

Package ‘cTMed’

November 22, 2024

Title Continuous Time Mediation

Version 1.0.1.9000

Description Calculates standard errors and confidence intervals for effects in continuous-time mediation models. This package extends the work of Deboeck and Preacher (2015) <[doi:10.1080/10705511.2014.973960](https://doi.org/10.1080/10705511.2014.973960)> and Ryan and Hamaker (2021) <[doi:10.1007/s11336-021-09767-0](https://doi.org/10.1007/s11336-021-09767-0)> by providing methods to generate standard errors and confidence intervals for the total, direct, and indirect effects in these models.

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

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

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends 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]
(<<https://orcid.org/0000-0003-4818-8420>>)

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

Contents

confint.ctmeddelta	3
confint.ctmedmc	4
DeltaBeta	6
DeltaBetaStd	10

DeltaIndirectCentral	14
DeltaMed	17
DeltaMedStd	20
DeltaTotalCentral	24
Direct	27
DirectStd	30
Indirect	32
IndirectCentral	35
IndirectStd	36
MCBeta	39
MCBetaStd	42
MCIndirectCentral	46
MCMed	49
MCMedStd	53
MCPHi	57
MCTotalCentral	59
Med	62
MedStd	65
plot.ctmeddelta	67
plot.ctmedmc	69
plot.ctmedmed	71
plot.ctmedtraj	72
PosteriorBeta	73
PosteriorIndirectCentral	75
PosteriorMed	78
PosteriorPhi	81
PosteriorTotalCentral	82
print.ctmeddelta	84
print.ctmedeffect	86
print.ctmedmc	88
print.ctmedmcphi	90
print.ctmedmed	91
print.ctmedtraj	92
summary.ctmeddelta	93
summary.ctmedmc	95
summary.ctmedmed	96
summary.ctmedposteriorphi	98
summary.ctmedtraj	98
Total	99
TotalCentral	101
TotalStd	103
Trajectory	105

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 data frame of confidence intervals.

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")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,
    -0.00600, -0.00033, 0.00110,
    0.00324, 0.00020, -0.00061,
    0.00040, 0.00374, 0.00016,
    -0.00022, -0.00273, -0.00016,
    0.00009, 0.00150, 0.00012,
```

```

-0.00151, 0.00016, 0.00389,
0.00103, -0.00007, -0.00283,
-0.00050, 0.00000, 0.00156,
-0.00600, -0.00022, 0.00103,
0.00644, 0.00031, -0.00119,
-0.00374, -0.00021, 0.00070,
-0.00033, -0.00273, -0.00007,
0.00031, 0.00287, 0.00013,
-0.00014, -0.00170, -0.00012,
0.00110, -0.00016, -0.00283,
-0.00119, 0.00013, 0.00297,
0.00063, -0.00004, -0.00177,
0.00324, 0.00009, -0.00050,
-0.00374, -0.00014, 0.00063,
0.00495, 0.00024, -0.00093,
0.00020, 0.00150, 0.00000,
-0.00021, -0.00170, -0.00004,
0.00024, 0.00214, 0.00012,
-0.00061, 0.00012, 0.00156,
0.00070, -0.00012, -0.00177,
-0.00093, 0.00012, 0.00223
),
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, level = 0.95)

# Range of time intervals -----
delta <- DeltaMed(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1:5,
  from = "x",
  to = "y",
  med = "m"
)
confint(delta, level = 0.95)

```

Description

Monte Carlo Method Confidence Intervals

Usage

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

Arguments

<code>object</code>	Object of class <code>ctmedmc</code> .
<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 data frame of confidence intervals.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
set.seed(42)
phi <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,
    -0.00600, -0.00033, 0.00110,
    0.00324, 0.00020, -0.00061,
    0.00040, 0.00374, 0.00016,
    -0.00022, -0.00273, -0.00016,
    0.00009, 0.00150, 0.00012,
    -0.00151, 0.00016, 0.00389,
    0.00103, -0.00007, -0.00283,
    -0.00050, 0.00000, 0.00156,
    -0.00600, -0.00022, 0.00103,
    0.00644, 0.00031, -0.00119,
  )
```

```

-0.00374, -0.00021, 0.00070,
-0.00033, -0.00273, -0.00007,
0.00031, 0.00287, 0.00013,
-0.00014, -0.00170, -0.00012,
0.00110, -0.00016, -0.00283,
-0.00119, 0.00013, 0.00297,
0.00063, -0.00004, -0.00177,
0.00324, 0.00009, -0.00050,
-0.00374, -0.00014, 0.00063,
0.00495, 0.00024, -0.00093,
0.00020, 0.00150, 0.00000,
-0.00021, -0.00170, -0.00004,
0.00024, 0.00214, 0.00012,
-0.00061, 0.00012, 0.00156,
0.00070, -0.00012, -0.00177,
-0.00093, 0.00012, 0.00223
),
nrow = 9
)

# Specific time interval -----
mc <- 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
)
confint(mc, level = 0.95)

# Range of time intervals -----
mc <- MCMed(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1:5,
  from = "x",
  to = "y",
  med = "m",
  R = 100L # use a large value for R in actual research
)
confint(mc, level = 0.95)

```

Description

This function computes the delta method sampling variance-covariance matrix for the elements of the matrix of lagged coefficients β 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

```
DeltaBeta(phi, vcov_phi_vec, delta_t, 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).
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 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 g using $\hat{\theta}$ as input can be expressed as:

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

where \mathbf{J} is the matrix of first-order derivatives of the function 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 $g(\hat{\theta})$ as follows:

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

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

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

Value

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

call Function call.

args Function arguments.

fun Function used ("DeltaBeta").

output A list the length of which is equal to the length of `delta_t`.

Each element in the output list has the following elements:

delta_t Time interval.

jacobian Jacobian matrix.

est Estimated elements of the matrix of lagged coefficients.

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

Author(s)

Ivan Jacob Agaloos Pesigan

References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. *Sociological Methodology*, 17, 37. doi:[10.2307/271028](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: [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPHI\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

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")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,
    -0.00600, -0.00033, 0.00110,
    0.00324, 0.00020, -0.00061,
    0.00040, 0.00374, 0.00016,
    -0.00022, -0.00273, -0.00016,
    0.00009, 0.00150, 0.00012,
    -0.00151, 0.00016, 0.00389,
    0.00103, -0.00007, -0.00283,
    -0.00050, 0.00000, 0.00156,
    -0.00600, -0.00022, 0.00103,
    0.00644, 0.00031, -0.00119,
    -0.00374, -0.00021, 0.00070,
    -0.00033, -0.00273, -0.00007,
    0.00031, 0.00287, 0.00013,
    -0.00014, -0.00170, -0.00012,
    0.00110, -0.00016, -0.00283,
    -0.00119, 0.00013, 0.00297,
    0.00063, -0.00004, -0.00177,
    0.00324, 0.00009, -0.00050,
    -0.00374, -0.00014, 0.00063,
    0.00495, 0.00024, -0.00093,
    0.00020, 0.00150, 0.00000,
    -0.00021, -0.00170, -0.00004,
    0.00024, 0.00214, 0.00012,
    -0.00061, 0.00012, 0.00156,
    0.00070, -0.00012, -0.00177,
    -0.00093, 0.00012, 0.00223
  ),
  nrow = 9
)

# Specific time interval -----
DeltaBeta(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1
)

# Range of time intervals -----
delta <- DeltaBeta(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1:5
)
plot(delta)

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

```

```
summary(delta)
confint(delta, level = 0.95)
plot(delta)
```

DeltaBetaStd

Delta Method Sampling Variance-Covariance Matrix for the Elements of the Standardized 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 standardized matrix of lagged coefficients β over a specific time interval Δt or a range of time intervals using the first-order stochastic differential equation model's drift matrix Φ and process noise covariance matrix Σ .

Usage

```
DeltaBetaStd(phi, vcov_phi_vec, sigma, vcov_sigma_vech, delta_t, 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 $\text{vec}(\Phi)$.
sigma	Numeric matrix. The process noise covariance matrix (Σ).
vcov_sigma_vech	Numeric matrix. The sampling variance-covariance matrix of $\text{vech}(\Sigma)$.
delta_t	Numeric. Time interval (Δt).
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 [TotalStd\(\)](#).

Delta Method:

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

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

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

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

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

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

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

Value

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

call Function call.

args Function arguments.

fun Function used ("DeltaBetaStd").

output A list the length of which is equal to the length of `delta_t`.

Each element in the output list has the following elements:

delta_t Time interval.

jacobian Jacobian matrix.

est Estimated elements of the matrix of lagged coefficients.

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

Author(s)

Ivan Jacob Agaloos Pesigan

References

- Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. *Sociological Methodology*, 17, 37. [doi:10.2307/271028](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: [DeltaBeta\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBETA\(\)](#), [MCBETAStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

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")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,
    -0.00600, -0.00033, 0.00110,
    0.00324, 0.00020, -0.00061,
    0.00040, 0.00374, 0.00016,
    -0.00022, -0.00273, -0.00016,
    0.00009, 0.00150, 0.00012,
    -0.00151, 0.00016, 0.00389,
    0.00103, -0.00007, -0.00283,
    -0.00050, 0.00000, 0.00156,
    -0.00600, -0.00022, 0.00103,
    0.00644, 0.00031, -0.00119,
    -0.00374, -0.00021, 0.00070,
    -0.00033, -0.00273, -0.00007,
    0.00031, 0.00287, 0.00013,
    -0.00014, -0.00170, -0.00012,
    0.00110, -0.00016, -0.00283,
    -0.00119, 0.00013, 0.00297,
    0.00063, -0.00004, -0.00177,
    0.00324, 0.00009, -0.00050,
    -0.00374, -0.00014, 0.00063,
    0.00495, 0.00024, -0.00093,
    0.00020, 0.00150, 0.00000,
    -0.00021, -0.00170, -0.00004,
    0.00024, 0.00214, 0.00012,
    -0.00061, 0.00012, 0.00156,
    0.00070, -0.00012, -0.00177,
    -0.00093, 0.00012, 0.00223
  ),
  nrow = 9
)
sigma <- matrix(
  data = c(
```

```

      0.24, 0.02, -0.05,
      0.02, 0.07, 0.02,
      -0.05, 0.02, 0.08
    ),
    nrow = 3
  )
vcov_sigma_vech <- matrix(
  data = c(
    0.00057, 0.00001, -0.00009,
    0.00000, 0.00000, 0.00001,
    0.00001, 0.00012, 0.00001,
    0.00000, -0.00002, 0.00000,
    -0.00009, 0.00001, 0.00014,
    0.00000, 0.00000, -0.00005,
    0.00000, 0.00000, 0.00000,
    0.00010, 0.00001, 0.00000,
    0.00000, -0.00002, 0.00000,
    0.00001, 0.00005, 0.00001,
    0.00001, 0.00000, -0.00005,
    0.00000, 0.00001, 0.00012
  ),
  nrow = 6
)

# Specific time interval -----
DeltaBetaStd(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  sigma = sigma,
  vcov_sigma_vech = vcov_sigma_vech,
  delta_t = 1
)

# Range of time intervals -----
delta <- DeltaBetaStd(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  sigma = sigma,
  vcov_sigma_vech = vcov_sigma_vech,
  delta_t = 1:5
)
plot(delta)

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

Usage

```
DeltaIndirectCentral(phi, vcov_phi_vec, delta_t, 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).
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 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 g using $\hat{\theta}$ as input can be expressed as:

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

where \mathbf{J} is the matrix of first-order derivatives of the function 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 $g(\hat{\theta})$ as follows:

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

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

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

Value

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

call Function call.

args Function arguments.

fun Function used ("DeltaIndirectCentral").

output A list the length of which is equal to the length of `delta_t`.

