# Package 'cTMed'

May 3, 2024
Title Continuous Time Mediation
Version 0.9.1
<b>Description</b> Calculates standard errors and confidence intervals for effects in continuous time mediation models.
<pre>URL https://github.com/jeksterslab/cTMed,    https://jeksterslab.github.io/cTMed/</pre>
BugReports https://github.com/jeksterslab/cTMed/issues License MIT + file LICENSE
Encoding UTF-8
<b>Roxygen</b> list(markdown = TRUE)
<b>Depends</b> R (>= $3.5.0$ )
LinkingTo Rcpp, RcppArmadillo
Imports Rcpp, numDeriv, parallel, ctsem
Suggests knitr, rmarkdown, testthat, simStateSpace, expm
RoxygenNote 7.3.1
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>
Contents
confint.ctmeddelta       2         confint.ctmedmc       3         DeltaBeta       6         DeltaIndirectCentral       1         DeltaMed       1         DeltaTotalCentral       1         Direct       2         Indirect       2

2 confint.ctmeddelta

Index		<b>79</b>
	TotalCentral	77
		75
		74
		73
	· · · · · · · · · · · · · · · · · · ·	71
		70
	· · · · · · · · · · · · · · · · · · ·	68
	print.ctmedmed	67
	print.ctmedmcphi	66
	print.ctmedmc	64
	print.ctmedeffect	62
	print.ctmeddelta	60
	PosteriorTotalCentral	58
	PosteriorPhi	57
	PosteriorMed	54
	PosteriorIndirectCentral	52
		51
	plot.ctmedmc	49
	plot.ctmeddelta	48
		45
		41
		39
		35
		31
	MCBeta	
	IndirectCentral	26

 ${\tt confint.ctmeddelta} \qquad \textit{Delta Method Confidence Intervals}$ 

# Description

Delta Method Confidence Intervals

# Usage

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

# 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.
	additional arguments.

confint.ctmeddelta 3

#### Value

Returns a matrix of 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>
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 ------

4 confint.ctmedmc

```
delta <- DeltaMed(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1,
 from = "x",
 to = "y",
 med = "m"
confint(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"
)
confint(delta)
```

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, ...)
```

# 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 matrix of confidence intervals.

confint.ctmedmc 5

#### 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
# 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)
# 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 \text{ \# use a large value for } R \text{ in actual research}
)
confint(mc)
```

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  $\boldsymbol{\beta}$  over a specific time-interval  $\Delta t$  or a range of time-intervals using the first-order stochastic differential equation model's drift matrix  $\boldsymbol{\Phi}$ .

# Usage

```
DeltaBeta(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 $\operatorname{vec}\left(\mathbf{\Phi}\right)$ .
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.

#### **Details**

See Total().

#### **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:

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

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

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

## **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,  $\boldsymbol{\eta}_{i,t}$  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,  $\boldsymbol{\Lambda}$  a matrix of factor loadings, and  $\boldsymbol{\Theta}$  the covariance matrix of  $\boldsymbol{\varepsilon}$ .

An alternative representation of the measurement error is given by

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

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 ctmeddelta which is a list with the following elements:

call Function call.

args Function arguments.

fun Function used (DeltaBeta).

**output** A list with length of length(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: DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), TestPhi(), TestStable(), Total(), TotalCentral()
```

```
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 -------
DeltaBeta(
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1
)
# Range of time-intervals -------
delta <- DeltaBeta(</pre>
 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)
```

10 DeltaIndirectCentral

plot(delta)

DeltaIndirectCentral Delta Method Sampling Variance-Covariance Matrix for the Indirect

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

### 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  $\text{vec}(\Phi)$ , 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:

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

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

DeltaIndirectCentral 11

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

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

# **Linear Stochastic Differential Equation Model:**

The measurement model is given by

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

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

An alternative representation of the measurement error is given by

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

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 ctmeddelta which is a list with the following elements:

call Function call.

args Function arguments.

fun Function used (DeltaIndirectCentral).

