

Package ‘cTMed’

March 22, 2024

Title Continuous Time Mediation

Version 0.0.0.9000

Description Calculates standard errors and confidence intervals
for the indirect effect in continuous time mediation models.

URL <https://github.com/jeksterslab/cTMed>,
<https://jeksterslab.github.io/cTMed/>

BugReports <https://github.com/jeksterslab/cTMed/issues>

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

Depends R (>= 3.5.0)

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp, numDeriv, parallel

Suggests knitr, rmarkdown, testthat, simStateSpace, expm

RoxygenNote 7.3.1

NeedsCompilation yes

Author Ivan Jacob Agaloos Pesigan [aut, cre, cph]
(<https://orcid.org/0000-0003-4818-8420>)

Maintainer Ivan Jacob Agaloos Pesigan <r.jeksterslab@gmail.com>

R topics documented:

confint.ctmeddelta	2
confint.ctmedmc	3
deboeck2015	4
deboeck2015phi	6
DeltaMed	7
Direct	10

Indirect	12
MCMed	15
MCPHi	18
Med	20
plot.ctmeddelta	23
plot.ctmedmc	24
plot.ctmedmed	25
PosteriorMed	26
print.ctmeddelta	28
print.ctmedeffect	29
print.ctmedmc	31
print.ctmedmcphi	32
print.ctmedmed	33
summary.ctmeddelta	35
summary.ctmedmc	36
summary.ctmedmed	37
TestPhi	39
TestStable	40
Total	41
Index	44

confint.ctmeddelta	<i>Delta Method Confidence Intervals</i>
--------------------	--

Description

Delta Method Confidence Intervals

Usage

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

Arguments

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

Value

Returns a matrix of confidence intervals.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```

data("deboeck2015phi", package = "cTMed")
phi <- deboeck2015phi$dynr$phi
vcov_phi_vec <- deboeck2015phi$dynr$vcov

# 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(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1:20,
  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.

<code>parm</code>	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.
<code>level</code>	the confidence level required.
<code>...</code>	additional arguments.

Value

Returns a matrix of confidence intervals.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
data("deboeck2015phi", package = "cTMed")
phi <- deboeck2015phi$dynr$phi
vcov_phi_vec <- deboeck2015phi$dynr$vcov

# Specific time-interval -----
mc <- MCMed(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1,
  from = "x",
  to = "y",
  med = "m",
  R = 5 # use a large value for R in actual research
)
confint(mc)

