# Package 'cTMed'

October 21, 2024	
Title Continuous Time Mediation	
Version 1.0.1	
<b>Description</b> Calculates standard errors and confidence intervals for effects in continuous-time mediation models. This package extends the work of Deboeck and Preacher (2015) <doi:10.1080 10705511.2014.973960=""> and Ryan and Hamaker (2021) <doi:10.1007 s11336-021-09767-0=""> by providing methods to generate standard errors and confidence intervals for the total, direct, and indirect effects in these models.</doi:10.1007></doi:10.1080>	
<pre>URL https://github.com/jeksterslab/cTMed,</pre>	
https://jeksterslab.github.io/cTMed/	
<pre>BugReports https://github.com/jeksterslab/cTMed/issues</pre>	
License GPL (>= 3)	
Encoding UTF-8	
<b>Depends</b> R (>= $3.5.0$ )	
LinkingTo Rcpp, RcppArmadillo	
Imports Rcpp, numDeriv, parallel, ctsem, simStateSpace	
Suggests knitr, rmarkdown, testthat, expm	
RoxygenNote 7.3.2	
NeedsCompilation yes	
Author Ivan Jacob Agaloos Pesigan [aut, cre, cph] ( <a href="https://orcid.org/0000-0003-4818-8420">https://orcid.org/0000-0003-4818-8420</a> )	
Maintainer Ivan Jacob Agaloos Pesigan <r.jeksterslab@gmail.com></r.jeksterslab@gmail.com>	
Repository CRAN	
<b>Date/Publication</b> 2024-10-21 11:50:29 UTC	
Contents	
confint.ctmeddelta	

2 confint.ctmeddelta

D	eltaBeta	6
D	eltaIndirectCentral	9
D	eltaMed	13
D	PeltaTotalCentral	16
D	rirect	19
	ndirect	22
Iı	ndirectCentral	24
N	ICBeta	26
N	1CIndirectCentral	29
N	ICMed	33
	ICPhi	36
	1CTotalCentral	38
	1ed	41
	lot.ctmeddelta	44
	lot.ctmedmc	46
-	lot.ctmedmed	47
_	lot.ctmedtraj	48
1	osteriorBeta	49
P	osteriorIndirectCentral	52
P	osteriorMed	55
	osteriorPhi	57
	osteriorTotalCentral	58
	rint.ctmeddelta	61
_	rint.ctmedeffect	63
-	rint.ctmedmc	64
1	rint.ctmedmcphi	66
	rint.ctmedmed	67
	rint.ctmedtraj	68
_	ımmary.ctmeddelta	69
	ımmary.ctmedmc	71
	ımmary.ctmedmed	73
	ummary.ctmedposteriorphi	74
	ımmary.ctmedtraj	75
	otal	76
	otalCentral	78
	rajectory	80
-		00
Index		82
confint	c.ctmeddelta	

# Description

Delta Method Confidence Intervals

confint.ctmeddelta 3

## Usage

```
## S3 method for class 'ctmeddelta'
confint(object, parm = NULL, level = 0.95, ...)
```

additional arguments.

## **Arguments**

object Object of class ctmeddelta.

parm a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.

level the confidence level required.

## Value

Returns a data frame of confidence intervals.

## Author(s)

Ivan Jacob Agaloos Pesigan

```
phi <- matrix(</pre>
 data = c(
   -0.357, 0.771, -0.450,
   0.0, -0.511, 0.729,
   0, 0, -0.693
 ),
 nrow = 3
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
vcov_phi_vec <- matrix(</pre>
 data = c(
    0.002704274, -0.001475275, 0.000949122,
    -0.001619422, 0.000885122, -0.000569404,
   0.00085493, -0.000465824, 0.000297815,
    -0.001475275, 0.004428442, -0.002642303,
    0.000980573, -0.00271817, 0.001618805,
    -0.000586921, 0.001478421, -0.000871547,
    0.000949122, -0.002642303, 0.006402668,
    -0.000697798, 0.001813471, -0.004043138,
    0.000463086, -0.001120949, 0.002271711,
    -0.001619422, 0.000980573, -0.000697798,
    0.002079286, -0.001152501, 0.000753,
    -0.001528701, 0.000820587, -0.000517524,
    0.000885122, -0.00271817, 0.001813471,
    -0.001152501, 0.00342605, -0.002075005,
     0.000899165, \ -0.002532849, \ 0.001475579, \\
    -0.000569404, 0.001618805, -0.004043138,
```

4 confint.ctmedmc

```
0.000753, -0.002075005, 0.004984032,
   -0.000622255, 0.001634917, -0.003705661,
   0.00085493, -0.000586921, 0.000463086,
   -0.001528701, 0.000899165, -0.000622255,
   0.002060076, -0.001096684, 0.000686386,
   -0.000465824, 0.001478421, -0.001120949,
   0.000820587, -0.002532849, 0.001634917,
   -0.001096684, 0.003328692, -0.001926088,
   0.000297815, -0.000871547, 0.002271711,
   -0.000517524, 0.001475579, -0.003705661,
   0.000686386, -0.001926088, 0.004726235
 ),
 nrow = 9
)
# Specific time interval ------
delta <- DeltaMed(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1,
 from = "x",
 to = "y",
 med = "m"
confint(delta, level = 0.95)
# Range of time intervals ------
delta <- DeltaMed(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1:5,
 from = "x",
 to = "y",
 med = "m"
confint(delta, level = 0.95)
```

confint.ctmedmc

Monte Carlo Method Confidence Intervals

## Description

Monte Carlo Method Confidence Intervals

## Usage

```
## S3 method for class 'ctmedmc'
confint(object, parm = NULL, level = 0.95, ...)
```

confint.ctmedmc 5

## **Arguments**

object Object of class ctmedmc.

parm a specification of which parameters are to be given confidence intervals, either

a vector of numbers or a vector of names. If missing, all parameters are consid-

ered.

level the confidence level required.

... additional arguments.

#### Value

Returns a data frame of confidence intervals.

#### Author(s)

Ivan Jacob Agaloos Pesigan

```
set.seed(42)
phi <- matrix(</pre>
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) \leftarrow rownames(phi) \leftarrow c("x", "m", "y")
vcov_phi_vec <- matrix(</pre>
  data = c(
    0.002704274, -0.001475275, 0.000949122,
    -0.001619422, 0.000885122, -0.000569404,
    0.00085493, -0.000465824, 0.000297815,
    -0.001475275, 0.004428442, -0.002642303,
    0.000980573, -0.00271817, 0.001618805,
    -0.000586921, 0.001478421, -0.000871547,
    0.000949122, -0.002642303, 0.006402668,
    -0.000697798, 0.001813471, -0.004043138,
    0.000463086, -0.001120949, 0.002271711,
    -0.001619422, 0.000980573, -0.000697798,
    0.002079286, -0.001152501, 0.000753,
    -0.001528701, 0.000820587, -0.000517524,
    0.000885122, -0.00271817, 0.001813471,
    -0.001152501, 0.00342605, -0.002075005,
    0.000899165, -0.002532849, 0.001475579,
    -0.000569404, 0.001618805, -0.004043138,
    0.000753, -0.002075005, 0.004984032,
    -0.000622255, 0.001634917, -0.003705661,
    0.00085493, -0.000586921, 0.000463086,
    -0.001528701, 0.000899165, -0.000622255,
```

6 DeltaBeta

```
0.002060076, -0.001096684, 0.000686386,
   -0.000465824, 0.001478421, -0.001120949,
   0.000820587, -0.002532849, 0.001634917,
   -0.001096684, 0.003328692, -0.001926088,
   0.000297815, -0.000871547, 0.002271711,
   -0.000517524, 0.001475579, -0.003705661,
   0.000686386, -0.001926088, 0.004726235
 ),
 nrow = 9
)
# Specific time interval ------
mc <- MCMed(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1,
 from = "x",
 to = "y",
 med = "m",
 R = 100L # use a large value for R in actual research
)
confint(mc, level = 0.95)
# Range of time intervals ------
mc <- MCMed(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1:5,
 from = "x",
 to = "y",
 med = "m",
 R = 100L # use a large value for R in actual research
)
confint(mc, level = 0.95)
```

DeltaBeta

Delta Method Sampling Variance-Covariance Matrix for the Elements of the Matrix of Lagged Coefficients Over a Specific Time Interval or a Range of Time Intervals

# Description

This function computes the delta method sampling variance-covariance matrix for the elements of the matrix of lagged coefficients  $\beta$  over a specific time interval  $\Delta t$  or a range of time intervals using the first-order stochastic differential equation model's drift matrix  $\Phi$ .

## Usage

```
DeltaBeta(phi, vcov_phi_vec, delta_t, ncores = NULL)
```

DeltaBeta 7

## Arguments

phi Numeric matrix. The drift matrix ( $\Phi$ ), phi should have row and column names

pertaining to the variables in the system.

vcov\_phi\_vec Numeric matrix. The sampling variance-covariance matrix of  $vec(\Phi)$ .

delta\_t Vector of positive numbers. Time interval ( $\Delta t$ ).

ncores Positive integer. Number of cores to use. If ncores = NULL, use a single core.

Consider using multiple cores when the length of delta\_t is long.

#### **Details**

See Total().

#### **Delta Method:**

Let  $\theta$  be  $\operatorname{vec}(\Phi)$ , that is, the elements of the  $\Phi$  matrix in vector form sorted column-wise. Let  $\hat{\theta}$  be  $\operatorname{vec}(\hat{\Phi})$ . By the multivariate central limit theory, the function g using  $\hat{\theta}$  as input can be expressed as:

$$\sqrt{n}\left(\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right) - \mathbf{g}\left(\boldsymbol{\theta}\right)\right) \xrightarrow{\mathrm{D}} \mathcal{N}\left(0, \mathbf{J}\boldsymbol{\Gamma}\mathbf{J}'\right)$$

where **J** is the matrix of first-order derivatives of the function **g** with respect to the elements of  $\theta$  and  $\Gamma$  is the asymptotic variance-covariance matrix of  $\hat{\theta}$ .

From the former, we can derive the distribution of  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right)$  as follows:

$$\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right) \approx \mathcal{N}\left(\mathbf{g}\left(\boldsymbol{\theta}\right), n^{-1}\mathbf{J}\mathbf{\Gamma}\mathbf{J}'\right)$$

The uncertainty associated with the estimator  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right)$  is, therefore, given by  $n^{-1}\mathbf{J}\Gamma\mathbf{J}'$ . When  $\Gamma$  is unknown, by substitution, we can use the estimated sampling variance-covariance matrix of  $\hat{\boldsymbol{\theta}}$ , that is,  $\hat{\mathbb{V}}\left(\hat{\boldsymbol{\theta}}\right)$  for  $n^{-1}\Gamma$ . Therefore, the sampling variance-covariance matrix of  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right)$  is given by

$$\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right) \approx \mathcal{N}\left(\mathbf{g}\left(\boldsymbol{\theta}\right), \mathbf{J}\hat{\mathbb{V}}\left(\hat{\boldsymbol{\theta}}\right) \mathbf{J}'\right).$$

## Value

Returns an object of class ctmeddelta which is a list with the following elements:

call Function call.

args Function arguments.

fun Function used ("DeltaBeta").

**output** A list the length of which is equal to the length of delta\_t.

Each element in the output list has the following elements:

**delta\_t** Time interval.

jacobian Jacobian matrix.

8 DeltaBeta

est Estimated elements of the matrix of lagged coefficients.

vcov Sampling variance-covariance matrix of estimated elements of the matrix of lagged coefficients.

#### Author(s)

Ivan Jacob Agaloos Pesigan

#### References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. Sociological Methodology, 17, 37. doi:10.2307/271028

Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. Structural Equation Modeling: A Multidisciplinary Journal, 23 (1), 61–75. doi:10.1080/10705511.2014.973960

Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. Psychometrika, 87 (1), 214–252. doi:10.1007/s11336021097670

#### See Also

```
Other Continuous Time Mediation Functions: DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()
```

```
phi <- matrix(</pre>
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
vcov_phi_vec <- matrix(</pre>
  data = c(
    0.002704274, -0.001475275, 0.000949122,
    -0.001619422, 0.000885122, -0.000569404,
    0.00085493, -0.000465824, 0.000297815,
    -0.001475275, 0.004428442, -0.002642303,
    0.000980573, -0.00271817, 0.001618805,
    -0.000586921, 0.001478421, -0.000871547,
    0.000949122, -0.002642303, 0.006402668,
    -0.000697798, 0.001813471, -0.004043138,
    0.000463086, -0.001120949, 0.002271711,
    -0.001619422, 0.000980573, -0.000697798,
    0.002079286, -0.001152501, 0.000753,
    -0.001528701, 0.000820587, -0.000517524,
```

```
0.000885122, -0.00271817, 0.001813471,
   -0.001152501, 0.00342605, -0.002075005,
   0.000899165, -0.002532849, 0.001475579,
   -0.000569404, 0.001618805, -0.004043138,
   0.000753, -0.002075005, 0.004984032,
   -0.000622255, 0.001634917, -0.003705661,
   0.00085493, -0.000586921, 0.000463086,
   -0.001528701, 0.000899165, -0.000622255,
   0.002060076, -0.001096684, 0.000686386,
   -0.000465824, 0.001478421, -0.001120949,
   0.000820587, -0.002532849, 0.001634917,
   -0.001096684, 0.003328692, -0.001926088,
   0.000297815, -0.000871547, 0.002271711,
   -0.000517524, 0.001475579, -0.003705661,
   0.000686386, -0.001926088, 0.004726235
 ),
 nrow = 9
)
# Specific time interval ------
DeltaBeta(
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1
)
# Range of time intervals -------
delta <- DeltaBeta(
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1:5
)
plot(delta)
# Methods -----
# DeltaBeta has a number of methods including
# print, summary, confint, and plot
print(delta)
summary(delta)
confint(delta, level = 0.95)
plot(delta)
```

DeltaIndirectCentral Delta Method Sampling Variance-Covariance Matrix for the Indirect

Effect Centrality Over a Specific Time Interval or a Range of Time

Intervals

## **Description**

This function computes the delta method sampling variance-covariance matrix for the indirect effect centrality over a specific time interval  $\Delta t$  or a range of time intervals using the first-order stochastic differential equation model's drift matrix  $\Phi$ .

## Usage

DeltaIndirectCentral(phi, vcov\_phi\_vec, delta\_t, ncores = NULL)

#### **Arguments**

phi Numeric matrix. The drift matrix ( $\Phi$ ). phi should have row and column names

pertaining to the variables in the system.

vcov\_phi\_vec Numeric matrix. The sampling variance-covariance matrix of  $vec(\Phi)$ .

delta\_t Vector of positive numbers. Time interval ( $\Delta t$ ).

ncores Positive integer. Number of cores to use. If ncores = NULL, use a single core.

Consider using multiple cores when the length of delta\_t is long.

#### **Details**

See IndirectCentral() more details.

#### **Delta Method:**

Let  $\theta$  be  $\operatorname{vec}(\Phi)$ , that is, the elements of the  $\Phi$  matrix in vector form sorted column-wise. Let  $\hat{\theta}$  be  $\operatorname{vec}(\hat{\Phi})$ . By the multivariate central limit theory, the function  $\mathbf{g}$  using  $\hat{\theta}$  as input can be expressed as:

$$\sqrt{n}\left(\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right) - \mathbf{g}\left(\boldsymbol{\theta}\right)\right) \xrightarrow{\mathrm{D}} \mathcal{N}\left(0, \mathbf{J}\boldsymbol{\Gamma}\mathbf{J}'\right)$$

where **J** is the matrix of first-order derivatives of the function **g** with respect to the elements of  $\theta$  and  $\Gamma$  is the asymptotic variance-covariance matrix of  $\hat{\theta}$ .

