Package 'RMediation'

June 20, 2026

Type Package

Title Mediation Analysis Confidence Intervals

Version 1.5.0

Date 2026-06-18

Author Davood Tofighi [aut, cre] (ORCID:
<https://orcid.org/0000-0001-8523-7776>)

Maintainer Davood Tofighi <dtofighi@gmail.com>

Description Computes confidence intervals for nonlinear functions of model parameters (e.g., product of k coefficients) in single-level and multilevel structural equation
models. Methods include the distribution of the product, Monte Carlo simulation, and bootstrap methods. It also performs the Model-Based Constrained Optimization (MBCO) procedure for hypothesis testing of indirect effects.
References:
Tofighi, D., and MacKinnon, D. P. (2011). RMediation: An R package for mediation analysis confidence intervals. Behavior Research Methods, 43, 692-700. <doi:10.3758/s13428-011-0076-x>;
Tofighi, D., and Kelley, K. (2020). Improved inference in mediation analysis: Introducing the model-based constrained optimization procedure.
Psychological Methods, 25(4), 496-515. <doi:10.1037/met0000259>;
Tofighi, D. (2020). Bootstrap Model-Based Constrained Optimization Tests of Indirect Effects. Frontiers in Psychology, 10, 2989.
<doi:10.3389/fpsyg.2019.02989>.

License GPL (>= 3) | file LICENSE

URL https://data-wise.github.io/rmediation/,
https://github.com/data-wise/rmediation

BugReports https://github.com/data-wise/rmediation/issues

Depends R (>= 4.1.0)

Imports checkmate (>= 2.1.0), graphics (>= 4.1.0), grDevices (>= 4.1.0), lavaan (>= 0.5-20), MASS (>= 7.3), methods, S7

Suggests knitr, medfit (>= 0.2.0), OpenMx (>= 2.13), rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Config/testthat/edition 3

Config/Needs/website pkgdown, quarto

NeedsCompilation no

Repository CRAN

Date/Publication 2026-06-20 02:40:02 UTC

Contents

cdf *Cumulative Distribution Function*

Description

Generic function for computing cumulative distribution function.

Usage

```
cdf(object, ...)
```

Arguments

`object`	A distribution object.
`...`	Additional arguments passed to methods.

ci *Confidence Interval*

Description

Generic function for computing confidence intervals.

Usage

```
ci(mu, ...)
```

Arguments

`mu`	A distribution object or numeric vector of means.
`...`	Additional arguments passed to methods.

`ci_mediation_data` *Confidence Interval for MediationData Objects*

Description

Computes confidence intervals for the indirect effect from a medfit MediationData object using RMediation's methods (DOP, Monte Carlo, etc.).

Usage

```
ci_mediation_data(mu, level = 0.95, type = "dop", n.mc = 1e+05, ...)

ci_serial_mediation_data(mu, level = 0.95, type = "MC", n.mc = 1e+05, ...)
```

Arguments

`mu`	A MediationData object from the medfit package
`level`	Confidence level (default 0.95 for 95% CI)
`type`	Type of CI method: "dop" (Distribution of Product), "MC" (Monte Carlo), or "asymp" (asymptotic normal)
`n.mc`	Number of Monte Carlo samples (for type = "MC")
`...`	Additional arguments passed to underlying methods

Details

This method extracts the a and b path coefficients from the MediationData object, along with their standard errors and covariance, and computes confidence intervals using RMediation's methods.

Method Options:

- **"dop"**: Distribution of Product method. Uses the exact or approximate distribution of the product of two normal random variables. Recommended for most applications.
- **"MC"**: Monte Carlo simulation. Samples from the joint distribution of a and b to estimate the CI. Use `n.mc` to control precision.
- **"asymp"**: Asymptotic normal approximation using the delta method. Fast but may be inaccurate for small samples or non-normal distributions.