# Range of time-intervals -----
mc <- MCMed(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1:20,
  from = "x",
  to = "y",
  med = "m",
  R = 5 # use a large value for R in actual research
)
confint(mc)
```

Description

The data was simulated using `simStateSpace::SimSSMVARFixed()` from a discrete-time vector autoregressive model given by

Usage

deboeck2015

Format

Dataframe with Five Columns:

id Individual ID.

time Time variable.

x X variable.

m M variable.

y Y variable.

Details

$$\mathbf{y}_{i,t} = \beta \mathbf{y}_{i,t-1} + \boldsymbol{\varepsilon}_{i,t}$$

where $\mathbf{y}_{i,t}$ and $\mathbf{y}_{i,t+1}$ represents a vector of observed variables X , M , and Y for individual i at time t and $t + 1$, $\boldsymbol{\varepsilon}_{i,t}$ a vector of normally distributed random noise with mean vector of zero and covariance matrix Ψ given by

$$\Psi = \begin{pmatrix} 0.10 & 0 & 0 \\ 0 & 0.10 & 0 \\ 0 & 0 & 0.10 \end{pmatrix}, \quad \text{and}$$

β is a matrix of lagged parameters given by

$$\beta = \begin{pmatrix} 0.70 & 0 & 0 \\ 0.50 & 0.60 & 0 \\ -0.10 & 0.40 & 0.50 \end{pmatrix}.$$

The mean vector $\boldsymbol{\mu}_0$ and covariance matrix Σ_0 of the initial condition are given by

$$\boldsymbol{\mu}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \text{and}$$

$$\Sigma_0 = \begin{pmatrix} 1 & 0.20 & 0.20 \\ 0.20 & 1 & 0.20 \\ 0.20 & 0.20 & 1 \end{pmatrix}.$$

References

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](https://doi.org/10.1080/10705511.2014.973960)

deboeck2015phi

Drift Matrix

Description

Parameter estimates and sampling variance-covariance matrix of the continuous-time vector autoregressive model drift matrix using the data set deboeck2015. The model was fitted using the dynr package.

Usage

deboeck2015phi

Format

List with Two Elements:

dynr Results using the dynr package.

ctsem Results using the ctsem package.

The dynr element is a list with the following elements:

phi The estimated drift matrix Φ .

vcov The estimated sampling variance-covariance matrix of $\text{vec}(\Phi)$.

The ctsem element is a list with the following elements:

posterior Posterior distribution.

posterior_phi Posterior distribution of the drift matrix Φ .

phi Posterior mean of the drift matrix Φ .

vcov Posterior variance-covariance matrix of $\text{vec}(\Phi)$.

References

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](https://doi.org/10.1080/10705511.2014.973960)

Ou, L., Hunter, M. D., & Chow, S.-M. (2019). What's for dynr: A package for linear and nonlinear dynamic modeling in R. *The R Journal*, 11(1), 91. [doi:10.32614/rj2019012](https://doi.org/10.32614/rj2019012)

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 Δt or a range of time-intervals using the first-order stochastic differential equation model's drift matrix Φ .

Usage

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

Arguments

phi	Numeric matrix. The drift matrix (Φ). phi should have row and column names pertaining to the variables in the system.
vcov_phi_vec	Numeric matrix. The sampling variance-covariance matrix of $\text{vec}(\Phi)$.
delta_t	Vector of positive numbers. Time interval (Δt).
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 θ be $\text{vec}(\Phi)$, that is, the elements of the Φ 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:

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

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

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

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

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

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

Linear Stochastic Differential Equation Model:

The measurement model is given by

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

where $\mathbf{y}_{i,t}$, $\boldsymbol{\eta}_{i,t}$, and $\boldsymbol{\varepsilon}_{i,t}$ are random variables and $\boldsymbol{\nu}$, $\mathbf{\Lambda}$, 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, $\mathbf{\Lambda}$ a matrix of factor loadings, and $\boldsymbol{\Theta}$ the covariance matrix of $\boldsymbol{\varepsilon}$.

An alternative representation of the measurement error is given by

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

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

The dynamic structure is given by

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

where $\boldsymbol{\iota}$ is a term which is unobserved and constant over time, $\boldsymbol{\Phi}$ is the drift matrix which represents the rate of change of the solution in the absence of any random fluctuations, $\boldsymbol{\Sigma}$ is the matrix of volatility or randomness in the process, and $d\mathbf{W}$ 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: [Direct\(\)](#), [Indirect\(\)](#), [MCMed\(\)](#), [MCPhi\(\)](#), [Med\(\)](#), [PosteriorMed\(\)](#), [TestPhi\(\)](#), [TestStable\(\)](#), [Total\(\)](#)

Examples

```
data("deboeck2015phi", package = "cTMed")
phi <- deboeck2015phi$dynr$phi
vcov_phi_vec <- deboeck2015phi$dynr$vcov

# Specific time-interval -----
DeltaMed(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1,
  from = "x",
  to = "y",
  med = "m"
)

# Range of time-intervals -----
delta <- DeltaMed(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1:20,
  from = "x",
  to = "y",
  med = "m"
)

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

```
plot(delta)
```

Direct

Direct Effect of X on Y Over a Specific Time-Interval

Description

This function computes the direct effect of the independent variable X on the dependent variable Y through mediator variables \mathbf{m} over a specific time-interval Δt using the first-order stochastic differential equation model's drift matrix Φ .

Usage

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

Arguments

phi	Numeric matrix. The drift matrix (Φ). phi should have row and column names pertaining to the variables in the system.
delta_t	Numeric. Time interval (Δt).
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

$$\text{Direct}_{\Delta t, i, j, \mathbf{m}} = \exp(\Delta t \mathbf{D} \Phi \mathbf{D})_{i, j}$$

where Φ 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 Φ , j the column index of X in Φ , and Δt the time-interval.

Linear Stochastic Differential Equation Model:

The measurement model is given by

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

where $\mathbf{y}_{i,t}$, $\boldsymbol{\eta}_{i,t}$, and $\boldsymbol{\varepsilon}_{i,t}$ are random variables and $\boldsymbol{\nu}$, $\mathbf{\Lambda}$, 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, $\mathbf{\Lambda}$ a matrix of factor loadings, and $\boldsymbol{\Theta}$ the covariance matrix of $\boldsymbol{\varepsilon}$.