From the former, we can derive the distribution of  $\mathbf{g}\left(\hat{\pmb{\theta}}\right)$  as follows:

$$\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right) \approx \mathcal{N}\left(\mathbf{g}\left(\boldsymbol{\theta}\right), n^{-1}\mathbf{J}\boldsymbol{\Gamma}\mathbf{J}'\right)$$

The uncertainty associated with the estimator  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right)$  is, therefore, given by  $n^{-1}\mathbf{J}\Gamma\mathbf{J}'$ . When  $\Gamma$  is unknown, by substitution, we can use the estimated sampling variance-covariance matrix of  $\hat{\boldsymbol{\theta}}$ , that is,  $\hat{\mathbb{V}}\left(\hat{\boldsymbol{\theta}}\right)$  for  $n^{-1}\Gamma$ . Therefore, the sampling variance-covariance matrix of  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right)$  is given by

$$\mathbf{g}\left(\hat{oldsymbol{ heta}}
ight)pprox\mathcal{N}\left(\mathbf{g}\left(oldsymbol{ heta}
ight),\mathbf{J}\hat{\mathbb{V}}\left(\hat{oldsymbol{ heta}}
ight)\mathbf{J}'
ight).$$

#### Value

Returns an object of class ctmeddelta which is a list with the following elements:

call Function call.

args Function arguments.

fun Function used ("DeltaIndirectCentral").

**output** A list the length of which is equal to the length of delta\_t.

Each element in the output list has the following elements:

delta\_t Time interval.

jacobian Jacobian matrix.

est Estimated indirect effect centrality.

vcov Sampling variance-covariance matrix of estimated indirect effect centrality.

## Author(s)

Ivan Jacob Agaloos Pesigan

#### References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. Sociological Methodology, 17, 37. doi:10.2307/271028

Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. Structural Equation Modeling: A Multidisciplinary Journal, 23 (1), 61–75. doi:10.1080/10705511.2014.973960

Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. Psychometrika, 87 (1), 214–252. doi:10.1007/s11336021097670

## See Also

Other Continuous Time Mediation Functions: DeltaBeta(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()

```
phi <- matrix(
   data = c(
     -0.357, 0.771, -0.450,
     0.0, -0.511, 0.729,
     0, 0, -0.693
   ),
   nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")
vcov_phi_vec <- matrix(</pre>
```

```
data = c(
   0.002704274, -0.001475275, 0.000949122,
   -0.001619422, 0.000885122, -0.000569404,
   0.00085493, -0.000465824, 0.000297815,
   -0.001475275, 0.004428442, -0.002642303,
   0.000980573, -0.00271817, 0.001618805,
   -0.000586921, 0.001478421, -0.000871547,
   0.000949122, -0.002642303, 0.006402668,
   -0.000697798, 0.001813471, -0.004043138,
   0.000463086, -0.001120949, 0.002271711,
   -0.001619422, 0.000980573, -0.000697798,
   0.002079286, -0.001152501, 0.000753,
   -0.001528701, 0.000820587, -0.000517524,
   0.000885122, -0.00271817, 0.001813471,
   -0.001152501, 0.00342605, -0.002075005,
   0.000899165, -0.002532849, 0.001475579,
   -0.000569404, 0.001618805, -0.004043138,
   0.000753, -0.002075005, 0.004984032,
   -0.000622255, 0.001634917, -0.003705661,
   0.00085493, -0.000586921, 0.000463086,
   -0.001528701, 0.000899165, -0.000622255,
   0.002060076, -0.001096684, 0.000686386,
   -0.000465824, 0.001478421, -0.001120949,
   0.000820587, -0.002532849, 0.001634917,
   -0.001096684, 0.003328692, -0.001926088,
   0.000297815, -0.000871547, 0.002271711,
   -0.000517524, 0.001475579, -0.003705661,
   0.000686386, -0.001926088, 0.004726235
 ),
 nrow = 9
)
# Specific time interval -------
DeltaIndirectCentral(
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1
)
# Range of time intervals -------
delta <- DeltaIndirectCentral(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1:5
)
plot(delta)
# Methods ------
# DeltaIndirectCentral has a number of methods including
# print, summary, confint, and plot
print(delta)
summary(delta)
confint(delta, level = 0.95)
```

DeltaMed 13

plot(delta)

Delta Method Sampling Variance-Covariance Matrix for the Total, Di-

rect, and Indirect Effects of X on Y Through M Over a Specific Time

Interval or a Range of Time Intervals

## **Description**

This function computes the delta method sampling variance-covariance matrix for the total, direct, and indirect effects of the independent variable X on the dependent variable Y through mediator variables  $\mathbf{m}$  over a specific time interval  $\Delta t$  or a range of time intervals using the first-order stochastic differential equation model's drift matrix  $\Phi$ .

#### Usage

DeltaMed(phi, vcov\_phi\_vec, delta\_t, from, to, med, ncores = NULL)

## **Arguments**

phi Numeric matrix. The drift matrix ( $\Phi$ ). phi should have row and column names

pertaining to the variables in the system.

vcov\_phi\_vec Numeric matrix. The sampling variance-covariance matrix of  $vec(\Phi)$ .

delta\_t Vector of positive numbers. Time interval  $(\Delta t)$ .

from Character string. Name of the independent variable X in phi.

to Character string. Name of the dependent variable Y in phi.

med Character vector. Name/s of the mediator variable/s in phi.

ncores Positive integer. Number of cores to use. If ncores = NULL, use a single core.

Consider using multiple cores when the length of delta\_t is long.

#### **Details**

See Total(), Direct(), and Indirect() for more details.

## **Delta Method:**

Let  $\theta$  be  $\text{vec}(\Phi)$ , that is, the elements of the  $\Phi$  matrix in vector form sorted column-wise. Let  $\hat{\theta}$  be  $\text{vec}(\hat{\Phi})$ . By the multivariate central limit theory, the function  $\mathbf{g}$  using  $\hat{\theta}$  as input can be expressed as:

$$\sqrt{n}\left(\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right) - \mathbf{g}\left(\boldsymbol{\theta}\right)\right) \xrightarrow{\mathrm{D}} \mathcal{N}\left(0, \mathbf{J}\boldsymbol{\Gamma}\mathbf{J}'\right)$$

where **J** is the matrix of first-order derivatives of the function **g** with respect to the elements of  $\theta$  and  $\Gamma$  is the asymptotic variance-covariance matrix of  $\hat{\theta}$ .

From the former, we can derive the distribution of  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right)$  as follows:

14 DeltaMed

$$\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right) \approx \mathcal{N}\left(\mathbf{g}\left(\boldsymbol{\theta}\right), n^{-1}\mathbf{J}\Gamma\mathbf{J}'\right)$$

The uncertainty associated with the estimator  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right)$  is, therefore, given by  $n^{-1}\mathbf{J}\Gamma\mathbf{J}'$ . When  $\Gamma$  is unknown, by substitution, we can use the estimated sampling variance-covariance matrix of  $\hat{\boldsymbol{\theta}}$ , that is,  $\hat{\mathbb{V}}\left(\hat{\boldsymbol{\theta}}\right)$  for  $n^{-1}\Gamma$ . Therefore, the sampling variance-covariance matrix of  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right)$  is given by

$$\mathbf{g}\left(\hat{oldsymbol{ heta}}
ight)pprox\mathcal{N}\left(\mathbf{g}\left(oldsymbol{ heta}
ight),\mathbf{J}\hat{\mathbb{V}}\left(\hat{oldsymbol{ heta}}
ight)\mathbf{J}'
ight).$$

## Value

Returns an object of class ctmeddelta which is a list with the following elements:

call Function call.

args Function arguments.

fun Function used ("DeltaMed").

**output** A list the length of which is equal to the length of delta\_t.

Each element in the output list has the following elements:

delta\_t Time interval.

jacobian Jacobian matrix.

est Estimated total, direct, and indirect effects.

vcov Sampling variance-covariance matrix of the estimated total, direct, and indirect effects.

## Author(s)

Ivan Jacob Agaloos Pesigan

#### References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. Sociological Methodology, 17, 37. doi:10.2307/271028

Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. Structural Equation Modeling: A Multidisciplinary Journal, 23 (1), 61–75. doi:10.1080/10705511.2014.973960

Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. Psychometrika, 87 (1), 214–252. doi:10.1007/s11336021097670

#### See Also

Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()

DeltaMed 15

```
phi <- matrix(</pre>
 data = c(
   -0.357, 0.771, -0.450,
   0.0, -0.511, 0.729,
   0, 0, -0.693
 ),
 nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
vcov_phi_vec <- matrix(</pre>
 data = c(
   0.002704274, -0.001475275, 0.000949122,
    -0.001619422, 0.000885122, -0.000569404,
   0.00085493, -0.000465824, 0.000297815,
    -0.001475275, 0.004428442, -0.002642303,
   0.000980573, -0.00271817, 0.001618805,
    -0.000586921, 0.001478421, -0.000871547,
   0.000949122, -0.002642303, 0.006402668,
    -0.000697798, 0.001813471, -0.004043138,
   0.000463086, -0.001120949, 0.002271711,
    -0.001619422, 0.000980573, -0.000697798,
   0.002079286, -0.001152501, 0.000753,
   -0.001528701, 0.000820587, -0.000517524,
   0.000885122, -0.00271817, 0.001813471,
    -0.001152501, 0.00342605, -0.002075005,
   0.000899165, -0.002532849, 0.001475579,
    -0.000569404, 0.001618805, -0.004043138,
   0.000753, -0.002075005, 0.004984032,
    -0.000622255, 0.001634917, -0.003705661,
   0.00085493, -0.000586921, 0.000463086,
    -0.001528701, 0.000899165, -0.000622255,
    0.002060076, -0.001096684, 0.000686386,
    -0.000465824, 0.001478421, -0.001120949,
   0.000820587, -0.002532849, 0.001634917,
    -0.001096684, 0.003328692, -0.001926088,
   0.000297815, -0.000871547, 0.002271711,
   -0.000517524, 0.001475579, -0.003705661,
   0.000686386, -0.001926088, 0.004726235
 ),
 nrow = 9
)
# Specific time interval -------
DeltaMed(
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1,
 from = "x",
 to = "y",
 med = "m"
)
```

16 DeltaTotalCentral

```
# Range of time intervals ------
delta <- DeltaMed(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1:5,
 from = "x",
 to = "y",
 med = "m"
plot(delta)
# Methods ------
# DeltaMed has a number of methods including
# print, summary, confint, and plot
print(delta)
summary(delta)
confint(delta, level = 0.95)
plot(delta)
```

DeltaTotalCentral

Delta Method Sampling Variance-Covariance Matrix for the Total Effect Centrality Over a Specific Time Interval or a Range of Time Intervals

## Description

This function computes the delta method sampling variance-covariance matrix for the total effect centrality over a specific time interval  $\Delta t$  or a range of time intervals using the first-order stochastic differential equation model's drift matrix  $\Phi$ .

## Usage

```
DeltaTotalCentral(phi, vcov_phi_vec, delta_t, ncores = NULL)
```

# Arguments

phi	Numeric matrix. The drift matrix $(\Phi)$ . phi should have row and column n pertaining to the variables in the system.	
vcov_phi_vec	Numeric matrix. The sampling variance-covariance matrix of $\operatorname{vec}\left(\mathbf{\Phi}\right)$ .	
delta_t	Vector of positive numbers. Time interval $(\Delta t)$ .	
ncores	Positive integer. Number of cores to use. If ncores = NULL, use a single core. Consider using multiple cores when the length of delta_t is long.	

DeltaTotalCentral 17

#### **Details**

See TotalCentral() more details.

## Delta Method:

Let  $\theta$  be  $\operatorname{vec}(\Phi)$ , that is, the elements of the  $\Phi$  matrix in vector form sorted column-wise. Let  $\hat{\theta}$  be  $\operatorname{vec}(\hat{\Phi})$ . By the multivariate central limit theory, the function  $\mathbf{g}$  using  $\hat{\theta}$  as input can be expressed as:

$$\sqrt{n}\left(\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right) - \mathbf{g}\left(\boldsymbol{\theta}\right)\right) \xrightarrow{\mathrm{D}} \mathcal{N}\left(0, \mathbf{J}\boldsymbol{\Gamma}\mathbf{J}'\right)$$

where **J** is the matrix of first-order derivatives of the function **g** with respect to the elements of  $\theta$  and  $\Gamma$  is the asymptotic variance-covariance matrix of  $\hat{\theta}$ .

From the former, we can derive the distribution of  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right)$  as follows:

$$\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right) \approx \mathcal{N}\left(\mathbf{g}\left(\boldsymbol{\theta}\right), n^{-1}\mathbf{J}\mathbf{\Gamma}\mathbf{J}'\right)$$

The uncertainty associated with the estimator  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right)$  is, therefore, given by  $n^{-1}\mathbf{J}\Gamma\mathbf{J}'$ . When  $\Gamma$  is unknown, by substitution, we can use the estimated sampling variance-covariance matrix of  $\hat{\boldsymbol{\theta}}$ , that is,  $\hat{\mathbb{V}}\left(\hat{\boldsymbol{\theta}}\right)$  for  $n^{-1}\Gamma$ . Therefore, the sampling variance-covariance matrix of  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right)$  is given by

$$\mathbf{g}\left(\hat{\boldsymbol{ heta}}\right) pprox \mathcal{N}\left(\mathbf{g}\left(\boldsymbol{ heta}
ight), \mathbf{J}\hat{\mathbb{V}}\left(\hat{\boldsymbol{ heta}}
ight) \mathbf{J}'
ight).$$

## Value

Returns an object of class ctmeddelta which is a list with the following elements:

call Function call.

args Function arguments.

**fun** Function used ("DeltaTotalCentral").

**output** A list the length of which is equal to the length of delta\_t.

Each element in the output list has the following elements:

delta\_t Time interval.

jacobian Jacobian matrix.

est Estimated total effect centrality.

vcov Sampling variance-covariance matrix of estimated total effect centrality.

## Author(s)

Ivan Jacob Agaloos Pesigan

18 DeltaTotalCentral

#### References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. Sociological Methodology, 17, 37. doi:10.2307/271028

Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. Structural Equation Modeling: A Multidisciplinary Journal, 23 (1), 61–75. doi:10.1080/10705511.2014.973960

Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. Psychometrika, 87 (1), 214–252. doi:10.1007/s11336021097670

#### See Also

Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaMed(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()

```
phi <- matrix(</pre>
  data = c(
    -0.357, 0.771, -0.450,
   0.0, -0.511, 0.729,
   0, 0, -0.693
  ),
  nrow = 3
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
vcov_phi_vec <- matrix(</pre>
  data = c(
    0.002704274, -0.001475275, 0.000949122,
    -0.001619422, 0.000885122, -0.000569404,
    0.00085493, -0.000465824, 0.000297815,
    -0.001475275, 0.004428442, -0.002642303,
    0.000980573, -0.00271817, 0.001618805,
    -0.000586921, 0.001478421, -0.000871547,
    0.000949122, -0.002642303, 0.006402668,
    -0.000697798, 0.001813471, -0.004043138,
    0.000463086, -0.001120949, 0.002271711,
    -0.001619422, 0.000980573, -0.000697798,
    0.002079286, -0.001152501, 0.000753,
    -0.001528701, 0.000820587, -0.000517524,
    0.000885122, -0.00271817, 0.001813471,
    -0.001152501, 0.00342605, -0.002075005,
    0.000899165, -0.002532849, 0.001475579,
    -0.000569404, 0.001618805, -0.004043138,
    0.000753, -0.002075005, 0.004984032,
    -0.000622255, 0.001634917, -0.003705661,
    0.00085493, -0.000586921, 0.000463086,
    -0.001528701, 0.000899165, -0.000622255,
    0.002060076, -0.001096684, 0.000686386,
```

Direct 19

```
-0.000465824, 0.001478421, -0.001120949,
   0.000820587, -0.002532849, 0.001634917,
   -0.001096684, 0.003328692, -0.001926088,
   0.000297815, -0.000871547, 0.002271711,
   -0.000517524, 0.001475579, -0.003705661,
   0.000686386, -0.001926088, 0.004726235
 ),
 nrow = 9
)
# Specific time interval ------
DeltaTotalCentral(
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1
# Range of time intervals ------
delta <- DeltaTotalCentral(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1:5
)
plot(delta)
# Methods -----
# DeltaTotalCentral has a number of methods including
# print, summary, confint, and plot
print(delta)
summary(delta)
confint(delta, level = 0.95)
plot(delta)
```

Direct

Direct Effect of X on Y Over a Specific Time Interval

## Description

This function computes the direct effect of the independent variable X on the dependent variable Y through mediator variables  $\mathbf{m}$  over a specific time interval  $\Delta t$  using the first-order stochastic differential equation model's drift matrix  $\mathbf{\Phi}$ .

