

Package: RMediation (via r-universe)

July 22, 2026

Type Package

Title Mediation Analysis Confidence Intervals

Version 1.6.1

Date 2026-06-30

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](https://doi.org/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](https://doi.org/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](https://doi.org/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), cubature (>= 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

Language en-US

LazyData true

Config/testthat/edition 3

Config/Needs/website pkgdown, quarto

Config/roxygen2/version 8.0.0

NeedsCompilation no

Repository <https://cran.r-universe.dev>

Date/Publication 2026-07-22 00:12:09 UTC

RemoteUrl <https://github.com/cran/RMediation>

RemoteRef HEAD

RemoteSha 1e1b8a1c14834278f43ca944d997838c2a8887b9

Contents

cdf	3
ci	3
ci_mediation_data	4
dist_quantile	5
mbco	6
MBCOResult	8
medci	8
memory_exp	12
pMC	13
pprodnormal	14
pprodnormal3	15
ProductNormal	17
ProductNormal2	17
ProductNormal3	18
qMC	18
qprodnormal	19
tidy	21
tidy.logLik	22
Index	23

cdf	<i>Cumulative Distribution Function</i>
-----	---

Description

Generic function for computing cumulative distribution function.

Usage

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

Arguments

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

Value

A numeric vector of cumulative probabilities for the supplied quantiles, as returned by the dispatched method.

Examples

```
pn <- ProductNormal(mu = c(0.5, 0.3), Sigma = diag(2) * 0.1)
cdf(pn, q = 0)
```

ci	<i>Confidence Interval</i>
----	----------------------------

Description

Generic function for computing confidence intervals.

Usage

```
ci(object, ...)
```

Arguments

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

Value

A confidence interval (numeric vector of lower/upper bounds, or a list) as returned by the dispatched method.

Examples

```
pn <- ProductNormal(mu = c(0.5, 0.3), Sigma = diag(2) * 0.1)
ci(pn)
```

ci_mediation_data	<i>Confidence Interval for MediationData Objects</i>
-------------------	--

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

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

Arguments

object	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 "asyp" (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.
- **"asyp"**: 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

<code>object</code>	A distribution object (e.g., <code>ProductNormal</code>).
<code>...</code>	Additional arguments passed to methods.

Value

A numeric vector of quantiles for the supplied probabilities, as returned by the dispatched method.

Examples

```
pn <- ProductNormal(mu = c(0.5, 0.3), Sigma = diag(2) * 0.1)
dist_quantile(pn, p = 0.025)
```

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.

Examples

```
res <- MBCOResult(statistic = 4.2, df = 1, p_value = 0.04, type = "asympt")
```

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

<code>mu.x</code>	mean of x
<code>mu.y</code>	mean of y
<code>se.x</code>	standard error (deviation) of x
<code>se.y</code>	standard error (deviation) of y
<code>rho</code>	correlation between x and y , where $-1 < \rho < 1$. The default value is 0.
<code>alpha</code>	significance level for the confidence interval. The default value is .05.
<code>type</code>	method used to compute confidence interval. It takes on the values "dop" (default), "MC", "asympt" or "all"
<code>plot</code>	when TRUE, plots the distribution of <code>n.mc</code> 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.
<code>plotCI</code>	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 <code>plot="TRUE"</code> . 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="asympt"` 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="asympt"` 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 `"asympt"`, `medci` returns a [list](#) that contains:

<code>CI</code>	a vector of lower and upper confidence limits (at significance level α),
<code>Estimate</code>	a point estimate of the quantity of interest,
<code>SE</code>	standard error of the quantity of interest,
<code>MC Error</code>	When <code>type="MC"</code> , 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 `"asympt"`, `medci` returns a [list](#) that contains:

$(1-\alpha)\%$ 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](#)

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](https://doi.org/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	<i>Probability (percentile) for the Monte Carlo Sampling Distribution of a nonlinear function of coefficients estimates</i>
-----	---

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	<i>Percentile for the Distribution of Product of Two Normal Variables</i>
-------------	---

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 <i>x</i>
mu.y	mean of <i>y</i>
se.x	standard error (deviation) of <i>x</i>
se.y	standard error (deviation) of <i>y</i>
rho	correlation between <i>x</i> and <i>y</i> , 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", 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:

<code>p</code>	probability (percentile) corresponding to quantile <code>q</code>
<code>error</code>	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")
```

pprodnormal3	<i>Cumulative Distribution Function for the Product of Three Normal Variables</i>
--------------	---

Description

Computes $P(X_1 * X_2 * X_3 \leq q)$ where (X_1, X_2, X_3) follows a trivariate normal distribution with mean vector `mean` and covariance matrix `cov`. Z is marginalized analytically through the conditional Gaussian structure $Z \mid (X, Y)$, and the remaining two-dimensional integral is evaluated with adaptive cubature (dimension reduction; see Tofighi, 2026).

Usage

