output in <code>output_folder</code> .
integrity	Logical. If <code>integrity = TRUE</code> , check for the output file integrity when <code>overwrite = FALSE</code> .
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 <code>manMCMedMiss:::SimSuffix()</code> .
overwrite	Logical. Overwrite existing output in <code>output_folder</code> .
integrity	Logical. If <code>integrity = TRUE</code> , check for the output file integrity when <code>overwrite = FALSE</code> .
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 <code>manMCMedMiss:::SimSuffix()</code> .
overwrite	Logical. Overwrite existing output in <code>output_folder</code> .
integrity	Logical. If <code>integrity = TRUE</code> , check for the output file integrity when <code>overwrite = FALSE</code> .
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 <code>manMCMedMiss:::SimSuffix()</code> .
overwrite	Logical. Overwrite existing output in <code>output_folder</code> .
integrity	Logical. If <code>integrity = TRUE</code> , check for the output file integrity when <code>overwrite = FALSE</code> .
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 <code>manMCMedMiss:::SimSuffix()</code> .
overwrite	Logical. Overwrite existing output in <code>output_folder</code> .
integrity	Logical. If <code>integrity = TRUE</code> , check for the output file integrity when <code>overwrite = FALSE</code> .
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	<i>Summarize Simulations</i>
-----	------------------------------

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	<i>Bind Results</i>
---------	---------------------

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	<i>Summarize Simulations - SimJointSigML</i>
---------------	--

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	<i>Summarize Simulations - SimMCMI</i>
---------	--

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	<i>Summarize Simulations - SimMCML</i>
---------	--

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	<i>Summarize Simulations - SimMINB</i>
---------	--

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	<i>Summarize Simulations - SimNBStackedMI</i>
----------------	---

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	<i>Descriptive Statistics of Type I Error Rate, Statistical Power, and Miss Rate</i>
-------------	--

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	<i>Conditional Inference Tree</i>
------	-----------------------------------

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

- * **Compression Functions**
 - Compress, [13](#)
- * **Conditional Inference Tree Functions**
 - Tree, [54](#)
- * **Confidence Interval Functions**
 - MCMI, [24](#)
 - MCML, [25](#)
 - MINB, [26](#)
 - NBMI, [27](#)
 - NBML, [28](#)
 - NBStackedMI, [29](#)
- * **Data Generation Functions**
 - AmputeData, [3](#)
 - GenData, [21](#)
 - ImputeData, [22](#)
- * **Figure Functions**
 - FigBoxPlot, [15](#)
 - FigScatterPlot, [16](#)
- * **Model Fitting Functions**
 - FitModelIndirect, [17](#)
 - FitModelMI, [17](#)
 - FitModelML, [19](#)
- * **Multiple Imputation Functions**
 - MICombine, [25](#)
- * **Significance Test Functions**
 - JointSigMI, [22](#)
 - JointSigML, [23](#)
- * **Table Functions**
 - TabDescribe, [54](#)
- * **ampute**
 - AmputeData, [3](#)
 - CheckAmputeData, [4](#)
 - SimAmputeData, [33](#)
- * **check**
 - Check, [4](#)
 - CheckAmputeData, [4](#)
 - CheckFitModelMI, [5](#)
 - CheckFitModelML, [6](#)
 - CheckGenData, [6](#)
 - CheckImputeData, [7](#)
 - CheckJointSigMI, [8](#)
 - CheckJointSigML, [8](#)
 - CheckMCMI, [9](#)
 - CheckMCML, [10](#)
 - CheckMINB, [10](#)
 - CheckNBMI, [11](#)
 - CheckNBML, [12](#)
 - CheckStackedMINB, [12](#)
 - SimImputeData, [37](#)
- * **ci**
 - CheckMCMI, [9](#)
 - CheckMCML, [10](#)
 - CheckMINB, [10](#)
 - CheckNBMI, [11](#)
 - CheckNBML, [12](#)
 - CheckStackedMINB, [12](#)
 - MCMI, [24](#)
 - MCML, [25](#)
 - MINB, [26](#)
 - NBMI, [27](#)
 - NBML, [28](#)
 - NBStackedMI, [29](#)
 - SimMCMI, [40](#)
 - SimMCML, [41](#)
 - SimMINB, [42](#)
 - SimNBMI, [43](#)
 - SimNBML, [44](#)
 - SimNBStackedMI, [45](#)
 - SumMCMI, [50](#)
 - SumMCML, [50](#)
 - SumMINB, [51](#)
 - SumNBMI, [52](#)
 - SumNBML, [52](#)
 - SumNBStackedMI, [53](#)
- * **compress**
 - Compress, [13](#)
- * **data**
 - elliott2007, [14](#)