**output** A list with length of length(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

12 DeltaIndirectCentral

#### 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(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), TestPhi(), TestStable(), 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>
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)
plot(delta)
```

DeltaMed

Delta Method Sampling Variance-Covariance Matrix for the 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 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  $\text{vec}(\Phi)$ .

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  $\operatorname{vec}(\Phi)$ , 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:

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

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

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

## **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  $\mathbf{y}_{i,t}$ ,  $\eta_{i,t}$ , 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 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

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

where  $\mathbf{z}_{i,t}$  is a vector of independent standard normal random variables and  $\left(\Theta^{\frac{1}{2}}\right)\left(\Theta^{\frac{1}{2}}\right)' = \Theta$ . 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 ctmeddelta which is a list with the following elements:

call Function call.

args Function arguments.

fun Function used (DeltaMed).

**output** A list with length of length(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(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), TestPhi(), TestStable(), 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>
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"
)
```

DeltaTotalCentral 17

```
# 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 names pertaining to the variables in the system.
vcov_phi_vec	Numeric matrix. The sampling variance-covariance matrix of $\operatorname{vec}\left(\mathbf{\Phi}\right)$ .
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 TotalCentral() 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  $g(\hat{\theta})$  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

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

# **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,  $\boldsymbol{\eta}_{i,t}$  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,  $\boldsymbol{\Lambda}$  a matrix of factor loadings, and  $\boldsymbol{\Theta}$  the covariance matrix of  $\boldsymbol{\varepsilon}$ .

An alternative representation of the measurement error is given by

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

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

DeltaTotalCentral 19

#### 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 with length of length(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(), DeltaMed(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), TestPhi(), TestStable(), Total(), TotalCentral()

```
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>
```

20 DeltaTotalCentral

```
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 -------
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)
```

Direct 21

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)

# **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

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

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,  $\boldsymbol{\eta}_{i,t}$  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,  $\boldsymbol{\Lambda}$  a matrix of factor loadings, and  $\boldsymbol{\Theta}$  the covariance matrix of  $\boldsymbol{\varepsilon}$ .

An alternative representation of the measurement error is given by

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

22 Direct

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 ("Direct").

output The direct 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(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), TestPhi(), TestStable(), Total(), TotalCentral()

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

Indirect 23

```
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>
  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.

24 Indirect

#### **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

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

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

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

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

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

Indirect 25

#### 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(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), TestPhi(), TestStable(), 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>
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",
```

26 IndirectCentral

```
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 ( $\Delta t$ ).

## Author(s)

Ivan Jacob Agaloos Pesigan

#### See Also

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

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
)
```

# **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 runs TestPhi() on the generated drift matrix  $\Phi$ . If the TestPhi() returns FALSE, the function generates a new drift matrix  $\Phi$  and runs the test recursively until TestPhi() 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.

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

The measurement model is given by

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

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

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 ctmedmc which is a list with the following elements:

```
call Function call.
```

**args** Function arguments.

fun Function used (MCBeta).

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

## See Also

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

```
set.seed(42)
phi <- matrix(
   data = c(
     -0.357, 0.771, -0.450,
     0.0, -0.511, 0.729,
     0, 0, -0.693
),</pre>
```

```
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)
```

MCIndirectCentral 31

```
# Methods -----
# MCBeta has a number of methods including
# print, summary, confint, and plot
print(mc)
summary(mc)
confint(mc, level = 0.95)
```

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

```
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 $\operatorname{vec}\left(\Phi\right)$ .
delta_t	Numeric. Time interval ( $\Delta t$ ).
R	Positive integer. Number of replications.
test_phi	Logical. If test_phi = TRUE, the function runs TestPhi() on the generated drift matrix $\Phi$ . If the TestPhi() returns FALSE, the function generates a new drift matrix $\Phi$ and runs the test recursively until TestPhi() 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.

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

An alternative representation of the measurement error is given by

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

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 ctmedmc which is a list with the following elements:

MCIndirectCentral 33

```
call Function call.
```

args Function arguments.

fun Function used (MCIndirectCentral).

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

#### See Also

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

```
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,
```

34 MCIndirectCentral

```
-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)
plot(mc)
```

MCMed 35

MCMed	Monte Carlo Sampling Distribution of Total, Direct, and Indirect Effects of X on Y Through M Over a Specific Time-Interval or a Range of Time-Intervals
	of time time vais

# **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 $\operatorname{vec}\left(\mathbf{\Phi}\right)$ .
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 runs TestPhi() on the generated drift matrix $\Phi$ . If the TestPhi() returns FALSE, the function generates a new drift matrix $\Phi$ and runs the test recursively until TestPhi() 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**

36

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.