An alternative representation of the measurement error is given by

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

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

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

where $\boldsymbol{\iota}$ is a term which is unobserved and constant over time, $\boldsymbol{\Phi}$ is the drift matrix which represents the rate of change of the solution in the absence of any random fluctuations, $\boldsymbol{\Sigma}$ is the matrix of volatility or randomness in the process, and $d\mathbf{W}$ 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](https://doi.org/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](https://doi.org/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](https://doi.org/10.1007/s11336021097670)

See Also

Other Continuous Time Mediation Functions: [DeltaMed\(\)](#), [Indirect\(\)](#), [MCMed\(\)](#), [MCPPhi\(\)](#), [Med\(\)](#), [PosteriorMed\(\)](#), [TestPhi\(\)](#), [TestStable\(\)](#), [Total\(\)](#)

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")
```

```

delta_t <- 1
Direct(
  phi = phi,
  delta_t = delta_t,
  from = "x",
  to = "y",
  med = "m"
)
phi <- matrix(
  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)
Direct(
  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 Δt using the first-order stochastic differential equation model's drift matrix Φ .

Usage

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

Arguments

<code>phi</code>	Numeric matrix. The drift matrix (Φ). <code>phi</code> should have row and column names pertaining to the variables in the system.
<code>delta_t</code>	Numeric. Time interval (Δt).
<code>from</code>	Character string. Name of the independent variable X in <code>phi</code> .
<code>to</code>	Character string. Name of the dependent variable Y in <code>phi</code> .
<code>med</code>	Character vector. Name/s of the mediator variable/s in <code>phi</code> .

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 Δt is given by

$$\text{Indirect}_{\Delta t} = \exp(\Delta t \Phi)_{i,j} - \exp(\Delta t \mathbf{D}_m \Phi \mathbf{D}_m)_{i,j}$$

where Φ 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 Φ , j the column index of X in Φ , and Δt the time-interval.

Linear Stochastic Differential Equation Model:

The measurement model is given by

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

where $\mathbf{y}_{i,t}$, $\boldsymbol{\eta}_{i,t}$, and $\boldsymbol{\varepsilon}_{i,t}$ are random variables and $\boldsymbol{\nu}$, $\mathbf{\Lambda}$, 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, $\mathbf{\Lambda}$ a matrix of factor loadings, and $\boldsymbol{\Theta}$ the covariance matrix of $\boldsymbol{\varepsilon}$.

An alternative representation of the measurement error is given by

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

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

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

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

Value

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

call Function call.

args Function arguments.

fun Function used ("Indirect").

output The indirect effect.

Author(s)

Ivan Jacob Agaloos Pesigan

References

- Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. *Sociological Methodology*, 17, 37. [doi:10.2307/271028](https://doi.org/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](https://doi.org/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](https://doi.org/10.1007/s11336021097670)

See Also

Other Continuous Time Mediation Functions: [DeltaMed\(\)](#), [Direct\(\)](#), [MCMed\(\)](#), [MCPHi\(\)](#), [Med\(\)](#), [PosteriorMed\(\)](#), [TestPhi\(\)](#), [TestStable\(\)](#), [Total\(\)](#)