- params, [30](#)
 - results, [30](#)
- * **figure**
 - FigBoxPlot, [15](#)
 - FigScatterPlot, [16](#)
- * **fit**
 - CheckFitModelMI, [5](#)
 - CheckFitModelML, [6](#)
 - FitModelIndirect, [17](#)
 - FitModelMI, [17](#)
 - FitModelML, [19](#)
 - SimFitModelMI, [34](#)
 - SimFitModelML, [35](#)
- * **gendata**
 - AmputeData, [3](#)
 - CheckAmputeData, [4](#)
 - CheckGenData, [6](#)
 - CheckImputeData, [7](#)
 - GenData, [21](#)
 - ImputeData, [22](#)
 - SimAmputeData, [33](#)
 - SimGenData, [36](#)
 - SimImputeData, [37](#)
- * **impute**
 - CheckImputeData, [7](#)
 - ImputeData, [22](#)
 - SimImputeData, [37](#)
- * **manMCMedMiss**
 - AmputeData, [3](#)
 - Check, [4](#)
 - CheckAmputeData, [4](#)
 - CheckFitModelMI, [5](#)
 - CheckFitModelML, [6](#)
 - CheckGenData, [6](#)
 - CheckImputeData, [7](#)
 - CheckJointSigMI, [8](#)
 - CheckJointSigML, [8](#)
 - CheckMCMI, [9](#)
 - CheckMCML, [10](#)
 - CheckMINB, [10](#)
 - CheckNBMI, [11](#)
 - CheckNBML, [12](#)
 - CheckStackedMINB, [12](#)
 - Compress, [13](#)
 - FigBoxPlot, [15](#)
 - FigScatterPlot, [16](#)
 - FitModelIndirect, [17](#)
 - FitModelMI, [17](#)
 - FitModelML, [19](#)
 - GenData, [21](#)
 - ImputeData, [22](#)
 - JointSigMI, [22](#)
 - JointSigML, [23](#)
 - MCMI, [24](#)
 - MCML, [25](#)
 - MICombine, [25](#)
 - MINB, [26](#)
 - NBBI, [27](#)
 - NBML, [28](#)
 - NBStackedMI, [29](#)
 - Sim, [32](#)
 - SimAmputeData, [33](#)
 - SimFitModelMI, [34](#)
 - SimFitModelML, [35](#)
 - SimFN, [36](#)
 - SimGenData, [36](#)
 - SimImputeData, [37](#)
 - SimJointSigMI, [38](#)
 - SimJointSigML, [39](#)
 - SimMCMI, [40](#)
 - SimMCML, [41](#)
 - SimMINB, [42](#)
 - SimNBBI, [43](#)
 - SimNBML, [44](#)
 - SimNBStackedMI, [45](#)
 - SimProj, [46](#)
 - Sum, [47](#)
 - SumBind, [47](#)
 - SumHit, [48](#)
 - SumJointSigMI, [48](#)
 - SumJointSigML, [49](#)
 - SumMCMI, [50](#)
 - SumMCML, [50](#)
 - SumMINB, [51](#)
 - SumNBBI, [52](#)
 - SumNBML, [52](#)
 - SumNBStackedMI, [53](#)
 - TabDescribe, [54](#)
 - Tree, [54](#)
- * **mediation**
 - AmputeData, [3](#)
 - CheckAmputeData, [4](#)
 - CheckFitModelMI, [5](#)
 - CheckFitModelML, [6](#)
 - CheckGenData, [6](#)
 - CheckJointSigMI, [8](#)