Value

A list with components:

`CI`	The confidence interval (lower, upper)
`Estimate`	Point estimate of indirect effect (a*b)
`SE`	Standard error of indirect effect
`type`	Method used for CI computation

See Also

ci for the generic function, MediationData for the input class, ProductNormal for the underlying distribution class

Examples

```
## Not run:
library(medfit)
library(RMediation)

# Fit mediation models
fit_m <- lm(M ~ X + C, data = mydata)
fit_y <- lm(Y ~ X + M + C, data = mydata)

# Extract mediation structure
med_data <- extract_mediation(fit_m, model_y = fit_y,
                              treatment = "X", mediator = "M")

# Compute CI using Distribution of Product
ci(med_data, type = "dop")

# Compute CI using Monte Carlo
ci(med_data, type = "MC", n.mc = 10000)

## End(Not run)
```

---

dist_quantile *Distribution Quantile Function*

Description

Compute quantiles for distribution objects. This function computes quantiles for product normal distributions, not data quantiles (use `stats::quantile` for data).

Usage

```
dist_quantile(object, ...)
```

Arguments

`object`	A distribution object (e.g., ProductNormal).
`...`	Additional arguments passed to methods.

`is_valid_for_computation` *Method to check if ProductNormal object is properly specified for computation*

Description

Method to check if ProductNormal object is properly specified for computation

Usage

```
is_valid_for_computation(object)
```

Arguments

`object`	A ProductNormal object

Value

TRUE if object is valid for computation methods

`mbco` *Model-based Constrained Optimization (MBCO) Chi-squared Test*

Description

This function computes asymptotic MBCO chi-squared test for a smooth function of model parameters including a function of indirect effects.

Usage

```
mbco(h0, h1, ...)
```

Arguments

`h0`	An `OpenMx` model estimated under a null hypothesis, which is a more constrained model
`h1`	An `OpenMx` model estimated under an alternative hypothesis, which is a less constrained model. This is usually a model hypothesized by a researcher.
`...`	Additional arguments.

Value

An object of class `MBCOResult` containing

`statistic`	asymptotic chi-squared test statistic value
`df`	chi-squared df
`p_value`	chi-squared p-value computed based on the method specified by the argument `type`
`type`	The type of test performed

Author(s)

Davood Tofighi <dtofighi@gmail.com>

Examples

```
## Not run:
data(memory_exp)
memory_exp$x <- as.numeric(memory_exp$x)-1 # manually creating dummy codes
endVar <- c('x', 'repetition', 'imagery', 'recall')
manifests <- c('x', 'repetition', 'imagery', 'recall')
full_model <- OpenMx::mxModel(
 "memory_example",
 type = "RAM",
 manifestVars = manifests,
 OpenMx::mxPath(
   from = "x",
   to = endVar,
   arrows = 1,
   free = TRUE,
   values = .2,
   labels = c("a1", "a2", "cp")
 ),
 OpenMx::mxPath(
   from = 'repetition',
  to = 'recall',
  arrows = 1,
   free = TRUE,
   values = .2,
   labels = 'b1'
 ),
 OpenMx::mxPath(
   from = 'imagery',
 to = 'recall',
 arrows = 1,
 free = TRUE,
 values = .2,
 labels = "b2"
),
OpenMx::mxPath(
 from = manifests,
 arrows = 2,
```

```
 free = TRUE,
 values = .8
),
OpenMx::mxPath(
 from = "one",
 to = endVar,
 arrows = 1,
 free = TRUE,
 values = .1
),
OpenMx::mxAlgebra(a1 * b1, name = "ind1"),
OpenMx::mxAlgebra(a2 * b2, name = "ind2"),
OpenMx::mxCI("ind1", type = "both"),
OpenMx::mxCI("ind2", type = "both"),
OpenMx::mxData(observed = memory_exp, type = "raw")
)
## Reduced  Model for indirect effect: a1*b1
null_model1 <- OpenMx::mxModel(
model= full_model,
name = "Null Model 1",
OpenMx::mxConstraint(ind1 == 0, name = "ind1_eq0_constr")
)
full_model <- OpenMx::mxTryHard(full_model, checkHess=FALSE, silent = TRUE )
null_model1 <- OpenMx::mxTryHard(null_model1, checkHess=FALSE, silent = TRUE )
mbco(null_model1,full_model)

## End(Not run)
```

`MBCOResult` *MBCO Result Class*

Description

A class representing the results of a Model-Based Constrained Optimization (MBCO) test.

