# Package 'cTMed'

September 20, 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,</pre>					
https://jeksterslab.github.io/cTMed/					
<pre>BugReports https://github.com/jeksterslab/cTMed/issues License GPL (&gt;= 3) Encoding UTF-8</pre>					
<b>Roxygen</b> list(markdown = TRUE)					
<b>Depends</b> R (>= $3.5.0$ )					
LinkingTo Rcpp, RcppArmadillo					
Imports Rcpp, numDeriv, parallel, ctsem, simStateSpace					
Suggests knitr, rmarkdown, testthat, expm					
RoxygenNote 7.3.2					
NeedsCompilation yes					
Author Ivan Jacob Agaloos Pesigan [aut, cre, cph] ( <a href="https://orcid.org/0000-0003-4818-8420">https://orcid.org/0000-0003-4818-8420</a> )					
Maintainer Ivan Jacob Agaloos Pesigan <r.jeksterslab@gmail.com></r.jeksterslab@gmail.com>					
Contents					
confint.ctmeddelta       confint.ctmedmc         DeltaBeta       DeltaIndirectCentral       1         DeltaMed       1         DeltaTotalCentral       1         Direct       2         Indirect       2					

2 confint.ctmeddelta

IndirectCentral	6
MCBeta	7
MCIndirectCentral	1
MCMed	5
MCPhi	9
MCTotalCentral	1
Med	5
plot.ctmeddelta	8
plot.ctmedmc	9
plot.ctmedmed	1
plot.ctmedtraj	2
PosteriorBeta	3
PosteriorIndirectCentral	5
PosteriorMed	8
PosteriorPhi	1
PosteriorTotalCentral	2
print.ctmeddelta	4
print.ctmedeffect	6
print.ctmedmc	8
print.ctmedmcphi	9
print.ctmedmed	1
print.ctmedtraj	2
summary.ctmeddelta	3
summary.ctmedmc	5
summary.ctmedmed	6
summary.ctmedposteriorphi	8
summary.ctmedtraj	8
Total	9
TotalCentral	1
Trajectory	3
8.	5

# Description

Delta Method Confidence Intervals

# Usage

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

3 confint.ctmeddelta

# **Arguments**

Object of class ctmeddelta. object a specification of which parameters are to be given confidence intervals, either parm a vector of numbers or a vector of names. If missing, all parameters are considered. the confidence level required. level

additional arguments.

#### Value

Returns a matrix of confidence intervals.

### Author(s)

Ivan Jacob Agaloos Pesigan

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

4 confint.ctmedmc

```
-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(
 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.

confint.ctmedmc 5

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.

1evel the confidence level required.
... additional arguments.

#### Value

Returns a matrix of confidence intervals.

## Author(s)

Ivan Jacob Agaloos Pesigan

```
set.seed(42)
phi <- matrix(</pre>
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) \leftarrow rownames(phi) \leftarrow c("x", "m", "y")
vcov_phi_vec <- matrix(</pre>
  data = c(
    0.002704274, -0.001475275, 0.000949122,
    -0.001619422, 0.000885122, -0.000569404,
    0.00085493, -0.000465824, 0.000297815,
    -0.001475275, 0.004428442, -0.002642303,
    0.000980573, -0.00271817, 0.001618805,
    -0.000586921, 0.001478421, -0.000871547,
    0.000949122, -0.002642303, 0.006402668,
    -0.000697798, 0.001813471, -0.004043138,
    0.000463086, -0.001120949, 0.002271711,
    -0.001619422, 0.000980573, -0.000697798,
    0.002079286, -0.001152501, 0.000753,
    -0.001528701, 0.000820587, -0.000517524,
    0.000885122, -0.00271817, 0.001813471,
    -0.001152501, 0.00342605, -0.002075005,
    0.000899165, -0.002532849, 0.001475579,
    -0.000569404, 0.001618805, -0.004043138,
    0.000753, -0.002075005, 0.004984032,
    -0.000622255, 0.001634917, -0.003705661,
    0.00085493, -0.000586921, 0.000463086,
    -0.001528701, 0.000899165, -0.000622255,
    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 # use a large value for R 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  $\beta$  over a specific time interval  $\Delta t$  or a range of time intervals using the first-order stochastic differential equation model's drift matrix  $\Phi$ .

## Usage

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

#### **Arguments**

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

pertaining to the variables in the system.

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

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

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

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

## **Details**

See Total().

#### **Delta Method:**

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

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

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

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

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

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

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

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

An alternative representation of the measurement error is given by

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

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

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 ("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(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()
```

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

10 DeltaIndirectCentral

```
phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1:5
)
plot(delta)

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

DeltaIndirectCentral

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

#### **Description**

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

## Usage

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

#### **Arguments**

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

pertaining to the variables in the system.

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

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

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

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

## **Details**

See IndirectCentral() more details.

# **Delta Method:**

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

DeltaIndirectCentral 11

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

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

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

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

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

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

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

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

#### Value

Returns an object of class 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).

12 DeltaIndirectCentral

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(), DeltaMed(), DeltaTotalCentral(), Direct(), Indirect(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), Med(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()

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

```
-0.001619422, 0.000980573, -0.000697798,
   0.002079286, -0.001152501, 0.000753,
   -0.001528701, 0.000820587, -0.000517524,
   0.000885122, -0.00271817, 0.001813471,
   -0.001152501, 0.00342605, -0.002075005,
   0.000899165, -0.002532849, 0.001475579,
   -0.000569404, 0.001618805, -0.004043138,
   0.000753, -0.002075005, 0.004984032,
   -0.000622255, 0.001634917, -0.003705661,
   0.00085493, -0.000586921, 0.000463086,
   -0.001528701, 0.000899165, -0.000622255,
   0.002060076, -0.001096684, 0.000686386,
   -0.000465824, 0.001478421, -0.001120949,
   0.000820587, -0.002532849, 0.001634917,
   -0.001096684, 0.003328692, -0.001926088,
   0.000297815, -0.000871547, 0.002271711,
   -0.000517524, 0.001475579, -0.003705661,
   0.000686386, -0.001926088, 0.004726235
 ),
 nrow = 9
)
# Specific time interval ------
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  $\mathbf{\Phi}$ .

### Usage

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

# **Arguments**

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

pertaining to the variables in the system.

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

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

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

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

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

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

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

### **Details**

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

# **Delta Method:**

Let  $\theta$  be  $\operatorname{vec}(\Phi)$ , 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

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

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

An alternative representation of the measurement error is given by

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

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 ("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(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()

```
phi <- matrix(</pre>
 data = c(
   -0.357, 0.771, -0.450,
   0.0, -0.511, 0.729,
   0, 0, -0.693
 ),
 nrow = 3
)
colnames(phi) \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
)
# Specific time interval -------
DeltaMed(
```

DeltaTotalCentral 17

```
phi = phi,
 vcov_phi_vec = vcov_phi_vec,
 delta_t = 1,
 from = "x",
 to = "y",
 med = "m"
# 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  $\mathbf{g}\left(\hat{\boldsymbol{\theta}}\right)$  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

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

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

The measurement model is given by

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

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

An alternative representation of the measurement error is given by

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

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(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()

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

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

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

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(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()

```
phi <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
),
  nrow = 3</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(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()
```

```
phi <- matrix(</pre>
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
delta_t <- 1
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(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()
```

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 tests the stability of the generated drift matrix  $\Phi$ . If the test returns FALSE, the function generates a new drift matrix  $\Phi$  and runs the test recursively until the test returns TRUE.

ncores Positive integer. Number of cores to use. If ncores = NULL, use a single core. Consider using multiple cores when number of replications R is a large value.

seed Random seed.