## Usage

```
Direct(phi, delta_t, from, to, med)
```

20 Direct

## **Arguments**

phi	Numeric matrix. The drift matrix $(\Phi)$ , phi should have row and column names pertaining to the variables in the system.
delta_t	Numeric. Time interval $(\Delta t)$ .
from	Character string. Name of the independent variable $X$ in phi.
to	Character string. Name of the dependent variable $Y$ in phi.
med	Character vector. Name/s of the mediator variable/s in phi.

## **Details**

The direct effect of the independent variable X on the dependent variable Y relative to some mediator variables  $\mathbf{m}$  is given by

$$\operatorname{Direct}_{\Delta t_{i,j,\mathbf{m}}} = \exp\left(\Delta t \mathbf{D} \mathbf{\Phi} \mathbf{D}\right)_{i,j}$$

where  $\Phi$  denotes the drift matrix,  $\mathbf{D}$  a diagonal matrix where the diagonal elements corresponding to mediator variables  $\mathbf{m}$  are set to zero and the rest to one, i the row index of Y in  $\Phi$ , j the column index of X in  $\Phi$ , and  $\Delta t$  the time interval.

#### **Linear Stochastic Differential Equation Model:**

The measurement model is given by

$$\mathbf{y}_{i,t} = oldsymbol{
u} + oldsymbol{\Lambda} oldsymbol{\eta}_{i,t} + oldsymbol{arepsilon}_{i,t}, \quad ext{with} \quad oldsymbol{arepsilon}_{i,t} \sim \mathcal{N}\left(\mathbf{0}, oldsymbol{\Theta}
ight)$$

where  $\mathbf{y}_{i,t}$ ,  $\eta_{i,t}$ , and  $\varepsilon_{i,t}$  are random variables and  $\nu$ ,  $\Lambda$ , and  $\Theta$  are model parameters.  $\mathbf{y}_{i,t}$  represents a vector of observed random variables,  $\eta_{i,t}$  a vector of latent random variables, and  $\varepsilon_{i,t}$  a vector of random measurement errors, at time t and individual t.  $\nu$  denotes a vector of intercepts,  $\Lambda$  a matrix of factor loadings, and  $\Theta$  the covariance matrix of  $\varepsilon$ .

An alternative representation of the measurement error is given by

$$oldsymbol{arepsilon}_{i,t} = oldsymbol{\Theta}^{rac{1}{2}} \mathbf{z}_{i,t}, \quad ext{with} \quad \mathbf{z}_{i,t} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}
ight)$$

where  $\mathbf{z}_{i,t}$  is a vector of independent standard normal random variables and  $\left(\mathbf{\Theta}^{\frac{1}{2}}\right)\left(\mathbf{\Theta}^{\frac{1}{2}}\right)' = \mathbf{\Theta}$ . The dynamic structure is given by

$$\mathrm{d} oldsymbol{\eta}_{i,t} = \left( oldsymbol{\iota} + oldsymbol{\Phi} oldsymbol{\eta}_{i,t} 
ight) \mathrm{d} t + oldsymbol{\Sigma}^{rac{1}{2}} \mathrm{d} \mathbf{W}_{i,t}$$

where  $\iota$  is a term which is unobserved and constant over time,  $\Phi$  is the drift matrix which represents the rate of change of the solution in the absence of any random fluctuations,  $\Sigma$  is the matrix of volatility or randomness in the process, and  $\mathrm{d}W$  is a Wiener process or Brownian motion, which represents random fluctuations.

## Value

Returns an object of class ctmedeffect which is a list with the following elements:

call Function call.

args Function arguments.

**fun** Function used ("Direct").

output The direct effect.

Direct 21

#### Author(s)

Ivan Jacob Agaloos Pesigan

## References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. Sociological Methodology, 17, 37. doi:10.2307/271028

Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. Structural Equation Modeling: A Multidisciplinary Journal, 23 (1), 61–75. doi:10.1080/10705511.2014.973960

Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. Psychometrika, 87 (1), 214–252. doi:10.1007/s11336021097670

#### See Also

```
Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()
```

```
phi <- matrix(</pre>
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
delta_t <- 1
Direct(
  phi = phi,
  delta_t = delta_t,
  from = "x",
  to = "y",
  med = "m"
phi <- matrix(</pre>
  data = c(
    -6, 5.5, 0, 0,
    1.25, -2.5, 5.9, -7.3,
    0, 0, -6, 2.5,
    5, 0, 0, -6
  ),
  nrow = 4
colnames(phi) <- rownames(phi) <- paste0("y", 1:4)</pre>
Direct(
```

22 Indirect

```
phi = phi,
  delta_t = delta_t,
  from = "y2",
  to = "y4",
  med = c("y1", "y3")
)
```

Indirect

Indirect Effect of X on Y Through M Over a Specific Time Interval

## Description

This function computes the indirect effect of the independent variable X on the dependent variable Y through mediator variables  $\mathbf{m}$  over a specific time interval  $\Delta t$  using the first-order stochastic differential equation model's drift matrix  $\mathbf{\Phi}$ .

#### Usage

```
Indirect(phi, delta_t, from, to, med)
```

## **Arguments**

phi	Numeric matrix. The drift matrix $(\Phi)$ . phi should have row and column names pertaining to the variables in the system.
delta_t	Numeric. Time interval ( $\Delta t$ ).
from	Character string. Name of the independent variable $X$ in phi.
to	Character string. Name of the dependent variable $Y$ in phi.
med	Character vector. Name/s of the mediator variable/s in phi.

## Details

The indirect effect of the independent variable X on the dependent variable Y relative to some mediator variables  $\mathbf{m}$  over a specific time interval  $\Delta t$  is given by

Indirect<sub>$$\Delta t$$</sub> = exp  $(\Delta t \mathbf{\Phi})_{i,j}$  - exp  $(\Delta t \mathbf{D_m} \mathbf{\Phi} \mathbf{D_m})_{i,j}$ 

where  $\Phi$  denotes the drift matrix,  $\mathbf{D_m}$  a matrix where the off diagonal elements are zeros and the diagonal elements are zero for the index/indices of mediator variables  $\mathbf{m}$  and one otherwise, i the row index of Y in  $\Phi$ , j the column index of X in  $\Phi$ , and  $\Delta t$  the time interval.

## **Linear Stochastic Differential Equation Model:**

The measurement model is given by

$$\mathbf{y}_{i,t} = \mathbf{\nu} + \mathbf{\Lambda} \boldsymbol{\eta}_{i,t} + \boldsymbol{arepsilon}_{i,t}, \quad ext{with} \quad \boldsymbol{arepsilon}_{i,t} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{\Theta}
ight)$$

where  $\mathbf{y}_{i,t}$ ,  $\boldsymbol{\eta}_{i,t}$ , and  $\boldsymbol{\varepsilon}_{i,t}$  are random variables and  $\boldsymbol{\nu}$ ,  $\boldsymbol{\Lambda}$ , and  $\boldsymbol{\Theta}$  are model parameters.  $\mathbf{y}_{i,t}$  represents a vector of observed random variables,  $\boldsymbol{\eta}_{i,t}$  a vector of latent random variables, and

Indirect 23

 $\varepsilon_{i,t}$  a vector of random measurement errors, at time t and individual i.  $\nu$  denotes a vector of intercepts,  $\Lambda$  a matrix of factor loadings, and  $\Theta$  the covariance matrix of  $\varepsilon$ .

An alternative representation of the measurement error is given by

$$\boldsymbol{\varepsilon}_{i,t} = \boldsymbol{\Theta}^{\frac{1}{2}} \mathbf{z}_{i,t}, \quad \text{with} \quad \mathbf{z}_{i,t} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)$$

where  $\mathbf{z}_{i,t}$  is a vector of independent standard normal random variables and  $\left(\Theta^{\frac{1}{2}}\right)\left(\Theta^{\frac{1}{2}}\right)' = \Theta$ . The dynamic structure is given by

$$\mathrm{d} oldsymbol{\eta}_{i,t} = \left( oldsymbol{\iota} + oldsymbol{\Phi} oldsymbol{\eta}_{i,t} 
ight) \mathrm{d} t + oldsymbol{\Sigma}^{rac{1}{2}} \mathrm{d} \mathbf{W}_{i,t}$$

where  $\iota$  is a term which is unobserved and constant over time,  $\Phi$  is the drift matrix which represents the rate of change of the solution in the absence of any random fluctuations,  $\Sigma$  is the matrix of volatility or randomness in the process, and  $\mathrm{d}W$  is a Wiener process or Brownian motion, which represents random fluctuations.

#### Value

Returns an object of class ctmedeffect which is a list with the following elements:

call Function call.

args Function arguments.

fun Function used ("Indirect").

output The indirect effect.

#### Author(s)

Ivan Jacob Agaloos Pesigan

#### References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. Sociological Methodology, 17, 37. doi:10.2307/271028

Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. Structural Equation Modeling: A Multidisciplinary Journal, 23 (1), 61–75. doi:10.1080/10705511.2014.973960

Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. Psychometrika, 87 (1), 214–252. doi:10.1007/s11336021097670

## See Also

Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Direct(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()

## **Examples**

```
phi <- matrix(</pre>
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) \leftarrow rownames(phi) \leftarrow c("x", "m", "y")
delta_t <- 1
Indirect(
  phi = phi,
  delta_t = delta_t,
  from = "x",
  to = "y",
  med = "m"
phi <- matrix(</pre>
  data = c(
    -6, 5.5, 0, 0,
   1.25, -2.5, 5.9, -7.3,
    0, 0, -6, 2.5,
   5, 0, 0, -6
  ),
  nrow = 4
)
colnames(phi) <- rownames(phi) <- paste0("y", 1:4)</pre>
Indirect(
  phi = phi,
  delta_t = delta_t,
  from = "y2",
  to = "y4",
  med = c("y1", "y3")
)
```

IndirectCentral

Indirect Effect Centrality

## **Description**

**Indirect Effect Centrality** 

## Usage

```
IndirectCentral(phi, delta_t)
```

## **Arguments**

phi	Numeric matrix. The drift matrix $(\Phi)$ , phi should have row and column names
	pertaining to the variables in the system.
delta_t	Vector of positive numbers. Time interval $(\Delta t)$ .

#### **Details**

Indirect effect centrality is the sum of all possible indirect effects between different pairs of variables in which a specific variable serves as the only mediator.

## Value

Returns an object of class ctmedmed which is a list with the following elements:

```
call Function call.args Function arguments.fun Function used ("IndirectCentral").output A matrix of indirect effect centrality.
```

## Author(s)

Ivan Jacob Agaloos Pesigan

#### References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. Sociological Methodology, 17, 37. doi:10.2307/271028

Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. Structural Equation Modeling: A Multidisciplinary Journal, 23 (1), 61–75. doi:10.1080/10705511.2014.973960

Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. Psychometrika, 87 (1), 214–252. doi:10.1007/s11336021097670

#### See Also

```
Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()
```

```
phi <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
),</pre>
```

26 MCBeta

```
nrow = 3
colnames(phi) \leftarrow rownames(phi) \leftarrow c("x", "m", "y")
# Specific time interval ------
IndirectCentral(
 phi = phi,
 delta_t = 1
)
# Range of time intervals ------
indirect_central <- IndirectCentral(</pre>
 phi = phi,
 delta_t = 1:30
plot(indirect_central)
# Methods ------
# IndirectCentral has a number of methods including
# print, summary, and plot
indirect_central <- IndirectCentral(</pre>
 phi = phi,
 delta_t = 1:5
print(indirect_central)
summary(indirect_central)
plot(indirect_central)
```

MCBeta

Monte Carlo Sampling Distribution for the Elements of the Matrix of Lagged Coefficients Over a Specific Time Interval or a Range of Time Intervals

## Description

This function generates a Monte Carlo method sampling distribution for the elements of the matrix of lagged coefficients  $\beta$  over a specific time interval  $\Delta t$  or a range of time intervals using the first-order stochastic differential equation model drift matrix  $\Phi$ .

## Usage

```
MCBeta(
   phi,
   vcov_phi_vec,
   delta_t,
   R,
   test_phi = TRUE,
   ncores = NULL,
   seed = NULL
)
```

MCBeta 27

#### **Arguments**

phi Numeric matrix. The drift matrix ( $\Phi$ ), phi should have row and column names

pertaining to the variables in the system.

vcov\_phi\_vec Numeric matrix. The sampling variance-covariance matrix of vec  $(\Phi)$ .

delta\_t Numeric. Time interval ( $\Delta t$ ).

R Positive integer. Number of replications.

test\_phi Logical. If test\_phi = TRUE, the function tests the stability of the generated

drift matrix  $\Phi$ . If the test returns FALSE, the function generates a new drift

matrix  $\Phi$  and runs the test recursively until the test returns TRUE.

ncores Positive integer. Number of cores to use. If ncores = NULL, use a single core.

Consider using multiple cores when number of replications R is a large value.

seed Random seed.

## **Details**

See Total().

#### **Monte Carlo Method:**

Let  $\theta$  be  $\operatorname{vec}(\Phi)$ , that is, the elements of the  $\Phi$  matrix in vector form sorted column-wise. Let  $\hat{\theta}$  be  $\operatorname{vec}(\hat{\Phi})$ . Based on the asymptotic properties of maximum likelihood estimators, we can assume that estimators are normally distributed around the population parameters.

$$\hat{oldsymbol{ heta}} \sim \mathcal{N}\left(oldsymbol{ heta}, \mathbb{V}\left(\hat{oldsymbol{ heta}}
ight)
ight)$$

Using this distributional assumption, a sampling distribution of  $\hat{\theta}$  which we refer to as  $\hat{\theta}^*$  can be generated by replacing the population parameters with sample estimates, that is,

$$\hat{oldsymbol{ heta}}^* \sim \mathcal{N}\left(\hat{oldsymbol{ heta}}, \hat{\mathbb{V}}\left(\hat{oldsymbol{ heta}}
ight)
ight).$$

Let  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right)$  be a parameter that is a function of the estimated parameters. A sampling distribution of  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right)$ , which we refer to as  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}^*\right)$ , can be generated by using the simulated estimates to calculate  $\mathbf{g}$ . The standard deviations of the simulated estimates are the standard errors. Percentiles corresponding to  $100\left(1-\alpha\right)\%$  are the confidence intervals.

#### Value

Returns an object of class ctmedmc which is a list with the following elements:

call Function call.

args Function arguments.

fun Function used ("MCBeta").

**output** A list the length of which is equal to the length of delta\_t.

Each element in the output list has the following elements:

**est** A vector of total, direct, and indirect effects.

thetahatstar A matrix of Monte Carlo total, direct, and indirect effects.