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")
delta_t <- 1
Indirect(
  phi = phi,
  delta_t = delta_t,
  from = "x",
  to = "y",
  med = "m"
)
phi <- matrix(
  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)
Indirect(
  phi = phi,
  delta_t = delta_t,
  from = "y2",
  to = "y4",
  med = c("y1", "y3")
)
```

MCMed

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

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} at a particular time-interval Δt using the first-order stochastic differential equation model drift matrix Φ .

Usage

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

Arguments

phi	Numeric matrix. The drift matrix (Φ). phi should have row and column names pertaining to the variables in the system.
vcov_phi_vec	Numeric matrix. The sampling variance-covariance matrix of $\text{vec}(\Phi)$.
delta_t	Numeric. Time interval (Δt).
from	Character string. Name of the independent variable X in phi.
to	Character string. Name of the dependent variable Y in phi.
med	Character vector. Name/s of the mediator variable/s in phi.
R	Positive integer. Number of replications.
test_phi	Logical. If test_phi = TRUE, the function runs TestPhi() on the generated drift matrix Φ . If the TestPhi() returns FALSE, the function generates a new drift matrix Φ and runs the test recursively until TestPhi() returns TRUE.
ncores	Positive integer. Number of cores to use. If ncores = NULL, use a single core. Consider using multiple cores when number of replications R is a large value.
seed	Random seed.

Details

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

Monte Carlo Method:

Let θ be $\text{vec}(\Phi)$, that is, the elements of the Φ 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.

$$\hat{\theta} \sim \mathcal{N}(\theta, \mathbb{V}(\hat{\theta}))$$

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,

$$\hat{\theta}^* \sim \mathcal{N}(\hat{\theta}, \hat{\mathbb{V}}(\hat{\theta})).$$

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

Linear Stochastic Differential Equation Model:

The measurement model is given by

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

where $\mathbf{y}_{i,t}$, $\boldsymbol{\eta}_{i,t}$, and $\boldsymbol{\varepsilon}_{i,t}$ are random variables and $\boldsymbol{\nu}$, $\mathbf{\Lambda}$, 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, $\mathbf{\Lambda}$ a matrix of factor loadings, and $\boldsymbol{\Theta}$ the covariance matrix of $\boldsymbol{\varepsilon}$.

An alternative representation of the measurement error is given by

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

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

The dynamic structure is given by

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

where $\boldsymbol{\iota}$ is a term which is unobserved and constant over time, $\boldsymbol{\Phi}$ is the drift matrix which represents the rate of change of the solution in the absence of any random fluctuations, $\boldsymbol{\Sigma}$ is the matrix of volatility or randomness in the process, and $d\mathbf{W}$ 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 (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](https://doi.org/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](https://doi.org/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](https://doi.org/10.1007/s11336021097670)

See Also

Other Continuous Time Mediation Functions: [DeltaMed\(\)](#), [Direct\(\)](#), [Indirect\(\)](#), [MCPHI\(\)](#), [Med\(\)](#), [PosteriorMed\(\)](#), [TestPhi\(\)](#), [TestStable\(\)](#), [Total\(\)](#)

Examples

```
data("deboeck2015phi", package = "cTMed")
phi <- deboeck2015phi$dynr$phi
vcov_phi_vec <- deboeck2015phi$dynr$vcov

# Specific time-interval -----
MCMed(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1,
  from = "x",
  to = "y",
  med = "m",
  R = 5 # use a large value for R in actual research
)

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

```

    delta_t = 1:20,
    from = "x",
    to = "y",
    med = "m",
    R = 5 # use a large value for R in actual research
)

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

```

MCPhi

*Generate Random Drift Matrices Using the Monte Carlo Method***Description**

This function generates random drift matrices Φ 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 (Φ). phi should have row and column names pertaining to the variables in the system.
vcov_phi_vec	Numeric matrix. The sampling variance-covariance matrix of $\text{vec}(\Phi)$.
R	Positive integer. Number of replications.
test_phi	Logical. If test_phi = TRUE, the function runs <code>TestPhi()</code> on the generated drift matrix Φ . If the <code>TestPhi()</code> returns FALSE, the function generates a new drift matrix Φ and runs the test recursively until <code>TestPhi()</code> 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 θ be $\text{vec}(\Phi)$, that is, the elements of the Φ 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.

$$\hat{\theta} \sim \mathcal{N}(\theta, \mathbb{V}(\hat{\theta}))$$

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,

$$\hat{\theta}^* \sim \mathcal{N}\left(\hat{\theta}, \hat{\mathbf{V}}\left(\hat{\theta}\right)\right).$$

Linear Stochastic Differential Equation Model:

The measurement model is given by

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

where $\mathbf{y}_{i,t}$, $\boldsymbol{\eta}_{i,t}$, and $\boldsymbol{\varepsilon}_{i,t}$ are random variables and $\boldsymbol{\nu}$, $\mathbf{\Lambda}$, 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, $\mathbf{\Lambda}$ a matrix of factor loadings, and $\boldsymbol{\Theta}$ the covariance matrix of $\boldsymbol{\varepsilon}$.