#### **Details**

See Total().

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

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

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

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

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

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

# **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  $\mathbf{y}_{i,t}$ ,  $\boldsymbol{\eta}_{i,t}$ , 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(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()
```

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

MCIndirectCentral

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

# **Description**

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

# Usage

```
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 $\operatorname{vec}\left(\mathbf{\Phi}\right)$ .
delta_t	Numeric. Time interval ( $\Delta t$ ).
R	Positive integer. Number of replications.
test_phi	Logical. If test_phi = TRUE, the function tests the stability of the generated drift matrix $\Phi$ . If the test returns FALSE, the function generates a new drift matrix $\Phi$ and runs the test recursively until the test returns TRUE.
ncores	Positive integer. Number of cores to use. If ncores = NULL, use a single core. Consider using multiple cores when number of replications R is a large value.
seed	Random seed.

#### **Details**

See IndirectCentral() for more details.

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

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

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

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

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

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

# **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(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()
```

```
set.seed(42)
phi <- matrix(</pre>
 data = c(
    -0.357, 0.771, -0.450,
   0.0, -0.511, 0.729,
   0, 0, -0.693
 ),
 nrow = 3
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
vcov_phi_vec <- matrix(</pre>
 data = c(
    0.002704274, -0.001475275, 0.000949122,
    -0.001619422, 0.000885122, -0.000569404,
    0.00085493, -0.000465824, 0.000297815,
    -0.001475275, 0.004428442, -0.002642303,
    0.000980573, -0.00271817, 0.001618805,
```

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

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

# Description

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

# Usage

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

# **Arguments** phi

P	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 ( $\Delta t$ ).
from	Character string. Name of the independent variable $X$ in phi.
to	Character string. Name of the dependent variable $Y$ in phi.
med	Character vector. Name/s of the mediator variable/s in phi.
R	Positive integer. Number of replications.
test_phi	Logical. If test_phi = TRUE, the function tests the stability of the generated drift matrix $\Phi$ . If the test returns FALSE, the function generates a new drift matrix $\Phi$ and runs the test recursively until the test returns TRUE.
ncores	Positive integer. Number of cores to use. If ncores = NULL, use a single core. Consider using multiple cores when number of replications R is a large value.
seed	Random seed.

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

## **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.

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

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(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()
```

```
set.seed(42)
phi <- matrix(</pre>
 data = c(
    -0.357, 0.771, -0.450,
   0.0, -0.511, 0.729,
   0, 0, -0.693
 ),
 nrow = 3
colnames(phi) <- rownames(phi) <- c("x", "m", "y")</pre>
vcov_phi_vec <- matrix(</pre>
 data = c(
    0.002704274, -0.001475275, 0.000949122,
    -0.001619422, 0.000885122, -0.000569404,
    0.00085493, -0.000465824, 0.000297815,
    -0.001475275, 0.004428442, -0.002642303,
    0.000980573, -0.00271817, 0.001618805,
```

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 tests the stability of the generated drift matrix $\Phi$ . If the test returns FALSE, the function generates a new drift matrix $\Phi$ and runs the test recursively until the test returns TRUE.
ncores	Positive integer. Number of cores to use. If ncores = NULL, use a single core. Consider using multiple cores when number of replications R is a large value.
seed	Random seed.

# **Details**

### **Monte Carlo Method:**

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

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

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

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

40 MCPhi

## **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

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

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 ("MCPhi").

output A list simulated drift matrices.

# Author(s)

Ivan Jacob Agaloos Pesigan

### See Also

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

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

MCTotalCentral 41

```
0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) \leftarrow rownames(phi) \leftarrow c("x", "m", "y")
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**

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

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

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

R Positive integer. Number of replications.

test\_phi Logical. If test\_phi = TRUE, the function tests the stability of the generated drift matrix  $\Phi$ . If the test returns FALSE, the function generates a new drift matrix  $\Phi$  and runs the test recursively until the test returns TRUE.

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

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  $\text{vec}(\Phi)$ , that is, the elements of the  $\Phi$  matrix in vector form sorted column-wise. Let  $\hat{\theta}$  be  $\text{vec}(\hat{\Phi})$ . 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

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

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

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

MCTotalCentral 43

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:

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(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()

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,
    -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(), IndirectCentral(), MCBeta(), MCIndirectCentral(), MCMed(), MCPhi(), MCTotalCentral(), PosteriorBeta(), PosteriorIndirectCentral(), PosteriorMed(), PosteriorPhi(), PosteriorTotalCentral(), Total(), TotalCentral(), Trajectory()

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

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

plot.ctmedtraj

Plot Method for an Object of Class ctmedtraj

### **Description**

Plot Method for an Object of Class ctmedtraj

# Usage

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

# Arguments

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

## 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")

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

PosteriorBeta 53

```
plot(traj)
```

PosteriorBeta Posterior Sampling Distribution for the Elements of the Matrix of

Lagged Coefficients Over a Specific Time Interval or a Range of Time

Intervals

## **Description**

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

### Usage

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

### **Arguments**

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

pertaining to the variables in the system.

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

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

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

#### **Details**

See Total().

### Value

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

call Function call.

args Function arguments.

fun Function used ("PosteriorBeta").

output A list 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.

54 PosteriorBeta

#### Author(s)

Ivan Jacob Agaloos Pesigan

#### References

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

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

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

#### See Also

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

```
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,
```

PosteriorIndirectCentral 55

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

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

56 PosteriorIndirectCentral

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.

## 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

PosteriorIndirectCentral 57

#### See Also

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

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

58 PosteriorMed

```
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  $\mathbf{\Phi}$ .