Each element in the output list has the following elements:

delta_t Time interval.

jacobian Jacobian matrix.

est Estimated indirect effect centrality.

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

Author(s)

Ivan Jacob Agaloos Pesigan

References

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

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

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

See Also

Other Continuous Time Mediation Functions: `DeltaBeta()`, `DeltaBetaStd()`, `DeltaMed()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `Direct()`, `DirectStd()`, `Indirect()`, `IndirectCentral()`, `IndirectStd()`, `MCBeta()`, `MCBetaStd()`, `MCIndirectCentral()`, `MCMed()`, `MCMedStd()`, `MCPhi()`, `MCTotalCentral()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorIndirectCentral()`, `PosteriorMed()`, `PosteriorPhi()`, `PosteriorTotalCentral()`, `Total()`, `TotalCentral()`, `TotalStd()`, `Trajectory()`

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")
  vcov_phi_vec <- matrix(
    data = c(
      0.002704274, -0.001475275, 0.000949122,
      -0.001619422, 0.000885122, -0.000569404,
      0.00085493, -0.000465824, 0.000297815,
      -0.001475275, 0.004428442, -0.002642303,
      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(
  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 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:

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

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

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

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

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

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

Value

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

call Function call.

args Function arguments.

fun Function used ("DeltaMed").

output A list the length of which is equal to the length of `delta_t`.

Each element in the output list has the following elements:

delta_t Time interval.

jacobian Jacobian matrix.

est Estimated total, direct, and indirect effects.

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

Author(s)

Ivan Jacob Agaloos Pesigan

References

- Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. *Sociological Methodology*, 17, 37. doi:10.2307/271028
- Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. *Structural Equation Modeling: A Multidisciplinary Journal*, 23 (1), 61–75. doi:10.1080/10705511.2014.973960
- Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. *Psychometrika*, 87 (1), 214–252. doi:10.1007/s11336021097670

See Also

Other Continuous Time Mediation Functions: [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBETA\(\)](#), [MCBETAStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

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")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,
    -0.00600, -0.00033, 0.00110,
    0.00324, 0.00020, -0.00061,
    0.00040, 0.00374, 0.00016,
    -0.00022, -0.00273, -0.00016,
    0.00009, 0.00150, 0.00012,
    -0.00151, 0.00016, 0.00389,
    0.00103, -0.00007, -0.00283,
    -0.00050, 0.00000, 0.00156,
    -0.00600, -0.00022, 0.00103,
    0.00644, 0.00031, -0.00119,
    -0.00374, -0.00021, 0.00070,
    -0.00033, -0.00273, -0.00007,
    0.00031, 0.00287, 0.00013,
    -0.00014, -0.00170, -0.00012,
    0.00110, -0.00016, -0.00283,
    -0.00119, 0.00013, 0.00297,
    0.00063, -0.00004, -0.00177,
    0.00324, 0.00009, -0.00050,
    -0.00374, -0.00014, 0.00063,
    0.00495, 0.00024, -0.00093,
    0.00020, 0.00150, 0.00000,
    -0.00021, -0.00170, -0.00004,
    0.00024, 0.00214, 0.00012,
    -0.00061, 0.00012, 0.00156,
    0.00070, -0.00012, -0.00177,
    -0.00093, 0.00012, 0.00223
  ),
  nrow = 9
)

# 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 = 1:5,
  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)

```

DeltaMedStd

Delta Method Sampling Variance-Covariance Matrix for the Standardized 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 standardized 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 and process noise covariance matrix Σ .

Usage

```

DeltaMedStd(
  phi,
  vcov_phi_vec,
  sigma,
  vcov_sigma_vech,

```

```

    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 $\text{vec}(\Phi)$.
sigma	Numeric matrix. The process noise covariance matrix (Σ).
vcov_sigma_vech	Numeric matrix. The sampling variance-covariance matrix of $\text{vech}(\Sigma)$.
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.
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 [TotalStd\(\)](#), [DirectStd\(\)](#), and [IndirectStd\(\)](#) for more details.

Delta Method:

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

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

where \mathbf{J} is the matrix of first-order derivatives of the function 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 $g(\hat{\theta})$ as follows:

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

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

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

Value

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

call Function call.

args Function arguments.

fun Function used ("DeltaMedStd").

output A list the length of which is equal to the length of `delta_t`.

Each element in the output list has the following elements:

delta_t Time interval.

jacobian Jacobian matrix.

est Estimated total, direct, and indirect effects.

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

Author(s)

Ivan Jacob Agaloos Pesigan

References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. *Sociological Methodology*, 17, 37. doi:[10.2307/271028](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: [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPHI\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

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")
```

```

vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,
    -0.00600, -0.00033, 0.00110,
    0.00324, 0.00020, -0.00061,
    0.00040, 0.00374, 0.00016,
    -0.00022, -0.00273, -0.00016,
    0.00009, 0.00150, 0.00012,
    -0.00151, 0.00016, 0.00389,
    0.00103, -0.00007, -0.00283,
    -0.00050, 0.00000, 0.00156,
    -0.00600, -0.00022, 0.00103,
    0.00644, 0.00031, -0.00119,
    -0.00374, -0.00021, 0.00070,
    -0.00033, -0.00273, -0.00007,
    0.00031, 0.00287, 0.00013,
    -0.00014, -0.00170, -0.00012,
    0.00110, -0.00016, -0.00283,
    -0.00119, 0.00013, 0.00297,
    0.00063, -0.00004, -0.00177,
    0.00324, 0.00009, -0.00050,
    -0.00374, -0.00014, 0.00063,
    0.00495, 0.00024, -0.00093,
    0.00020, 0.00150, 0.00000,
    -0.00021, -0.00170, -0.00004,
    0.00024, 0.00214, 0.00012,
    -0.00061, 0.00012, 0.00156,
    0.00070, -0.00012, -0.00177,
    -0.00093, 0.00012, 0.00223
  ),
  nrow = 9
)
sigma <- matrix(
  data = c(
    0.24, 0.02, -0.05,
    0.02, 0.07, 0.02,
    -0.05, 0.02, 0.08
  ),
  nrow = 3
)
vcov_sigma_vech <- matrix(
  data = c(
    0.00057, 0.00001, -0.00009,
    0.00000, 0.00000, 0.00001,
    0.00001, 0.00012, 0.00001,
    0.00000, -0.00002, 0.00000,
    -0.00009, 0.00001, 0.00014,
    0.00000, 0.00000, -0.00005,
    0.00000, 0.00000, 0.00000,
    0.00010, 0.00001, 0.00000,
    0.00000, -0.00002, 0.00000,
    0.00001, 0.00005, 0.00001,
    0.00001, 0.00000, -0.00005,
  )

```

```

      0.00000, 0.00001, 0.00012
    ),
    nrow = 6
  )

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

# Range of time intervals -----
delta <- DeltaMedStd(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  sigma = sigma,
  vcov_sigma_vech = vcov_sigma_vech,
  delta_t = 1:5,
  from = "x",
  to = "y",
  med = "m"
)
plot(delta)

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

Usage

```
DeltaTotalCentral(phi, vcov_phi_vec, delta_t, 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).

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 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 g using $\hat{\theta}$ as input can be expressed as:

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

where \mathbf{J} is the matrix of first-order derivatives of the function 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 $g(\hat{\theta})$ as follows:

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

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

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

Value

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

call Function call.

args Function arguments.

fun Function used ("DeltaTotalCentral").

output A list the length of which is equal to the length of `delta_t`.

Each element in the output list has the following elements:

delta_t Time interval.

jacobian Jacobian matrix.

est Estimated total effect centrality.

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

Author(s)

Ivan Jacob Agaloos Pesigan

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: [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

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")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,
    -0.00600, -0.00033, 0.00110,
    0.00324, 0.00020, -0.00061,
    0.00040, 0.00374, 0.00016,
    -0.00022, -0.00273, -0.00016,
    0.00009, 0.00150, 0.00012,
    -0.00151, 0.00016, 0.00389,
    0.00103, -0.00007, -0.00283,
  )
```

```

-0.00050, 0.00000, 0.00156,
-0.00600, -0.00022, 0.00103,
0.00644, 0.00031, -0.00119,
-0.00374, -0.00021, 0.00070,
-0.00033, -0.00273, -0.00007,
0.00031, 0.00287, 0.00013,
-0.00014, -0.00170, -0.00012,
0.00110, -0.00016, -0.00283,
-0.00119, 0.00013, 0.00297,
0.00063, -0.00004, -0.00177,
0.00324, 0.00009, -0.00050,
-0.00374, -0.00014, 0.00063,
0.00495, 0.00024, -0.00093,
0.00020, 0.00150, 0.00000,
-0.00021, -0.00170, -0.00004,
0.00024, 0.00214, 0.00012,
-0.00061, 0.00012, 0.00156,
0.00070, -0.00012, -0.00177,
-0.00093, 0.00012, 0.00223
),
nrow = 9
)

# Specific time interval -----
DeltaTotalCentral(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1
)

# Range of time intervals -----
delta <- DeltaTotalCentral(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1:5
)
plot(delta)

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

```

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 $\left(\boldsymbol{\Theta}^{\frac{1}{2}}\right) \left(\boldsymbol{\Theta}^{\frac{1}{2}}\right)' = \boldsymbol{\Theta}$.