28 MCBeta

#### Author(s)

Ivan Jacob Agaloos Pesigan

#### References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. Sociological Methodology, 17, 37. doi:10.2307/271028

Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. Structural Equation Modeling: A Multidisciplinary Journal, 23 (1), 61–75. doi:10.1080/10705511.2014.973960

Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. Psychometrika, 87 (1), 214–252. doi:10.1007/s11336021097670

#### See Also

```
Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), McIndirectCentral(), McMed(), McPhi(), McTotalCentral(), Med(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()
```

```
set.seed(42)
phi <- matrix(</pre>
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
vcov_phi_vec <- matrix(</pre>
  data = c(
    0.002704274, -0.001475275, 0.000949122,
    -0.001619422, 0.000885122, -0.000569404,
    0.00085493, -0.000465824, 0.000297815,
    -0.001475275, 0.004428442, -0.002642303,
    0.000980573, -0.00271817, 0.001618805,
    -0.000586921, 0.001478421, -0.000871547,
    0.000949122, -0.002642303, 0.006402668,
    -0.000697798, 0.001813471, -0.004043138,
    0.000463086, -0.001120949, 0.002271711,
    -0.001619422, 0.000980573, -0.000697798,
    0.002079286, -0.001152501, 0.000753,
    -0.001528701, 0.000820587, -0.000517524,
    0.000885122, -0.00271817, 0.001813471,
    -0.001152501, 0.00342605, -0.002075005,
    0.000899165, -0.002532849, 0.001475579,
    -0.000569404, 0.001618805, -0.004043138,
```

```
0.000753, -0.002075005, 0.004984032,
   -0.000622255, 0.001634917, -0.003705661,
   0.00085493, -0.000586921, 0.000463086,
   -0.001528701, 0.000899165, -0.000622255,
    0.002060076, \ -0.001096684, \ 0.000686386, \\
   -0.000465824, 0.001478421, -0.001120949,
   0.000820587, -0.002532849, 0.001634917,
   -0.001096684, 0.003328692, -0.001926088,
   0.000297815, -0.000871547, 0.002271711,
   -0.000517524, 0.001475579, -0.003705661,
   0.000686386, -0.001926088, 0.004726235
 ),
 nrow = 9
)
# Specific time interval ------
MCBeta(
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1,
 R = 100L # use a large value for R in actual research
)
# Range of time intervals ------
mc <- MCBeta(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1:5,
 R = 100L # use a large value for R in actual research
)
plot(mc)
# Methods -----
# MCBeta has a number of methods including
# print, summary, confint, and plot
print(mc)
summary(mc)
confint(mc, level = 0.95)
plot(mc)
```

MCIndirectCentral

Monte Carlo Sampling Distribution of Indirect Effect Centrality Over a Specific Time Interval or a Range of Time Intervals

## **Description**

This function generates a Monte Carlo method sampling distribution of the indirect effect centrality at a particular time interval  $\Delta t$  using the first-order stochastic differential equation model drift matrix  $\Phi$ .

## Usage

30

```
MCIndirectCentral(
  phi,
  vcov_phi_vec,
  delta_t,
  R,
  test_phi = TRUE,
  ncores = NULL,
  seed = NULL
)
```

## **Arguments**

phi Numeric matrix. The drift matrix ( $\Phi$ ). phi should have row and column names

pertaining to the variables in the system.

vcov\_phi\_vec Numeric matrix. The sampling variance-covariance matrix of vec  $(\Phi)$ .

delta\_t Numeric. Time interval ( $\Delta t$ ).

R Positive integer. Number of replications.

test\_phi Logical. If test\_phi = TRUE, the function tests the stability of the generated

drift matrix  $\Phi$ . If the test returns FALSE, the function generates a new drift

matrix  $\Phi$  and runs the test recursively until the test returns TRUE.

ncores Positive integer. Number of cores to use. If ncores = NULL, use a single core.

Consider using multiple cores when number of replications R is a large value.

seed Random seed.

#### **Details**

See IndirectCentral() for more details.

## **Monte Carlo Method:**

Let  $\theta$  be  $\operatorname{vec}(\Phi)$ , that is, the elements of the  $\Phi$  matrix in vector form sorted column-wise. Let  $\hat{\theta}$  be  $\operatorname{vec}(\hat{\Phi})$ . Based on the asymptotic properties of maximum likelihood estimators, we can assume that estimators are normally distributed around the population parameters.

$$\hat{oldsymbol{ heta}} \sim \mathcal{N}\left(oldsymbol{ heta}, \mathbb{V}\left(\hat{oldsymbol{ heta}}
ight)
ight)$$

Using this distributional assumption, a sampling distribution of  $\hat{\theta}$  which we refer to as  $\hat{\theta}^*$  can be generated by replacing the population parameters with sample estimates, that is,

$$\hat{oldsymbol{ heta}}^* \sim \mathcal{N}\left(\hat{oldsymbol{ heta}}, \hat{\mathbb{V}}\left(\hat{oldsymbol{ heta}}
ight)
ight).$$

Let  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right)$  be a parameter that is a function of the estimated parameters. A sampling distribution of  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right)$ , which we refer to as  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}^*\right)$ , can be generated by using the simulated estimates to calculate  $\mathbf{g}$ . The standard deviations of the simulated estimates are the standard errors. Percentiles corresponding to  $100 \left(1-\alpha\right)\%$  are the confidence intervals.

#### Value

Returns an object of class ctmedmc which is a list with the following elements:

call Function call.

args Function arguments.

fun Function used ("MCIndirectCentral").

**output** A list the length of which is equal to the length of delta\_t.

Each element in the output list has the following elements:

est A vector of indirect effect centrality.

thetahatstar A matrix of Monte Carlo indirect effect centrality.

## Author(s)

Ivan Jacob Agaloos Pesigan

#### References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. Sociological Methodology, 17, 37. doi:10.2307/271028

Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. Structural Equation Modeling: A Multidisciplinary Journal, 23 (1), 61–75. doi:10.1080/10705511.2014.973960

Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. Psychometrika, 87 (1), 214–252. doi:10.1007/s11336021097670

#### See Also

Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()

```
set.seed(42)
phi <- matrix(
    data = c(
        -0.357, 0.771, -0.450,
        0.0, -0.511, 0.729,
        0, 0, -0.693
    ),
    nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")
vcov_phi_vec <- matrix(
    data = c(
        0.002704274, -0.001475275, 0.000949122,</pre>
```

```
-0.001619422, 0.000885122, -0.000569404,
   0.00085493, -0.000465824, 0.000297815,
   -0.001475275, 0.004428442, -0.002642303,
   0.000980573, -0.00271817, 0.001618805,
   -0.000586921, 0.001478421, -0.000871547,
   0.000949122, -0.002642303, 0.006402668,
   -0.000697798, 0.001813471, -0.004043138,
   0.000463086, -0.001120949, 0.002271711,
   -0.001619422, 0.000980573, -0.000697798,
   0.002079286, -0.001152501, 0.000753,
   -0.001528701, 0.000820587, -0.000517524,
   0.000885122, -0.00271817, 0.001813471,
   -0.001152501, 0.00342605, -0.002075005,
   0.000899165, -0.002532849, 0.001475579,
   -0.000569404, 0.001618805, -0.004043138,
   0.000753, -0.002075005, 0.004984032,
   -0.000622255, 0.001634917, -0.003705661,
   0.00085493, -0.000586921, 0.000463086,
   -0.001528701, 0.000899165, -0.000622255,
   0.002060076, -0.001096684, 0.000686386,
   -0.000465824, 0.001478421, -0.001120949,
   0.000820587, -0.002532849, 0.001634917,
   -0.001096684, 0.003328692, -0.001926088,
   0.000297815, -0.000871547, 0.002271711,
   -0.000517524, 0.001475579, -0.003705661,
    0.000686386 , -0.001926088 , \ 0.004726235 \\
 ),
 nrow = 9
)
# Specific time interval ------
MCIndirectCentral(
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1,
 R = 100L # use a large value for R in actual research
)
# Range of time intervals ------
mc <- MCIndirectCentral(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1:5,
 R = 100L # use a large value for R in actual research
)
plot(mc)
# Methods -----
# MCIndirectCentral has a number of methods including
# print, summary, confint, and plot
print(mc)
summary(mc)
confint(mc, level = 0.95)
```

MCMed 33

plot(mc)

MCMed Monte Carlo Sampling Distribution of Total, Direct, and Indirect Ef-

fects of X on Y Through M Over a Specific Time Interval or a Range of

Time Intervals

## **Description**

This function generates a Monte Carlo method sampling distribution of the total, direct and indirect effects of the independent variable X on the dependent variable Y through mediator variables  $\mathbf{m}$  over a specific time interval  $\Delta t$  or a range of time intervals using the first-order stochastic differential equation model drift matrix  $\mathbf{\Phi}$ .

## Usage

```
MCMed(
    phi,
    vcov_phi_vec,
    delta_t,
    from,
    to,
    med,
    R,
    test_phi = TRUE,
    ncores = NULL,
    seed = NULL
)
```

## Arguments

phi Numeric matrix. The drift matrix	$(\Phi)$ . phi should have row and column names
--------------------------------------	---

pertaining to the variables in the system.

vcov\_phi\_vec Numeric matrix. The sampling variance-covariance matrix of  $vec(\Phi)$ .

delta\_t Numeric. Time interval ( $\Delta t$ ).

from Character string. Name of the independent variable X in phi. to Character string. Name of the dependent variable Y in phi. med Character vector. Name/s of the mediator variable/s in phi.

R Positive integer. Number of replications.

test\_phi Logical. If test\_phi = TRUE, the function tests the stability of the generated

drift matrix  $\Phi$ . If the test returns FALSE, the function generates a new drift

matrix  $\Phi$  and runs the test recursively until the test returns TRUE.

ncores Positive integer. Number of cores to use. If ncores = NULL, use a single core.

Consider using multiple cores when number of replications R is a large value.

seed Random seed.

#### **Details**

See Total(), Direct(), and Indirect() for more details.

#### **Monte Carlo Method:**

Let  $\theta$  be  $\operatorname{vec}(\Phi)$ , that is, the elements of the  $\Phi$  matrix in vector form sorted column-wise. Let  $\hat{\theta}$  be  $\operatorname{vec}(\hat{\Phi})$ . Based on the asymptotic properties of maximum likelihood estimators, we can assume that estimators are normally distributed around the population parameters.

$$\hat{oldsymbol{ heta}} \sim \mathcal{N}\left(oldsymbol{ heta}, \mathbb{V}\left(\hat{oldsymbol{ heta}}
ight)
ight)$$

Using this distributional assumption, a sampling distribution of  $\hat{\theta}$  which we refer to as  $\hat{\theta}^*$  can be generated by replacing the population parameters with sample estimates, that is,

$$\hat{oldsymbol{ heta}}^* \sim \mathcal{N}\left(\hat{oldsymbol{ heta}}, \hat{\mathbb{V}}\left(\hat{oldsymbol{ heta}}
ight)
ight).$$

Let  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right)$  be a parameter that is a function of the estimated parameters. A sampling distribution of  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right)$ , which we refer to as  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}^*\right)$ , can be generated by using the simulated estimates to calculate  $\mathbf{g}$ . The standard deviations of the simulated estimates are the standard errors. Percentiles corresponding to  $100\left(1-\alpha\right)\%$  are the confidence intervals.

#### Value

Returns an object of class ctmedmc which is a list with the following elements:

call Function call.

args Function arguments.

fun Function used ("MCMed").

output A list with length of length(delta\_t).

Each element in the output list has the following elements:

est A vector of total, direct, and indirect effects.

thetahatstar A matrix of Monte Carlo total, direct, and indirect effects.

## Author(s)

Ivan Jacob Agaloos Pesigan

#### References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. Sociological Methodology, 17, 37. doi:10.2307/271028

Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. Structural Equation Modeling: A Multidisciplinary Journal, 23 (1), 61–75. doi:10.1080/10705511.2014.973960

Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. Psychometrika, 87 (1), 214–252. doi:10.1007/s11336021097670

MCMed 35

#### See Also

Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCPhi(), MCTotalCentral(), Med(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()

```
set.seed(42)
phi <- matrix(</pre>
  data = c(
   -0.357, 0.771, -0.450,
   0.0, -0.511, 0.729,
   0, 0, -0.693
  ).
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
vcov_phi_vec <- matrix(</pre>
  data = c(
    0.002704274, -0.001475275, 0.000949122,
    -0.001619422, 0.000885122, -0.000569404,
    0.00085493, -0.000465824, 0.000297815,
    -0.001475275, 0.004428442, -0.002642303,
    0.000980573, -0.00271817, 0.001618805,
    -0.000586921, 0.001478421, -0.000871547,
    0.000949122, -0.002642303, 0.006402668,
    -0.000697798, 0.001813471, -0.004043138,
    0.000463086, -0.001120949, 0.002271711,
    -0.001619422, 0.000980573, -0.000697798,
    0.002079286, -0.001152501, 0.000753,
    -0.001528701, 0.000820587, -0.000517524,
    0.000885122, -0.00271817, 0.001813471,
    -0.001152501, 0.00342605, -0.002075005,
    0.000899165, -0.002532849, 0.001475579,
    -0.000569404, 0.001618805, -0.004043138,
    0.000753, -0.002075005, 0.004984032,
    -0.000622255, 0.001634917, -0.003705661,
    0.00085493, -0.000586921, 0.000463086,
    -0.001528701, 0.000899165, -0.000622255,
    0.002060076, -0.001096684, 0.000686386,
    -0.000465824, 0.001478421, -0.001120949,
    0.000820587, -0.002532849, 0.001634917,
    -0.001096684, 0.003328692, -0.001926088,
    0.000297815, -0.000871547, 0.002271711,
    -0.000517524, 0.001475579, -0.003705661,
    0.000686386, -0.001926088, 0.004726235
  ),
  nrow = 9
)
# Specific time interval --------
```

36 MCPhi

```
MCMed(
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1,
 from = "x",
 to = "y",
 med = "m",
 R = 100L # use a large value for R in actual research
)
# Range of time intervals ------
mc <- MCMed(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1:5,
 from = "x",
 to = "y",
 med = "m",
 R = 100L # use a large value for R in actual research
plot(mc)
# Methods -----
# MCMed has a number of methods including
# print, summary, confint, and plot
print(mc)
summary(mc)
confint(mc, level = 0.95)
```

MCPhi

Generate Random Drift Matrices Using the Monte Carlo Method

## **Description**

This function generates random drift matrices  $\Phi$  using the Monte Carlo method.

## Usage

```
MCPhi(phi, vcov_phi_vec, R, test_phi = TRUE, ncores = NULL, seed = NULL)
```

## **Arguments**

phi	Numeric matrix. The drift matrix $(\Phi)$ , phi should have row and column names pertaining to the variables in the system.
vcov_phi_vec	Numeric matrix. The sampling variance-covariance matrix of $\operatorname{vec}\left(\mathbf{\Phi}\right)$ .
R	Positive integer. Number of replications.

MCPhi 37

test\_phi Logical. If test\_phi = TRUE, the function tests the stability of the generated

drift matrix  $\Phi$ . If the test returns FALSE, the function generates a new drift

matrix  $\Phi$  and runs the test recursively until the test returns TRUE.

ncores Positive integer. Number of cores to use. If ncores = NULL, use a single core.

Consider using multiple cores when number of replications R is a large value.

seed Random seed.

### **Details**

#### **Monte Carlo Method:**

Let  $\theta$  be  $\operatorname{vec}(\Phi)$ , that is, the elements of the  $\Phi$  matrix in vector form sorted column-wise. Let  $\hat{\theta}$  be  $\operatorname{vec}(\hat{\Phi})$ . Based on the asymptotic properties of maximum likelihood estimators, we can assume that estimators are normally distributed around the population parameters.

$$\hat{oldsymbol{ heta}} \sim \mathcal{N}\left(oldsymbol{ heta}, \mathbb{V}\left(\hat{oldsymbol{ heta}}
ight)
ight)$$

Using this distributional assumption, a sampling distribution of  $\hat{\theta}$  which we refer to as  $\hat{\theta}^*$  can be generated by replacing the population parameters with sample estimates, that is,

$$\hat{oldsymbol{ heta}}^* \sim \mathcal{N}\left(\hat{oldsymbol{ heta}}, \hat{\mathbb{V}}\left(\hat{oldsymbol{ heta}}
ight)
ight).$$

#### Value

Returns an object of class ctmedmc which is a list with the following elements:

call Function call.

args Function arguments.

fun Function used ("MCPhi").

output A list simulated drift matrices.

### Author(s)

Ivan Jacob Agaloos Pesigan

# See Also

Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCTotalCentral(), Med(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()

```
set.seed(42)
phi <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,</pre>
```

38 MCTotalCentral

```
0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
  phi = phi,
  vcov_phi_vec = 0.1 * diag(9),
  R = 100L # use a large value for R in actual research
)
phi <- matrix(</pre>
  data = c(
    -6, 5.5, 0, 0,
    1.25, -2.5, 5.9, -7.3,
    0, 0, -6, 2.5,
    5, 0, 0, -6
  ),
  nrow = 4
)
colnames(phi) <- rownames(phi) <- paste0("y", 1:4)</pre>
MCPhi(
  phi = phi,
  vcov_phi_vec = 0.1 * diag(16),
  R = 100L, # use a large value for R in actual research
  test_phi = FALSE
)
```

MCTotalCentral

Monte Carlo Sampling Distribution of Total Effect Centrality Over a Specific Time Interval or a Range of Time Intervals

# Description

This function generates a Monte Carlo method sampling distribution of the total effect centrality at a particular time interval  $\Delta t$  using the first-order stochastic differential equation model drift matrix  $\Phi$ .

