

# Package ‘RMediation’

November 20, 2025

**Type** Package

**Title** Mediation Analysis Confidence Intervals

**Version** 1.3.0

**Date** 2025-11-18

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

**Maintainer** Davood Tofighi <[dtofighi@gmail.com](mailto: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), base (>= 4.1.0), stats (>= 4.1.0)

**Imports** graphics (>= 4.1.0), lavaan (>= 0.5-20), e1071 (>= 1.6-7),  
OpenMx (>= 2.13), grDevices (>= 4.1.0), MASS (>= 7.3), modelr  
(>= 0.1.8), generics (>= 0.0.2), checkmate (>= 2.1.0)

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**Encoding** UTF-8  
**LazyData** true  
**RoxygenNote** 7.3.3  
**Config/testthat/edition** 3  
**NeedsCompilation** no  
**Repository** CRAN  
**Date/Publication** 2025-11-20 10:20:19 UTC

Contents

ci . . . . .	2
mbco . . . . .	5
medci . . . . .	7
memory_exp . . . . .	9
pMC . . . . .	10
pprodnormal . . . . .	11
qMC . . . . .	13
qprodnormal . . . . .	14
tidy.logLik . . . . .	15
<b>Index</b>	<b>17</b>

---

ci	<i>CI for a nonlinear function of coefficients estimates</i>
----	--

---

Description

This function returns a  $(1 - \alpha)\%$  confidence interval (CI) for a well-defined nonlinear function of the coefficients in single-level and multilevel structural equation models. The ci function uses the Monte Carlo (type="MC") and the asymptotic normal theory (type="asyp") with the multivariate delta standard error (Asymptotic-Delta) method (Sobel, 1982) to compute a CI. In addition, for each of the methods, when a user specifies plot=TRUE and plotCI=TRUE, a plot of the sampling distribution of the quantity of interest in the quant argument and an overlaid plot of the CI will be produced. When type="all" and plot=TRUE, two overlaid plots of the sampling distributions corresponding to each method will be produced; when plotCI=TRUE, then the overlaid plots of the CIs for both methods will be displayed as well.

Usage

```
ci(  
  mu,  
  Sigma,  
  quant,  
  alpha = 0.05,  
  type = "MC",
```

```

    plot = FALSE,
    plotCI = FALSE,
    n.mc = 1e+06,
    H0 = FALSE,
    mu0 = NULL,
    Sigma0 = NULL,
    ...
)

```

## Arguments

mu	(1) a <a href="#">vector</a> 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> . Or, (2) a <a href="#">lavaan</a> object.
Sigma	either a covariance matrix or a <a href="#">vector</a> 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 <a href="#">formula</a> that <b>must</b> 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>
alpha	significance level for the CI. The default value is .05.
type	method used to compute a CI. It takes on the values "MC" (default) for Monte Carlo, "asyp" for Asymptotic–Delta, or "all" that produces CIs using both methods.
plot	when TRUE, plot the approximate sampling distribution of the quantity of interest using the specified method(s) in the argument type. The default value is FALSE. When <code>type="all"</code> , superimposed density plots generated by both methods are displayed.
plotCI	when TRUE, overlays a CI plot with error bars on the density plot of the sampling distribution of quant. When <code>type="all"</code> , the superimposed CI plots generated by both methods are added to the density plots. Note that to obtain a CI plot, one must also specify <code>plot="TRUE"</code> . The default value is FALSE.
n.mc	Monte Carlo sample size. The default sample size is 1e+6.
H0	False. If TRUE, it will estimate the sampling distribution of $H_0 : f(\mathbf{b}) = 0$ . See the arguments mu0 and Sigma0.
mu0	a <a href="#">vector</a> of means (e.g., coefficient estimates) for the normal random variables that satisfy the null hypothesis $H_0 : f(\mathbf{b}) = 0$ . If it is not provided, smallest z value of mu is set to zero.
Sigma0	either a covariance matrix or a <a href="#">vector</a> that stacks all the columns of the lower triangle variance–covariance matrix one underneath the other. If it is not provided, then Sigma is used instead.
...	additional arguments.

**Value**

When type is "MC" or "asyp", ci 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.