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 ("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: [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

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")
)

```

DirectStd

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

Description

This function computes the standardized 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 and process noise covariance matrix Σ .

Usage

```
DirectStd(phi, sigma, 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.
sigma	Numeric matrix. The process noise covariance matrix (Σ).
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 standardized direct effect of the independent variable X on the dependent variable Y relative to some mediator variables \mathbf{m} is given by

$$\text{Direct}_{\Delta t_{i,j,\mathbf{m}}}^* = \mathbf{S} \left(\exp(\Delta t \mathbf{D} \Phi \mathbf{D})_{i,j} \right) \mathbf{S}^{-1}$$

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 , \mathbf{S} a diagonal matrix with model-implied standard deviations on the diagonals, and Δt the time interval.

Value

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

call Function call.

args Function arguments.

fun Function used ("DirectStd").

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: [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBETA\(\)](#), [MCBETAStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

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")
  sigma <- matrix(
    data = c(
      0.24, 0.02, -0.05,
      0.02, 0.07, 0.02,
      -0.05, 0.02, 0.08
    ),
    nrow = 3
  )
  delta_t <- 1
  DirectStd(
    phi = phi,
    sigma = sigma,
    delta_t = delta_t,
    from = "x",
    to = "y",
    med = "m"
  )

```

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

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 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
- 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()`, `DeltaBetaStd()`, `DeltaIndirectCentral()`, `DeltaMed()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `Direct()`, `DirectStd()`, `IndirectCentral()`, `IndirectStd()`, `MCBETA()`, `MCBETAStd()`, `MCIndirectCentral()`, `MCMed()`, `MCMedStd()`, `MCPHI()`, `MCTotalCentral()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorIndirectCentral()`, `PosteriorMed()`, `PosteriorPhi()`, `PosteriorTotalCentral()`, `Total()`, `TotalCentral()`, `TotalStd()`, `Trajectory()`

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")
  )

```

IndirectCentral	<i>Indirect Effect Centrality</i>
-----------------	-----------------------------------

Description

Indirect Effect Centrality

Usage

```
IndirectCentral(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	Vector of positive numbers. Time interval (Δt).

Details

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

Value

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

call Function call.

args Function arguments.

fun Function used ("IndirectCentral").

output A matrix of indirect effect centrality.

Author(s)

Ivan Jacob Agaloos Pesigan

References

- Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. *Sociological Methodology*, 17, 37. doi:[10.2307/271028](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: [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectStd\(\)](#), [MCBETA\(\)](#), [MCBETAStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

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")

# Specific time interval -----
IndirectCentral(
  phi = phi,
  delta_t = 1
)

# Range of time intervals -----
indirect_central <- IndirectCentral(
  phi = phi,
  delta_t = 1:30
)
plot(indirect_central)

# Methods -----
# IndirectCentral has a number of methods including
# print, summary, and plot
indirect_central <- IndirectCentral(
  phi = phi,
  delta_t = 1:5
)
print(indirect_central)
summary(indirect_central)
plot(indirect_central)
```

IndirectStd

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

Description

This function computes the standardized 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 and process noise covariance matrix Σ .

Usage

```
IndirectStd(phi, sigma, 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.
sigma	Numeric matrix. The process noise covariance matrix (Σ).
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 standardized 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

$$\text{Indirect}_{\Delta t}^* = \text{Total}_{\Delta t}^* - \text{Direct}_{\Delta t}^*$$

where $\text{Total}_{\Delta t}^*$ and $\text{Direct}_{\Delta t}^*$ are standardized total and direct effects for time interval Δt .

Value

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

call Function call.

args Function arguments.

fun Function used ("IndirectStd").

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: [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [MCBETA\(\)](#), [MCBETAStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

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")
sigma <- matrix(
  data = c(
    0.24, 0.02, -0.05,
    0.02, 0.07, 0.02,
    -0.05, 0.02, 0.08
  ),
  nrow = 3
)
delta_t <- 1
IndirectStd(
  phi = phi,
  sigma = sigma,
  delta_t = delta_t,
  from = "x",
  to = "y",
  med = "m"
)
```

MCBeta	<i>Monte Carlo Sampling Distribution for the Elements of the Matrix of Lagged Coefficients Over a Specific Time Interval or a Range of Time Intervals</i>
--------	---

Description

This function generates a Monte Carlo method sampling distribution for the elements of the matrix of lagged coefficients β over a specific time interval Δt or a range of time intervals using the first-order stochastic differential equation model drift matrix Φ .

Usage

```
MCBeta(
  phi,
  vcov_phi_vec,
  delta_t,
  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).
R	Positive integer. Number of replications.
test_phi	Logical. If test_phi = TRUE, the function tests the stability of the generated drift matrix Φ . If the test returns FALSE, the function generates a new drift matrix Φ 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 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,

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

Let $g\left(\hat{\theta}\right)$ be a parameter that is a function of the estimated parameters. A sampling distribution of $g\left(\hat{\theta}\right)$, which we refer to as $g\left(\hat{\theta}^*\right)$, 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.

Value

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

call Function call.

args Function arguments.

fun Function used ("MCBeta").

output A list the length of which is equal to the length of `delta_t`.

Each element in the output list has the following elements:

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

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

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: [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBetaStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

Examples

```

set.seed(42)
phi <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,
    -0.00600, -0.00033, 0.00110,
    0.00324, 0.00020, -0.00061,
    0.00040, 0.00374, 0.00016,
    -0.00022, -0.00273, -0.00016,
    0.00009, 0.00150, 0.00012,
    -0.00151, 0.00016, 0.00389,
    0.00103, -0.00007, -0.00283,
    -0.00050, 0.00000, 0.00156,
    -0.00600, -0.00022, 0.00103,
    0.00644, 0.00031, -0.00119,
    -0.00374, -0.00021, 0.00070,
    -0.00033, -0.00273, -0.00007,
    0.00031, 0.00287, 0.00013,
    -0.00014, -0.00170, -0.00012,
    0.00110, -0.00016, -0.00283,
    -0.00119, 0.00013, 0.00297,
    0.00063, -0.00004, -0.00177,
    0.00324, 0.00009, -0.00050,
    -0.00374, -0.00014, 0.00063,
    0.00495, 0.00024, -0.00093,
    0.00020, 0.00150, 0.00000,
    -0.00021, -0.00170, -0.00004,
    0.00024, 0.00214, 0.00012,
    -0.00061, 0.00012, 0.00156,
    0.00070, -0.00012, -0.00177,
    -0.00093, 0.00012, 0.00223
  ),
  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(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1:5,
  R = 100L # use a large value for R in actual research
)
plot(mc)

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

```

MCBetaStd

Monte Carlo Sampling Distribution for the Elements of the Standardized 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 standardized matrix of lagged coefficients β over a specific time interval Δt or a range of time intervals using the first-order stochastic differential equation model drift matrix Φ and process noise covariance matrix Σ .

Usage

```

MCBetaStd(
  phi,
  vcov_phi_vec,
  sigma,
  vcov_sigma_vech,
  delta_t,
  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.

<code>vcov_phi_vec</code>	Numeric matrix. The sampling variance-covariance matrix of $\text{vec}(\Phi)$.
<code>sigma</code>	Numeric matrix. The process noise covariance matrix (Σ).
<code>vcov_sigma_vech</code>	Numeric matrix. The sampling variance-covariance matrix of $\text{vech}(\Sigma)$.
<code>delta_t</code>	Numeric. Time interval (Δt).
<code>R</code>	Positive integer. Number of replications.
<code>test_phi</code>	Logical. If <code>test_phi = TRUE</code> , the function tests the stability of the generated drift matrix Φ . If the test returns <code>FALSE</code> , the function generates a new drift matrix Φ and runs the test recursively until the test returns <code>TRUE</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 <code>R</code> is a large value.
<code>seed</code>	Random seed.

Details

See [TotalStd\(\)](#).

Monte Carlo Method:

Let θ be a vector that combines $\text{vec}(\Phi)$, that is, the elements of the Φ matrix in vector form sorted column-wise and $\text{vech}(\Sigma)$, that is, the unique elements of the Σ matrix in vector form sorted column-wise. Let $\hat{\theta}$ be a vector that combines $\{\text{vec}(\hat{\Phi}) \text{ and } \text{vech}(\hat{\Sigma})\}$. 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.

Value

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

call Function call.

args Function arguments.

fun Function used ("MCBetaStd").

output A list the length of which is equal to the length of `delta_t`.

Each element in the output list has the following elements:

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

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

Author(s)

Ivan Jacob Agaloos Pesigan

References

- Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. *Sociological Methodology*, 17, 37. doi:10.2307/271028
- Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. *Structural Equation Modeling: A Multidisciplinary Journal*, 23 (1), 61–75. doi:10.1080/10705511.2014.973960
- Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. *Psychometrika*, 87 (1), 214–252. doi:10.1007/s11336021097670

See Also

Other Continuous Time Mediation Functions: [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

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")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,
    -0.00600, -0.00033, 0.00110,
    0.00324, 0.00020, -0.00061,
    0.00040, 0.00374, 0.00016,
    -0.00022, -0.00273, -0.00016,
    0.00009, 0.00150, 0.00012,
    -0.00151, 0.00016, 0.00389,
    0.00103, -0.00007, -0.00283,
    -0.00050, 0.00000, 0.00156,
    -0.00600, -0.00022, 0.00103,
    0.00644, 0.00031, -0.00119,
    -0.00374, -0.00021, 0.00070,
    -0.00033, -0.00273, -0.00007,
    0.00031, 0.00287, 0.00013,
    -0.00014, -0.00170, -0.00012,
    0.00110, -0.00016, -0.00283,
  )
)
```

```

-0.00119, 0.00013, 0.00297,
0.00063, -0.00004, -0.00177,
0.00324, 0.00009, -0.00050,
-0.00374, -0.00014, 0.00063,
0.00495, 0.00024, -0.00093,
0.00020, 0.00150, 0.00000,
-0.00021, -0.00170, -0.00004,
0.00024, 0.00214, 0.00012,
-0.00061, 0.00012, 0.00156,
0.00070, -0.00012, -0.00177,
-0.00093, 0.00012, 0.00223
),
nrow = 9
)
sigma <- matrix(
  data = c(
    0.24, 0.02, -0.05,
    0.02, 0.07, 0.02,
    -0.05, 0.02, 0.08
  ),
  nrow = 3
)
vcov_sigma_vech <- matrix(
  data = c(
    0.00057, 0.00001, -0.00009,
    0.00000, 0.00000, 0.00001,
    0.00001, 0.00012, 0.00001,
    0.00000, -0.00002, 0.00000,
    -0.00009, 0.00001, 0.00014,
    0.00000, 0.00000, -0.00005,
    0.00000, 0.00000, 0.00000,
    0.00010, 0.00001, 0.00000,
    0.00000, -0.00002, 0.00000,
    0.00001, 0.00005, 0.00001,
    0.00001, 0.00000, -0.00005,
    0.00000, 0.00001, 0.00012
  ),
  nrow = 6
)

# Specific time interval -----
MCBetaStd(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  sigma = sigma,
  vcov_sigma_vech = vcov_sigma_vech,
  delta_t = 1,
  R = 100L # use a large value for R in actual research
)

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

```

```

vcov_phi_vec = vcov_phi_vec,
sigma = sigma,
vcov_sigma_vech = vcov_sigma_vech,
delta_t = 1:5,
R = 100L # use a large value for R in actual research
)
plot(mc)

# Methods -----
# MCBetaStd 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 Δt using the first-order stochastic differential equation model drift matrix Φ .