# Usage

```
MCTotalCentral(
   phi,
   vcov_phi_vec,
   delta_t,
   R,
   test_phi = TRUE,
   ncores = NULL,
   seed = NULL
)
```

MCTotalCentral 39

#### **Arguments**

phi Numeric matrix. The drift matrix ( $\Phi$ ), phi should have row and column names

pertaining to the variables in the system.

vcov\_phi\_vec Numeric matrix. The sampling variance-covariance matrix of vec  $(\Phi)$ .

delta\_t Numeric. Time interval ( $\Delta t$ ).

R Positive integer. Number of replications.

test\_phi Logical. If test\_phi = TRUE, the function tests the stability of the generated

drift matrix  $\Phi$ . If the test returns FALSE, the function generates a new drift

matrix  $\Phi$  and runs the test recursively until the test returns TRUE.

ncores Positive integer. Number of cores to use. If ncores = NULL, use a single core.

Consider using multiple cores when number of replications R is a large value.

seed Random seed.

#### **Details**

See TotalCentral() for more details.

#### **Monte Carlo Method:**

Let  $\theta$  be  $\operatorname{vec}(\Phi)$ , that is, the elements of the  $\Phi$  matrix in vector form sorted column-wise. Let  $\hat{\theta}$  be  $\operatorname{vec}(\hat{\Phi})$ . Based on the asymptotic properties of maximum likelihood estimators, we can assume that estimators are normally distributed around the population parameters.

$$\hat{oldsymbol{ heta}} \sim \mathcal{N}\left(oldsymbol{ heta}, \mathbb{V}\left(\hat{oldsymbol{ heta}}
ight)
ight)$$

Using this distributional assumption, a sampling distribution of  $\hat{\theta}$  which we refer to as  $\hat{\theta}^*$  can be generated by replacing the population parameters with sample estimates, that is,

$$\hat{oldsymbol{ heta}}^* \sim \mathcal{N}\left(\hat{oldsymbol{ heta}}, \hat{\mathbb{V}}\left(\hat{oldsymbol{ heta}}
ight)
ight).$$

Let  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right)$  be a parameter that is a function of the estimated parameters. A sampling distribution of  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right)$ , which we refer to as  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}^*\right)$ , can be generated by using the simulated estimates to calculate  $\mathbf{g}$ . The standard deviations of the simulated estimates are the standard errors. Percentiles corresponding to  $100\left(1-\alpha\right)\%$  are the confidence intervals.

#### Value

Returns an object of class ctmedmc which is a list with the following elements:

call Function call.

**args** Function arguments.

fun Function used ("MCTotalCentral").

**output** A list the length of which is equal to the length of delta\_t.

Each element in the output list has the following elements:

**est** A vector of total effect centrality.

thetahatstar A matrix of Monte Carlo total effect centrality.

40 MCTotalCentral

#### Author(s)

Ivan Jacob Agaloos Pesigan

#### References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. Sociological Methodology, 17, 37. doi:10.2307/271028

Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. Structural Equation Modeling: A Multidisciplinary Journal, 23 (1), 61–75. doi:10.1080/10705511.2014.973960

Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. Psychometrika, 87 (1), 214–252. doi:10.1007/s11336021097670

#### See Also

```
Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), Med(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()
```

```
set.seed(42)
phi <- matrix(</pre>
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
vcov_phi_vec <- matrix(</pre>
  data = c(
    0.002704274, -0.001475275, 0.000949122,
    -0.001619422, 0.000885122, -0.000569404,
    0.00085493, -0.000465824, 0.000297815,
    -0.001475275, 0.004428442, -0.002642303,
    0.000980573, -0.00271817, 0.001618805,
    -0.000586921, 0.001478421, -0.000871547,
    0.000949122, -0.002642303, 0.006402668,
    -0.000697798, 0.001813471, -0.004043138,
    0.000463086, -0.001120949, 0.002271711,
    -0.001619422, 0.000980573, -0.000697798,
    0.002079286, -0.001152501, 0.000753,
    -0.001528701, 0.000820587, -0.000517524,
    0.000885122, -0.00271817, 0.001813471,
    -0.001152501, 0.00342605, -0.002075005,
    0.000899165, -0.002532849, 0.001475579,
    -0.000569404, 0.001618805, -0.004043138,
```

Med 41

```
0.000753, -0.002075005, 0.004984032,
   -0.000622255, 0.001634917, -0.003705661,
   0.00085493, -0.000586921, 0.000463086,
   -0.001528701, 0.000899165, -0.000622255,
    0.002060076, \ -0.001096684, \ 0.000686386, \\
   -0.000465824, 0.001478421, -0.001120949,
   0.000820587, -0.002532849, 0.001634917,
   -0.001096684, 0.003328692, -0.001926088,
   0.000297815, -0.000871547, 0.002271711,
   -0.000517524, 0.001475579, -0.003705661,
   0.000686386, -0.001926088, 0.004726235
 ),
 nrow = 9
)
# Specific time interval ------
MCTotalCentral(
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1,
 R = 100L # use a large value for R in actual research
)
# Range of time intervals ------
mc <- MCTotalCentral(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1:5,
 R = 100L # use a large value for R in actual research
)
plot(mc)
# Methods ------
# MCTotalCentral has a number of methods including
# print, summary, confint, and plot
print(mc)
summary(mc)
confint(mc, level = 0.95)
plot(mc)
```

Med

Total, Direct, and Indirect Effects of X on Y Through M Over a Specific Time Interval or a Range of Time Intervals

### **Description**

This function computes the total, direct, and indirect effects of the independent variable X on the dependent variable Y through mediator variables  $\mathbf{m}$  over a specific time interval  $\Delta t$  or a range of time intervals using the first-order stochastic differential equation model's drift matrix  $\mathbf{\Phi}$ .

42 Med

#### **Usage**

Med(phi, delta\_t, from, to, med)

#### **Arguments**

delta_t Vector of positive numbers. Time interval $(\Delta t)$ .  from Character string. Name of the independent variable $X$ in phi.  to Character string. Name of the dependent variable $Y$ in phi.  med Character vector. Name/s of the mediator variable/s in phi.	phi	Numeric matrix. The drift matrix $(\Phi)$ , phi should have row and column names pertaining to the variables in the system.
to Character string. Name of the dependent variable $Y$ in phi.	delta_t	Vector of positive numbers. Time interval $(\Delta t)$ .
	from	Character string. Name of the independent variable $X$ in phi.
med Character vector. Name/s of the mediator variable/s in phi.	to	Character string. Name of the dependent variable $Y$ in phi.
	med	Character vector. Name/s of the mediator variable/s in phi.

#### **Details**

See Total(), Direct(), and Indirect() for more details.

#### **Linear Stochastic Differential Equation Model:**

The measurement model is given by

$$\mathbf{y}_{i,t} = \mathbf{\nu} + \mathbf{\Lambda} \boldsymbol{\eta}_{i,t} + \boldsymbol{arepsilon}_{i,t}, \quad ext{with} \quad \boldsymbol{arepsilon}_{i,t} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{\Theta}
ight)$$

where  $\mathbf{y}_{i,t}$ ,  $\eta_{i,t}$ , and  $\varepsilon_{i,t}$  are random variables and  $\nu$ ,  $\Lambda$ , and  $\Theta$  are model parameters.  $\mathbf{y}_{i,t}$  represents a vector of observed random variables,  $\eta_{i,t}$  a vector of latent random variables, and  $\varepsilon_{i,t}$  a vector of random measurement errors, at time t and individual t.  $\nu$  denotes a vector of intercepts,  $\Lambda$  a matrix of factor loadings, and  $\Theta$  the covariance matrix of  $\varepsilon$ .

An alternative representation of the measurement error is given by

$$\boldsymbol{\varepsilon}_{i,t} = \boldsymbol{\Theta}^{\frac{1}{2}} \mathbf{z}_{i,t}, \quad \text{with} \quad \mathbf{z}_{i,t} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)$$

where  $\mathbf{z}_{i,t}$  is a vector of independent standard normal random variables and  $\left(\Theta^{\frac{1}{2}}\right)\left(\Theta^{\frac{1}{2}}\right)' = \Theta$ . The dynamic structure is given by

$$\mathrm{d} \boldsymbol{\eta}_{i,t} = \left( \boldsymbol{\iota} + \boldsymbol{\Phi} \boldsymbol{\eta}_{i,t} \right) \mathrm{d} t + \boldsymbol{\Sigma}^{\frac{1}{2}} \mathrm{d} \mathbf{W}_{i,t}$$

where  $\iota$  is a term which is unobserved and constant over time,  $\Phi$  is the drift matrix which represents the rate of change of the solution in the absence of any random fluctuations,  $\Sigma$  is the matrix of volatility or randomness in the process, and  $\mathrm{d}W$  is a Wiener process or Brownian motion, which represents random fluctuations.

# Value

Returns an object of class ctmedmed which is a list with the following elements:

call Function call.

args Function arguments.

fun Function used ("Med").

output A matrix of total, direct, and indirect effects.

Med 43

#### Author(s)

Ivan Jacob Agaloos Pesigan

#### References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. Sociological Methodology, 17, 37. doi:10.2307/271028

Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. Structural Equation Modeling: A Multidisciplinary Journal, 23 (1), 61–75. doi:10.1080/10705511.2014.973960

Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. Psychometrika, 87 (1), 214–252. doi:10.1007/s11336021097670

#### See Also

Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()

```
phi <- matrix(</pre>
 data = c(
   -0.357, 0.771, -0.450,
   0.0, -0.511, 0.729,
   0, 0, -0.693
 ),
 nrow = 3
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
# Specific time interval -------
Med(
 phi = phi,
 delta_t = 1,
 from = "x",
 to = "y",
 med = "m"
)
# Range of time intervals -----
med <- Med(
 phi = phi,
 delta_t = 1:30,
 from = "x",
 to = "y",
 med = "m"
plot(med)
```

44 plot.ctmeddelta

```
# Methods ------
# Med has a number of methods including
# print, summary, and plot
med <- Med(
    phi = phi,
    delta_t = 1:5,
    from = "x",
    to = "y",
    med = "m"
)
print(med)
summary(med)
plot(med)</pre>
```

plot.ctmeddelta

Plot Method for an Object of Class ctmeddelta

# Description

Plot Method for an Object of Class ctmeddelta

# Usage

```
## S3 method for class 'ctmeddelta'
plot(x, alpha = 0.05, col = NULL, ...)
```

# Arguments

Χ	Object of class ctmeddelta.
alpha	Numeric. Significance level
col	Character vector. Optional argument. Character vector of colors.
	Additional arguments.

# Value

Displays plots of point estimates and confidence intervals.

# Author(s)

Ivan Jacob Agaloos Pesigan

plot.ctmeddelta 45

```
phi <- matrix(</pre>
 data = c(
   -0.357, 0.771, -0.450,
   0.0, -0.511, 0.729,
   0, 0, -0.693
 nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
vcov_phi_vec <- matrix(</pre>
 data = c(
   0.002704274, -0.001475275, 0.000949122,
    -0.001619422, 0.000885122, -0.000569404,
   0.00085493, -0.000465824, 0.000297815,
    -0.001475275, 0.004428442, -0.002642303,
   0.000980573, -0.00271817, 0.001618805,
    -0.000586921, 0.001478421, -0.000871547,
    0.000949122, -0.002642303, 0.006402668,
    -0.000697798, 0.001813471, -0.004043138,
    0.000463086, -0.001120949, 0.002271711,
    -0.001619422, 0.000980573, -0.000697798,
   0.002079286, -0.001152501, 0.000753,
   -0.001528701, 0.000820587, -0.000517524,
   0.000885122, -0.00271817, 0.001813471,
    -0.001152501, 0.00342605, -0.002075005,
   0.000899165, -0.002532849, 0.001475579,
    -0.000569404, 0.001618805, -0.004043138,
   0.000753, -0.002075005, 0.004984032,
    -0.000622255, 0.001634917, -0.003705661,
   0.00085493, -0.000586921, 0.000463086,
    -0.001528701, 0.000899165, -0.000622255,
    0.002060076, -0.001096684, 0.000686386,
    -0.000465824, 0.001478421, -0.001120949,
   0.000820587, -0.002532849, 0.001634917,
    -0.001096684, 0.003328692, -0.001926088,
   0.000297815, -0.000871547, 0.002271711,
   -0.000517524, 0.001475579, -0.003705661,
    0.000686386, \ -0.001926088, \ 0.004726235 
 ),
 nrow = 9
)
# Range of time intervals ------
delta <- DeltaMed(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1:5,
 from = "x",
 to = "y",
 med = "m"
)
```

46 plot.ctmedmc

```
plot(delta)
```

plot.ctmedmc

Plot Method for an Object of Class ctmedmc

### **Description**

Plot Method for an Object of Class ctmedmc

# Usage

```
## S3 method for class 'ctmedmc'
plot(x, alpha = 0.05, col = NULL, ...)
```

### **Arguments**

```
    x Object of class ctmedmc.
    alpha Numeric. Significance level
    col Character vector. Optional argument. Character vector of colors.
    ... Additional arguments.
```

#### Value

Displays plots of point estimates and confidence intervals.

# Author(s)

Ivan Jacob Agaloos Pesigan

```
set.seed(42)
phi <- matrix(
    data = c(
        -0.357, 0.771, -0.450,
        0.0, -0.511, 0.729,
        0, 0, -0.693
    ),
    nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")
vcov_phi_vec <- matrix(
    data = c(
        0.002704274, -0.001475275, 0.000949122,
        -0.001619422, 0.000885122, -0.000569404,
        0.00085493, -0.000465824, 0.000297815,
        -0.001475275, 0.004428442, -0.002642303,</pre>
```

plot.ctmedmed 47

```
0.000980573, -0.00271817, 0.001618805,
    -0.000586921, 0.001478421, -0.000871547,
   0.000949122, -0.002642303, 0.006402668,
   -0.000697798, 0.001813471, -0.004043138,
    0.000463086, \ -0.001120949, \ 0.002271711, \\
   -0.001619422, 0.000980573, -0.000697798,
   0.002079286, -0.001152501, 0.000753,
    -0.001528701, 0.000820587, -0.000517524,
   0.000885122, -0.00271817, 0.001813471,
    -0.001152501, 0.00342605, -0.002075005,
   0.000899165, -0.002532849, 0.001475579,
    -0.000569404, 0.001618805, -0.004043138,
   0.000753, -0.002075005, 0.004984032,
    -0.000622255, 0.001634917, -0.003705661,
   0.00085493, -0.000586921, 0.000463086,
    -0.001528701, 0.000899165, -0.000622255,
   0.002060076, -0.001096684, 0.000686386,
    -0.000465824, 0.001478421, -0.001120949,
   0.000820587, -0.002532849, 0.001634917,
    -0.001096684, 0.003328692, -0.001926088,
   0.000297815, -0.000871547, 0.002271711,
    -0.000517524, 0.001475579, -0.003705661,
   0.000686386, -0.001926088, 0.004726235
 ),
 nrow = 9
)
# Range of time intervals -------
mc <- MCMed(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1:5,
 from = "x",
 to = "y",
 med = "m",
 R = 100L # use a large value for R in actual research
plot(mc)
```

plot.ctmedmed

Plot Method for an Object of Class ctmedmed

### Description

Plot Method for an Object of Class ctmedmed

#### Usage

```
## S3 method for class 'ctmedmed'
plot(x, col = NULL, legend_pos = "topright", ...)
```

48 plot.ctmedtraj

## **Arguments**

X	Object of class ctmedmed.
col	Character vector. Optional argument. Character vector of colors.
legend_pos	Character vector. Optional argument. Legend position.
	Additional arguments.