**MCMed** 

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

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.

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

An alternative representation of the measurement error is given by

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

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 ctmedmc which is a list with the following elements:

MCMed 37

```
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

#### See Also

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

```
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,
```

38 MCMed

```
-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 -------
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)
```

MCPhi 39

```
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.
test_phi	Logical. If test_phi = TRUE, the function runs TestPhi() on the generated drift matrix $\Phi$ . If the TestPhi() returns FALSE, the function generates a new drift matrix $\Phi$ and runs the test recursively until TestPhi() 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,

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

40 MCPhi

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

The measurement model is given by

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

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,  $\boldsymbol{\eta}_{i,t}$  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,  $\boldsymbol{\Lambda}$  a matrix of factor loadings, and  $\boldsymbol{\Theta}$  the covariance matrix of  $\boldsymbol{\varepsilon}$ .

An alternative representation of the measurement error is given by

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

where  $\mathbf{z}_{i,t}$  is a vector of independent standard normal random variables and  $\left(\boldsymbol{\Theta}^{\frac{1}{2}}\right)' = \boldsymbol{\Theta}$ . 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 a list of 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(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), TestPhi(), TestStable(), Total(), TotalCentral()

```
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")</pre>
```

MCTotalCentral 41

```
MCPhi(
  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
)
```

#### **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 runs TestPhi() on the generated

drift matrix  $\Phi$ . If the TestPhi() returns FALSE, the function generates a new

drift matrix  $\Phi$  and runs the test recursively until TestPhi() 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.

#### **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  $\mathbf{y}_{i,t}$ ,  $\eta_{i,t}$ , and  $\varepsilon_{i,t}$  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  $\varepsilon_{i,t}$  a vector of random measurement errors, at time t and individual i.  $\boldsymbol{\nu}$  denotes a vector of intercepts,  $\boldsymbol{\Lambda}$  a matrix of factor loadings, and  $\boldsymbol{\Theta}$  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  $\left(\mathbf{\Theta}^{\frac{1}{2}}\right)\left(\mathbf{\Theta}^{\frac{1}{2}}\right)'=\mathbf{\Theta}.$ 

MCTotalCentral 43

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 ctmedmc which is a list with the following elements:

call Function call.

args Function arguments.

fun Function used (MCTotalCentral).

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

#### See Also

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

44 MCTotalCentral

```
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,
    \hbox{\tt -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 -------
MCTotalCentral(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1,
  R = 100L # use a large value for R in actual research
)
```

Med 45

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  $\mathbf{\Phi}$ .

# Usage

```
Med(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	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.

#### **Details**

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

46 Med

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

The measurement model is given by

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

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.

#### 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

Med 47

#### See Also

Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), TestPhi(), TestStable(), 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>
# 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)
# 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)
```

48 plot.ctmeddelta

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**

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

#### 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,
```

plot.ctmedmc 49

```
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"
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.

50 plot.ctmedmc

## 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
# Range of time-intervals -------
mc <- MCMed(</pre>
 phi = phi,
 vcov_phi_vec = vcov_phi_vec,
```

plot.ctmedmed 51

```
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", ...)
```

# 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.

# 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")

# Range of time-intervals ------
med <- Med(
  phi = phi,
  delta_t = 1:5,</pre>
```

52 PosteriorIndirectCentral

```
from = "x",
to = "y",
med = "m"
)
plot(med)
```

#### 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.

# 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 with length of length(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.