Usage

```

MCIndirectCentral(
  phi,
  vcov_phi_vec,
  delta_t,
  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).
R	Positive integer. Number of replications.

test_phi	Logical. If test_phi = TRUE, the function tests the stability of the generated drift matrix Φ . If the test returns FALSE, the function generates a new drift matrix Φ 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 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.

Value

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

call Function call.

args Function arguments.

fun Function used ("MCIndirectCentral").

output A list the length of which is equal to the length of `delta_t`.

Each element in the output list has the following elements:

est A vector of indirect effect centrality.

thetahatstar A matrix of Monte Carlo indirect effect centrality.

Author(s)

Ivan Jacob Agaloos Pesigan

References

- Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. *Sociological Methodology*, 17, 37. doi:10.2307/271028
- Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. *Structural Equation Modeling: A Multidisciplinary Journal*, 23 (1), 61–75. doi:10.1080/10705511.2014.973960
- Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. *Psychometrika*, 87 (1), 214–252. doi:10.1007/s11336021097670

See Also

Other Continuous Time Mediation Functions: [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

Examples

```
set.seed(42)
phi <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,
    -0.00600, -0.00033, 0.00110,
    0.00324, 0.00020, -0.00061,
    0.00040, 0.00374, 0.00016,
    -0.00022, -0.00273, -0.00016,
    0.00009, 0.00150, 0.00012,
    -0.00151, 0.00016, 0.00389,
    0.00103, -0.00007, -0.00283,
    -0.00050, 0.00000, 0.00156,
    -0.00600, -0.00022, 0.00103,
    0.00644, 0.00031, -0.00119,
    -0.00374, -0.00021, 0.00070,
    -0.00033, -0.00273, -0.00007,
    0.00031, 0.00287, 0.00013,
    -0.00014, -0.00170, -0.00012,
    0.00110, -0.00016, -0.00283,
    -0.00119, 0.00013, 0.00297,
    0.00063, -0.00004, -0.00177,
    0.00324, 0.00009, -0.00050,
  )
```



```

-0.00374, -0.00014, 0.00063,
0.00495, 0.00024, -0.00093,
0.00020, 0.00150, 0.00000,
-0.00021, -0.00170, -0.00004,
0.00024, 0.00214, 0.00012,
-0.00061, 0.00012, 0.00156,
0.00070, -0.00012, -0.00177,
-0.00093, 0.00012, 0.00223
),
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(
  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

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} over a specific time interval Δt or a range of time intervals 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 tests the stability of the generated drift matrix Φ . If the test returns FALSE, the function generates a new drift matrix Φ and runs the test recursively until the test returns TRUE.
ncores	Positive integer. Number of cores to use. If ncores = NULL, use a single core. Consider using multiple cores when number of replications R is a large value.
seed	Random seed.

Details

See [Total\(\)](#), [Direct\(\)](#), and [Indirect\(\)](#) for more details.

Monte Carlo Method:

Let 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.

Value

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

call Function call.

args Function arguments.

fun Function used ("MCMed").

output A list with length of `length(delta_t)`.

Each element in the output list has the following elements:

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

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

Author(s)

Ivan Jacob Agaloos Pesigan

References

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

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

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

See Also

Other Continuous Time Mediation Functions: `DeltaBeta()`, `DeltaBetaStd()`, `DeltaIndirectCentral()`, `DeltaMed()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `Direct()`, `DirectStd()`, `Indirect()`, `IndirectCentral()`, `IndirectStd()`, `MCBeta()`, `MCBetaStd()`, `MCIndirectCentral()`, `MCMedStd()`, `MCPhi()`, `MCTotalCentral()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorIndirectCentral()`, `PosteriorMed()`, `PosteriorPhi()`, `PosteriorTotalCentral()`, `Total()`, `TotalCentral()`, `TotalStd()`, `Trajectory()`

Examples

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

```

      0, 0, -0.693
    ),
    nrow = 3
  )
  colnames(phi) <- rownames(phi) <- c("x", "m", "y")
  vcov_phi_vec <- matrix(
    data = c(
      0.00843, 0.00040, -0.00151,
      -0.00600, -0.00033, 0.00110,
      0.00324, 0.00020, -0.00061,
      0.00040, 0.00374, 0.00016,
      -0.00022, -0.00273, -0.00016,
      0.00009, 0.00150, 0.00012,
      -0.00151, 0.00016, 0.00389,
      0.00103, -0.00007, -0.00283,
      -0.00050, 0.00000, 0.00156,
      -0.00600, -0.00022, 0.00103,
      0.00644, 0.00031, -0.00119,
      -0.00374, -0.00021, 0.00070,
      -0.00033, -0.00273, -0.00007,
      0.00031, 0.00287, 0.00013,
      -0.00014, -0.00170, -0.00012,
      0.00110, -0.00016, -0.00283,
      -0.00119, 0.00013, 0.00297,
      0.00063, -0.00004, -0.00177,
      0.00324, 0.00009, -0.00050,
      -0.00374, -0.00014, 0.00063,
      0.00495, 0.00024, -0.00093,
      0.00020, 0.00150, 0.00000,
      -0.00021, -0.00170, -0.00004,
      0.00024, 0.00214, 0.00012,
      -0.00061, 0.00012, 0.00156,
      0.00070, -0.00012, -0.00177,
      -0.00093, 0.00012, 0.00223
    ),
    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(
    phi = phi,
    vcov_phi_vec = vcov_phi_vec,

```

```

    delta_t = 1:5,
    from = "x",
    to = "y",
    med = "m",
    R = 100L # use a large value for R in actual research
  )
plot(mc)

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

```

MCMedStd

Monte Carlo Sampling Distribution of Standardized 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 standardized 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 drift matrix Φ and process noise covariance matrix Σ .

Usage

```

MCMedStd(
  phi,
  vcov_phi_vec,
  sigma,
  vcov_sigma_vech,
  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 $\text{vec}(\Phi)$.
sigma	Numeric matrix. The process noise covariance matrix (Σ).
vcov_sigma_vech	Numeric matrix. The sampling variance-covariance matrix of $\text{vech}(\Sigma)$.
delta_t	Numeric. Time interval (Δt).
from	Character string. Name of the independent variable X in ϕ .
to	Character string. Name of the dependent variable Y in ϕ .
med	Character vector. Name/s of the mediator variable/s in ϕ .
R	Positive integer. Number of replications.
test_phi	Logical. If <code>test_phi = TRUE</code> , the function tests the stability of the generated drift matrix Φ . If the test returns <code>FALSE</code> , the function generates a new drift matrix Φ and runs the test recursively until the test returns <code>TRUE</code> .
ncores	Positive integer. Number of cores to use. If <code>ncores = NULL</code> , use a single core. Consider using multiple cores when number of replications <code>R</code> is a large value.
seed	Random seed.

Details

See [TotalStd\(\)](#), [DirectStd\(\)](#), and [IndirectStd\(\)](#) for more details.

Monte Carlo Method:

Let θ be a vector that combines $\text{vec}(\Phi)$, that is, the elements of the Φ matrix in vector form sorted column-wise and $\text{vech}(\Sigma)$, that is, the unique elements of the Σ matrix in vector form sorted column-wise. Let $\hat{\theta}$ be a vector that combines $\{\text{vec}(\hat{\Phi})$ and $\text{vech}(\hat{\Sigma})\}$. 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 $\mathbf{g}(\hat{\theta})$ be a parameter that is a function of the estimated parameters. A sampling distribution of $\mathbf{g}(\hat{\theta})$, which we refer to as $\mathbf{g}(\hat{\theta}^*)$, 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 $100(1 - \alpha)\%$ are the confidence intervals.

Value

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

call Function call.

args Function arguments.

fun Function used ("MCMedStd").

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: [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

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")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,
    -0.00600, -0.00033, 0.00110,
    0.00324, 0.00020, -0.00061,
    0.00040, 0.00374, 0.00016,
    -0.00022, -0.00273, -0.00016,
    0.00009, 0.00150, 0.00012,
```

```

-0.00151, 0.00016, 0.00389,
0.00103, -0.00007, -0.00283,
-0.00050, 0.00000, 0.00156,
-0.00600, -0.00022, 0.00103,
0.00644, 0.00031, -0.00119,
-0.00374, -0.00021, 0.00070,
-0.00033, -0.00273, -0.00007,
0.00031, 0.00287, 0.00013,
-0.00014, -0.00170, -0.00012,
0.00110, -0.00016, -0.00283,
-0.00119, 0.00013, 0.00297,
0.00063, -0.00004, -0.00177,
0.00324, 0.00009, -0.00050,
-0.00374, -0.00014, 0.00063,
0.00495, 0.00024, -0.00093,
0.00020, 0.00150, 0.00000,
-0.00021, -0.00170, -0.00004,
0.00024, 0.00214, 0.00012,
-0.00061, 0.00012, 0.00156,
0.00070, -0.00012, -0.00177,
-0.00093, 0.00012, 0.00223
),
nrow = 9
)
sigma <- matrix(
  data = c(
    0.24, 0.02, -0.05,
    0.02, 0.07, 0.02,
    -0.05, 0.02, 0.08
  ),
  nrow = 3
)
vcov_sigma_vech <- matrix(
  data = c(
    0.00057, 0.00001, -0.00009,
    0.00000, 0.00000, 0.00001,
    0.00001, 0.00012, 0.00001,
    0.00000, -0.00002, 0.00000,
    -0.00009, 0.00001, 0.00014,
    0.00000, 0.00000, -0.00005,
    0.00000, 0.00000, 0.00000,
    0.00010, 0.00001, 0.00000,
    0.00000, -0.00002, 0.00000,
    0.00001, 0.00005, 0.00001,
    0.00001, 0.00000, -0.00005,
    0.00000, 0.00001, 0.00012
  ),
  nrow = 6
)

# Specific time interval -----
MCMedStd(
  phi = phi,