### Usage

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

# Arguments

phi	List of numeric matrices. Each element of the list is a sample from the posterior distribution of the drift matrix ( $\Phi$ ). Each matrix should have row and column names pertaining to the variables in the system.
delta_t	Numeric. Time interval ( $\Delta t$ ).
from	Character string. Name of the independent variable $X$ in phi.
to	Character string. Name of the dependent variable $Y$ in phi.

PosteriorMed 59

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

#### See Also

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

60 PosteriorMed

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

PosteriorPhi 61

```
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

### See Also

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

62 PosteriorTotalCentral

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 poster.
---

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.

### Author(s)

Ivan Jacob Agaloos Pesigan

PosteriorTotalCentral 63

#### References

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

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

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

#### See Also

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

```
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,
```

64 print.ctmeddelta

```
-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

# Usage

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

print.ctmeddelta 65

# 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,
    -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,
```

66 print.ctmedeffect

```
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, ...)
```

# Arguments

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

print.ctmedeffect 67

## 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",
 med = "m"
)
print(indirect_dt)
```

68 print.ctmedmc

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(</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,
```

print.ctmedmcphi 69

```
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)
```

70 print.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 71

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 ------
```

72 print.ctmedtraj

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

print.ctmedtraj

Print Method for Object of Class ctmedtraj

# Description

Print Method for Object of Class ctmedtraj

# Usage

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

# Arguments

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

## Value

Returns a data frame of simulated data.

## Author(s)

Ivan Jacob Agaloos Pesigan

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

summary.ctmeddelta 73

```
time = 150,
phi = phi,
med = "m"
)
print(traj)
```

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(
  data = c(
    0.002704274, -0.001475275, 0.000949122,</pre>
```

74 summary.ctmeddelta

```
-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)
```

summary.ctmedmc 75

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,
    0.000949122, -0.002642303, 0.006402668,
    -0.000697798, 0.001813471, -0.004043138,
```

76 summary.ctmedmed

```
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)
```

summary.ctmedmed 77

## **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"
```

78 summary.ctmedtraj

```
)
summary(med)
```

```
summary.ctmedposteriorphi
```

Summary Method for Object of Class ctmedposteriorphi

# Description

Summary Method for Object of Class ctmedposteriorphi

#### Usage

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

# Arguments

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

#### Value

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

## Author(s)

Ivan Jacob Agaloos Pesigan

summary.ctmedtraj

Summary Method for an Object of Class ctmedtraj

## **Description**

Summary Method for an Object of Class ctmedtraj

#### **Usage**

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

# Arguments

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

Total 79

## Value

Returns a data frame of simulated data.

# Author(s)

Ivan Jacob Agaloos Pesigan

## **Examples**

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

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

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**

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 ( $\Delta t$ ).

80 Total

#### **Details**

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

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

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

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

The measurement model is given by

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

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

TotalCentral 81

#### See Also

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

## **Examples**

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

82 TotalCentral

#### Arguments

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

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

#### Author(s)

Ivan Jacob Agaloos Pesigan

#### See Also

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

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

Trajectory 83

Trajectory	Simulate Trajectories of Variables	

## **Description**

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

#### Usage

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

## Arguments

mu0	Numeric vector. Initial values of the variables.
time	Positive integer. Number of time points.
phi	Numeric matrix. The drift matrix ( $\Phi$ ). phi should have row and column names pertaining to the variables in the system.
med	Character vector. Name/s of the mediator variable/s in phi.

## Value

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