PosteriorIndirectCentral 53

#### 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(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), TestPhi(), TestStable(), 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>
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,
```

54 PosteriorMed

```
-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)
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

#### **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  $\Phi$ .

PosteriorMed 55

# 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 with length of length(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

56 PosteriorMed

#### See Also

Other Continuous Time Mediation Functions: DeltaBeta(), DeltaIndirectCentral(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorIndirectCentral(), PosteriorPhi(), PosteriorTotalCentral(), TestPhi(), TestStable(), 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) \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,
    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,
```

PosteriorPhi 57

```
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)
```

PosteriorPhi

Extract the Posterior Samples of the Drift Matrix

# 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.

## Author(s)

Ivan Jacob Agaloos Pesigan

58 PosteriorTotalCentral

#### See Also

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

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)

#### **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 with length of length(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.

PosteriorTotalCentral 59

#### 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(), TestPhi(), TestStable(), 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>
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,
```

60 print.ctmeddelta

```
-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,
 delta_t = 1
)
# Range of time-intervals ------
posterior <- PosteriorTotalCentral(</pre>
 phi = phi,
 delta_t = 1:5
)
# Methods ------
# PosteriorTotalCentral has a number of methods including
# print, summary, confint, and plot
print(posterior)
summary(posterior)
confint(posterior, level = 0.95)
plot(posterior)
```

print.ctmeddelta

Print Method for Object of Class ctmeddelta

# **Description**

Print Method for Object of Class ctmeddelta

print.ctmeddelta 61

#### 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

Returns a matrix of time-interval, 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) <- 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,
```

62 print.ctmedeffect

```
-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(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"
print(delta)
```

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, ...)
```

print.ctmedeffect 63

# **Arguments**

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

#### Value

Returns the effects.

# 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>
delta_t <- 1
# Time-Interval of One ------
## Total Effect -----
total_dt <- Total(</pre>
 phi = phi,
 delta_t = delta_t
print(total_dt)
## Direct Effect -------
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",
```

print.ctmedmc

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

#### Value

Returns a matrix of estimates, standard errors, number of Monte Carlo replications, 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,</pre>
```

print.ctmedmc 65

```
-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 -------
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.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.
```

#### Value

Returns the structure of the output.

# 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")
mc <- MCPhi(
    phi = phi,
    vcov_phi_vec = 0.1 * diag(9),
    R = 100L # use a large value for R in actual research
)
print(mc)</pre>
```

print.ctmedmed 67

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

Returns a matrix of effects.

# 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>
# Specific time-interval ------
med <- Med(</pre>
 phi = phi,
 delta_t = 1,
 from = "x",
 to = "y",
 med = "m"
print(med)
# Range of time-intervals ------
```

68 summary.ctmeddelta

```
med <- Med(
    phi = phi,
    delta_t = 1:5,
    from = "x",
    to = "y",
    med = "m"
)
print(med)</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, ...)
```

# Arguments

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

#### Value

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

# 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")
vcov_phi_vec <- matrix(</pre>
```

summary.ctmeddelta 69

```
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"
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)
```

70 summary.ctmedmc

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.
```

#### Value

Returns a matrix of effects, time-interval, estimates, standard errors, test statistics, p-values, 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,
```

summary.ctmedmed 71

```
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 ------
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
)
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)
```

72 summary.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

```
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"
```

TestPhi 73

```
)
summary(med)
```

TestPhi

Test the Drift Matrix

#### **Description**

Both have to be true for the function to return TRUE.

- Test that the real part of all eigenvalues of  $\Phi$  is less than zero.
- Test that the diagonal values of  $\Phi$  are between 0 to negative inifinity.

# Usage

```
TestPhi(phi)
```

## **Arguments**

phi

Numeric matrix. The drift matrix  $(\Phi)$ .

# 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(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), TestStable(), Total(), TotalCentral()
```

```
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")
TestPhi(phi = phi)
phi <- matrix(
  data = c(
    -6, 5.5, 0, 0,
    1.25, -2.5, 5.9, -7.3,</pre>
```

74 TestStable

```
0, 0, -6, 2.5,
5, 0, 0, -6
),
nrow = 4
)
colnames(phi) <- rownames(phi) <- paste0("y", 1:4)
TestPhi(phi = phi)</pre>
```

TestStable

Test Stability

## **Description**

The function computes the eigenvalues of the input matrix x. It checks if the real part of all eigenvalues is negative. If all eigenvalues have negative real parts, the system is considered stable.

# Usage

TestStable(x)

#### **Arguments**

Х

Numeric matrix.