```



```

vcov_phi_vec = vcov_phi_vec,
sigma = sigma,
vcov_sigma_vech = vcov_sigma_vech,
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 <- MCMedStd(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  sigma = sigma,
  vcov_sigma_vech = vcov_sigma_vech,
  delta_t = 1:5,
  from = "x",
  to = "y",
  med = "m",
  R = 100L # use a large value for R in actual research
)
plot(mc)

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

```

MCPhi

Generate Random Drift Matrices Using the Monte Carlo Method

Description

This function generates random drift matrices 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 tests the stability of the generated drift matrix Φ . If the test returns FALSE, the function generates a new drift matrix Φ 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 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,

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

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\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

Examples

```
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")
  MCPHi(
    phi = phi,
    vcov_phi_vec = 0.1 * diag(9),
    R = 100L # use a large value for R in actual research
  )
  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 = 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 Δt using the first-order stochastic differential equation model drift matrix Φ .

Usage

```

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

```

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>vcov_phi_vec</code>	Numeric matrix. The sampling variance-covariance matrix of $\text{vec}(\Phi)$.
<code>delta_t</code>	Numeric. Time interval (Δt).
<code>R</code>	Positive integer. Number of replications.
<code>test_phi</code>	Logical. If <code>test_phi = TRUE</code> , the function tests the stability of the generated drift matrix Φ . If the test returns <code>FALSE</code> , the function generates a new drift matrix Φ and runs the test recursively until the test returns <code>TRUE</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 <code>R</code> is a large value.
<code>seed</code>	Random seed.

Details

See [TotalCentral\(\)](#) 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.

Value

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

call Function call.

args Function arguments.

fun Function used ("MCTotalCentral").

output A list the length of which is equal to the length of `delta_t`.

Each element in the output list has the following elements:

est A vector of total effect centrality.

thetahatstar A matrix of Monte Carlo total effect centrality.

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\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

Examples

```
set.seed(42)
phi <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,
    -0.00600, -0.00033, 0.00110,
    0.00324, 0.00020, -0.00061,
    0.00040, 0.00374, 0.00016,
    -0.00022, -0.00273, -0.00016,
    0.00009, 0.00150, 0.00012,
    -0.00151, 0.00016, 0.00389,
    0.00103, -0.00007, -0.00283,
    -0.00050, 0.00000, 0.00156,
    -0.00600, -0.00022, 0.00103,
    0.00644, 0.00031, -0.00119,
    -0.00374, -0.00021, 0.00070,
    -0.00033, -0.00273, -0.00007,
    0.00031, 0.00287, 0.00013,
    -0.00014, -0.00170, -0.00012,
```

```

    0.00110, -0.00016, -0.00283,
    -0.00119, 0.00013, 0.00297,
    0.00063, -0.00004, -0.00177,
    0.00324, 0.00009, -0.00050,
    -0.00374, -0.00014, 0.00063,
    0.00495, 0.00024, -0.00093,
    0.00020, 0.00150, 0.00000,
    -0.00021, -0.00170, -0.00004,
    0.00024, 0.00214, 0.00012,
    -0.00061, 0.00012, 0.00156,
    0.00070, -0.00012, -0.00177,
    -0.00093, 0.00012, 0.00223
  ),
  nrow = 9
)

# Specific time interval -----
MCTotalCentral(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1,
  R = 100L # use a large value for R in actual research
)

# Range of time intervals -----
mc <- MCTotalCentral(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1:5,
  R = 100L # use a large value for R in actual research
)
plot(mc)

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

```

Med

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

Description

This function computes the total, direct, and indirect effects of the independent variable X on the dependent variable Y through mediator variables \mathbf{m} over a specific time interval 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

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

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 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

See Also

Other Continuous Time Mediation Functions: [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBETA\(\)](#), [MCBETAStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

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")

# 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)
```

MedStd	<i>Standardized Total, Direct, and Indirect Effects of X on Y Through M Over a Specific Time Interval or a Range of Time Intervals</i>
--------	--

Description

This function computes the standardized 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 and process noise covariance matrix Σ .

Usage

```
MedStd(phi, sigma, 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.
sigma	Numeric matrix. The process noise covariance matrix (Σ).
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 [TotalStd\(\)](#), [DirectStd\(\)](#), and [IndirectStd\(\)](#) for more details.

Value

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

call Function call.

args Function arguments.

fun Function used ("MedStd").

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: [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

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")
sigma <- matrix(
  data = c(
    0.24, 0.02, -0.05,
    0.02, 0.07, 0.02,
    -0.05, 0.02, 0.08
  ),
  nrow = 3
)
```

```

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

# Range of time intervals -----
med <- MedStd(
  phi = phi,
  sigma = sigma,
  delta_t = 1:30,
  from = "x",
  to = "y",
  med = "m"
)
plot(med)

# Methods -----
# MedStd has a number of methods including
# print, summary, and plot
med <- MedStd(
  phi = phi,
  sigma = sigma,
  delta_t = 1:5,
  from = "x",
  to = "y",
  med = "m"
)
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, col = NULL, ...)
```

Arguments

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

Value

Displays plots of point estimates and confidence intervals.

Author(s)

Ivan Jacob Agaloos Pesigan

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")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,
    -0.00600, -0.00033, 0.00110,
    0.00324, 0.00020, -0.00061,
    0.00040, 0.00374, 0.00016,
    -0.00022, -0.00273, -0.00016,
    0.00009, 0.00150, 0.00012,
    -0.00151, 0.00016, 0.00389,
    0.00103, -0.00007, -0.00283,
    -0.00050, 0.00000, 0.00156,
    -0.00600, -0.00022, 0.00103,
    0.00644, 0.00031, -0.00119,
    -0.00374, -0.00021, 0.00070,
    -0.00033, -0.00273, -0.00007,
    0.00031, 0.00287, 0.00013,
    -0.00014, -0.00170, -0.00012,
    0.00110, -0.00016, -0.00283,
    -0.00119, 0.00013, 0.00297,
    0.00063, -0.00004, -0.00177,
    0.00324, 0.00009, -0.00050,
    -0.00374, -0.00014, 0.00063,
    0.00495, 0.00024, -0.00093,
    0.00020, 0.00150, 0.00000,
    -0.00021, -0.00170, -0.00004,
    0.00024, 0.00214, 0.00012,
  )
)
```

```

      -0.00061, 0.00012, 0.00156,
      0.00070, -0.00012, -0.00177,
      -0.00093, 0.00012, 0.00223
    ),
    nrow = 9
  )

  # Range of time intervals -----
  delta <- DeltaMed(
    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.

Value

Displays plots of point estimates and confidence intervals.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```

set.seed(42)
phi <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,
    -0.00600, -0.00033, 0.00110,
    0.00324, 0.00020, -0.00061,
    0.00040, 0.00374, 0.00016,
    -0.00022, -0.00273, -0.00016,
    0.00009, 0.00150, 0.00012,
    -0.00151, 0.00016, 0.00389,
    0.00103, -0.00007, -0.00283,
    -0.00050, 0.00000, 0.00156,
    -0.00600, -0.00022, 0.00103,
    0.00644, 0.00031, -0.00119,
    -0.00374, -0.00021, 0.00070,
    -0.00033, -0.00273, -0.00007,
    0.00031, 0.00287, 0.00013,
    -0.00014, -0.00170, -0.00012,
    0.00110, -0.00016, -0.00283,
    -0.00119, 0.00013, 0.00297,
    0.00063, -0.00004, -0.00177,
    0.00324, 0.00009, -0.00050,
    -0.00374, -0.00014, 0.00063,
    0.00495, 0.00024, -0.00093,
    0.00020, 0.00150, 0.00000,
    -0.00021, -0.00170, -0.00004,
    0.00024, 0.00214, 0.00012,
    -0.00061, 0.00012, 0.00156,
    0.00070, -0.00012, -0.00177,
    -0.00093, 0.00012, 0.00223
  ),
  nrow = 9
)

# Range of time intervals -----
mc <- MCMed(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1:5,
  from = "x",
  to = "y",
  med = "m",

```

```

    R = 100L # use a large value for R in actual research
  )
  plot(mc)

```

plot.ctmedmed

Plot Method for an Object of Class ctmedmed

Description

Plot Method for an Object of Class ctmedmed

Usage

```

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

```

Arguments

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

Value

Displays plots of point estimates and confidence intervals.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```

phi <- matrix(
  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,

```

```

    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.

Value

Displays trajectory plots of 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")

traj <- Trajectory(

```



```

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

plot(traj)

```

PosteriorBeta	<i>Posterior Sampling Distribution for the Elements of the Matrix of Lagged Coefficients Over a Specific Time Interval or a Range of Time Intervals</i>
---------------	---

Description

This function generates a posterior sampling distribution for the elements of the matrix of lagged coefficients β over a specific time interval Δt or a range of time intervals using the first-order stochastic differential equation model drift matrix Φ .

Usage

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

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

Details

See [Total\(\)](#).

Value

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

call Function call.

args Function arguments.

fun Function used ("PosteriorBeta").

output A list the length of which is equal to the length of `delta_t`.