```
call Function call.args Function arguments.fun Function used ("Trajectory").output A data frame of simulated data.
```

#### See Also

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

84 Trajectory

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

# **Index**

DeltaBeta, 6 DeltaIndirectCentral, 10 DeltaMed, 13 DeltaTotalCentral, 17 Direct, 21 Indirect, 23 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 Med, 45 PosteriorTotalCentral, 41 Med, 45 PosteriorTotalCentral, 55 PosteriorTotalCentral, 26 MCBeta, 27 MCIndirectCentral, 41 Med, 45 PosteriorDeta, 53 PosteriorDeta, 53 PosteriorDeta, 53 PosteriorDeta, 53 PosteriorDeta, 54 PosteriorDetalCentral, 62 Total, 79 TotalCentral, 81 Trajectory, 83 * methods confint.ctmeddelta, 2 confint.ctmeddelta, 2 confint.ctmeddelta, 48 plot.ctmeddelta, 48 plot.ctmeddelta, 48 plot.ctmeddelta, 48 plot.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 73 summary.ctmedded, 71 DeltaBeta, 6 DeltaBeta, 6 DeltaIndirectCentral, 10 DeltaBeta, 6 DeltaIndirectCentral, 17 Direct, 21 Indirect, 23 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39 MCIndirectCentral, 31 MCMed, 35 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39 MCIndirectCentral, 31	* Continuous Time Mediation Functions	Med, 45
DeltaIndirectCentral, 10 DeltaWed, 13 DeltaTotalCentral, 17 Direct, 21 Indirect, 23 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 Med, 45 PosteriorPostalCentral, 41 Med, 45 PosteriorBeta, 53 PosteriorBeta, 53 PosteriorPhi, 61 PosteriorTotalCentral, 55 PosteriorBeta, 53 PosteriorBeta, 53 PosteriorTotalCentral, 62 Total, 79 TotalCentral, 81 Trajectory, 83 **methods Confint.ctmeddelta, 2 confint.ctmedmc, 4 plot.ctmedmc, 4 plot.ctmedmc, 4 plot.ctmedmc, 4 plot.ctmeddelta, 48 plot.ctmeddelta, 48 plot.ctmeddelta, 48 plot.ctmeddelta, 49 plot.ctmeddelta, 64 print.ctmedeffect, 66 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 73 summary.ctmeddelta,		
DeltaMed, 13 DeltaTotalCentral, 17 Direct, 21 Indirect, 23 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 Med, 45 PosteriorFlat, 61 PosteriorTotalCentral, 81 Trajectory, 83 McMed, 35 McMed, 35 PosteriorBeta, 53 PosteriorBeta, 53 PosteriorBeta, 62 Total, 79 TotalCentral, 62 Total, 79 TotalCentral, 62 PosteriorMed, 58 PosteriorMed, 58 PosteriorTotalCentral, 62 Total, 79 TotalCentral, 81 Trajectory, 83 * methods  confint.ctmeddelta, 2 confint.ctmeddelta, 2 confint.ctmeddelta, 48 plot.ctmeddelta, 48 plot.ctmeddelta, 48 plot.ctmeddelta, 48 plot.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 71 print.ctmeddelta, 73 summary.ctmedmed, 71 print.ctmeddelta, 73 summary.ctmedmed, 75 summary.ctmedmed, 76 summary.ctmedded, 76 summary.ctmedded, 77 print.ctmedic, 75 summary.ctmedded, 76 summary.ctmedded, 77 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39 MCIndirectCentral, 31 MCMed, 35 MCIndirectCentral, 31 MCMed, 35 MCIndirectCentral, 31	*	•
DeltaTotalCentral, 17 Direct, 21 Indirect, 23 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 Med, 35 MCNed, 35 MCNed, 45 PosteriorBeta, 53 PosteriorIndirectCentral, 55 PosteriorTotalCentral, 62 Total, 79 TotalCentral, 41 Med, 45 PosteriorBeta, 53 PosteriorIndirectCentral, 55 PosteriorTotalCentral, 62 Total, 79 TotalCentral, 81 Trajectory, 83 **methods confint.ctmeddelta, 2 confint.ctmeddelta, 2 confint.ctmeddelta, 2 plot.ctmeddelta, 48 plot.ctmeddelta, 48 plot.ctmeddelta, 48 plot.ctmeddelta, 64 print.ctmeddelta, 67 print.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmeddelta, 78 summary.ctmeddelta, 79 summary.ctmeddelta,		•
Direct, 21 Indirect, 23 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39 MCTotalCentral, 41 Med, 45 PosteriorBeta, 53 PosteriorIndirectCentral, 55 PosteriorMed, 58 PosteriorPhi, 61 PosteriorTotalCentral, 81 Trajectory, 83 * beta DeltaBeta, 6 MCBeta, 27 PosteriorBeta, 53 * beta DeltaBeta, 6 MCBeta, 27 PosteriorBeta, 53 * beta DeltaBeta, 6 MCBeta, 27 DeltaIndirectCentral, 10 DeltaMed, 13 DeltaTotalCentral, 26 MCBeta, 27 MCIndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39  PosteriorTotalCentral, 62 Total, 79 Trajectory, 83 * birect, 21 Indirect, 23 IndirectCentral, 10 DeltaMed, 13 Summary.ctmeddelta, 73 Summary.ctmeddelta, 73 Summary.ctmeddelta, 73 Summary.ctmeddelta, 73 Summary.ctmeddend, 76 Summary.ctmeddend, 76 Summary.ctmeddend, 76 Summary.ctmeddend, 76 Summary.ctmeddend, 78 Network DeltaTotalCentral, 10 DeltaTotalCentral, 11 IndirectCentral, 26 MCIndirectCentral, 31 MCMed, 35 MCIndirectCentral, 31		
Indirect, 23 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39 MCTotalCentral, 41 Med, 45 PosteriorBeta, 53 PosteriorPhi, 61 PosteriorTotalCentral, 82 Total, 79 TotalCentral, 83 PosteriorTotalCentral, 62 Total, 79 TotalCentral, 81 Trajectory, 83 * methods confint.ctmeddelta, 2 confint.ctmeddelta, 2 confint.ctmeddelta, 48 plot.ctmeddelta, 48 plot.ctmedmed, 51 Trajectory, 83 * beta DeltaBeta, 6 MCBeta, 27 PosteriorBeta, 53 * cTMed DeltaBeta, 6 DeltaIndirectCentral, 10 DeltaMed, 13 DeltaTotalCentral, 17 Direct, 21 Indirect, 23 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39  TotalCentral, 81 Trajectory, 83 * methods confint.ctmeddelta, 2 confint.ctmeddelta, 2 plot.ctmeddelta, 48 plot.ctmedmed, 51 print.ctmedderfect, 66 print.ctmedderia, 52 print.ctmeddelta, 64 print.ctmeddeffect, 66 print.ctmedmed, 71 print.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmedded, 76 summary.ctmedmed, 76 summary.ctmedderj, 78 * metwork DeltaIndirectCentral, 10 DeltaTotalCentral, 17 IndirectCentral, 26 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39		· · · · · · · · · · · · · · · · · · ·
IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39 MCTotalCentral, 41 Med, 45 PosteriorBeta, 53 PosteriorPhi, 61 PosteriorTotalCentral, 62 Total, 79 TotalCentral, 81 PosteriorTotalCentral, 62 Total, 79 TotalCentral, 81 Trajectory, 83 * methods confint.ctmeddelta, 2 confint.ctmeddelta, 2 confint.ctmeddelta, 48 plot.ctmedmed, 49 plot.ctmeddelta, 48 plot.ctmeddelta, 48 plot.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddeffect, 66 print.ctmeddeffect, 66 print.ctmedmed, 71 print.ctmedded, 72 posteriorBeta, 53 * cTMed DeltaBeta, 6 DeltaIndirectCentral, 10 DeltaMed, 13 DeltaTotalCentral, 17 Direct, 21 Indirect, 23 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39  TotalCentral, 81 Trajectory, 83 * effects Direct, 21 Total, 79 Trajectory, 83 * methods confint.ctmeddelta, 2 confint.ctmeddelta, 2 plot.ctmeddelta, 48 plot.ctmedmed, 51 plot.ctmeddelta, 64 print.ctmedderj, 52 print.ctmeddend, 75 summary.ctmeddend, 77 summary.ctmedded, 75 summary.ctmedded, 76 summary.ctmedmed, 76 summary.ctmedderj, 78 summary.ctmedtraj, 78 * network DeltaTotalCentral, 10 DeltaTotalCentral, 17 IndirectCentral, 26 MCIndirectCentral, 31		
MCBeta, 27  MCIndirectCentral, 31  MCMed, 35  MCPhi, 39  MCTotalCentral, 41  Med, 45  PosteriorBeta, 53  PosteriorPhi, 61  PosteriorTotalCentral, 62  Total, 79  TotalCentral, 81  Trajectory, 83  * methods  confint.ctmeddelta, 2  confint.ctmeddelta, 4  plot.ctmeddelta, 48  plot.ctmeddelta, 49  plot.ctmedderaj, 52  print.ctmeddelta, 64  print.ctmeddelta, 64  print.ctmedmed, 51  plot.ctmedderaj, 52  print.ctmeddelta, 64  print.ctmedmed, 68  print.ctmedmed, 71  peltaBeta, 6  MCBeta, 27  PosteriorBeta, 53  * cTMed  DeltaIndirectCentral, 10  DeltaMed, 13  DeltaTotalCentral, 17  Direct, 21  Indirect, 23  Indirect, 23  IndirectCentral, 26  MCBeta, 27  MCIndirectCentral, 31  MCMed, 35  MCPhi, 39  Trajectory, 83  * effects  Direct, 21  Indirect, 23  Seffects  Direct, 21  Indirect, 21  Indirect, 23  Seffects  Direct, 21  Indirect, 21  Indirect, 21  Indirect, 23  Seffects  Direct, 21  Indirect, 23  Seffects  Direct, 21  Indirect, 21  Indirect, 23  Indirect, 23  Indirect, 23  Indirect, 21  Indirect, 21  Indirect, 21  Indirect, 23  Indirect, 21  Indirect, 21  Indirect, 23  Indirect, 21  Indirect, 21  Indirect, 21  Indirect, 21  Indirect, 23  Indirect, 23  Indirect, 23  Indirect, 23  Indirect, 26  MCIndirectCentral, 31		· · · · · · · · · · · · · · · · · · ·
MCIndirectCentral, 31 MCMed, 35 MCPhi, 39 MCTotalCentral, 41 Med, 45 PosteriorBeta, 53 PosteriorHoid, 61 PosteriorTotalCentral, 62 Total, 79 TotalCentral, 81 Trajectory, 83 * beta DeltaBeta, 6 MCBeta, 27 PosteriorBeta, 53 * cTMed DeltaBeta, 6 DeltaIndirectCentral, 10 DeltaMed, 13 DeltaTotalCentral, 26 MCBeta, 27 MCIndirect, 21 Indirect, 23 Indirect, 23 Indirect, 23 IndirectCentral, 31 MCMed, 35 MCPhi, 39  * effects Direct, 21 Indirect, 23 Indirect, 23 Indirect, 23 IndirectCentral, 31 MCMed, 35 MCPhi, 39  * effects Direct, 21 Indirect, 23 Indirect, 23 IndirectCentral, 41 Med, 45 Total, 79 Trajectory, 83  * methods confint.ctmeddelta, 2 confint.ctmeddelta, 4 plot.ctmeddelta, 48 plot.ctmeddelta, 49 plot.ctmeddelta, 51 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmedmed, 71 print.ctmedmed, 71 print.ctmedmed, 71 print.ctmeddelta, 73 summary.ctmeddelta, 74 summa		,
MCMed, 35 MCPhi, 39 MCTotalCentral, 41 Med, 45 PosteriorBeta, 53 PosteriorIndirectCentral, 55 PosteriorPhi, 61 PosteriorTotalCentral, 62 Total, 79 TotalCentral, 81 Trajectory, 83 * methods PotalCentral, 81 Trajectory, 83 * beta DeltaBeta, 6 MCBeta, 27 PosteriorBeta, 53 * cTMed DeltaBeta, 6 DeltaIndirectCentral, 10 DeltaMed, 13 DeltaMed, 13 DeltaTotalCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39  Direct, 21 IndirectCentral, 26 MCIndirectCentral, 31	,	•
MCPhi, 39  MCTotalCentral, 41  Med, 45  PosteriorBeta, 53  PosteriorPhi, 61  PosteriorPhi, 61  PosteriorTotalCentral, 62  Total, 79  TotalCentral, 81  Trajectory, 83  * methods  confint.ctmeddelta, 2  confint.ctmeddelta, 48  plot.ctmeddmc, 49  plot.ctmeddmc, 49  plot.ctmeddmc, 51  Trajectory, 83  * beta  DeltaBeta, 6  MCBeta, 27  PosteriorBeta, 53  * cTMed  DeltaBeta, 6  DeltaIndirectCentral, 10  DeltaMed, 13  DeltaMed, 13  DeltaTotalCentral, 26  MCBeta, 27  MCIndirectCentral, 26  MCBeta, 27  MCIndirectCentral, 31  MCMed, 35  MCPhi, 39  IndirectCentral, 26  MCIndirectCentral, 31		Direct. 21
MCTotalCentral, 41 Med, 45 Med, 45 PosteriorBeta, 53 PosteriorIndirectCentral, 55 PosteriorPhi, 61 PosteriorTotalCentral, 62 Total, 79 Trajectory, 83 * methods  confint.ctmeddelta, 2 confint.ctmeddelta, 48 plot.ctmeddelta, 48 plot.ctmeddelta, 48 plot.ctmeddelta, 48 plot.ctmeddelta, 49 plot.ctmeddelta, 49 plot.ctmeddelta, 51 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 67 print.ctmeddelta, 68 print.ctmeddelta, 70 print.ctmeddelta, 71 print.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmeddelta, 75 summary.ctmeddelta, 76 summary.ctmeddelta, 76 summary.ctmeddelta, 78 summary.ctmeddelta, 79 summary.ctmeddelta, 79 summary.ctmeddelta, 79 summary.ctmeddelta, 79 summary.ctmeddelta, 79 summary.ctmedtraj, 78 * network DeltaIndirectCentral, 10 DeltaTotalCentral, 17 IndirectCentral, 26 MCIndirectCentral, 31	•	,