An alternative representation of the measurement error is given by

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

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

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

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

Value

Returns a list of simulated drift matrices.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Continuous Time Mediation Functions: [DeltaMed\(\)](#), [Direct\(\)](#), [Indirect\(\)](#), [MCMed\(\)](#), [Med\(\)](#), [PosteriorMed\(\)](#), [TestPhi\(\)](#), [TestStable\(\)](#), [Total\(\)](#)

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")
MCPhi(
  phi = phi,
  vcov_phi_vec = 0.1 * diag(9),
  R = 5
)
phi <- matrix(
  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)
MCPhi(
  phi = phi,
  vcov_phi_vec = 0.1 * diag(16),
  R = 5,
  test_phi = FALSE
)

```

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 Δt or a range of time-intervals using the first-order stochastic differential equation model's drift matrix Φ .

Usage

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

Arguments

phi	Numeric matrix. The drift matrix (Φ). phi should have row and column names pertaining to the variables in the system.
delta_t	Vector of positive numbers. Time interval (Δt).
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.

Linear Stochastic Differential Equation Model:

The measurement model is given by

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

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

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

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

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

where $\boldsymbol{\iota}$ is a term which is unobserved and constant over time, $\boldsymbol{\Phi}$ is the drift matrix which represents the rate of change of the solution in the absence of any random fluctuations, $\boldsymbol{\Sigma}$ is the matrix of volatility or randomness in the process, and $d\mathbf{W}$ 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](https://doi.org/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](https://doi.org/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](https://doi.org/10.1007/s11336021097670)

See Also

Other Continuous Time Mediation Functions: [DeltaMed\(\)](#), [Direct\(\)](#), [Indirect\(\)](#), [MCMed\(\)](#), [MCPhi\(\)](#), [PosteriorMed\(\)](#), [TestPhi\(\)](#), [TestStable\(\)](#), [Total\(\)](#)

Examples

```
# -----
# Example 1 -----
# -----
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")

# 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 = 1:20,
  from = "x",
  to = "y",
  med = "m"
)
plot(med)

# -----
# Example 2 -----
# -----
phi <- matrix(
  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)
```

```

# Specific time-interval -----
Med(
  phi = phi,
  delta_t = 1,
  from = "y2",
  to = "y4",
  med = c("y1", "y3")
)

# Range of time-intervals -----
med <- Med(
  phi = phi,
  delta_t = seq(from = 0, to = 5, length.out = 500),
  from = "y2",
  to = "y4",
  med = c("y1", "y3")
)

# Methods -----
# Med has a number of methods including
# print, summary, and plot
print(med)
summary(med)
plot(med)

```

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, ...)
```

Arguments

x	Object of class ctmeddelta.
alpha	Numeric. Significance level
...	Additional arguments.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```

data("deboeck2015phi", package = "cTMed")
phi <- deboeck2015phi$dynr$phi
vcov_phi_vec <- deboeck2015phi$dynr$vcov

# Range of time-intervals -----
delta <- DeltaMed(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1:20,
  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, ...)

```

Arguments

x	Object of class ctmedmc.
alpha	Numeric. Significance level
...	Additional arguments.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```

data("deboeck2015phi", package = "cTMed")
phi <- deboeck2015phi$dynr$phi
vcov_phi_vec <- deboeck2015phi$dynr$vcov

# Range of time-intervals -----
mc <- MCMed(
  phi = phi,

```



```

vcov_phi_vec = vcov_phi_vec,
delta_t = 1:20,
from = "x",
to = "y",
med = "m",
R = 5 # 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, ...)

```

Arguments

x	Object of class ctmedmed.
...	Additional arguments.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```

# -----
# Example 1 -----
# -----
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 = 1:20,
    from = "x",
    to = "y",
    med = "m"
  )
plot(med)

# -----
# Example 2 -----
# -----
phi <- matrix(
  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)

# Range of time-intervals -----
med <- Med(
  phi = phi,
  delta_t = seq(from = 0, to = 5, length.out = 500),
  from = "y2",
  to = "y4",
  med = c("y1", "y3")
)
plot(med)

```

PosteriorMed

*Posterior Distribution of Total, Direct, and Indirect Effects of X on Y
Through M Over a Specific Time-Interval*

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} at a particular time-interval Δt using the posterior distribution of the first-order stochastic differential equation model drift matrix Φ .

Usage

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

Arguments

<code>phi</code>	List of numeric matrices. Each element of the list is a sample from the posterior distribution of the drift matrix (Φ). Each matrix should have row and column names pertaining to the variables in the system.
<code>delta_t</code>	Numeric. Time interval (Δt).
<code>from</code>	Character string. Name of the independent variable X in <code>phi</code> .
<code>to</code>	Character string. Name of the dependent variable Y in <code>phi</code> .
<code>med</code>	Character vector. Name/s of the mediator variable/s in <code>phi</code> .
<code>ncores</code>	Positive integer. Number of cores to use. If <code>ncores = NULL</code> , 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: `DeltaMed()`, `Direct()`, `Indirect()`, `MCMed()`, `MCPHi()`, `Med()`, `TestPhi()`, `TestStable()`, `Total()`

Examples