Each element in the output list has the following elements:

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

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

Author(s)

Ivan Jacob Agaloos Pesigan

References

- Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. *Sociological Methodology*, 17, 37. doi:[10.2307/271028](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: [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

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")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,
    -0.00600, -0.00033, 0.00110,
    0.00324, 0.00020, -0.00061,
    0.00040, 0.00374, 0.00016,
    -0.00022, -0.00273, -0.00016,
    0.00009, 0.00150, 0.00012,
    -0.00151, 0.00016, 0.00389,
    0.00103, -0.00007, -0.00283,
    -0.00050, 0.00000, 0.00156,
    -0.00600, -0.00022, 0.00103,
    0.00644, 0.00031, -0.00119,
    -0.00374, -0.00021, 0.00070,
    -0.00033, -0.00273, -0.00007,
    0.00031, 0.00287, 0.00013,
    -0.00014, -0.00170, -0.00012,
    0.00110, -0.00016, -0.00283,
  )
)
```

```

-0.00119, 0.00013, 0.00297,
0.00063, -0.00004, -0.00177,
0.00324, 0.00009, -0.00050,
-0.00374, -0.00014, 0.00063,
0.00495, 0.00024, -0.00093,
0.00020, 0.00150, 0.00000,
-0.00021, -0.00170, -0.00004,
0.00024, 0.00214, 0.00012,
-0.00061, 0.00012, 0.00156,
0.00070, -0.00012, -0.00177,
-0.00093, 0.00012, 0.00223
),
nrow = 9
)

phi <- MCPHi(
  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(
  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 Δt or a range of time intervals using the posterior distribution of the first-order stochastic differential equation model drift matrix Φ .

Usage

```
PosteriorIndirectCentral(phi, delta_t, 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>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 [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 the length of which is equal to the length of `delta_t`.

Each element in the output list has the following elements:

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

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

Author(s)

Ivan Jacob Agaloos Pesigan

References

- Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. *Sociological Methodology*, 17, 37. [doi:10.2307/271028](#)
- Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. *Structural Equation Modeling: A Multidisciplinary Journal*, 23 (1), 61–75. [doi:10.1080/10705511.2014.973960](#)
- Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. *Psychometrika*, 87 (1), 214–252. [doi:10.1007/s11336021097670](#)

See Also

Other Continuous Time Mediation Functions: [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBETA\(\)](#), [MCBETAStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPHI\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

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")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,
    -0.00600, -0.00033, 0.00110,
    0.00324, 0.00020, -0.00061,
    0.00040, 0.00374, 0.00016,
    -0.00022, -0.00273, -0.00016,
    0.00009, 0.00150, 0.00012,
    -0.00151, 0.00016, 0.00389,
    0.00103, -0.00007, -0.00283,
    -0.00050, 0.00000, 0.00156,
    -0.00600, -0.00022, 0.00103,
    0.00644, 0.00031, -0.00119,
    -0.00374, -0.00021, 0.00070,
    -0.00033, -0.00273, -0.00007,
    0.00031, 0.00287, 0.00013,
    -0.00014, -0.00170, -0.00012,
    0.00110, -0.00016, -0.00283,
    -0.00119, 0.00013, 0.00297,
    0.00063, -0.00004, -0.00177,
    0.00324, 0.00009, -0.00050,
    -0.00374, -0.00014, 0.00063,
    0.00495, 0.00024, -0.00093,
    0.00020, 0.00150, 0.00000,
    -0.00021, -0.00170, -0.00004,
    0.00024, 0.00214, 0.00012,
    -0.00061, 0.00012, 0.00156,
    0.00070, -0.00012, -0.00177,
    -0.00093, 0.00012, 0.00223
  ),
  nrow = 9
)

phi <- MCPHI(
```

```

    phi = phi,
    vcov_phi_vec = vcov_phi_vec,
    R = 1000L
)$output

# Specific time interval -----
PosteriorIndirectCentral(
  phi = phi,
  delta_t = 1
)

# Range of time intervals -----
posterior <- PosteriorIndirectCentral(
  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 Δt or a range of time intervals 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

phi	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.
delta_t	Numeric. Time interval (Δt).
from	Character string. Name of the independent variable X in phi.

<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 the length of which is equal to the length of `delta_t`.

Each element in the output list has the following elements:

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

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

Author(s)

Ivan Jacob Agaloos Pesigan

References

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

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

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

See Also

Other Continuous Time Mediation Functions: [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

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")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,
    -0.00600, -0.00033, 0.00110,
    0.00324, 0.00020, -0.00061,
    0.00040, 0.00374, 0.00016,
    -0.00022, -0.00273, -0.00016,
    0.00009, 0.00150, 0.00012,
    -0.00151, 0.00016, 0.00389,
    0.00103, -0.00007, -0.00283,
    -0.00050, 0.00000, 0.00156,
    -0.00600, -0.00022, 0.00103,
    0.00644, 0.00031, -0.00119,
    -0.00374, -0.00021, 0.00070,
    -0.00033, -0.00273, -0.00007,
    0.00031, 0.00287, 0.00013,
    -0.00014, -0.00170, -0.00012,
    0.00110, -0.00016, -0.00283,
    -0.00119, 0.00013, 0.00297,
    0.00063, -0.00004, -0.00177,
    0.00324, 0.00009, -0.00050,
    -0.00374, -0.00014, 0.00063,
    0.00495, 0.00024, -0.00093,
    0.00020, 0.00150, 0.00000,
    -0.00021, -0.00170, -0.00004,
    0.00024, 0.00214, 0.00012,
    -0.00061, 0.00012, 0.00156,
    0.00070, -0.00012, -0.00177,
    -0.00093, 0.00012, 0.00223
  ),
  nrow = 9
)

phi <- MCPHi(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  R = 1000L
)$output

# Specific time interval -----
PosteriorMed(
  phi = phi,

```



```

    delta_t = 1,
    from = "x",
    to = "y",
    med = "m"
  )

  # Range of time intervals -----
  posterior <- PosteriorMed(
    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.

Value

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

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Continuous Time Mediation Functions: [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBETA\(\)](#), [MCBETAStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPHI\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

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

Description

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

Usage

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

Arguments

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

Details

See [TotalCentral\(\)](#) for more details.

Value

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

call Function call.

args Function arguments.

fun Function used ("PosteriorTotalCentral").

output A list the length of which is equal to the length of `delta_t`.

Each element in the output list has the following elements:

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

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

Author(s)

Ivan Jacob Agaloos Pesigan

References

- Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. *Sociological Methodology*, 17, 37. doi:10.2307/271028
- Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. *Structural Equation Modeling: A Multidisciplinary Journal*, 23 (1), 61–75. doi:10.1080/10705511.2014.973960
- Ryan, O., & Hamaker, E. L. (2021). Time to intervene: A continuous-time approach to network analysis and centrality. *Psychometrika*, 87 (1), 214–252. doi:10.1007/s11336021097670

See Also

Other Continuous Time Mediation Functions: [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

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")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,
    -0.00600, -0.00033, 0.00110,
    0.00324, 0.00020, -0.00061,
    0.00040, 0.00374, 0.00016,
    -0.00022, -0.00273, -0.00016,
    0.00009, 0.00150, 0.00012,
    -0.00151, 0.00016, 0.00389,
    0.00103, -0.00007, -0.00283,
    -0.00050, 0.00000, 0.00156,
    -0.00600, -0.00022, 0.00103,
    0.00644, 0.00031, -0.00119,
    -0.00374, -0.00021, 0.00070,
    -0.00033, -0.00273, -0.00007,
    0.00031, 0.00287, 0.00013,
    -0.00014, -0.00170, -0.00012,
    0.00110, -0.00016, -0.00283,
  )
)
```

```

-0.00119, 0.00013, 0.00297,
0.00063, -0.00004, -0.00177,
0.00324, 0.00009, -0.00050,
-0.00374, -0.00014, 0.00063,
0.00495, 0.00024, -0.00093,
0.00020, 0.00150, 0.00000,
-0.00021, -0.00170, -0.00004,
0.00024, 0.00214, 0.00012,
-0.00061, 0.00012, 0.00156,
0.00070, -0.00012, -0.00177,
-0.00093, 0.00012, 0.00223
),
nrow = 9
)

phi <- MCPHi(
  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(
  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, ...)
```

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

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

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")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,
    -0.00600, -0.00033, 0.00110,
    0.00324, 0.00020, -0.00061,
    0.00040, 0.00374, 0.00016,
    -0.00022, -0.00273, -0.00016,
    0.00009, 0.00150, 0.00012,
    -0.00151, 0.00016, 0.00389,
    0.00103, -0.00007, -0.00283,
    -0.00050, 0.00000, 0.00156,
    -0.00600, -0.00022, 0.00103,
    0.00644, 0.00031, -0.00119,
    -0.00374, -0.00021, 0.00070,
    -0.00033, -0.00273, -0.00007,
    0.00031, 0.00287, 0.00013,
    -0.00014, -0.00170, -0.00012,
    0.00110, -0.00016, -0.00283,
    -0.00119, 0.00013, 0.00297,
  )
)
```

```

    0.00063, -0.00004, -0.00177,
    0.00324, 0.00009, -0.00050,
    -0.00374, -0.00014, 0.00063,
    0.00495, 0.00024, -0.00093,
    0.00020, 0.00150, 0.00000,
    -0.00021, -0.00170, -0.00004,
    0.00024, 0.00214, 0.00012,
    -0.00061, 0.00012, 0.00156,
    0.00070, -0.00012, -0.00177,
    -0.00093, 0.00012, 0.00223
  ),
  nrow = 9
)

# 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 = 1:5,
  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

Prints 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

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

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```

set.seed(42)
phi <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,

```



```

-0.00600, -0.00033, 0.00110,
0.00324, 0.00020, -0.00061,
0.00040, 0.00374, 0.00016,
-0.00022, -0.00273, -0.00016,
0.00009, 0.00150, 0.00012,
-0.00151, 0.00016, 0.00389,
0.00103, -0.00007, -0.00283,
-0.00050, 0.00000, 0.00156,
-0.00600, -0.00022, 0.00103,
0.00644, 0.00031, -0.00119,
-0.00374, -0.00021, 0.00070,
-0.00033, -0.00273, -0.00007,
0.00031, 0.00287, 0.00013,
-0.00014, -0.00170, -0.00012,
0.00110, -0.00016, -0.00283,
-0.00119, 0.00013, 0.00297,
0.00063, -0.00004, -0.00177,
0.00324, 0.00009, -0.00050,
-0.00374, -0.00014, 0.00063,
0.00495, 0.00024, -0.00093,
0.00020, 0.00150, 0.00000,
-0.00021, -0.00170, -0.00004,
0.00024, 0.00214, 0.00012,
-0.00061, 0.00012, 0.00156,
0.00070, -0.00012, -0.00177,
-0.00093, 0.00012, 0.00223
),
nrow = 9
)

# Specific time interval -----
mc <- 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
)
print(mc)

# Range of time intervals -----
mc <- MCMed(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1:5,
  from = "x",
  to = "y",
  med = "m",
  R = 100L # use a large value for R in actual research
)
print(mc)

```

print.ctmedmcphi	<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

Prints a list of drift matrices.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
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)
```