### Value

Displays plots of point estimates and confidence intervals.

# Author(s)

Ivan Jacob Agaloos Pesigan

# **Examples**

```
phi <- matrix(</pre>
 data = c(
   -0.357, 0.771, -0.450,
   0.0, -0.511, 0.729,
   0, 0, -0.693
 ),
 nrow = 3
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
# Range of time intervals -----
med <- Med(
 phi = phi,
 delta_t = 1:5,
 from = "x",
 to = "y",
 med = "m"
)
plot(med)
```

plot.ctmedtraj

Plot Method for an Object of Class ctmedtraj

# **Description**

Plot Method for an Object of Class ctmedtraj

#### Usage

```
## S3 method for class 'ctmedtraj'
plot(x, legend_pos = "topright", total = TRUE, ...)
```

PosteriorBeta 49

# **Arguments**

Χ	Object of class ctmedtraj.
legend_pos	Character vector. Optional argument. Legend position.
total	Logical. If total = TRUE, include the total effect trajectory. If total = FALSE, exclude the total effect trajectory.
	Additional arguments.

### Value

Displays trajectory plots of the effects.

# Author(s)

Ivan Jacob Agaloos Pesigan

# **Examples**

```
phi <- matrix(</pre>
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
traj <- Trajectory(</pre>
  mu0 = c(3, 3, -3),
  time = 150,
  phi = phi,
  med = "m"
)
plot(traj)
```

PosteriorBeta

Posterior Sampling Distribution for the Elements of the Matrix of Lagged Coefficients Over a Specific Time Interval or a Range of Time Intervals

# **Description**

This function generates a posterior sampling distribution for the elements of the matrix of lagged coefficients  $\beta$  over a specific time interval  $\Delta t$  or a range of time intervals using the first-order stochastic differential equation model drift matrix  $\Phi$ .

50 PosteriorBeta

#### Usage

```
PosteriorBeta(phi, delta_t, ncores = NULL)
```

#### **Arguments**

phi Numeric matrix. The drift matrix ( $\Phi$ ), phi should have row and column names

pertaining to the variables in the system.

delta\_t Numeric. Time interval ( $\Delta t$ ).

ncores Positive integer. Number of cores to use. If ncores = NULL, use a single core.

Consider using multiple cores when number of replications R is a large value.

#### **Details**

See Total().

#### Value

Returns an object of class ctmedmc which is a list with the following elements:

call Function call.

args Function arguments.

**fun** Function used ("PosteriorBeta").

**output** A list the length of which is equal to the length of delta\_t.

Each element in the output list has the following elements:

est A vector of total, direct, and indirect effects.

thetahatstar A matrix of Monte Carlo total, direct, and indirect effects.

### Author(s)

Ivan Jacob Agaloos Pesigan

#### References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. Sociological Methodology, 17, 37. doi:10.2307/271028

Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. Structural Equation Modeling: A Multidisciplinary Journal, 23 (1), 61–75. doi:10.1080/10705511.2014.973960

Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. Psychometrika, 87 (1), 214–252. doi:10.1007/s11336021097670

#### See Also

```
Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()
```

PosteriorBeta 51

```
phi <- matrix(</pre>
  data = c(
   -0.357, 0.771, -0.450,
   0.0, -0.511, 0.729,
   0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
vcov_phi_vec <- matrix(</pre>
  data = c(
    0.002704274, -0.001475275, 0.000949122,
    -0.001619422, 0.000885122, -0.000569404,
    0.00085493, -0.000465824, 0.000297815,
    -0.001475275, 0.004428442, -0.002642303,
    0.000980573, -0.00271817, 0.001618805,
    -0.000586921, 0.001478421, -0.000871547,
    0.000949122, -0.002642303, 0.006402668,
    -0.000697798, 0.001813471, -0.004043138,
    0.000463086, -0.001120949, 0.002271711,
    -0.001619422, 0.000980573, -0.000697798,
    0.002079286, -0.001152501, 0.000753,
    -0.001528701, 0.000820587, -0.000517524,
    0.000885122, -0.00271817, 0.001813471,
    -0.001152501, 0.00342605, -0.002075005,
    0.000899165, -0.002532849, 0.001475579,
    -0.000569404, 0.001618805, -0.004043138,
    0.000753, -0.002075005, 0.004984032,
    -0.000622255, 0.001634917, -0.003705661,
    0.00085493, -0.000586921, 0.000463086,
    -0.001528701, 0.000899165, -0.000622255,
    0.002060076, -0.001096684, 0.000686386,
    -0.000465824, 0.001478421, -0.001120949,
    0.000820587, -0.002532849, 0.001634917,
    -0.001096684, 0.003328692, -0.001926088,
    0.000297815, -0.000871547, 0.002271711,
    -0.000517524, 0.001475579, -0.003705661,
     0.000686386, \ -0.001926088, \ 0.004726235 
  ),
  nrow = 9
)
phi <- MCPhi(</pre>
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  R = 1000L
)$output
# Specific time interval -------
PosteriorBeta(
  phi = phi,
```

52 PosteriorIndirectCentral

### PosteriorIndirectCentral

Posterior Distribution of the Indirect Effect Centrality Over a Specific Time Interval or a Range of Time Intervals

# **Description**

This function generates a posterior distribution of the indirect effect centrality over a specific time interval  $\Delta t$  or a range of time intervals using the posterior distribution of the first-order stochastic differential equation model drift matrix  $\Phi$ .

# Usage

```
PosteriorIndirectCentral(phi, delta_t, ncores = NULL)
```

# **Arguments**

phi	List of numeric matrices. Each element of the list is a sample from the posterior distribution of the drift matrix ( $\Phi$ ). Each matrix should have row and column names pertaining to the variables in the system.
delta_t	Numeric. Time interval $(\Delta t)$ .
ncores	Positive integer. Number of cores to use. If ncores = NULL, use a single core. Consider using multiple cores when number of replications R is a large value.

#### **Details**

See TotalCentral() for more details.

PosteriorIndirectCentral 53

#### Value

Returns an object of class ctmedmc which is a list with the following elements:

call Function call.

args Function arguments.

fun Function used ("PosteriorIndirectCentral").

**output** A list the length of which is equal to the length of delta\_t.

Each element in the output list has the following elements:

est Mean of the posterior distribution of the total, direct, and indirect effects.

thetahatstar Posterior distribution of the total, direct, and indirect effects.

### Author(s)

Ivan Jacob Agaloos Pesigan

#### References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. Sociological Methodology, 17, 37. doi:10.2307/271028

Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. Structural Equation Modeling: A Multidisciplinary Journal, 23 (1), 61–75. doi:10.1080/10705511.2014.973960

Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. Psychometrika, 87 (1), 214–252. doi:10.1007/s11336021097670

#### See Also

Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorBeta(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()

```
phi <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")
vcov_phi_vec <- matrix(
  data = c(
    0.002704274, -0.001475275, 0.000949122,
    -0.001619422, 0.000885122, -0.000569404,</pre>
```

54 PosteriorIndirectCentral

```
0.00085493, -0.000465824, 0.000297815,
   -0.001475275, 0.004428442, -0.002642303,
   0.000980573, -0.00271817, 0.001618805,
   -0.000586921, 0.001478421, -0.000871547,
    0.000949122, \ -0.002642303, \ 0.006402668, \\
   -0.000697798, 0.001813471, -0.004043138,
   0.000463086, -0.001120949, 0.002271711,
   -0.001619422, 0.000980573, -0.000697798,
   0.002079286, -0.001152501, 0.000753,
   -0.001528701, 0.000820587, -0.000517524,
   0.000885122, -0.00271817, 0.001813471,
   -0.001152501, 0.00342605, -0.002075005,
   0.000899165, -0.002532849, 0.001475579,
   -0.000569404, 0.001618805, -0.004043138,
   0.000753, -0.002075005, 0.004984032,
   -0.000622255, 0.001634917, -0.003705661,
   0.00085493, -0.000586921, 0.000463086,
   -0.001528701, 0.000899165, -0.000622255,
   0.002060076, -0.001096684, 0.000686386,
   -0.000465824, 0.001478421, -0.001120949,
   0.000820587, -0.002532849, 0.001634917,
   -0.001096684, 0.003328692, -0.001926088,
   0.000297815, -0.000871547, 0.002271711,
   -0.000517524, 0.001475579, -0.003705661,
   0.000686386, -0.001926088, 0.004726235
 ),
 nrow = 9
)
phi <- MCPhi(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 R = 1000L
)$output
# Specific time interval -------
PosteriorIndirectCentral(
 phi = phi,
 delta_t = 1
# Range of time intervals -----
posterior <- PosteriorIndirectCentral(</pre>
 phi = phi,
 delta_t = 1:5
)
# Methods -----
# PosteriorIndirectCentral has a number of methods including
# print, summary, confint, and plot
print(posterior)
summary(posterior)
confint(posterior, level = 0.95)
```

PosteriorMed 55

plot(posterior)

PosteriorMed	Posterior Distribution of Total, Direct, and Indirect Effects of X on Y Through M Over a Specific Time Interval or a Range of Time Intervals
	Through M Over a Specific Time Interval or a Range of Time Intervals

## **Description**

This function generates a posterior distribution of the total, direct and indirect effects of the independent variable X on the dependent variable Y through mediator variables  $\mathbf{m}$  over a specific time interval  $\Delta t$  or a range of time intervals using the posterior distribution of the first-order stochastic differential equation model drift matrix  $\Phi$ .

# Usage

```
PosteriorMed(phi, delta_t, from, to, med, ncores = NULL)
```

### **Arguments**

phi	List of numeric matrices. Each element of the list is a sample from the posterior distribution of the drift matrix ( $\Phi$ ). Each matrix should have row and column names pertaining to the variables in the system.
delta_t	Numeric. Time interval $(\Delta t)$ .
from	Character string. Name of the independent variable $X$ in phi.
to	Character string. Name of the dependent variable $Y$ in phi.
med	Character vector. Name/s of the mediator variable/s in phi.
ncores	Positive integer. Number of cores to use. If ncores = NULL, use a single core. Consider using multiple cores when number of replications R is a large value.

## **Details**

See Total(), Direct(), and Indirect() for more details.

### Value

Returns an object of class ctmedmc which is a list with the following elements:

call Function call.

args Function arguments.

fun Function used ("PosteriorMed").

output A list the length of which is equal to the length of delta\_t.

Each element in the output list has the following elements:

est Mean of the posterior distribution of the total, direct, and indirect effects.

thetahatstar Posterior distribution of the total, direct, and indirect effects.

56 PosteriorMed

#### Author(s)

Ivan Jacob Agaloos Pesigan

#### References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. Sociological Methodology, 17, 37. doi:10.2307/271028

Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. Structural Equation Modeling: A Multidisciplinary Journal, 23 (1), 61–75. doi:10.1080/10705511.2014.973960

Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. Psychometrika, 87 (1), 214–252. doi:10.1007/s11336021097670

#### See Also

Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()

```
phi <- matrix(</pre>
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
vcov_phi_vec <- matrix(</pre>
  data = c(
    0.002704274, -0.001475275, 0.000949122,
    -0.001619422, 0.000885122, -0.000569404,
    0.00085493, -0.000465824, 0.000297815,
    -0.001475275, 0.004428442, -0.002642303,
    0.000980573, -0.00271817, 0.001618805,
    -0.000586921, 0.001478421, -0.000871547,
    0.000949122, -0.002642303, 0.006402668,
    -0.000697798, 0.001813471, -0.004043138,
    0.000463086, -0.001120949, 0.002271711,
    -0.001619422, 0.000980573, -0.000697798,
    0.002079286, -0.001152501, 0.000753,
    -0.001528701, 0.000820587, -0.000517524,
    0.000885122, -0.00271817, 0.001813471,
    -0.001152501, 0.00342605, -0.002075005,
    0.000899165, -0.002532849, 0.001475579,
    -0.000569404, 0.001618805, -0.004043138,
    0.000753, -0.002075005, 0.004984032,
```

PosteriorPhi 57

```
-0.000622255, 0.001634917, -0.003705661,
   0.00085493, -0.000586921, 0.000463086,
   -0.001528701, 0.000899165, -0.000622255,
   0.002060076, -0.001096684, 0.000686386,
   -0.000465824, 0.001478421, -0.001120949,
   0.000820587, -0.002532849, 0.001634917,
   -0.001096684, 0.003328692, -0.001926088,
   0.000297815, -0.000871547, 0.002271711,
   -0.000517524, 0.001475579, -0.003705661,
   0.000686386, -0.001926088, 0.004726235
 ),
 nrow = 9
)
phi <- MCPhi(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 R = 1000L
)$output
# Specific time interval ------
PosteriorMed(
 phi = phi,
 delta_t = 1,
 from = "x",
 to = "y",
 med = "m"
)
# Range of time intervals -----
posterior <- PosteriorMed(</pre>
 phi = phi,
 delta_t = 1:5,
 from = "x",
 to = "y",
 med = "m"
# Methods -----
# PosteriorMed has a number of methods including
# print, summary, confint, and plot
print(posterior)
summary(posterior)
confint(posterior, level = 0.95)
plot(posterior)
```

58 PosteriorTotalCentral

## **Description**

The function extracts the posterior samples of the drift matrix from a fitted model from the ctsem::ctStanFit() function.

# Usage

```
PosteriorPhi(object)
```

### **Arguments**

object

Object of class ctStanFit. Output of the ctsem::ctStanFit() function.

#### Value

Returns an object of class ctmedposteriorphi which is a list drift matrices sampled from the posterior distribution.

### Author(s)

Ivan Jacob Agaloos Pesigan

#### See Also

Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()

PosteriorTotalCentral Posterior Distribution of the Total Effect Centrality Over a Specific
Time Interval or a Range of Time Intervals

# **Description**

This function generates a posterior distribution of the total effect centrality over a specific time interval  $\Delta t$  or a range of time intervals using the posterior distribution of the first-order stochastic differential equation model drift matrix  $\Phi$ .

### Usage

```
PosteriorTotalCentral(phi, delta_t, ncores = NULL)
```

PosteriorTotalCentral 59

### **Arguments**

phi List of numeric matrices. Each element of the list is a sample from the posterior

distribution of the drift matrix ( $\Phi$ ). Each matrix should have row and column

names pertaining to the variables in the system.

delta\_t Numeric. Time interval ( $\Delta t$ ).

ncores Positive integer. Number of cores to use. If ncores = NULL, use a single core.

Consider using multiple cores when number of replications R is a large value.

#### **Details**

See TotalCentral() for more details.

#### Value

Returns an object of class ctmedmc which is a list with the following elements:

call Function call.

args Function arguments.

fun Function used ("PosteriorTotalCentral").

**output** A list the length of which is equal to the length of delta\_t.

Each element in the output list has the following elements:

est Mean of the posterior distribution of the total, direct, and indirect effects.

thetahatstar Posterior distribution of the total, direct, and indirect effects.