Usage

```
MBCOResult(
  statistic = integer(0),
  df = integer(0),
  p_value = integer(0),
  type = character(0)
)
```

Arguments

`statistic`	Numeric test statistic value.
`df`	Numeric degrees of freedom.
`p_value`	Numeric p-value.
`type`	Character string indicating the type of test.

`medci` *Confidence Interval for the Mediated Effect*

Description

Produces confidence intervals for the mediated effect and the product of two normal random variables

Produces confidence intervals for the mediated effect and the product of two normal random variables

Usage

```
medci(
  mu.x,
  mu.y,
  se.x,
  se.y,
  rho = 0,
  alpha = 0.05,
  type = "dop",
  plot = FALSE,
  plotCI = FALSE,
  n.mc = 1e+05,
  ...
)

medci(
  mu.x,
  mu.y,
  se.x,
  se.y,
  rho = 0,
  alpha = 0.05,
  type = "dop",
  plot = FALSE,
  plotCI = FALSE,
  n.mc = 1e+05,
  ...
)
```

Arguments

`mu.x`	mean of x
`mu.y`	mean of y
`se.x`	standard error (deviation) of x
`se.y`	standard error (deviation) of y

rho	correlation between x and y, where -1 <rho < 1. The default value is 0.
alpha	significance level for the confidence interval. The default value is .05.
type	method used to compute confidence interval. It takes on the values "dop" (default), "MC", "asymp" or "all"
plot	when TRUE, plots the distribution of n.mc data points from the distribution of product of two normal random variables using the density estimates provided by the function density. The default value is FALSE.
plotCI	when TRUE, overlays a confidence interval with error bars on the plot for the mediated effect. Note that to obtain the CI plot, one must also specify plot="TRUE". The default value is FALSE.
n.mc	when type="MC", n.mc determines the sample size for the Monte Carlo method. The default sample size is 1E5.
...	additional arguments to be passed on to the function.

Details

This function returns a confidence interval at significance level α for the mediated effect (product of two normal random variables). To obtain a confidence interval using a specific method, the argument type should be specified. The default is type="dop", which uses the code we wrote in R to implement the distribution of product of the coefficients method described by Meeker and Escobar (1994) to evaluate the CDF of the distribution of product. type="MC" uses the Monte Carlo approach to compute the confidence interval (Tofighi & MacKinnon, 2011). type="asymp" produces the asymptotic normal confidence interval. Note that except for the Monte Carlo method, the standard error for the indirect effect is based on the analytical results by Craig (1936):

$$SE = \sqrt{\sigma_y^2\mu_x^2 + \sigma_x^2\mu_y^2 + 2\mu_x\mu_y\rho\sigma_x\sigma_y + \sigma_x^2\sigma_y^2 + \sigma_x^2\sigma_y^2\rho^2}$$

where σ_x and σ_y are the standard errors of x and y, respectively. In addition, the estimate of the indirect effect is $\mu_x\mu_y + \sigma_{xy}$, where σ_{xy} is the covariance between x and y. The argument type="all" prints confidence intervals using all available methods.

This function returns a $(1-\alpha)\%$ confidence interval for the mediated effect (product of two normal random variables). To obtain a confidence interval using a specific method, the argument type should be specified. The default is type="dop", which uses the code we wrote in R to implement the distribution of product of the coefficients method described by Meeker and Escobar (1994) to evaluate the CDF of the distribution of product. type="MC" uses the Monte Carlo approach to compute the confidence interval (Tofighi & MacKinnon, 2011). type="asymp" produces the asymptotic normal confidence interval. Note that except for the Monte Carlo method, the standard error for the indirect effect is based on the analytical results by Craig (1936):

$$\sqrt{(se.y^2\mu.x^2 + se.x^2\mu.y^2 + 2\mu.x\mu.y\rho se.x se.y + se.x^2 se.y^2 + se.x^2 se.y^2\rho^2)}$$

. In addition, the estimate of indirect effect is $\mu.x\mu.y + \sigma.xy$; type="all" prints confidence intervals using all four options.