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

Prints a matrix of 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")

# 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 = 1:5,
  from = "x",
  to = "y",
  med = "m"
)
print(med)
```

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

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

Prints a data frame of simulated data.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

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

traj <- Trajectory(
  mu0 = c(3, 3, -3),
```

```

    time = 150,
    phi = phi,
    med = "m"
  )

  print(traj)

```

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

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")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,

```

```

-0.00600, -0.00033, 0.00110,
0.00324, 0.00020, -0.00061,
0.00040, 0.00374, 0.00016,
-0.00022, -0.00273, -0.00016,
0.00009, 0.00150, 0.00012,
-0.00151, 0.00016, 0.00389,
0.00103, -0.00007, -0.00283,
-0.00050, 0.00000, 0.00156,
-0.00600, -0.00022, 0.00103,
0.00644, 0.00031, -0.00119,
-0.00374, -0.00021, 0.00070,
-0.00033, -0.00273, -0.00007,
0.00031, 0.00287, 0.00013,
-0.00014, -0.00170, -0.00012,
0.00110, -0.00016, -0.00283,
-0.00119, 0.00013, 0.00297,
0.00063, -0.00004, -0.00177,
0.00324, 0.00009, -0.00050,
-0.00374, -0.00014, 0.00063,
0.00495, 0.00024, -0.00093,
0.00020, 0.00150, 0.00000,
-0.00021, -0.00170, -0.00004,
0.00024, 0.00214, 0.00012,
-0.00061, 0.00012, 0.00156,
0.00070, -0.00012, -0.00177,
-0.00093, 0.00012, 0.00223
),
nrow = 9
)

# 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 = 1:5,
  from = "x",
  to = "y",
  med = "m"
)
summary(delta)

```

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

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

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
set.seed(42)
phi <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")
vcov_phi_vec <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151,
    -0.00600, -0.00033, 0.00110,
    0.00324, 0.00020, -0.00061,
    0.00040, 0.00374, 0.00016,
    -0.00022, -0.00273, -0.00016,
    0.00009, 0.00150, 0.00012,
    -0.00151, 0.00016, 0.00389,
    0.00103, -0.00007, -0.00283,
```

```

-0.00050, 0.00000, 0.00156,
-0.00600, -0.00022, 0.00103,
0.00644, 0.00031, -0.00119,
-0.00374, -0.00021, 0.00070,
-0.00033, -0.00273, -0.00007,
0.00031, 0.00287, 0.00013,
-0.00014, -0.00170, -0.00012,
0.00110, -0.00016, -0.00283,
-0.00119, 0.00013, 0.00297,
0.00063, -0.00004, -0.00177,
0.00324, 0.00009, -0.00050,
-0.00374, -0.00014, 0.00063,
0.00495, 0.00024, -0.00093,
0.00020, 0.00150, 0.00000,
-0.00021, -0.00170, -0.00004,
0.00024, 0.00214, 0.00012,
-0.00061, 0.00012, 0.00156,
0.00070, -0.00012, -0.00177,
-0.00093, 0.00012, 0.00223
),
nrow = 9
)

# Specific time interval -----
mc <- 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
)
summary(mc)

# Range of time intervals -----
mc <- MCMed(
  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)

```


Description

Summary Method for an Object of Class ctmedmed

Usage

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

Arguments

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

Value

Returns a matrix of effects.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
phi <- matrix(
  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 = 1:5,
  from = "x",
  to = "y",
  med = "m"
)
```

```
)
summary(med)
```

```
summary.ctmedposteriorphi
```

Summary Method for Object of Class ctmedposteriorphi

Description

Summary Method for Object of Class ctmedposteriorphi

Usage

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

Arguments

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

Value

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

Author(s)

Ivan Jacob Agaloos Pesigan

```
summary.ctmedtraj
```

Summary Method for an Object of Class ctmedtraj

Description

Summary Method for an Object of Class ctmedtraj

Usage

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

Arguments

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

Value

Returns a data frame of simulated data.

Author(s)

Ivan Jacob Agaloos Pesigan

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)
```

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
- 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\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

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
)
```

TotalCentral	<i>Total Effect Centrality</i>
--------------	--------------------------------

Description

Total Effect Centrality

Usage

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

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>	Vector of positive numbers. Time interval (Δt).

Details

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

Value

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

call Function call.

args Function arguments.

fun Function used ("TotalCentral").

output A matrix of total effect centrality.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. *Sociological Methodology*, 17, 37. doi:[10.2307/271028](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: [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

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")

  # Specific time interval -----
  TotalCentral(
    phi = phi,
    delta_t = 1
  )

  # Range of time intervals -----
  total_central <- TotalCentral(
    phi = phi,
    delta_t = 1:30
  )
  plot(total_central)

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

```

TotalStd

Standardized Total Effect Matrix Over a Specific Time Interval

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 and process noise covariance matrix Σ .

Usage

```
TotalStd(phi, sigma, delta_t)
```

Arguments

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

Details

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

$$\text{Total}_{\Delta t}^* = \mathbf{S} (\exp(\Delta t \mathbf{\Phi})) \mathbf{S}^{-1}$$

where $\mathbf{\Phi}$ denotes the drift matrix, \mathbf{S} a diagonal matrix with model-implied standard deviations on the diagonals and Δt the time interval.

Value

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

call Function call.

args Function arguments.

fun Function used ("TotalStd").

output The matrix of total effects.

Author(s)

Ivan Jacob Agaloos Pesigan

References

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

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

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

See Also

Other Continuous Time Mediation Functions: [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [Trajectory\(\)](#)

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")
sigma <- matrix(
  data = c(
    0.24, 0.02, -0.05,
    0.02, 0.07, 0.02,
    -0.05, 0.02, 0.08
  ),
  nrow = 3
)
delta_t <- 1
TotalStd(
  phi = phi,
  sigma = sigma,
  delta_t = delta_t
)

```

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

<code>mu0</code>	Numeric vector. Initial values of the variables.
<code>time</code>	Positive integer. Number of time points.
<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>med</code>	Character vector. Name/s of the mediator variable/s in <code>phi</code> .

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\(\)](#), [DeltaBetaStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBETA\(\)](#), [MCBETAStd\(\)](#), [MCIndirectCentral\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorMed\(\)](#), [PosteriorPhi\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalStd\(\)](#)

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"
)
plot(traj)

# Methods -----
# Trajectory has a number of methods including
# print, summary, and plot

traj <- Trajectory(
  mu0 = c(3, 3, -3),
  time = 25,
  phi = phi,
  med = "m"
)
print(traj)
summary(traj)
plot(traj)
```

Index

* Continuous Time Mediation Functions

- DeltaBeta, [6](#)
- DeltaBetaStd, [10](#)
- DeltaIndirectCentral, [14](#)
- DeltaMed, [17](#)
- DeltaMedStd, [20](#)
- DeltaTotalCentral, [24](#)
- Direct, [27](#)
- DirectStd, [30](#)
- Indirect, [32](#)
- IndirectCentral, [35](#)
- IndirectStd, [36](#)
- MCBETA, [39](#)
- MCBETAStd, [42](#)
- MCIndirectCentral, [46](#)
- MCMed, [49](#)
- MCMedStd, [53](#)
- MCPHI, [57](#)
- MCTotalCentral, [59](#)
- Med, [62](#)
- MedStd, [65](#)
- PosteriorBeta, [73](#)
- PosteriorIndirectCentral, [75](#)
- PosteriorMed, [78](#)
- PosteriorPhi, [81](#)
- PosteriorTotalCentral, [82](#)
- Total, [99](#)
- TotalCentral, [101](#)
- TotalStd, [103](#)
- Trajectory, [105](#)

* beta

- DeltaBeta, [6](#)
- DeltaBetaStd, [10](#)
- MCBETA, [39](#)
- MCBETAStd, [42](#)
- PosteriorBeta, [73](#)

* cTMed

- DeltaBeta, [6](#)
- DeltaBetaStd, [10](#)

- DeltaIndirectCentral, [14](#)
- DeltaMed, [17](#)
- DeltaMedStd, [20](#)
- DeltaTotalCentral, [24](#)
- Direct, [27](#)
- DirectStd, [30](#)
- Indirect, [32](#)
- IndirectCentral, [35](#)
- IndirectStd, [36](#)
- MCBETA, [39](#)
- MCBETAStd, [42](#)
- MCIndirectCentral, [46](#)
- MCMed, [49](#)
- MCMedStd, [53](#)
- MCPHI, [57](#)
- MCTotalCentral, [59](#)
- Med, [62](#)
- MedStd, [65](#)
- PosteriorBeta, [73](#)
- PosteriorIndirectCentral, [75](#)
- PosteriorMed, [78](#)
- PosteriorPhi, [81](#)
- PosteriorTotalCentral, [82](#)
- Total, [99](#)
- TotalCentral, [101](#)
- TotalStd, [103](#)
- Trajectory, [105](#)

* delta

- DeltaBeta, [6](#)
- DeltaBetaStd, [10](#)
- DeltaIndirectCentral, [14](#)
- DeltaMed, [17](#)
- DeltaMedStd, [20](#)
- DeltaTotalCentral, [24](#)

* effects

- Direct, [27](#)
- DirectStd, [30](#)
- Indirect, [32](#)
- IndirectCentral, [35](#)