Med, 45 PosteriorBeta, 53 PosteriorIndirectCentral, 55 PosteriorMed, 58 PosteriorTotalCentral, 62 Total, 79 PosteriorPhi, 61 PosteriorTotalCentral, 62 Total, 79 Totalcentral, 81 Total, 79 PosteriorTotalCentral, 62 Poltameded, 49 Polt.ctmeddelta, 48 Polt.ctmedded, 51 Trajectory, 83  * beta  DeltaBeta, 6 MCBeta, 27 PosteriorBeta, 53  * cTMed  DeltaBeta, 6 DeltaIndirectCentral, 10 DeltaBeta, 6 DeltaIndirectCentral, 17 Direct, 21 Indirect, 23 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39  Total, 79 Trajectory, 83  * methods  confint.ctmeddelta, 2  confint.ctmeddelta, 2  plot.ctmeddelta, 48 plot.ctmeddelta, 51 print.ctmeddelta, 64 print.ctmeddeffect, 66 print.ctmeddeffect, 66 print.ctmeddend, 70 print.ctmeddedlta, 73 summary.ctmeddelta, 74 summary.ctmeddelta, 74 summary.ctmeddelta, 74 summary.ctmeddelta, 64 print.ctmeddelta, 70 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 73 summary.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmeddelta, 75 summary.ctmeddelta, 7		,
PosteriorBeta, 53 PosteriorIndirectCentral, 55 PosteriorMed, 58 PosteriorTotalCentral, 62 Total, 79 TotalCentral, 81 Trajectory, 83 * beta DeltaBeta, 6 MCBeta, 27 DeltaIndirectCentral, 10 DeltaMed, 13 DeltaMed, 13 DeltaTotalCentral, 26 MCBeta, 27 MCIndirectCentral, 26 MCBeta, 27 Direct, 21 Indirect, 23 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39  Trajectory, 83 * methods confint.ctmeddelta, 2 confint.ctmeddelta, 4 plot.ctmedmc, 4 plot.ctmedmc, 49 plot.ctmedmc, 51 plot.ctmedtraj, 52 print.ctmeddelta, 64 print.ctmeddeffect, 66 print.ctmeddmcphi, 70 print.ctmedmcphi, 70 print.ctmedmcphi, 70 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmedmed, 76 summary.ctmeddraj, 78 summary.ctmedtraj, 78 IndirectCentral, 31 MCMed, 35 MCIndirectCentral, 31  MCIndirectCentral, 31 MCIndirectCentral, 31		,
PosteriorIndirectCentral, 55 PosteriorMed, 58 PosteriorPhi, 61 PosteriorTotalCentral, 62 Total, 79 TotalCentral, 81 Trajectory, 83 PosteriorBeta, 6 MCBeta, 27 PosteriorBeta, 6 DeltaBeta, 6 DeltaIndirectCentral, 10 DeltaBeta, 71 DeltaBeta, 8 DeltaTotalCentral, 17 Direct, 21 Indirect, 23 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39  * methods confint.ctmeddelta, 2 confint.ctmedmc, 4 plot.ctmeddelta, 48 plot.ctmedmed, 51 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmedmc, 68 print.ctmedmcd, 70 print.ctmedmcd, 71 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmedmed, 76 summary.ctmeddelta, 78 summary.ctmeddelta, 78 summary.ctmeddelta, 78 indirectCentral, 10 DeltaTotalCentral, 10 DeltaTotalCentral, 17 IndirectCentral, 26 MCIndirectCentral, 31	,	
PosteriorMed, 58 PosteriorPhi, 61 Confint.ctmeddelta, 2 Confint.ctmedmc, 4 PosteriorTotalCentral, 62 Total, 79 TotalCentral, 81 Trajectory, 83 PosteriorBeta, 6 MCBeta, 27 PosteriorBeta, 53 PoltaIndirectCentral, 10 DeltaMed, 13 DeltaTotalCentral, 17 Direct, 21 Indirect, 23 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39  Confint.ctmeddelta, 2 Confint.ctmedmc, 4 Delt.ctmedmc, 49 Delt.ctmedmed, 51 Delt.ctmeddelta, 64 Delt.ctmeddelta, 64 Delt.ctmeddelta, 64 Delt.ctmeddelta, 64 Delt.ctmedmc, 68 Delt.ctmedmc, 68 Delt.ctmeddelta, 70 Drint.ctmedmc, 71 Drint.ctmedmed, 71 Drint.ctmeddelta, 73 Summary.ctmeddelta, 73 Summary.ctmeddelta, 73 Summary.ctmedmed, 76 Summary.ctmedmed, 76 DeltaIndirectCentral, 17 DeltaTotalCentral, 10 DeltaTotalCentral, 10 DeltaTotalCentral, 17 IndirectCentral, 26 MCIndirectCentral, 26 MCIndirectCentral, 31		•
PosteriorTotalCentral, 62 Total, 79 TotalCentral, 81 Trajectory, 83  * beta DeltaBeta, 6 MCBeta, 27 DeltaBeta, 6 DeltaIndirectCentral, 10 DeltaTotalCentral, 17 Direct, 21 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39  plot.ctmeddelta, 48 plot.ctmedmc, 49 plot.ctmedmed, 51 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 68 print.ctmedmcphi, 70 print.ctmedmed, 71 print.ctmedtraj, 72 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmedmed, 76 summary.ctmedded, 76 pletaIndirectCentral, 78 IndirectCentral, 26 MCIndirectCentral, 10 DeltaTotalCentral, 17 IndirectCentral, 26 MCIndirectCentral, 31		confint.ctmeddelta, 2
PosteriorTotalCentral, 62 Total, 79 TotalCentral, 81 Trajectory, 83  * beta DeltaBeta, 6 MCBeta, 27 DeltaBeta, 6 DeltaIndirectCentral, 10 DeltaTotalCentral, 17 Direct, 21 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39  plot.ctmeddelta, 48 plot.ctmedmc, 49 plot.ctmedmed, 51 print.ctmeddelta, 64 print.ctmeddelta, 64 print.ctmeddelta, 68 print.ctmedmcphi, 70 print.ctmedmed, 71 print.ctmedtraj, 72 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmedmed, 76 summary.ctmedded, 76 pletaIndirectCentral, 78 IndirectCentral, 26 MCIndirectCentral, 10 DeltaTotalCentral, 17 IndirectCentral, 26 MCIndirectCentral, 31	PosteriorPhi, 61	confint.ctmedmc, 4