Value

A vector of lower confidence limit and upper confidence limit. When type is "prodclin" (default), "DOP", "MC" or "asymp", medci returns a list that contains:

CI	a vector of lower and upper confidence limits (at significance level α),
Estimate	a point estimate of the quantity of interest,
SE	standard error of the quantity of interest,
MC Error	When type="MC", error of the Monte Carlo estimate.

Note that when type="all", medci returns a list of *four* objects, each of which a list that contains the results produced by each method as described above.

A vector of lower confidence limit and upper confidence limit. When type is "prodclin" (default), "DOP", "MC" or "asymp", medci returns a list that contains:

(\eqn{1-\alpha})% CI	a vector of lower and upper confidence limits,
Estimate	a point estimate of the quantity of interest,
SE	standard error of the quantity of interest,
MC Error	When type="MC", error of the Monte Carlo estimate.

Note that when type="all", medci returns a list of *four* objects, each of which a list that contains the results produced by each method as described above.

Author(s)

Davood Tofighi <dtofighi@gmail.com>

References

Craig, C. C. (1936). On the frequency function of xy. *The Annals of Mathematical Statistics*, **7**, 1–15.

MacKinnon, D. P., Fritz, M. S., Williams, J., and Lockwood, C. M. (2007). Distribution of the product confidence limits for the indirect effect: Program PRODCLIN. *Behavior Research Methods*, **39**, 384–389.

Meeker, W. and Escobar, L. (1994). An algorithm to compute the CDF of the product of two normal random variables. *Communications in Statistics: Simulation and Computation*, **23**, 271–280.

Tofighi, D. and MacKinnon, D. P. (2011). RMediation: An R package for mediation analysis confidence intervals. *Behavior Research Methods*, **43**, 692–700. doi:10.3758/s134280110076x

Craig, C. C. (1936). On the frequency function of xy. *The Annals of Mathematical Statistics*, **7**, 1–15.

MacKinnon, D. P., Fritz, M. S., Williams, J., and Lockwood, C. M. (2007). Distribution of the product confidence limits for the indirect effect: Program PRODCLIN. *Behavior Research Methods*, **39**, 384–389.

Meeker, W. and Escobar, L. (1994). An algorithm to compute the CDF of the product of two normal random variables. *Communications in Statistics: Simulation and Computation*, **23**, 271–280.

Tofighi, D. and MacKinnon, D. P. (2011). RMediation: An R package for mediation analysis confidence intervals. *Behavior Research Methods*, **43**, 692–700. doi:10.3758/s134280110076x

See Also

qprodnormal pprodnormal ci RMediation-package

qprodnormal pprodnormal ci RMediation-package

Examples

```
## Example 1
res <- medci(
  mu.x = .2, mu.y = .4, se.x = 1, se.y = 1, rho = 0, alpha = .05,
  type = "dop", plot = TRUE, plotCI = TRUE
)
## Example 2
res <- medci(mu.x = .2, mu.y = .4, se.x = 1, se.y = 1, rho = 0,
 alpha = .05, type = "all", plot = TRUE, plotCI = TRUE)
## Example 1
res <- medci(
  mu.x = .2, mu.y = .4, se.x = 1, se.y = 1, rho = 0, alpha = .05,
  type = "dop", plot = TRUE, plotCI = TRUE
)
## Example 2
res <- medci(mu.x = .2, mu.y = .4, se.x = 1, se.y = 1, rho = 0,
 alpha = .05, type = "all", plot = TRUE, plotCI = TRUE)
```

---

memory_exp *Memory Experiment Data Description from MacKinnon et al., 2018*

Description

Data were obtained from eight replicated experiments. The data were collected on the first day of class as part of the first Dr. MacKinnon's (2018) classroom teaching. The pedagogical value of the experiment was that students would have first-hand knowledge of the experiment thereby increasing their understanding of course concepts. Permission to use the data was obtained from the university Institutional Review Board.