```
pprodnormal3(q, mean, cov, method = "hcubature", tol = 1e-06)

p_prod3(...)
```

Arguments

q	Numeric scalar quantile.
mean	Numeric vector of means of length 3.
cov	3x3 covariance matrix. Must be positive-definite, except that a single zero-variance component is permitted (the problem then reduces to the two-variable product-normal case). Perfect correlation between two components ($ \rho = 1$) or an indefinite matrix is rejected.
method	Integration method. Currently only "hcubature" is supported.
tol	Numeric tolerance passed to the integration routine; must be strictly positive.
...	Additional arguments (unused; present for the p_prod3 deprecated alias).

Value

Probability $P(X_1 * X_2 * X_3 \leq q)$ as a numeric scalar in $[0, 1]$.

Note

p_prod3() is a superseded alias for pprodnormal3(), kept for backward compatibility; use pprodnormal3() in new code.

Note

Numerical accuracy can degrade as the standardized (X, Y) correlation approaches ± 1 (the bivariate block becomes ill-conditioned). For a fully degenerate point mass (all variances zero) with zero means, $q == 0$ returns 0.5 by the mid-distribution convention rather than 0 or 1.

See Also

[pprodnormal\(\)](#) and [qprodnormal\(\)](#) for the two-variable product-normal CDF/quantile; [Product-Normal3](#) for the corresponding S7 class.

Examples

```
Sigma <- diag(3)
pprodnormal3(q = 0, mean = c(0, 0, 0), cov = Sigma)
```

ProductNormal	<i>ProductNormal Class</i>
---------------	----------------------------

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.

Examples

```
pn <- ProductNormal(mu = c(0.5, 0.3), Sigma = diag(2) * 0.1)
```

ProductNormal2	<i>Enhanced ProductNormal constructor with better validation</i>
----------------	--

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)

Examples

```
pn <- ProductNormal2(mu = c(0.5, 0.3), Sigma = diag(2) * 0.1)
```

ProductNormal3	<i>ProductNormal3 Class</i>
----------------	-----------------------------

Description

Represents the distribution of the product of three normal random variables. Intended for sequential indirect effects of the form $a_1 * a_2 * b$, where (a_1, a_2, b) is asymptotically trivariate normal.

Usage

```
ProductNormal3(mu = integer(0), Sigma = integer(0), method = character(0))
```

Arguments

mu	Numeric vector of means of length 3: $c(a1_hat, a2_hat, b_hat)$.
Sigma	3x3 asymptotic covariance matrix of (a_1, a_2, b) .
method	Integration method. Currently only "hcubature" is supported.

Examples

```
obj <- ProductNormal3(
  mu = c(0.5, 0.3, 0.2),
  Sigma = diag(3),
  method = "hcubature"
)
obj
cdf(obj, q = 1)
## Not run:
confint(obj, level = 0.95)

## End(Not run)
```

qMC	<i>Quantile for the Monte Carlo Sampling Distribution of a nonlinear function of coefficients estimates</i>
-----	---

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., <code>mu=c(b1=.1, b2=3)</code> ; otherwise, the coefficient names are assigned automatically as follows: <code>b1, b2, . . .</code>
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., <code>~b1*b2*b3*b4</code> . The names of coefficients must conform to the names provided in the argument mu or to the default names, i.e., <code>b1, b2, . . .</code>
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 $-1 < \text{rho} < 1$. 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", 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

Examples

```
## Not run:
tidy(mbco_result)

## End(Not run)
```

tidy.logLik	<i>Creates a data.frame for a log-likelihood object</i>
-------------	---

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)
```

Index

- * **data**
 - memory_exp, 12
- * **distribution**
 - pMC, 13
 - qMC, 18
- * **mediation**
 - medci, 8
- * **regression**
 - pMC, 13
 - qMC, 18
- * **sets**
 - memory_exp, 12

cdf, 3
ci, 3, 5, 11
ci_mediation_data, 4
ci_serial_mediation_data
 (ci_mediation_data), 4

data.frame, 22
density, 9
dist_quantile, 5

factor, 12
formula, 13, 19

list, 10, 11, 15, 20
logLik, 22

mbco, 6
MBCOResult, 8
medci, 8, 15, 21
MediationData, 5
memory_exp, 12

p_prod3 (pprodnormal3), 15
pMC, 13
pprodnormal, 11, 14
pprodnormal(), 16
pprodnormal3, 15
ProductNormal, 5, 17
ProductNormal2, 17
ProductNormal3, 16, 18

qMC, 18
qprodnormal, 11, 19
qprodnormal(), 16

tidy, 21
tidy.logLik, 22

vector, 13, 19