#### 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(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), TestPhi(), Total(), TotalCentral()
```

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

Total 75

```
-6, 5.5, 0, 0,

1.25, -2.5, 5.9, -7.3,

0, 0, -6, 2.5,

5, 0, 0, -6

),

nrow = 4

)

TestStable(x)
```

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.

delta\_t

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

#### **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,  $\boldsymbol{\eta}_{i,t}$  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,  $\boldsymbol{\Lambda}$  a matrix of factor loadings, and  $\boldsymbol{\Theta}$  the covariance matrix of  $\boldsymbol{\varepsilon}$ .

An alternative representation of the measurement error is given by

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

76 Total

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(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), TestPhi(), TestStable(), TotalCentral()

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

TotalCentral 77

```
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**

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 ( $\Delta t$ ).

# 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(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), TestPhi(), TestStable(), Total()
```

78 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>
# Specific time-interval ------
TotalCentral(
 phi = phi,
 delta_t = 1
)
# Range of time-intervals ------
total_central <- TotalCentral(</pre>
 phi = phi,
 delta_t = 1:30
plot(total_central)
# Methods ------
# IndirectCentral has a number of methods including
# print, summary, and plot
total_central <- TotalCentral(</pre>
 phi = phi,
 delta_t = 1:5
)
print(total_central)
summary(total_central)
plot(total_central)
```

# **Index**

* Continuous Time Mediation Functions	PosteriorIndirectCentral, 52
DeltaBeta, 6	PosteriorMed, 54
DeltaIndirectCentral, 10	PosteriorPhi, 57
DeltaMed, 13	PosteriorTotalCentral, 58
DeltaTotalCentral, 17	TestPhi, 73
Direct, 21	TestStable, 74
Indirect, 23	Total, 75
IndirectCentral, 26	TotalCentral, 77
MCBeta, 27	* effects
MCIndirectCentral, 31	Direct, 21
MCMed, 35	Indirect, 23
MCPhi, 39	Med, 45
MCTotalCentral, 41	Total, 75
Med, 45	* methods
PosteriorIndirectCentral, 52	confint.ctmeddelta, 2
PosteriorMed, 54	confint.ctmedmc, 4
PosteriorPhi, 57	plot.ctmeddelta,48
PosteriorTotalCentral, 58	plot.ctmedmc, 49
TestPhi, 73	plot.ctmedmed, 51
TestStable, 74	print.ctmeddelta,60
Total, 75	print.ctmedeffect, 62
TotalCentral, 77	print.ctmedmc,64
* beta	print.ctmedmcphi,66
DeltaBeta, 6	print.ctmedmed, 67
MCBeta, 27	summary.ctmeddelta,68
* cTMed	summary.ctmedmc,70
DeltaBeta, 6	summary.ctmedmed,72
DeltaIndirectCentral, 10	* network
DeltaMed, 13	DeltaIndirectCentral, 10
DeltaTotalCentral, 17	DeltaTotalCentral, 17
Direct, 21	<pre>IndirectCentral, 26</pre>
Indirect, 23	MCIndirectCentral, 31
<pre>IndirectCentral, 26</pre>	MCTotalCentral, 41
MCBeta, 27	PosteriorIndirectCentral, 52
MCIndirectCentral, 31	PosteriorTotalCentral, 58
MCMed, 35	TotalCentral, 77
MCPhi, 39	* path
MCTotalCentral, 41	DeltaMed, 13
Med, 45	MCMed, 35