Usage

```
data(memory_exp)
```

Format

A data frame with 369 rows and 5 variables:

study Replication ID, ranges from 1 to 8

repetition Use of repetition rehearsal technique, ranges from 1 to 12

recall Total words recalled out of 20 words, ranges from 3 to 20

imagery Use of imagery rehearsal technique on a 1 to 9 scale

x A factor with two levels: "repetition" (primary rehearsal) or "imagery" (secondary rehearsal)

Note

If you use the data set, please cite the original article by MacKinnon et al. (2018) cited below.

Source

doi:10.1037/met0000174.supp

References

MacKinnon, D. P., Valente, M. J., & Wurpts, I. C. (2018). Benchmark validation of statistical models: Application to mediation analysis of imagery and memory. Psychological Methods, 23, 654–671. doi:10.1037/met0000174

`pMC` *Probability (percentile) for the Monte Carlo Sampling Distribution of a nonlinear function of coefficients estimates*

Description

This function returns a probability corresponding to the quantile `q`.

Usage

```
pMC(q, mu, Sigma, quant, lower.tail = TRUE, n.mc = 1e+05, ...)
```

Arguments

`q`	quantile
`mu`	a vector of means (e.g., coefficient estimates) for the normal random variables. A user can assign a name to each mean value, e.g., `mu=c(b1=.1,b2=3)`; otherwise, the coefficient names are assigned automatically as follows: `b1,b2,...`.
`Sigma`	either a covariance matrix or a vector that stacks all the columns of the lower triangle variance–covariance matrix one underneath the other.
`quant`	quantity of interest, which is a nonlinear/linear function of the model parameters. Argument `quant` is a formula that **must** start with the symbol "tilde" (`~`): e.g., `~b1*b2*b3*b4`. The names of coefficients must conform to the names provided in the argument `mu` or to the default names, i.e., `b1,b2,...`.
`lower.tail`	logical; if `TRUE` (default), the probability is $P[quant < q]$; otherwise, $P[quant > q]$
`n.mc`	Monte Carlo sample size (default: 1e5). Larger values provide more precision but take longer to compute.
`...`	additional arguments.

Value

scalar probability value.

Author(s)

Davood Tofighi <dtofighi@gmail.com>

References

Tofighi, D. and MacKinnon, D. P. (2011). RMediation: An R package for mediation analysis confidence intervals. *Behavior Research Methods*, **43**, 692–700. doi:10.3758/s13428-011-0076-x

Examples

```
pMC(.2,
  mu = c(b1 = 1, b2 = .7, b3 = .6, b4 = .45), Sigma = c(.05, 0, 0, 0, .05, 0, 0, .03, 0, .03),
  quant = ~ b1 * b2 * b3 * b4
)
```

---

pprodnormal *Percentile for the Distribution of Product of Two Normal Variables*

Description

Generates percentiles (100 based quantiles) for the distribution of product of two normal random variables and the mediated effect

Usage

```
pprodnormal(
  q,
  mu.x,
  mu.y,
  se.x = 1,
  se.y = 1,
  rho = 0,
  lower.tail = TRUE,
  type = "dop",
  n.mc = 1e+05
)
```

Arguments

q	quantile or value of the product
mu.x	mean of x
mu.y	mean of y
se.x	standard error (deviation) of x
se.y	standard error (deviation) of y
rho	correlation between x and y, where -1 <rho < 1. The default value is 0.

lower.tail	logical; if TRUE (default), the probability is $P(XY \leq q)$; otherwise, $P(XY > q)$
type	method used to compute confidence interval. It takes on the values "dop" (default), "MC", "asymp" or "all"
n.mc	Monte Carlo sample size when type="MC" (default: 1e5). Larger values provide more precision but take longer to compute.