TotalCentral, 81 Trajectory, 83  * beta  DeltaBeta, 6 MCBeta, 27 PosteriorBeta, 53  * cTMed  DeltaBeta, 6 DeltaIndirectCentral, 10 Direct, 21 Indirect, 23 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39  * beta  plot.ctmedmed, 51 plot.ctmedtraj, 52 print.ctmeddelta, 64 print.ctmeddect, 66 print.ctmedmc, 68 print.ctmedmch, 70 print.ctmedmed, 71 print.ctmedtraj, 72 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmedmed, 76 summary.ctmedposteriorphi, 78 summary.ctmedposteriorphi, 78 pltaIndirectCentral, 26 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39  MCIndirectCentral, 31  MCIndirectCentral, 31  MCIndirectCentral, 31  MCIndirectCentral, 31		plot.ctmeddelta,48
TotalCentral, 81 Trajectory, 83  * beta  DeltaBeta, 6 MCBeta, 27 PosteriorBeta, 53  * cTMed  DeltaBeta, 6 DeltaIndirectCentral, 10 Direct, 21 Indirect, 23 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39  * beta  plot.ctmedmed, 51 plot.ctmedtraj, 52 print.ctmeddelta, 64 print.ctmeddect, 66 print.ctmedmc, 68 print.ctmedmch, 70 print.ctmedmed, 71 print.ctmedtraj, 72 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmedmed, 76 summary.ctmedposteriorphi, 78 summary.ctmedposteriorphi, 78 pltaIndirectCentral, 26 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39  MCIndirectCentral, 31  MCIndirectCentral, 31  MCIndirectCentral, 31  MCIndirectCentral, 31	· · · · · · · · · · · · · · · · · · ·	•
Trajectory, 83  * beta  DeltaBeta, 6 MCBeta, 27 PosteriorBeta, 53  * cTMed  DeltaBeta, 6 DeltaIndirectCentral, 10 Direct, 21 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39  plot.ctmedtraj, 52 print.ctmeddelta, 64 print.ctmedeffect, 66 print.ctmedmc, 68 print.ctmedmcphi, 70 print.ctmedmcd, 71 print.ctmedtraj, 72 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmedmcd, 75 summary.ctmedmed, 76 summary.ctmedposteriorphi, 78 summary.ctmedtraj, 78  * network  DeltaIndirectCentral, 10 DeltaTotalCentral, 17 IndirectCentral, 26 MCIndirectCentral, 31 MCIndirectCentral, 31 MCIndirectCentral, 31 MCIndirectCentral, 31		
* beta print.ctmeddelta, 64 DeltaBeta, 6 MCBeta, 27 PosteriorBeta, 53  * cTMed print.ctmedmcphi, 70 DeltaBeta, 6 DeltaIndirectCentral, 10 DeltaMed, 13 DeltaTotalCentral, 17 Direct, 21 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39  print.ctmeddelta, 64 print.ctmeddec, 68 print.ctmedmcphi, 70 print.ctmedmcphi, 70 print.ctmeddedta, 71 print.ctmeddedta, 71 print.ctmeddedta, 71 print.ctmeddedta, 73 print.ctmeddeco, 71 print.ctmeddeco, 71 print.ctmeddeco, 70 print.ctmeddelta, 70 print.ctmeddeco, 68 print.ctmeddelta, 64 print.ctmeddelta, 70 print.ctmedmc, 68 print.ctmedmc, 68 print.ctmedmc, 68 print.ctmedmc, 68 print.ctmedmc, 71 print.ctmedmc, 71 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmedmc, 75 summary.ctmedmc, 75 summary.ctmeddelta, 73 summary.ctmeddelta, 74 summary.ctmeddelta	•	·
DeltaBeta, 6 MCBeta, 27 PosteriorBeta, 53  **cTMed DeltaBeta, 6 DeltaIndirectCentral, 10 Direct, 21 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39  print.ctmedeffect, 66 print.ctmedmc, 68 print.ctmedmcphi, 70 print.ctmedmed, 71 print.ctmedtraj, 72 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmedmc, 75 summary.ctmedmed, 76 summary.ctmedposteriorphi, 78 summary.ctmedtraj, 78  **network  DeltaIndirectCentral, 10 DeltaTotalCentral, 10 DeltaTotalCentral, 17 IndirectCentral, 26 MCIndirectCentral, 31 MCIndirectCentral, 31 MCIndirectCentral, 31	* beta	· · · · · · · · · · · · · · · · · · ·
MCBeta, 27 PosteriorBeta, 53  * cTMed  DeltaBeta, 6 DeltaIndirectCentral, 10 DeltaTotalCentral, 17 Direct, 21 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39  print.ctmedmc, 68 print.ctmedmcphi, 70 print.ctmedded, 71 print.ctmedtraj, 72 summary.ctmeddelta, 73 summary.ctmeddelta, 73 summary.ctmedmc, 75 summary.ctmedmed, 76 summary.ctmedposteriorphi, 78 summary.ctmedtraj, 78 * network  DeltaIndirectCentral, 10 DeltaTotalCentral, 17 IndirectCentral, 26 MCIndirectCentral, 31 MCIndirectCentral, 31 MCIndirectCentral, 31	DeltaBeta, 6	•
PosteriorBeta, 53  * cTMed  DeltaBeta, 6  DeltaIndirectCentral, 10  Direct, 21  IndirectCentral, 26  MCBeta, 27  MCIndirectCentral, 31  MCMed, 35  MCPhi, 39  print.ctmedmcphi, 70  print.ctmedmed, 71  print.ctmeddraj, 72  summary.ctmeddelta, 73  summary.ctmeddelta, 73  summary.ctmedmed, 76  summary.ctmedposteriorphi, 78  summary.ctmedposteriorphi, 78  summary.ctmedtraj, 78  * network  DeltaIndirectCentral, 10  DeltaTotalCentral, 17  IndirectCentral, 26  MCIndirectCentral, 31  MCIndirectCentral, 31	MCBeta, 27	·
*cTMed print.ctmedmed, 71 DeltaBeta, 6 print.ctmedtraj, 72 DeltaIndirectCentral, 10 summary.ctmeddelta, 73 DeltaMed, 13 summary.ctmedmed, 75 DeltaTotalCentral, 17 summary.ctmedmed, 76 Direct, 21 summary.ctmedposteriorphi, 78 Indirect, 23 summary.ctmedtraj, 78 IndirectCentral, 26 *network  MCBeta, 27 DeltaIndirectCentral, 10 MCIndirectCentral, 31 DeltaTotalCentral, 17 MCMed, 35 MCPhi, 39 MCIndirectCentral, 31		
DeltaBeta, 6 DeltaIndirectCentral, 10 DeltaMed, 13 DeltaTotalCentral, 17 Direct, 21 Indirect, 23 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39  print.ctmedtraj, 72 summary.ctmeddelta, 73 summary.ctmedmc, 75 summary.ctmedmed, 76 summary.ctmedposteriorphi, 78 summary.ctmedtraj, 78 * network DeltaIndirectCentral, 10 DeltaTotalCentral, 17 IndirectCentral, 26 MCIndirectCentral, 31 MCIndirectCentral, 31	* cTMed	· · · · · · · · · · · · · · · · · · ·