```

data("deboeck2015phi", package = "cTMed")
phi <- deboeck2015phi$ctsem$posterior_phi

# Specific time-interval -----
PosteriorMed(
  phi = phi,
  delta_t = 1,
  from = "x",
  to = "y",
  med = "m"
)

# Range of time-intervals -----
posterior <- PosteriorMed(
  phi = phi,
  delta_t = 1:20,
  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)

```

print.ctmeddelta	<i>Print Method for Object of Class ctmeddelta</i>
------------------	--

Description

Print Method for Object of Class ctmeddelta

Usage

```

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

```

Arguments

x	an object of class ctmeddelta.
alpha	Numeric vector. Significance level α .
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

Examples

```
data("deboeck2015phi", package = "cTMed")
phi <- deboeck2015phi$dynr$phi
vcov_phi_vec <- deboeck2015phi$dynr$vcov

# Specific time-interval -----
delta <- DeltaMed(
  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(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1:20,
  from = "x",
  to = "y",
  med = "m"
)
print(delta)
```

print.ctmedeffect	<i>Print Method for Object of Class ctmedeffect</i>
-------------------	---

Description

Print Method for Object of Class ctmedeffect

Usage

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

Arguments

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

Value

Returns the effects.

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")
delta_t <- 1

# Time-Interval of One -----

## Total Effect -----
total_dt <- Total(
  phi = phi,
  delta_t = delta_t
)
print(total_dt)

## Direct Effect -----
direct_dt <- Direct(
  phi = phi,
  delta_t = delta_t,
  from = "x",
  to = "y",
  med = "m"
)
print(direct_dt)

## Indirect Effect -----
indirect_dt <- Indirect(
  phi = phi,
  delta_t = delta_t,
  from = "x",
  to = "y",

```

```

    med = "m"
  )
  print(indirect_dt)

```

print.ctmedmc	<i>Print Method for Object of Class ctmedmc</i>
---------------	---

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 α .
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

Examples

```

data("deboeck2015phi", package = "cTMed")
phi <- deboeck2015phi$dynr$phi
vcov_phi_vec <- deboeck2015phi$dynr$vcov

# Specific time-interval -----
mc <- MCMed(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1,
  from = "x",
  to = "y",
  med = "m",
  R = 5 # use a large value for R in actual research

```

```

)
print(mc)

# Range of time-intervals -----
mc <- MCMed(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1:20,
  from = "x",
  to = "y",
  med = "m",
  R = 5 # use a large value for R in actual research
)
print(mc)

```

print.ctmedmcphi	<i>Print Method for Object of Class ctmedmcphi</i>
------------------	--

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

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")
mc <- MCPHi(
  phi = phi,
  vcov_phi_vec = 0.1 * diag(9),
  R = 5
)
print(mc)

```

print.ctmedmed	<i>Print Method for Object of Class ctmedmed</i>
----------------	--

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

Examples

```

# -----
# Example 1 -----
# -----
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")

# Specific time-interval -----
med <- Med(
  phi = phi,
  delta_t = 1,
  from = "x",
  to = "y",
  med = "m"
)
print(med)

# Range of time-intervals -----
med <- Med(
  phi = phi,
  delta_t = 1:20,
  from = "x",
  to = "y",
  med = "m"
)
print(med)

# -----
# Example 2 -----
# -----
phi <- matrix(
  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)

# Specific time-interval -----
med <- Med(
  phi = phi,
  delta_t = 1,

```

```

    from = "y2",
    to = "y4",
    med = c("y1", "y3")
  )
  print(med)

# Range of time-intervals -----
med <- Med(
  phi = phi,
  delta_t = 1:10,
  from = "y2",
  to = "y4",
  med = c("y1", "y3")
)
print(med)

```

summary.ctmeddelta	<i>Summary Method for an Object of Class ctmeddelta</i>
--------------------	---

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 α .
...	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

Examples

```
data("deboeck2015phi", package = "cTMed")
phi <- deboeck2015phi$dynr$phi
vcov_phi_vec <- deboeck2015phi$dynr$vcov