INDEX

Med, 45	MCBeta, 8, 12, 15, 19, 22, 25, 26, 27, 33, 37,
PosteriorMed, 54	40, 43, 47, 53, 56, 58, 59, 73, 74, 76,
* test	77
TestPhi, 73	MCIndirectCentral, 8, 12, 15, 19, 22, 25, 26,
TestStable, 74	29, 31, 37, 40, 43, 47, 53, 56, 58, 59,
* uncertainty	73, 74, 76, 77
DeltaBeta, 6	MCMed, 8, 12, 15, 19, 22, 25, 26, 29, 33, 35, 40,
DeltaIndirectCentral, 10	43, 47, 53, 56, 58, 59, 73, 74, 76, 77
DeltaMed, 13	MCPhi, 8, 12, 15, 19, 22, 25, 26, 29, 33, 37, 39,
DeltaTotalCentral, 17	43, 47, 53, 56, 58, 59, 73, 74, 76, 77
MCBeta, 27	MCTotalCentral, 8, 12, 15, 19, 22, 25, 26, 29,
MCIndirectCentral, 31	33, 37, 40, 41, 47, 53, 56, 58, 59, 73,
MCMed, 35	74, 76, 77
MCPhi, 39	Med, 8, 12, 15, 19, 22, 25, 26, 29, 33, 37, 40,
MCTotalCentral, 41	43, 45, 53, 56, 58, 59, 73, 74, 76, 77
PosteriorIndirectCentral, 52	
PosteriorMed, 54	plot.ctmeddelta,48
PosteriorPhi, 57	plot.ctmedmc, 49
PosteriorTotalCentral, 58	plot.ctmedmed, 51
	PosteriorIndirectCentral, 8, 12, 15, 19,
confint.ctmeddelta, 2	22, 25, 26, 29, 33, 37, 40, 43, 47, 52,
confint.ctmedmc, 4	56, 58, 59, 73, 74, 76, 77
ctsem::ctStanFit(), 57	PosteriorMed, 8, 12, 15, 19, 22, 25, 26, 29,
Della-Dela ( 12 15 10 22 25 26 20 22	33, 37, 40, 43, 47, 53, 54, 58, 59, 73,
DeltaBeta, 6, 12, 15, 19, 22, 25, 26, 29, 33,	74, 76, 77
<i>37</i> , <i>40</i> , <i>43</i> , <i>47</i> , <i>53</i> , <i>56</i> , <i>58</i> , <i>59</i> , <i>73</i> , <i>74</i> ,	PosteriorPhi, 8, 12, 15, 19, 22, 25, 26, 29,
76, 77	33, 37, 40, 43, 47, 53, 56, 57, 59, 73,
DeltaIndirectCentral, 8, 10, 15, 19, 22, 25,	74, 76, 77
26, 29, 33, 37, 40, 43, 47, 53, 56, 58, 59, 73, 74, 76, 77	PosteriorTotalCentral, 8, 12, 15, 19, 22,
DeltaMed, 8, 12, 13, 19, 22, 25, 26, 29, 33, 37,	25, 26, 29, 33, 37, 40, 43, 47, 53, 56,
40, 43, 47, 53, 56, 58, 59, 73, 74, 76,	58, 58, 73, 74, 76, 77
77	print.ctmeddelta,60
DeltaTotalCentral, 8, 12, 15, 17, 22, 25, 26,	print.ctmedeffect, 62
29, 33, 37, 40, 43, 47, 53, 56, 58, 59,	print.ctmedmc,64
73, 74, 76, 77	print.ctmedmcphi,66
Direct, 8, 12, 15, 19, 21, 25, 26, 29, 33, 37,	print.ctmedmed,67
40, 43, 47, 53, 56, 58, 59, 73, 74, 76,	
77	summary.ctmeddelta,68
Direct(), 14, 36, 45, 55	summary.ctmedmc, 70
bii ccc(), 14, 50, 45, 55	summary.ctmedmed, 71
Indirect, 8, 12, 15, 19, 22, 23, 26, 29, 33, 37,	
40, 43, 47, 53, 56, 58, 59, 73, 74, 76,	TestPhi, 8, 12, 15, 19, 22, 25, 26, 29, 33, 37,
77	40, 43, 47, 53, 56, 58, 59, 73, 74, 76,
Indirect(), 14, 36, 45, 55	77
IndirectCentral, 8, 12, 15, 19, 22, 25, 26,	TestPhi(), 28, 31, 35, 39, 42
29, 33, 37, 40, 43, 47, 53, 56, 58, 59,	TestStable, 8, 12, 15, 19, 22, 25, 26, 29, 33,
73, 74, 76, 77	37, 40, 43, 47, 53, 56, 58, 59, 73, 74,
<pre>IndirectCentral(), 10, 32</pre>	76, 77

INDEX 81

```
Total, 8, 12, 15, 19, 22, 25, 26, 29, 33, 37, 40, 43, 47, 53, 56, 58, 59, 73, 74, 75, 77

Total(), 7, 14, 28, 36, 45, 55

TotalCentral, 8, 12, 15, 19, 22, 25, 26, 29, 33, 37, 40, 43, 47, 53, 56, 58, 59, 73, 74, 76, 77

TotalCentral(), 18, 42, 52, 58
```