DeltaMed, 13 DeltaTotalCentral, 17 Direct, 21 Indirect, 23 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39  summary.ctmedmed, 76 summary.ctmedposteriorphi, 78 summary.ctmedtraj, 78 **network DeltaIndirectCentral, 10 DeltaTotalCentral, 17 IndirectCentral, 26 MCIndirectCentral, 31 MCIndirectCentral, 31	DeltaBeta, 6	· · · · · · · · · · · · · · · · · · ·
DeltaTotalCentral, 17 Direct, 21 Summary.ctmedmed, 76 Summary.ctmedposteriorphi, 78 Indirect, 23 IndirectCentral, 26 MCBeta, 27 MCIndirectCentral, 31 MCMed, 35 MCPhi, 39 Summary.ctmedtraj, 78 * network DeltaIndirectCentral, 10 DeltaTotalCentral, 17 IndirectCentral, 26 MCIndirectCentral, 31	DeltaIndirectCentral, 10	summary.ctmeddelta,73
Direct, 21 summary.ctmedposteriorphi, 78 Indirect, 23 summary.ctmedtraj, 78 IndirectCentral, 26 * network  MCBeta, 27 DeltaIndirectCentral, 10 MCIndirectCentral, 31 DeltaTotalCentral, 17 MCMed, 35 IndirectCentral, 26 MCPhi, 39 MCIndirectCentral, 31	DeltaMed, 13	summary.ctmedmc, 75
Indirect, 23 summary.ctmedtraj, 78 IndirectCentral, 26 * network  MCBeta, 27 DeltaIndirectCentral, 10  MCIndirectCentral, 31 DeltaTotalCentral, 17  MCMed, 35 IndirectCentral, 26  MCPhi, 39 MCIndirectCentral, 31	DeltaTotalCentral, 17	summary.ctmedmed, 76
Indirect, 23 summary.ctmedtraj, 78 IndirectCentral, 26 * network  MCBeta, 27 DeltaIndirectCentral, 10  MCIndirectCentral, 31 DeltaTotalCentral, 17  MCMed, 35 IndirectCentral, 26  MCPhi, 39 MCIndirectCentral, 31	Direct, 21	summary.ctmedposteriorphi, 78
IndirectCentral, 26  MCBeta, 27  MCIndirectCentral, 31  MCMed, 35  MCPhi, 39  * network  DeltaIndirectCentral, 10  DeltaTotalCentral, 17  IndirectCentral, 26  MCIndirectCentral, 31	Indirect, 23	summary.ctmedtraj, 78
MCIndirectCentral, 31  MCMed, 35  MCPhi, 39  DeltaTotalCentral, 17  IndirectCentral, 26  MCIndirectCentral, 31	<pre>IndirectCentral, 26</pre>	
MCMed, 35 IndirectCentral, 26 MCPhi, 39 MCIndirectCentral, 31	MCBeta, 27	DeltaIndirectCentral, 10
MCPhi, 39 MCIndirectCentral, 31	MCIndirectCentral, 31	DeltaTotalCentral, 17
·	MCMed, 35	<pre>IndirectCentral, 26</pre>
MCTotalCentral, 41 MCTotalCentral, 41	MCPhi, 39	MCIndirectCentral, 31
	MCTotalCentral, 41	MCTotalCentral, 41

86 INDEX

PosteriorIndirectCentral, 55 PosteriorTotalCentral, 62	<pre>IndirectCentral(), 10, 32</pre>
TotalCentral, 81	MCBeta, 8, 12, 16, 19, 22, 25, 26, 27, 33, 37,
* path	40, 43, 47, 54, 57, 59, 61, 63, 81–83
DeltaMed, 13	MCIndirectCentral, 8, 12, 16, 19, 22, 25, 26,
MCMed, 35	29, 31, 37, 40, 43, 47, 54, 57, 59, 61,
Med, 45	63, 81–83
PosteriorMed, 58	MCMed, 8, 12, 16, 19, 22, 25, 26, 29, 33, 35, 40,
Trajectory, 83	43, 47, 54, 57, 59, 61, 63, 81–83
* uncertainty	MCPhi, 8, 12, 16, 19, 22, 25, 26, 29, 33, 37, 39,
DeltaBeta, 6	43, 47, 54, 57, 59, 61, 63, 81–83
DeltaIndirectCentral, 10	MCTotalCentral, 8, 12, 16, 19, 22, 25, 26, 29,
DeltaMed, 13	33, 37, 40, 41, 47, 54, 57, 59, 61, 63,
DeltaTotalCentral, 17	81–83
MCBeta, 27	Med, 8, 12, 16, 19, 22, 25, 26, 29, 33, 37, 40,
MCIndirectCentral, 31	43, 45, 54, 57, 59, 61, 63, 81–83
MCMed, 35	
MCPhi, 39	plot.ctmeddelta,48
MCTotalCentral, 41	plot.ctmedmc, 49
PosteriorBeta, 53	plot.ctmedmed, 51
PosteriorIndirectCentral, 55	plot.ctmedtraj,52
PosteriorMed, 58	PosteriorBeta, <i>8</i> , <i>12</i> , <i>16</i> , <i>19</i> , <i>22</i> , <i>25</i> , <i>26</i> , <i>29</i> ,
PosteriorPhi, 61	33, 37, 40, 43, 47, 53, 57, 59, 61, 63,
PosteriorTotalCentral, 62	81–83
confint.ctmeddelta,2	PosteriorIndirectCentral, 8, 12, 16, 19,
confint.ctmedmc, 4	22, 25, 26, 29, 33, 37, 40, 43, 47, 54,
ctsem::ctStanFit(), 61	55, 59, 61, 63, 81–83
	PosteriorMed, 8, 12, 16, 19, 22, 25, 26, 29,
DeltaBeta, 6, 12, 16, 19, 22, 25, 26, 29, 33,	33, 37, 40, 43, 47, 54, 57, 58, 61, 63,
37, 40, 43, 47, 54, 57, 59, 61, 63,	81–83
81–83	PosteriorPhi, 8, 12, 16, 19, 22, 25, 26, 29,
DeltaIndirectCentral, 8, 10, 16, 19, 22, 25,	33, 37, 40, 43, 47, 54, 57, 59, 61, 63,
26, 29, 33, 37, 40, 43, 47, 54, 57, 59,	81–83
61, 63, 81–83	PosteriorTotalCentral, <i>8</i> , <i>12</i> , <i>16</i> , <i>19</i> , <i>22</i> ,
DeltaMed, 8, 12, 13, 19, 22, 25, 26, 29, 33, 37, 40, 43, 47, 54, 57, 59, 61, 63, 81-83	25, 26, 29, 33, 37, 40, 43, 47, 54, 57,
40, 43, 47, 34, 37, 39, 61, 63, 61–63 DeltaTotalCentral, 8, 12, 16, 17, 22, 25, 26,	59, 61, 62, 81–83
29, 33, 37, 40, 43, 47, 54, 57, 59, 61,	print.ctmeddelta,64
63, 81–83	print.ctmedeffect,66
Direct, 8, 12, 16, 19, 21, 25, 26, 29, 33, 37,	print.ctmedmc,68
40, 43, 47, 54, 57, 59, 61, 63, 81–83	print.ctmedmcphi,69
Direct(), 14, 36, 45, 59	print.ctmedmed,71
D11 CCt(), 14, 50, 45, 5)	print.ctmedtraj,72
Indirect, 8, 12, 16, 19, 22, 23, 26, 29, 33, 37,	
40, 43, 47, 54, 57, 59, 61, 63, 81–83	summary.ctmeddelta,73
Indirect(), 14, 36, 45, 59	summary.ctmedmc,75
IndirectCentral, 8, 12, 16, 19, 22, 25, 26,	summary.ctmedmed,76
29, 33, 37, 40, 43, 47, 54, 57, 59, 61,	$\operatorname{summary.ctmedposteriorphi}, 78$
63, 81–83	summary.ctmedtraj,78

INDEX 87

```
Total, 8, 12, 16, 19, 22, 25, 26, 29, 33, 37, 40, 43, 47, 54, 57, 59, 61, 63, 79, 82, 83  
Total(), 7, 14, 28, 36, 45, 53, 59  
TotalCentral, 8, 12, 16, 19, 22, 25, 26, 29, 33, 37, 40, 43, 47, 54, 57, 59, 61, 63, 81, 81, 83  
TotalCentral(), 18, 42, 56, 62  
Trajectory, 8, 12, 16, 19, 22, 25, 26, 29, 33, 37, 40, 43, 47, 54, 57, 59, 61, 63, 81, 82, 83
```