#### Author(s)

Ivan Jacob Agaloos Pesigan

#### References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. Sociological Methodology, 17, 37. doi:10.2307/271028

Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. Structural Equation Modeling: A Multidisciplinary Journal, 23 (1), 61–75. doi:10.1080/10705511.2014.973960

Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. Psychometrika, 87 (1), 214–252. doi:10.1007/s11336021097670

## See Also

Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), Total(), TotalCentral(), Trajectory()

60 PosteriorTotalCentral

```
phi <- matrix(</pre>
  data = c(
   -0.357, 0.771, -0.450,
   0.0, -0.511, 0.729,
   0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
vcov_phi_vec <- matrix(</pre>
  data = c(
    0.002704274, -0.001475275, 0.000949122,
    -0.001619422, 0.000885122, -0.000569404,
    0.00085493, -0.000465824, 0.000297815,
    -0.001475275, 0.004428442, -0.002642303,
    0.000980573, -0.00271817, 0.001618805,
    -0.000586921, 0.001478421, -0.000871547,
    0.000949122, -0.002642303, 0.006402668,
    -0.000697798, 0.001813471, -0.004043138,
    0.000463086, -0.001120949, 0.002271711,
    -0.001619422, 0.000980573, -0.000697798,
    0.002079286, -0.001152501, 0.000753,
    -0.001528701, 0.000820587, -0.000517524,
    0.000885122, -0.00271817, 0.001813471,
    -0.001152501, 0.00342605, -0.002075005,
    0.000899165, -0.002532849, 0.001475579,
    -0.000569404, 0.001618805, -0.004043138,
    0.000753, -0.002075005, 0.004984032,
    -0.000622255, 0.001634917, -0.003705661,
    0.00085493, -0.000586921, 0.000463086,
    -0.001528701, 0.000899165, -0.000622255,
    0.002060076, -0.001096684, 0.000686386,
    -0.000465824, 0.001478421, -0.001120949,
    0.000820587, -0.002532849, 0.001634917,
    -0.001096684, 0.003328692, -0.001926088,
    0.000297815, -0.000871547, 0.002271711,
    -0.000517524, 0.001475579, -0.003705661,
   0.000686386, -0.001926088, 0.004726235
  ),
  nrow = 9
)
phi <- MCPhi(</pre>
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  R = 1000L
)$output
# Specific time interval -------
PosteriorTotalCentral(
  phi = phi,
```

print.ctmeddelta 61

print.ctmeddelta

Print Method for Object of Class ctmeddelta

# Description

Print Method for Object of Class ctmeddelta

### Usage

```
## S3 method for class 'ctmeddelta'
print(x, alpha = 0.05, digits = 4, ...)
```

# Arguments

x an object of class ctmeddelta. alpha Numeric vector. Significance level  $\alpha$ . digits Integer indicating the number of decimal places to display. . . . further arguments.

# Value

Prints a list of matrices of time intervals, estimates, standard errors, test statistics, p-values, and confidence intervals.

#### Author(s)

Ivan Jacob Agaloos Pesigan

62 print.ctmeddelta

```
phi <- matrix(</pre>
 data = c(
   -0.357, 0.771, -0.450,
   0.0, -0.511, 0.729,
   0, 0, -0.693
 nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
vcov_phi_vec <- matrix(</pre>
 data = c(
   0.002704274, -0.001475275, 0.000949122,
    -0.001619422, 0.000885122, -0.000569404,
   0.00085493, -0.000465824, 0.000297815,
    -0.001475275, 0.004428442, -0.002642303,
   0.000980573, -0.00271817, 0.001618805,
    -0.000586921, 0.001478421, -0.000871547,
    0.000949122, -0.002642303, 0.006402668,
    -0.000697798, 0.001813471, -0.004043138,
    0.000463086, -0.001120949, 0.002271711,
    -0.001619422, 0.000980573, -0.000697798,
   0.002079286, -0.001152501, 0.000753,
   -0.001528701, 0.000820587, -0.000517524,
   0.000885122, -0.00271817, 0.001813471,
    -0.001152501, 0.00342605, -0.002075005,
   0.000899165, -0.002532849, 0.001475579,
    -0.000569404, 0.001618805, -0.004043138,
   0.000753, -0.002075005, 0.004984032,
    -0.000622255, 0.001634917, -0.003705661,
   0.00085493, -0.000586921, 0.000463086,
    -0.001528701, 0.000899165, -0.000622255,
    0.002060076, -0.001096684, 0.000686386,
    -0.000465824, 0.001478421, -0.001120949,
   0.000820587, -0.002532849, 0.001634917,
    -0.001096684, 0.003328692, -0.001926088,
   0.000297815, -0.000871547, 0.002271711,
   -0.000517524, 0.001475579, -0.003705661,
   0.000686386, -0.001926088, 0.004726235
 ),
 nrow = 9
)
# Specific time interval -------
delta <- DeltaMed(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1,
 from = "x",
 to = "y",
 med = "m"
)
```

print.ctmedeffect 63

```
print(delta)

# Range of time intervals ------
delta <- DeltaMed(
    phi = phi,
    vcov_phi_vec = vcov_phi_vec,
    delta_t = 1:5,
    from = "x",
    to = "y",
    med = "m"
)
print(delta)</pre>
```

print.ctmedeffect

Print Method for Object of Class ctmedeffect

# Description

Print Method for Object of Class ctmedeffect

# Usage

```
## S3 method for class 'ctmedeffect'
print(x, digits = 4, ...)
```

# **Arguments**

x an object of class ctmedeffect.digits Integer indicating the number of decimal places to display.... further arguments.

## Value

Prints the effects.

### Author(s)

Ivan Jacob Agaloos Pesigan

```
phi <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
),
  nrow = 3</pre>
```

64 print.ctmedmc

```
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
delta_t <- 1
# Time Interval of One ------
## Total Effect ------
total_dt <- Total(</pre>
 phi = phi,
 delta_t = delta_t
print(total_dt)
direct_dt <- Direct(</pre>
 phi = phi,
 delta_t = delta_t,
 from = "x",
 to = "y",
 med = "m"
)
print(direct_dt)
## Indirect Effect ------
indirect_dt <- Indirect(</pre>
 phi = phi,
 delta_t = delta_t,
 from = "x",
 to = "y",
 med = "m"
)
print(indirect_dt)
```

print.ctmedmc

Print Method for Object of Class ctmedmc

## **Description**

Print Method for Object of Class ctmedmc

### Usage

```
## S3 method for class 'ctmedmc'
print(x, alpha = 0.05, digits = 4, ...)
```

# Arguments

x an object of class ctmedmc.

alpha Numeric vector. Significance level  $\alpha$ .

print.ctmedmc 65

```
digits Integer indicating the number of decimal places to display. . . . further arguments.
```

### Value

Prints a list of matrices of time intervals, estimates, standard errors, number of Monte Carlo replications, and confidence intervals.

#### Author(s)

Ivan Jacob Agaloos Pesigan

```
set.seed(42)
phi <- matrix(</pre>
  data = c(
   -0.357, 0.771, -0.450,
   0.0, -0.511, 0.729,
   0, 0, -0.693
  ),
  nrow = 3
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
vcov_phi_vec <- matrix(</pre>
  data = c(
    0.002704274, -0.001475275, 0.000949122,
    -0.001619422, 0.000885122, -0.000569404,
   0.00085493, -0.000465824, 0.000297815,
    -0.001475275, 0.004428442, -0.002642303,
    0.000980573, -0.00271817, 0.001618805,
    -0.000586921, 0.001478421, -0.000871547,
    0.000949122, -0.002642303, 0.006402668,
    -0.000697798, 0.001813471, -0.004043138,
    0.000463086, -0.001120949, 0.002271711,
    -0.001619422, 0.000980573, -0.000697798,
    0.002079286, -0.001152501, 0.000753,
    -0.001528701, 0.000820587, -0.000517524,
    0.000885122, -0.00271817, 0.001813471,
    -0.001152501, 0.00342605, -0.002075005,
    0.000899165, -0.002532849, 0.001475579,
    -0.000569404, 0.001618805, -0.004043138,
    0.000753, -0.002075005, 0.004984032,
    -0.000622255, 0.001634917, -0.003705661,
    0.00085493, -0.000586921, 0.000463086,
    -0.001528701, 0.000899165, -0.000622255,
    0.002060076, -0.001096684, 0.000686386,
    -0.000465824, 0.001478421, -0.001120949,
    0.000820587, -0.002532849, 0.001634917,
    -0.001096684, 0.003328692, -0.001926088,
    0.000297815, -0.000871547, 0.002271711,
    -0.000517524, 0.001475579, -0.003705661,
```

print.ctmedmcphi

```
0.000686386, -0.001926088, 0.004726235
 ),
 nrow = 9
)
# Specific time interval ------
mc <- MCMed(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1,
 from = "x",
 to = "y",
 med = "m"
 R = 100L # use a large value for R in actual research
print(mc)
# Range of time intervals ------
mc <- MCMed(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1:5,
 from = "x",
 to = "y",
 med = "m"
 R = 100L # use a large value for R in actual research
print(mc)
```

print.ctmedmcphi

Print Method for Object of Class ctmedmcphi

# **Description**

Print Method for Object of Class ctmedmcphi

## Usage

```
## S3 method for class 'ctmedmcphi'
print(x, digits = 4, ...)
```

# Arguments

```
x an object of class ctmedmcphi.digits Integer indicating the number of decimal places to display.further arguments.
```

print.ctmedmed 67

# Value

Prints a list of drift matrices.

#### Author(s)

Ivan Jacob Agaloos Pesigan

# **Examples**

```
set.seed(42)
phi <- matrix(</pre>
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) \leftarrow rownames(phi) \leftarrow c("x", "m", "y")
mc <- MCPhi(</pre>
  phi = phi,
  vcov_phi_vec = 0.1 * diag(9),
  R = 100L # use a large value for R in actual research
)
print(mc)
```

print.ctmedmed

Print Method for Object of Class ctmedmed

# Description

Print Method for Object of Class ctmedmed

### Usage

```
## S3 method for class 'ctmedmed'
print(x, digits = 4, ...)
```

## **Arguments**

```
x an object of class ctmedmed.digits Integer indicating the number of decimal places to display.... further arguments.
```

# Value

Prints a matrix of effects.

68 print.ctmedtraj

### Author(s)

Ivan Jacob Agaloos Pesigan

# **Examples**

```
phi <- matrix(</pre>
 data = c(
   -0.357, 0.771, -0.450,
   0.0, -0.511, 0.729,
   0, 0, -0.693
 ),
 nrow = 3
)
colnames(phi) \leftarrow rownames(phi) \leftarrow c("x", "m", "y")
# Specific time interval ------
med <- Med(</pre>
 phi = phi,
 delta_t = 1,
 from = "x",
 to = "y",
 med = "m"
)
print(med)
# Range of time intervals ------
med <- Med(
 phi = phi,
 delta_t = 1:5,
 from = "x",
 to = "y",
 med = "m"
)
print(med)
```

print.ctmedtraj

Print Method for Object of Class ctmedtraj

# Description

Print Method for Object of Class ctmedtraj

# Usage

```
## S3 method for class 'ctmedtraj'
print(x, ...)
```

summary.ctmeddelta 69

# **Arguments**

```
x an object of class ctmedtraj.... further arguments.
```

### Value

Prints a data frame of simulated data.

# Author(s)

Ivan Jacob Agaloos Pesigan

# **Examples**

```
phi <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")

traj <- Trajectory(
  mu0 = c(3, 3, -3),
    time = 150,
    phi = phi,
    med = "m"
)

print(traj)</pre>
```

summary.ctmeddelta

Summary Method for an Object of Class ctmeddelta

# Description

Summary Method for an Object of Class ctmeddelta

# Usage

```
## S3 method for class 'ctmeddelta'
summary(object, alpha = 0.05, ...)
```

70 summary.ctmeddelta

## **Arguments**

```
object Object of class ctmeddelta. 
alpha Numeric vector. Significance level \alpha. 
... additional arguments.
```

#### Value

Returns a data frame of effects, time intervals, estimates, standard errors, test statistics, p-values, and confidence intervals.

### Author(s)

Ivan Jacob Agaloos Pesigan

```
phi <- matrix(</pre>
 data = c(
   -0.357, 0.771, -0.450,
   0.0, -0.511, 0.729,
   0, 0, -0.693
 ),
 nrow = 3
colnames(phi) \leftarrow rownames(phi) \leftarrow c("x", "m", "y")
vcov_phi_vec <- matrix(</pre>
 data = c(
    0.002704274, -0.001475275, 0.000949122,
    -0.001619422, 0.000885122, -0.000569404,
   0.00085493, -0.000465824, 0.000297815,
    -0.001475275, 0.004428442, -0.002642303,
    0.000980573, -0.00271817, 0.001618805,
    -0.000586921, 0.001478421, -0.000871547,
    0.000949122, -0.002642303, 0.006402668,
    -0.000697798, 0.001813471, -0.004043138,
    0.000463086, -0.001120949, 0.002271711,
    -0.001619422, 0.000980573, -0.000697798,
   0.002079286, -0.001152501, 0.000753,
    -0.001528701, 0.000820587, -0.000517524,
    0.000885122, -0.00271817, 0.001813471,
    -0.001152501, 0.00342605, -0.002075005,
    0.000899165, -0.002532849, 0.001475579,
    -0.000569404, 0.001618805, -0.004043138,
    0.000753, -0.002075005, 0.004984032,
    -0.000622255, 0.001634917, -0.003705661,
    0.00085493, -0.000586921, 0.000463086,
    -0.001528701, 0.000899165, -0.000622255,
    0.002060076, -0.001096684, 0.000686386,
    -0.000465824, 0.001478421, -0.001120949,
    0.000820587, -0.002532849, 0.001634917,
    -0.001096684, 0.003328692, -0.001926088,
```

summary.ctmedmc 71

```
0.000297815, -0.000871547, 0.002271711,
   -0.000517524, 0.001475579, -0.003705661,
   0.000686386, -0.001926088, 0.004726235
 ),
 nrow = 9
)
# Specific time interval ------
delta <- DeltaMed(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1,
 from = "x",
 to = "y",
 med = "m"
summary(delta)
# Range of time intervals ------
delta <- DeltaMed(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1:5,
 from = "x",
 to = "y",
 med = "m"
summary(delta)
```

summary.ctmedmc

Summary Method for an Object of Class ctmedmc

# **Description**

Summary Method for an Object of Class ctmedmc

## Usage

```
## S3 method for class 'ctmedmc'
summary(object, alpha = 0.05, ...)
```

# Arguments

```
object Object of class ctmedmc.  
alpha Numeric vector. Significance level \alpha.  
additional arguments.
```

72 summary.ctmedmc

### Value

Returns a data frame of effects, time intervals, estimates, standard errors, number of Monte Carlo replications, and confidence intervals.

#### Author(s)

Ivan Jacob Agaloos Pesigan

```
set.seed(42)
phi <- matrix(</pre>
  data = c(
   -0.357, 0.771, -0.450,
   0.0, -0.511, 0.729,
   0, 0, -0.693
  ),
  nrow = 3
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
vcov_phi_vec <- matrix(</pre>
  data = c(
    0.002704274, -0.001475275, 0.000949122,
    -0.001619422, 0.000885122, -0.000569404,
   0.00085493, -0.000465824, 0.000297815,
    -0.001475275, 0.004428442, -0.002642303,
    0.000980573, -0.00271817, 0.001618805,
    -0.000586921, 0.001478421, -0.000871547,
   0.000949122, -0.002642303, 0.006402668,
    -0.000697798, 0.001813471, -0.004043138,
    0.000463086, -0.001120949, 0.002271711,
    -0.001619422, 0.000980573, -0.000697798,
    0.002079286, -0.001152501, 0.000753,
    -0.001528701, 0.000820587, -0.000517524,
    0.000885122, -0.00271817, 0.001813471,
    -0.001152501, 0.00342605, -0.002075005,
    0.000899165, -0.002532849, 0.001475579,
    -0.000569404, 0.001618805, -0.004043138,
    0.000753, -0.002075005, 0.004984032,
    -0.000622255, 0.001634917, -0.003705661,
    0.00085493, -0.000586921, 0.000463086,
    -0.001528701, 0.000899165, -0.000622255,
    0.002060076, -0.001096684, 0.000686386,
    -0.000465824, 0.001478421, -0.001120949,
    0.000820587, -0.002532849, 0.001634917,
    -0.001096684, 0.003328692, -0.001926088,
    0.000297815, -0.000871547, 0.002271711,
    -0.000517524, 0.001475579, -0.003705661,
    0.000686386, -0.001926088, 0.004726235
  ),
  nrow = 9
)
```

summary.ctmedmed 73

```
# Specific time interval ------
mc <- MCMed(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1,
 from = "x",
 to = "y",
 med = "m",
 R = 100L \text{ \# use a large value for } R \text{ in actual research}
)
summary(mc)
# Range of time intervals ------
mc <- MCMed(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1:5,
 from = "x",
 to = "y",
 med = "m",
 R = 100L # use a large value for R in actual research
)
summary(mc)
```

summary.ctmedmed

Summary Method for an Object of Class ctmedmed

# Description

Summary Method for an Object of Class ctmedmed

# Usage

```
## S3 method for class 'ctmedmed'
summary(object, digits = 4, ...)
```

# **Arguments**

object an object of class ctmedmed.

digits Integer indicating the number of decimal places to display.