Details

This function returns the percentile (probability) and the associated error for the distribution of product of mediated effect (two normal random variables). To obtain a percentile using a specific method, the argument type should be specified. The default method is type="dop", which is based on the method described by Meeker and Escobar (1994) to evaluate the CDF of the distribution of product of two normal random variables. type="MC" uses the Monte Carlo approach (Tofighi & MacKinnon, 2011). type="all" prints percentiles using all three options. For the method type="dop", the error is the modulus of absolute error for the numerical integration (for more information see Meeker and Escobar, 1994). For type="MC", the error refers to the Monte Carlo error.

Value

An object of the type list that contains the following values:

p	probability (percentile) corresponding to quantile q
error	estimate of the absolute error

Author(s)

Davood Tofighi <dtofighi@gmail.com>

References

Tofighi, D. and MacKinnon, D. P. (2011). RMediation: An R package for mediation analysis confidence intervals. *Behavior Research Methods*, **43**, 692–700. doi:10.3758/s13428-011-0076-x

See Also

medci RMediation-package

Examples

```
pprodnormal(q = 0, mu.x = .5, mu.y = .3, se.x = 1, se.y = 1, rho = 0, type = "all")
```

ProductNormal *ProductNormal Class*

Description

Represents the distribution of the product of normal random variables.

Usage

```
ProductNormal(mu = integer(0), Sigma = integer(0))
```

Arguments

mu	Numeric vector of means.
Sigma	Covariance matrix.

ProductNormal2 *Enhanced ProductNormal constructor with better validation*

Description

Enhanced ProductNormal constructor with better validation

Usage

```
ProductNormal2(mu, Sigma, validate = TRUE)
```

Arguments

mu	Numeric vector of means
Sigma	Covariance matrix
validate	Whether to run additional validation (default: TRUE)

`ProductNormal_from_lavaan` *Utility function to create ProductNormal from lavaan parameter estimates*

Description

Utility function to create ProductNormal from lavaan parameter estimates

Usage

```
ProductNormal_from_lavaan(lavaan_model, param_names)
```

Arguments

`lavaan_model`	A fitted lavaan model object
`param_names`	Names of parameters to include in the product (e.g., c("a", "b"))

Value

A ProductNormal object

`qMC` *Quantile for the Monte Carlo Sampling Distribution of a nonlinear function of coefficients estimates*

Description

This function returns a quantile corresponding to the probability `p`.

Usage

```
qMC(p, mu, Sigma, quant, n.mc = 1e+05, ...)
```

Arguments

`p`	probability.
`mu`	a vector of means (e.g., coefficient estimates) for the normal random variables. A user can assign a name to each mean value, e.g., `mu=c(b1=.1,b2=3)`; otherwise, the coefficient names are assigned automatically as follows: `b1,b2,...`.
`Sigma`	either a covariance matrix or a vector that stacks all the columns of the lower triangle variance–covariance matrix one underneath the other.

quant	quantity of interest, which is a nonlinear/linear function of the model parameters. Argument quant is a formula that **must** start with the symbol "tilde" (~): e.g., ~b1*b2*b3*b4. The names of coefficients must conform to the names provided in the argument mu or to the default names, i.e., b1,b2,....
n.mc	Monte Carlo sample size (default: 1e5). Larger values provide more precision but take longer to compute.
...	additional arguments.

Value

scalar quantile value.