# Specific time-interval -----
delta <- DeltaMed(
  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(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1:20,
  from = "x",
  to = "y",
  med = "m"
)
summary(delta)
```

summary.ctmedmc

Summary Method for an Object of Class ctmedmc

Description

Summary Method for an Object of Class ctmedmc

Usage

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

Arguments

object	Object of class ctmedmc.
alpha	Numeric vector. Significance level α .
...	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

Examples

```
data("deboeck2015phi", package = "cTMed")
phi <- deboeck2015phi$dynr$phi
vcov_phi_vec <- deboeck2015phi$dynr$vcov

# Specific time-interval -----
mc <- MCMed(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1,
  from = "x",
  to = "y",
  med = "m",
  R = 5 # use a large value for R in actual research
)
summary(mc)

# Range of time-intervals -----
mc <- MCMed(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1:20,
  from = "x",
  to = "y",
  med = "m",
  R = 5 # use a large value for R in actual research
)
summary(mc)
```

summary.ctmedmed

Summary Method for an Object of Class ctmedmed

Description

Summary Method for an Object of Class ctmedmed

Usage

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

Arguments

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

Value

Returns a matrix of effects.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
# -----
# Example 1 -----
# -----
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")

# 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 = 1:20,
  from = "x",
  to = "y",
  med = "m"
)
summary(med)

# -----
# Example 2 -----
# -----
```

```

phi <- matrix(
  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)

# Specific time-interval -----
med <- Med(
  phi = phi,
  delta_t = 1,
  from = "y2",
  to = "y4",
  med = c("y1", "y3")
)
summary(med)

# Range of time-intervals -----
med <- Med(
  phi = phi,
  delta_t = 1:10,
  from = "y2",
  to = "y4",
  med = c("y1", "y3")
)
summary(med)

```

TestPhi

Test the Drift Matrix

Description

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

- Test that the real part of all eigenvalues of Φ is less than zero.
- Test that the diagonal values of Φ are between 0 to negative infinity.

Usage

```
TestPhi(phi)
```

Arguments

phi Numeric matrix. The drift matrix (Φ).

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Continuous Time Mediation Functions: [DeltaMed\(\)](#), [Direct\(\)](#), [Indirect\(\)](#), [MCMed\(\)](#), [MCPPhi\(\)](#), [Med\(\)](#), [PosteriorMed\(\)](#), [TestStable\(\)](#), [Total\(\)](#)

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")
TestPhi(phi = phi)
phi <- matrix(
  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)
TestPhi(phi = phi)
```

TestStable

Test Stability

Description

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

Usage

```
TestStable(x)
```

Arguments

`x` Numeric matrix.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Continuous Time Mediation Functions: [DeltaMed\(\)](#), [Direct\(\)](#), [Indirect\(\)](#), [MCMed\(\)](#), [MCPHi\(\)](#), [Med\(\)](#), [PosteriorMed\(\)](#), [TestPhi\(\)](#), [Total\(\)](#)

Examples

```
x <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
)
TestStable(x)
x <- matrix(
  data = c(
    -6, 5.5, 0, 0,
    1.25, -2.5, 5.9, -7.3,
    0, 0, -6, 2.5,
    5, 0, 0, -6
  ),
  nrow = 4
)
TestStable(x)
```

Total	<i>Total Effect Matrix Over a Specific Time-Interval</i>
-------	--

Description

This function computes the total effects matrix over a specific time-interval Δt using the first-order stochastic differential equation model’s drift matrix Φ .

Usage

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

Arguments

- phi Numeric matrix. The drift matrix (Φ). phi should have row and column names pertaining to the variables in the system.
- delta_t Numeric. Time interval (Δt).

Details

The total effect matrix over a specific time-interval Δt is given by

$$\text{Total}_{\Delta t} = \exp(\Delta t \Phi)$$

where Φ denotes the drift matrix, and Δt the time-interval.

Linear Stochastic Differential Equation Model:

The measurement model is given by

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

where $\mathbf{y}_{i,t}$, $\boldsymbol{\eta}_{i,t}$, and $\boldsymbol{\varepsilon}_{i,t}$ are random variables and $\boldsymbol{\nu}$, $\mathbf{\Lambda}$, 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, $\mathbf{\Lambda}$ a matrix of factor loadings, and $\boldsymbol{\Theta}$ the covariance matrix of $\boldsymbol{\varepsilon}$.

An alternative representation of the measurement error is given by

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

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

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

where $\boldsymbol{\iota}$ is a term which is unobserved and constant over time, Φ is the drift matrix which represents the rate of change of the solution in the absence of any random fluctuations, $\boldsymbol{\Sigma}$ is the matrix of volatility or randomness in the process, and $d\mathbf{W}$ 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](https://doi.org/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](https://doi.org/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](https://doi.org/10.1007/s11336021097670)

See Also

Other Continuous Time Mediation Functions: [DeltaMed\(\)](#), [Direct\(\)](#), [Indirect\(\)](#), [MCMed\(\)](#), [MCPPhi\(\)](#), [Med\(\)](#), [PosteriorMed\(\)](#), [TestPhi\(\)](#), [TestStable\(\)](#)

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")
delta_t <- 1
Total(
  phi = phi,
  delta_t = delta_t
)
phi <- matrix(
  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)
Total(
  phi = phi,
  delta_t = delta_t
)
```