When type="all", ci returns a [list](#) of two 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**

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**

[medci RMediation-package](#)

**Examples**

```
ci(
  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, type = "all", plot = TRUE, plotCI = TRUE
)
# An Example of Conservative Null Sampling Distribution
ci(c(b1 = .3, b2 = .4, b3 = .3), c(.01, 0, 0, .01, 0, .02),
  quant = ~ b1 * b2 * b3, type = "mc", plot = TRUE, plotCI = TRUE,
  H0 = TRUE, mu0 = c(b1 = .3, b2 = .4, b3 = 0)
)
# An Example of Less Conservative Null Sampling Distribution
ci(c(b1 = .3, b2 = .4, b3 = .3), c(.01, 0, 0, .01, 0, .02),
  quant = ~ b1 * b2 * b3, type = "mc", plot = TRUE, plotCI = TRUE,
  H0 = TRUE, mu0 = c(b1 = 0, b2 = .4, b3 = 0.1)
)
```

**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 = NULL,
  h1 = NULL,
  R = 10L,
  type = "asyp",
  alpha = 0.05,
  checkHess = "No",
  checkSE = "No",
  optim = "SLSQP",
  precision = 1e-09
)
```

**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.
R	The number of bootstrap draws.
type	If 'asyp' (default), the asymptotic MBCO chi-squares test comparing fit of h0 and h1. If 'parametric', the parametric bootstrap MBCO chi-squared test is computed. If 'semi', the semi-parametric MBCO chi-squared is computed.
alpha	Significance level with the default value of .05
checkHess	If 'No' (default), the Hessian matrix would not be calculated.
checkSE	if 'No' (default), the standard errors would not be calculated.
optim	Choose optimizer available in OpenMx. The default optimizer is "SLSQP". Other optimizer choices are available. See <a href="#">mxOption</a> for more details.
precision	Functional precision. The default value is set to 1e-9. See <a href="#">mxOption</a> for more details.

**Value**

A [list](#) that contains

chisq                    asymptotic chi-squared test statistic value

df	chi-squared df
p	chi-squared p-value computed based on the method specified by the argument type

**Author(s)**

Davood Tofighi <dtofighi@gmail.com>

**Examples**

```
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)

```

---

medci

---

*Confidence Interval for the Mediated Effect*


---

## Description

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,
  ...
)

```

## 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", "asyp" 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 <a href="#">density</a> . 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  $(1 - \alpha)\%$  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="asyp" 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):

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

. 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 "asyp", 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

## 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)
```

---

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 on a 1 to 9 scale

- recall** Total words recalled out of 20 words
- imagery** Use of imagery rehearsal technique on a 1 to 9 scale
- x** A **factor** with two levels: repetition or primary rehearsal = 0, imagery or secondary rehearsal = 1

**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](https://doi.org/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+06, ...)

**Arguments**

q	quantile
mu	a <b>vector</b> 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 <b>vector</b> 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 <b>formula</b> that <b>must</b> 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. The default sample size is 1e+6.
...	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 $x$

<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>lower.tail</code>	logical; if TRUE (default), the probability is $P[X*Y < q]$ ; otherwise, $P[X*Y > q]$
<code>type</code>	method used to compute confidence interval. It takes on the values "dop" (default), "MC", "asympt" or "all"
<code>n.mc</code>	when <code>type="MC"</code> , <code>n.mc</code> determines the sample size for the Monte Carlo method. The default sample size is 1E5.

### 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 $q$
<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")
```

---

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+06, ...)
```

### Arguments

$p$	probability.
$\mu$	a <b>vector</b> 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: $b1, b2, \dots$
$\Sigma$	either a covariance matrix or a <b>vector</b> 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 <b>formula</b> that <b>must</b> start with the symbol "tilde" ( $\sim$ ): e.g., $\sim b1*b2*b3*b4$ . The names of coefficients must conform to the names provided in the argument $\mu$ or to the default names, i.e., $b1, b2, \dots$
$n.mc$	Monte Carlo sample size. The default sample size is $1e+6$ .
$\dots$	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 $P[X * Y < q]$ ; otherwise, $P[X * Y > q]$
type	method used to compute confidence interval. It takes on the values "dop" (default), "MC", "asyp" or "all"
n.mc	when type="MC", n.mc determines the sample size for the Monte Carlo method. The default sample size is 1E5.

**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:

<code>q</code>	quantile corresponding to probability <code>p</code>
<code>error</code>	estimate of the absolute error

### Author(s)

Davood Tofighi <[dtofighi@gmail.com](mailto: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.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**

<code>x</code>	A log-likelihood object, typically returned by <a href="#">logLik</a> .
<code>...</code>	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](mailto:dtofighi@gmail.com)>

**See Also**

[logLik](#)

**Examples**

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



# Index

- \* **data**
  - memory\_exp, 9
- \* **distribution**
  - ci, 2
  - pMC, 10
  - qMC, 13
- \* **mediation**
  - medci, 7
- \* **regression**
  - ci, 2
  - pMC, 10
  - qMC, 13
- \* **sets**
  - memory\_exp, 9

ci, 2, 9

data.frame, 16  
density, 8

factor, 10  
formula, 3, 10, 13

lavaan, 3  
list, 4, 5, 8, 12, 15  
logLik, 16

mbco, 5  
medci, 4, 7, 12, 15  
memory\_exp, 9  
mxOption, 5

pMC, 10  
pprodnormal, 9, 11

qMC, 13  
qprodnormal, 9, 14

tidy.logLik, 15

vector, 3, 10, 13