Author(s)

Davood Tofighi <dtofighi@gmail.com>

References

Tofighi, D. and MacKinnon, D. P. (2011). RMediation: An R package for mediation analysis confidence intervals. *Behavior Research Methods*, **43**, 692–700. doi:10.3758/s13428-011-0076-x

Examples

```
qMC(.05,
  mu = c(b1 = 1, b2 = .7, b3 = .6, b4 = .45), Sigma = c(.05, 0, 0, 0, .05, 0, 0, .03, 0, .03),
   quant = ~ b1 * b2 * b3 * b4
)
```

---

qprodnormal *Quantile for the Distribution of Product of Two Normal Variables*

Description

Generates quantiles for the distribution of product of two normal random variables

Usage

```
qprodnormal(
  p,
  mu.x,
  mu.y,
  se.x,
  se.y,
  rho = 0,
  lower.tail = TRUE,
  type = "dop",
  n.mc = 1e+05
)
```

Arguments

p	probability
mu.x	mean of x
mu.y	mean of y
se.x	standard error (deviation) of x
se.y	standard error (deviation) of y
rho	correlation between x and y, where -1 <rho < 1. The default value is 0.
lower.tail	logical; if TRUE (default), the probability is $P(XY \leq q)$; otherwise, $P(XY > q)$
type	method used to compute confidence interval. It takes on the values "dop" (default), "MC", "asymp" or "all"
n.mc	Monte Carlo sample size when type="MC" (default: 1e5). Larger values provide more precision but take longer to compute.

Details

This function returns a quantile and the associated error (accuracy) corresponding the requested percentile (probability) p of the distribution of product of mediated effect (product of two normal random variables). To obtain a quantile using a specific method, the argument type should be specified. The default method is type="dop", which uses the method described by Meeker and Escobar (1994) to evaluate the CDF of the distribution of product of two normal variables. type="MC" uses the Monte Carlo approach (Tofighi & MacKinnon, 2011). type="all" prints quantiles using all three options. For the method type="dop", the error is the modulus of absolute error for the numerical integration (for more information see Meeker and Escobar, 1994). For type="MC", the error refers to the Monte Carlo error.

Value

An object of the type list that contains the following values:

q	quantile corresponding to probability p
error	estimate of the absolute error

Author(s)

Davood Tofighi <dtofighi@gmail.com>

References

Tofighi, D. and MacKinnon, D. P. (2011). RMediation: An R package for mediation analysis confidence intervals. *Behavior Research Methods*, **43**, 692–700. doi:10.3758/s13428-011-0076-x

See Also

medci RMediation-package

Examples

```
## lower tail
qprodnormal(
  p = .1, mu.x = .5, mu.y = .3, se.x = 1, se.y = 1, rho = 0,
  lower.tail = TRUE, type = "all"
)
## upper tail
qprodnormal(
  p = .1, mu.x = .5, mu.y = .3, se.x = 1, se.y = 1, rho = 0,
  lower.tail = FALSE, type = "all"
)
```

---

tidy *Tidy generic function*

Description

Tidy generic function

Usage

```
tidy(x, ...)
```

Arguments

x	An object to tidy
...	Additional arguments passed to methods

Value

A tibble or data.frame with tidy output

---

tidy.logLik *Creates a data.frame for a log-likelihood object*

Description

Creates a data.frame for a log-likelihood object

Usage

```
## S3 method for class 'logLik'
tidy(x, ...)
```

Arguments

x	x A log-likelihood object, typically returned by logLik.
...	Additional arguments (not used)

Value

A data.frame with columns:

term The term name

estimate The log-likelihood value

df The degrees of freedom

Author(s)

Davood Tofighi <dtofighi@gmail.com>

See Also

logLik

Examples

```
fit <- lm(mpg ~ wt, data = mtcars)
logLik_fit <- logLik(fit)
tidy(logLik_fit)
```

---

utils_validation *Enhanced validation and utility functions for ProductNormal class*

Description

Enhanced validation and utility functions for ProductNormal class

---

validate_ProductNormal *Additional validation for ProductNormal objects*

Description

Additional validation for ProductNormal objects

Usage

```
validate_ProductNormal(object, verbose = FALSE)
```

Arguments

`object`	A ProductNormal object
`verbose`	Whether to show detailed validation messages

Value

TRUE if valid, throws error if invalid

Index