Index

* Continuous Time Mediation Functions

- DeltaMed, 7
- Direct, 10
- Indirect, 12
- MCMed, 15
- MCPhi, 18
- Med, 20
- PosteriorMed, 26
- TestPhi, 39
- TestStable, 40
- Total, 41

* cTMed

- DeltaMed, 7
- Direct, 10
- Indirect, 12
- MCMed, 15
- MCPhi, 18
- Med, 20
- PosteriorMed, 26
- TestPhi, 39
- TestStable, 40
- Total, 41

* data

- deboeck2015, 4
- deboeck2015phi, 6

* effects

- Direct, 10
- Indirect, 12
- Med, 20
- Total, 41

* main

- DeltaMed, 7
- MCMed, 15
- Med, 20

* methods

- confint.ctmeddelta, 2
- confint.ctmedmc, 3
- plot.ctmeddelta, 23
- plot.ctmedmc, 24

- plot.ctmedmed, 25
- print.ctmeddelta, 28
- print.ctmedeffect, 29
- print.ctmedmc, 31
- print.ctmedmcphi, 32
- print.ctmedmed, 33
- summary.ctmeddelta, 35
- summary.ctmedmc, 36
- summary.ctmedmed, 37

* test

- TestPhi, 39
- TestStable, 40

* uncertainty

- DeltaMed, 7
- MCMed, 15
- MCPhi, 18
- PosteriorMed, 26

- confint.ctmeddelta, 2
- confint.ctmedmc, 3

- deboeck2015, 4
- deboeck2015phi, 6
- DeltaMed, 7, 11, 14, 17, 19, 22, 27, 40, 41, 43
- Direct, 9, 10, 14, 17, 19, 22, 27, 40, 41, 43
- Direct(), 7, 16, 21, 27

- Indirect, 9, 11, 12, 17, 19, 22, 27, 40, 41, 43
- Indirect(), 7, 16, 21, 27

- MCMed, 9, 11, 14, 15, 19, 22, 27, 40, 41, 43
- MCPhi, 9, 11, 14, 17, 18, 22, 27, 40, 41, 43
- Med, 9, 11, 14, 17, 19, 20, 27, 40, 41, 43

- plot.ctmeddelta, 23
- plot.ctmedmc, 24
- plot.ctmedmed, 25
- PosteriorMed, 9, 11, 14, 17, 19, 22, 26, 40, 41, 43
- print.ctmeddelta, 28
- print.ctmedeffect, 29

`print.ctmedmc`, [31](#)
`print.ctmedmcphi`, [32](#)
`print.ctmedmed`, [33](#)

`simStateSpace::SimSSMVARFixed()`, [4](#)
`summary.ctmeddelta`, [35](#)
`summary.ctmedmc`, [36](#)
`summary.ctmedmed`, [37](#)

`TestPhi`, [9](#), [11](#), [14](#), [17](#), [19](#), [22](#), [27](#), [39](#), [41](#), [43](#)
`TestPhi()`, [15](#), [18](#)
`TestStable`, [9](#), [11](#), [14](#), [17](#), [19](#), [22](#), [27](#), [40](#), [40](#),
[43](#)
`Total`, [9](#), [11](#), [14](#), [17](#), [19](#), [22](#), [27](#), [40](#), [41](#), [41](#)
`Total()`, [7](#), [16](#), [21](#), [27](#)