- CheckJointSigML, 8
- CheckMCMCMI, 9
- CheckMCML, 10
- CheckMINB, 10
- CheckNBMI, 11
- CheckNBML, 12
- CheckStackedMINB, 12
- FitModelIndirect, 17
- FitModelMI, 17
- FitModelML, 19
- GenData, 21
- JointSigMI, 22
- JointSigML, 23
- MCMCMI, 24
- MCML, 25
- MINB, 26
- NBBI, 27
- NBML, 28
- NBStackedMI, 29
- SimAmputeData, 33
- SimFitModelMI, 34
- SimFitModelML, 35
- SimGenData, 36
- SimJointSigMI, 38
- SimJointSigML, 39
- SimMCMCMI, 40
- SimMCML, 41
- SimMINB, 42
- SimNBBI, 43
- SimNBML, 44
- SimNBStackedMI, 45
- SumJointSigMI, 48
- SumJointSigML, 49
- SumMCMCMI, 50
- SumMCML, 50
- SumMINB, 51
- SumNBBI, 52
- SumNBML, 52
- SumNBStackedMI, 53
- * mi**
 - CheckFitModelMI, 5
 - CheckJointSigMI, 8
 - CheckMCMCMI, 9
 - CheckMINB, 10
 - CheckNBBI, 11
 - CheckStackedMINB, 12
 - FitModelMI, 17
 - JointSigMI, 22
- MCMCMI, 24
- MICombine, 25
- MINB, 26
- NBBI, 27
- NBStackedMI, 29
- SimFitModelMI, 34
- SimJointSigMI, 38
- SimMCMCMI, 40
- SimMINB, 42
- SimNBBI, 43
- SimNBStackedMI, 45
- SumJointSigMI, 48
- SumMCMCMI, 50
- SumMINB, 51
- SumNBBI, 52
- SumNBStackedMI, 53
- * ml**
 - CheckFitModelML, 6
 - CheckJointSigML, 8
 - CheckMCML, 10
 - CheckNBML, 12
 - FitModelIndirect, 17
 - FitModelML, 19
 - JointSigML, 23
 - MCML, 25
 - NBML, 28
 - SimFitModelML, 35
 - SimJointSigML, 39
 - SimMCML, 41
 - SimNBML, 44
 - SumJointSigML, 49
 - SumMCML, 50
 - SumNBML, 52
- * normal**
 - CheckGenData, 6
 - GenData, 21
 - SimGenData, 36
- * parameters**
 - params, 30
- * results**
 - results, 30
- * simulation**
 - Sim, 32
 - SimAmputeData, 33
 - SimFitModelMI, 34
 - SimFitModelML, 35
 - SimFN, 36
 - SimGenData, 36

- SimImputeData, 37
- SimJointSigMI, 38
- SimJointSigML, 39
- SimMCMCMI, 40
- SimMCML, 41
- SimMINB, 42
- SimNBMI, 43
- SimNBML, 44
- SimNBStackedMI, 45
- SimProj, 46
- SumMCMCMI, 50
- SumMINB, 51
- SumNBMI, 52
- SumNBStackedMI, 53
- * **summary**
 - Sum, 47
 - SumBind, 47
 - SumHit, 48
 - SumJointSigMI, 48
 - SumJointSigML, 49
 - SumMCMCMI, 50
 - SumMCML, 50
 - SumMINB, 51
 - SumNBMI, 52
 - SumNBML, 52
 - SumNBStackedMI, 53
- * **table**
 - TabDescribe, 54
- * **test**
 - CheckJointSigMI, 8
 - CheckJointSigML, 8
 - JointSigMI, 22
 - JointSigML, 23
 - SimJointSigMI, 38
 - SimJointSigML, 39
 - SumJointSigMI, 48
 - SumJointSigML, 49
- * **tree**
 - Tree, 54
- AmputeData, 3, 21, 22
- Check, 4
- CheckAmputeData, 4
- CheckFitModelMI, 5
- CheckFitModelML, 6
- CheckGenData, 6
- CheckImputeData, 7
- CheckJointSigMI, 8
- CheckJointSigML, 8
- CheckMCMCMI, 9
- CheckMCML, 10
- CheckMINB, 10
- CheckNBMI, 11
- CheckNBML, 12
- CheckStackedMINB, 12
- Compress, 13
- elliott2007, 14
- FigBoxPlot, 15, 16
- FigScatterPlot, 15, 16
- FitModelIndirect, 17, 19, 20
- FitModelMI, 17, 17, 20
- FitModelML, 17, 19, 19
- GenData, 4, 21, 22
- ImputeData, 4, 21, 22
- JointSigMI, 22, 24
- JointSigML, 23, 23
- MCMCMI, 24, 25, 27–30
- MCML, 24, 25, 27–30
- MICombine, 25
- MINB, 24, 25, 26, 28–30
- NBBI, 24, 25, 27, 27, 29, 30
- NBML, 24, 25, 27, 28, 28, 30
- NBStackedMI, 24, 25, 27–29, 29
- params, 30
- results, 30
- Sim, 32
- SimAmputeData, 33
- SimFitModelMI, 34
- SimFitModelML, 35
- SimFN, 36
- SimGenData, 36
- SimImputeData, 37
- SimJointSigMI, 38
- SimJointSigML, 39
- SimMCMCMI, 40
- SimMCML, 41
- SimMINB, 42
- SimNBBI, 43

SimNBML, [44](#)
SimNBStackedMI, [45](#)
SimProj, [46](#)
Sum, [47](#)
SumBind, [47](#)
SumHit, [48](#)
SumJointSigMI, [48](#)
SumJointSigML, [49](#)
SumMCMI, [50](#)
SumMCML, [50](#)
SumMINB, [51](#)
SumNBMI, [52](#)
SumNBML, [52](#)
SumNBStackedMI, [53](#)

TabDescribe, [54](#)
Tree, [54](#)