... further arguments.

#### Value

Returns a matrix of effects.

## Author(s)

Ivan Jacob Agaloos Pesigan

## **Examples**

```
phi <- matrix(</pre>
 data = c(
   -0.357, 0.771, -0.450,
   0.0, -0.511, 0.729,
   0, 0, -0.693
 ),
 nrow = 3
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
# Specific time interval ------
med <- Med(
 phi = phi,
 delta_t = 1,
 from = "x",
 to = "y",
 med = "m"
summary(med)
# Range of time intervals ------
med <- Med(
 phi = phi,
 delta_t = 1:5,
 from = "x",
 to = "y",
 med = "m"
summary(med)
```

summary.ctmedposteriorphi

Summary Method for Object of Class ctmedposteriorphi

# **Description**

Summary Method for Object of Class ctmedposteriorphi

## Usage

```
## S3 method for class 'ctmedposteriorphi'
summary(object, ...)
```

summary.ctmedtraj 75

# **Arguments**

```
object an object of class ctmedposteriorphi.
... further arguments.
```

## Value

Returns a list of the posterior means (in matrix form) and covariance matrix.

## Author(s)

Ivan Jacob Agaloos Pesigan

summary.ctmedtraj

Summary Method for an Object of Class ctmedtraj

# Description

Summary Method for an Object of Class ctmedtraj

# Usage

```
## S3 method for class 'ctmedtraj'
summary(object, ...)
```

# **Arguments**

```
object an object of class ctmedtraj. ... further arguments.
```

#### Value

Returns a data frame of simulated data.

# Author(s)

Ivan Jacob Agaloos Pesigan

```
phi <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
```

76 Total

```
traj <- Trajectory(</pre>
  mu0 = c(3, 3, -3),
  time = 150,
  phi = phi,
  med = "m"
)
summary(traj)
```

Total

Total Effect Matrix Over a Specific Time Interval

# Description

This function computes the total effects matrix over a specific time interval  $\Delta t$  using the first-order stochastic differential equation model's drift matrix  $\Phi$ .

## Usage

```
Total(phi, delta_t)
```

#### **Arguments**

phi Numeric matrix. The drift matrix ( $\Phi$ ). phi should have row and column names

pertaining to the variables in the system.

Numeric. Time interval ( $\Delta t$ ). delta\_t

# **Details**

The total effect matrix over a specific time interval  $\Delta t$  is given by

$$Total_{\Delta t} = \exp\left(\Delta t \mathbf{\Phi}\right)$$

where  $\Phi$  denotes the drift matrix, and  $\Delta t$  the time interval.

## **Linear Stochastic Differential Equation Model:**

The measurement model is given by

$$\mathbf{y}_{i,t} = \boldsymbol{\nu} + \boldsymbol{\Lambda} \boldsymbol{\eta}_{i,t} + \boldsymbol{\varepsilon}_{i,t}, \quad ext{with} \quad \boldsymbol{\varepsilon}_{i,t} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Theta}\right)$$

where  $\mathbf{y}_{i,t}$ ,  $\boldsymbol{\eta}_{i,t}$ , and  $\boldsymbol{\varepsilon}_{i,t}$  are random variables and  $\boldsymbol{\nu}$ ,  $\boldsymbol{\Lambda}$ , and  $\boldsymbol{\Theta}$  are model parameters.  $\mathbf{y}_{i,t}$ represents a vector of observed random variables,  $\eta_{i,t}$  a vector of latent random variables, and  $\boldsymbol{\varepsilon}_{i,t}$  a vector of random measurement errors, at time t and individual i.  $\boldsymbol{\nu}$  denotes a vector of intercepts,  $\Lambda$  a matrix of factor loadings, and  $\Theta$  the covariance matrix of  $\varepsilon$ .

An alternative representation of the measurement error is given by

$$\boldsymbol{\varepsilon}_{i,t} = \boldsymbol{\Theta}^{\frac{1}{2}} \mathbf{z}_{i,t}, \quad \text{with} \quad \mathbf{z}_{i,t} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)$$

Total 77

where  $\mathbf{z}_{i,t}$  is a vector of independent standard normal random variables and  $\left(\Theta^{\frac{1}{2}}\right)\left(\Theta^{\frac{1}{2}}\right)' = \Theta$ . The dynamic structure is given by

$$\mathrm{d} oldsymbol{\eta}_{i,t} = \left( oldsymbol{\iota} + oldsymbol{\Phi} oldsymbol{\eta}_{i,t} 
ight) \mathrm{d} t + oldsymbol{\Sigma}^{rac{1}{2}} \mathrm{d} \mathbf{W}_{i,t}$$

where  $\iota$  is a term which is unobserved and constant over time,  $\Phi$  is the drift matrix which represents the rate of change of the solution in the absence of any random fluctuations,  $\Sigma$  is the matrix of volatility or randomness in the process, and  $\mathrm{d}W$  is a Wiener process or Brownian motion, which represents random fluctuations.

#### Value

Returns an object of class ctmedeffect which is a list with the following elements:

call Function call.

args Function arguments.

fun Function used ("Total").

**output** The matrix of total effects.

#### Author(s)

Ivan Jacob Agaloos Pesigan

#### References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. Sociological Methodology, 17, 37. doi:10.2307/271028

Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. Structural Equation Modeling: A Multidisciplinary Journal, 23 (1), 61–75. doi:10.1080/10705511.2014.973960

Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. Psychometrika, 87 (1), 214–252. doi:10.1007/s11336021097670

## See Also

Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), TotalCentral(), Trajectory()

```
phi <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
),
  nrow = 3</pre>
```

78 TotalCentral

```
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
delta_t <- 1
Total(
  phi = phi,
  delta_t = delta_t
phi <- matrix(</pre>
  data = c(
    -6, 5.5, 0, 0,
    1.25, -2.5, 5.9, -7.3,
    0, 0, -6, 2.5,
    5, 0, 0, -6
  ),
 nrow = 4
)
colnames(phi) <- rownames(phi) <- paste0("y", 1:4)</pre>
Total(
  phi = phi,
  delta_t = delta_t
```

TotalCentral

Total Effect Centrality

# **Description**

**Total Effect Centrality** 

# Usage

```
TotalCentral(phi, delta_t)
```

# Arguments

Numeric matrix. The drift matrix ( $\Phi$ ). phi should have row and column names pertaining to the variables in the system.

delta\_t Vector of positive numbers. Time interval ( $\Delta t$ ).

# **Details**

The total effect centrality of a variable is the sum of the total effects of a variable on all other variables at a particular time interval.

TotalCentral 79

#### Value

Returns an object of class ctmedmed which is a list with the following elements:

```
call Function call.args Function arguments.fun Function used ("TotalCentral").output A matrix of total effect centrality.
```

## Author(s)

Ivan Jacob Agaloos Pesigan

#### References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. Sociological Methodology, 17, 37. doi:10.2307/271028

Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. Structural Equation Modeling: A Multidisciplinary Journal, 23 (1), 61–75. doi:10.1080/10705511.2014.973960

Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. Psychometrika, 87 (1), 214–252. doi:10.1007/s11336021097670

## See Also

Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), Trajectory()

80 Trajectory

```
phi = phi,
  delta_t = 1:30
)
plot(total_central)

# Methods -------
# TotalCentral has a number of methods including
# print, summary, and plot
total_central <- TotalCentral(
  phi = phi,
  delta_t = 1:5
)
print(total_central)
summary(total_central)
plot(total_central)</pre>
```

Trajectory

Simulate Trajectories of Variables

## **Description**

This function simulates trajectories of variables without measurement error or process noise. Total corresponds to the total effect and Direct corresponds to the portion of the total effect where the indirect effect is removed.

# Usage

```
Trajectory(mu0, time, phi, med)
```

## **Arguments**

mu0 Numeric vector. Initial values of the variables.

time Positive integer. Number of time points.

phi Numeric matrix. The drift matrix ( $\Phi$ ), phi should have row and column names

pertaining to the variables in the system.

med Character vector. Name/s of the mediator variable/s in phi.

#### Value

Returns an object of class ctmedtraj which is a list with the following elements:

```
call Function call.
```

args Function arguments.

fun Function used ("Trajectory").

output A data frame of simulated data.

Trajectory 81

# See Also

Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral()

```
phi <- matrix(</pre>
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
traj <- Trajectory(</pre>
  mu0 = c(3, 3, -3),
  time = 150,
  phi = phi,
  med = "m"
plot(traj)
# Trajectory has a number of methods including
# print, summary, and plot
traj <- Trajectory(</pre>
  mu0 = c(3, 3, -3),
  time = 25,
  phi = phi,
  med = "m"
print(traj)
summary(traj)
plot(traj)
```

# **Index**

Continuous Time Mediation Functions	Med, 41
DeltaBeta, 6	PosteriorBeta,49
DeltaIndirectCentral, 9	PosteriorIndirectCentral, 52
DeltaMed, 13	PosteriorMed, 55
DeltaTotalCentral, 16	PosteriorPhi, 57
Direct, 19	PosteriorTotalCentral, 58
Indirect, 22	Total, 76
<pre>IndirectCentral, 24</pre>	TotalCentral, 78
MCBeta, 26	Trajectory, 80
MCIndirectCentral, 29	* delta
MCMed, 33	DeltaBeta, 6
MCPhi, 36	DeltaIndirectCentral, 9
MCTotalCentral, 38	DeltaMed, 13
Med, 41	DeltaTotalCentral, 16
PosteriorBeta, 49	* effects
PosteriorIndirectCentral, 52	Direct, 19
PosteriorMed, 55	Indirect, 22
PosteriorPhi, 57	<pre>IndirectCentral, 24</pre>
PosteriorTotalCentral, 58	Med, 41
Total, 76	Total, 76
TotalCentral, 78	TotalCentral, 78
Trajectory, $80$	Trajectory, 80
beta	* mc
DeltaBeta, 6	MCBeta, 26
MCBeta, 26	MCIndirectCentral, 29
PosteriorBeta, 49	MCMed, 33
cTMed	MCPhi, 36
DeltaBeta, 6	MCTotalCentral, 38
DeltaIndirectCentral, 9	* methods
DeltaMed, 13	<pre>confint.ctmeddelta, 2</pre>
DeltaTotalCentral, 16	confint.ctmedmc,4
Direct, 19	plot.ctmeddelta,44
Indirect, 22	plot.ctmedmc, 46
<pre>IndirectCentral, 24</pre>	plot.ctmedmed, 47
MCBeta, 26	plot.ctmedtraj,48
MCIndirectCentral, 29	print.ctmeddelta,61
MCMed, 33	print.ctmedeffect, 63
MCPhi, 36	print.ctmedmc, 64
MCTotalCentral, 38	print.ctmedmcphi, 66

INDEX 83

print.ctmedmed,67	Indirect, 8, 11, 14, 18, 21, 22, 25, 28, 31, 35,
print.ctmedtraj,68	37, 40, 43, 50, 53, 56, 58, 59, 77, 79
summary.ctmeddelta,69	81
summary.ctmedmc,71	Indirect(), <i>13</i> , <i>34</i> , <i>42</i> , <i>55</i>
summary.ctmedmed, 73	IndirectCentral, 8, 11, 14, 18, 21, 23, 24,
summary.ctmedposteriorphi, 74	28, 31, 35, 37, 40, 43, 50, 53, 56, 58,
summary.ctmedtraj, 75	59, 77, 79, 81
* network	IndirectCentral(), $10, 30$
DeltaIndirectCentral, 9	
DeltaTotalCentral, 16	MCBeta, 8, 11, 14, 18, 21, 23, 25, 26, 31, 35,
<pre>IndirectCentral, 24</pre>	37, 40, 43, 50, 53, 56, 58, 59, 77, 79
MCIndirectCentral, 29	81
MCTotalCentral, 38	MCIndirectCentral, 8, 11, 14, 18, 21, 23, 25
PosteriorIndirectCentral, 52	28, 29, 35, 37, 40, 43, 50, 53, 56, 58,
PosteriorTotalCentral, 58	59, 77, 79, 81
TotalCentral, 78	MCMed, 8, 11, 14, 18, 21, 23, 25, 28, 31, 33, 37
* path	40, 43, 50, 53, 56, 58, 59, 77, 79, 81
DeltaMed, 13	MCPhi, 8, 11, 14, 18, 21, 23, 25, 28, 31, 35, 36
MCMed, 33	40, 43, 50, 53, 56, 58, 59, 77, 79, 81
Med, 41	MCTotalCentral, 8, 11, 14, 18, 21, 23, 25, 28
PosteriorMed, 55	31, 35, 37, 38, 43, 50, 53, 56, 58, 59
Trajectory, 80	77, 79, 81
* posterior	Med, 8, 11, 14, 18, 21, 23, 25, 28, 31, 35, 37,
PosteriorBeta, 49	40, 41, 50, 53, 56, 58, 59, 77, 79, 81
PosteriorIndirectCentral, 52	
PosteriorMed, 55	plot.ctmeddelta,44
PosteriorPhi, 57	plot.ctmedmc,46
PosteriorTotalCentral, 58	plot.ctmedmed, 47
	plot.ctmedtraj,48
<pre>confint.ctmeddelta, 2</pre>	PosteriorBeta, 8, 11, 14, 18, 21, 23, 25, 28,
confint.ctmedmc, 4	31, 35, 37, 40, 43, 49, 53, 56, 58, 59
<pre>ctsem::ctStanFit(), 58</pre>	77, 79, 81
<del></del>	PosteriorIndirectCentral, 8, 11, 14, 18,
DeltaBeta, 6, 11, 14, 18, 21, 23, 25, 28, 31,	21, 23, 25, 28, 31, 35, 37, 40, 43, 50
35, 37, 40, 43, 50, 53, 56, 58, 59, 77,	52, 56, 58, 59, 77, 79, 81
79, 81	PosteriorMed, 8, 11, 14, 18, 21, 23, 25, 28,
DeltaIndirectCentral, 8, 9, 14, 18, 21, 23,	31, 35, 37, 40, 43, 50, 53, 55, 58, 59
25, 28, 31, 35, 37, 40, 43, 50, 53, 56,	77, 79, 81
58, 59, 77, 79, 81	PosteriorPhi, 8, 11, 14, 18, 21, 23, 25, 28,
DeltaMed, 8, 11, 13, 18, 21, 23, 25, 28, 31, 35,	31, 35, 37, 40, 43, 50, 53, 56, 57, 59,
37, 40, 43, 50, 53, 56, 58, 59, 77, 79,	<i>77</i> , <i>79</i> , <i>81</i>
81	PosteriorTotalCentral, 8, 11, 14, 18, 21,
DeltaTotalCentral, 8, 11, 14, 16, 21, 23, 25,	23, 25, 28, 31, 35, 37, 40, 43, 50, 53,
28, 31, 35, 37, 40, 43, 50, 53, 56, 58,	56, 58, 58, 77, 79, 81
59, 77, 79, 81	print.ctmeddelta,61
Direct, 8, 11, 14, 18, 19, 23, 25, 28, 31, 35,	print.ctmedeffect,63
37, 40, 43, 50, 53, 56, 58, 59, 77, 79,	print.ctmedmc,64
81	print.ctmedmcphi, 66
Direct(), 13, 34, 42, 55	print.ctmedmed.67

INDEX

```
print.ctmedtraj, 68
summary.ctmeddelta,69
summary.ctmedmc, 71
summary.ctmedmed, 73
summary.ctmedposteriorphi, 74
summary.ctmedtraj, 75
Total, 8, 11, 14, 18, 21, 23, 25, 28, 31, 35, 37,
         40, 43, 50, 53, 56, 58, 59, 76, 79, 81
Total(), 7, 13, 27, 34, 42, 50, 55
TotalCentral, 8, 11, 14, 18, 21, 23, 25, 28,
         31, 35, 37, 40, 43, 50, 53, 56, 58, 59,
         77, 78, 81
TotalCentral(), 17, 39, 52, 59
Trajectory, 8, 11, 14, 18, 21, 23, 25, 28, 31,
         35, 37, 40, 43, 50, 53, 56, 58, 59, 77,
         79, 80
```