- IndirectStd, 36
- Med, 62
- MedStd, 65
- Total, 99
- TotalCentral, 101
- TotalStd, 103
- Trajectory, 105
- * mc**
 - MCBeta, 39
 - MCBetaStd, 42
 - MCIndirectCentral, 46
 - MCMed, 49
 - MCMedStd, 53
 - MCPhi, 57
 - MCTotalCentral, 59
- * methods**
 - confint.ctmeddelta, 3
 - confint.ctmedmc, 4
 - plot.ctmeddelta, 67
 - plot.ctmedmc, 69
 - plot.ctmedmed, 71
 - plot.ctmedtraj, 72
 - print.ctmeddelta, 84
 - print.ctmedeffect, 86
 - print.ctmedmc, 88
 - print.ctmedmcphi, 90
 - print.ctmedmed, 91
 - print.ctmedtraj, 92
 - summary.ctmeddelta, 93
 - summary.ctmedmc, 95
 - summary.ctmedmed, 96
 - summary.ctmedposteriorphi, 98
 - summary.ctmedtraj, 98
- * network**
 - DeltaIndirectCentral, 14
 - DeltaTotalCentral, 24
 - IndirectCentral, 35
 - MCIndirectCentral, 46
 - MCTotalCentral, 59
 - PosteriorIndirectCentral, 75
 - PosteriorTotalCentral, 82
 - TotalCentral, 101
- * path**
 - DeltaMed, 17
 - DeltaMedStd, 20
 - MCMed, 49
 - MCMedStd, 53
 - Med, 62
 - MedStd, 65
 - PosteriorMed, 78
 - Trajectory, 105
- * posterior**
 - PosteriorBeta, 73
 - PosteriorIndirectCentral, 75
 - PosteriorMed, 78
 - PosteriorPhi, 81
 - PosteriorTotalCentral, 82
- confint.ctmeddelta, 3
- confint.ctmedmc, 4
- ctsem::ctStanFit(), 81
- DeltaBeta, 6, 12, 15, 19, 22, 26, 29, 31, 34, 36, 38, 40, 44, 48, 51, 55, 58, 61, 64, 66, 74, 77, 79, 82, 83, 101, 102, 104, 106
- DeltaBetaStd, 8, 10, 15, 19, 22, 26, 29, 31, 34, 36, 38, 40, 44, 48, 51, 55, 58, 61, 64, 66, 74, 77, 79, 82, 83, 101, 102, 104, 106
- DeltaIndirectCentral, 8, 12, 14, 19, 22, 26, 29, 31, 34, 36, 38, 40, 44, 48, 51, 55, 58, 61, 64, 66, 74, 77, 79, 82, 83, 101, 102, 104, 106
- DeltaMed, 8, 12, 15, 17, 22, 26, 29, 31, 34, 36, 38, 40, 44, 48, 51, 55, 58, 61, 64, 66, 74, 77, 79, 82, 83, 101, 102, 104, 106
- DeltaMedStd, 8, 12, 15, 19, 20, 26, 29, 31, 34, 36, 38, 40, 44, 48, 51, 55, 58, 61, 64, 66, 74, 77, 79, 82, 83, 101, 102, 104, 106
- DeltaTotalCentral, 8, 12, 15, 19, 22, 24, 29, 31, 34, 36, 38, 40, 44, 48, 51, 55, 58, 61, 64, 66, 74, 77, 79, 82, 83, 101, 102, 104, 106
- Direct, 8, 12, 15, 19, 22, 26, 27, 31, 34, 36, 38, 40, 44, 48, 51, 55, 58, 61, 64, 66, 74, 77, 79, 82, 83, 101, 102, 104, 106
- Direct(), 17, 50, 63, 79
- DirectStd, 8, 12, 15, 19, 22, 26, 29, 30, 34, 36, 38, 40, 44, 48, 51, 55, 58, 61, 64, 66, 74, 77, 79, 82, 83, 101, 102, 104, 106
- DirectStd(), 21, 54, 65
- Indirect, 8, 12, 15, 19, 22, 26, 29, 31, 32, 36, 38, 40, 44, 48, 51, 55, 58, 61, 64, 66, 74, 77, 79, 82, 83, 101, 102, 104, 106

- Indirect(), [17](#), [50](#), [63](#), [79](#)
- IndirectCentral, [8](#), [12](#), [15](#), [19](#), [22](#), [26](#), [29](#),
[31](#), [34](#), [35](#), [38](#), [40](#), [44](#), [48](#), [51](#), [55](#), [58](#),
[61](#), [64](#), [66](#), [74](#), [77](#), [79](#), [82](#), [83](#), [101](#),
[102](#), [104](#), [106](#)
- IndirectCentral(), [14](#), [47](#)
- IndirectStd, [8](#), [12](#), [15](#), [19](#), [22](#), [26](#), [29](#), [31](#), [34](#),
[36](#), [36](#), [40](#), [44](#), [48](#), [51](#), [55](#), [58](#), [61](#), [64](#),
[66](#), [74](#), [77](#), [79](#), [82](#), [83](#), [101](#), [102](#), [104](#),
[106](#)
- IndirectStd(), [21](#), [54](#), [65](#)
- MCBeta, [8](#), [12](#), [15](#), [19](#), [22](#), [26](#), [29](#), [31](#), [34](#), [36](#),
[38](#), [39](#), [44](#), [48](#), [51](#), [55](#), [58](#), [61](#), [64](#), [66](#),
[74](#), [77](#), [79](#), [82](#), [83](#), [101](#), [102](#), [104](#), [106](#)
- MCBetaStd, [8](#), [12](#), [15](#), [19](#), [22](#), [26](#), [29](#), [31](#), [34](#),
[36](#), [38](#), [40](#), [42](#), [48](#), [51](#), [55](#), [58](#), [61](#), [64](#),
[66](#), [74](#), [77](#), [79](#), [82](#), [83](#), [101](#), [102](#), [104](#),
[106](#)
- MCIndirectCentral, [8](#), [12](#), [15](#), [19](#), [22](#), [26](#), [29](#),
[31](#), [34](#), [36](#), [38](#), [40](#), [44](#), [46](#), [51](#), [55](#), [58](#),
[61](#), [64](#), [66](#), [74](#), [77](#), [79](#), [82](#), [83](#), [101](#),
[102](#), [104](#), [106](#)
- MCMed, [8](#), [12](#), [15](#), [19](#), [22](#), [26](#), [29](#), [31](#), [34](#), [36](#), [38](#),
[40](#), [44](#), [48](#), [49](#), [55](#), [58](#), [61](#), [64](#), [66](#), [74](#),
[77](#), [79](#), [82](#), [83](#), [101](#), [102](#), [104](#), [106](#)
- MCMedStd, [8](#), [12](#), [15](#), [19](#), [22](#), [26](#), [29](#), [31](#), [34](#), [36](#),
[38](#), [40](#), [44](#), [48](#), [51](#), [53](#), [58](#), [61](#), [64](#), [66](#),
[74](#), [77](#), [79](#), [82](#), [83](#), [101](#), [102](#), [104](#), [106](#)
- MCPHi, [8](#), [12](#), [15](#), [19](#), [22](#), [26](#), [29](#), [31](#), [34](#), [36](#), [38](#),
[40](#), [44](#), [48](#), [51](#), [55](#), [57](#), [61](#), [64](#), [66](#), [74](#),
[77](#), [79](#), [82](#), [83](#), [101](#), [102](#), [104](#), [106](#)
- MCTotalCentral, [8](#), [12](#), [15](#), [19](#), [22](#), [26](#), [29](#), [31](#),
[34](#), [36](#), [38](#), [40](#), [44](#), [48](#), [51](#), [55](#), [58](#), [59](#),
[64](#), [66](#), [74](#), [77](#), [79](#), [82](#), [83](#), [101](#), [102](#),
[104](#), [106](#)
- Med, [8](#), [12](#), [15](#), [19](#), [22](#), [26](#), [29](#), [31](#), [34](#), [36](#), [38](#),
[40](#), [44](#), [48](#), [51](#), [55](#), [58](#), [61](#), [62](#), [66](#), [74](#),
[77](#), [79](#), [82](#), [83](#), [101](#), [102](#), [104](#), [106](#)
- MedStd, [8](#), [12](#), [15](#), [19](#), [22](#), [26](#), [29](#), [31](#), [34](#), [36](#),
[38](#), [40](#), [44](#), [48](#), [51](#), [55](#), [58](#), [61](#), [64](#), [65](#),
[74](#), [77](#), [79](#), [82](#), [83](#), [101](#), [102](#), [104](#), [106](#)
- plot.ctmeddelta, [67](#)
- plot.ctmedmc, [69](#)
- plot.ctmedmed, [71](#)
- plot.ctmedtraj, [72](#)
- PosteriorBeta, [8](#), [12](#), [15](#), [19](#), [22](#), [26](#), [29](#), [31](#),
[34](#), [36](#), [38](#), [40](#), [44](#), [48](#), [51](#), [55](#), [58](#), [61](#),
[64](#), [66](#), [73](#), [77](#), [79](#), [82](#), [83](#), [101](#), [102](#),
[104](#), [106](#)
- PosteriorIndirectCentral, [8](#), [12](#), [15](#), [19](#),
[22](#), [26](#), [29](#), [31](#), [34](#), [36](#), [38](#), [40](#), [44](#), [48](#),
[51](#), [55](#), [58](#), [61](#), [64](#), [66](#), [74](#), [75](#), [79](#), [82](#),
[83](#), [101](#), [102](#), [104](#), [106](#)
- PosteriorMed, [8](#), [12](#), [15](#), [19](#), [22](#), [26](#), [29](#), [31](#),
[34](#), [36](#), [38](#), [40](#), [44](#), [48](#), [51](#), [55](#), [58](#), [61](#),
[64](#), [66](#), [74](#), [77](#), [78](#), [82](#), [83](#), [101](#), [102](#),
[104](#), [106](#)
- PosteriorPhi, [8](#), [12](#), [15](#), [19](#), [22](#), [26](#), [29](#), [31](#),
[34](#), [36](#), [38](#), [40](#), [44](#), [48](#), [51](#), [55](#), [58](#), [61](#),
[64](#), [66](#), [74](#), [77](#), [79](#), [81](#), [83](#), [101](#), [102](#),
[104](#), [106](#)
- PosteriorTotalCentral, [8](#), [12](#), [15](#), [19](#), [22](#),
[26](#), [29](#), [31](#), [34](#), [36](#), [38](#), [40](#), [44](#), [48](#), [51](#),
[55](#), [58](#), [61](#), [64](#), [66](#), [74](#), [77](#), [79](#), [82](#), [82](#),
[101](#), [102](#), [104](#), [106](#)
- print.ctmeddelta, [84](#)
- print.ctmedeffect, [86](#)
- print.ctmedmc, [88](#)
- print.ctmedmcphi, [90](#)
- print.ctmedmed, [91](#)
- print.ctmedtraj, [92](#)
- summary.ctmeddelta, [93](#)
- summary.ctmedmc, [95](#)
- summary.ctmedmed, [96](#)
- summary.ctmedposteriorphi, [98](#)
- summary.ctmedtraj, [98](#)
- Total, [8](#), [12](#), [15](#), [19](#), [22](#), [26](#), [29](#), [31](#), [34](#), [36](#), [38](#),
[40](#), [44](#), [48](#), [51](#), [55](#), [58](#), [61](#), [64](#), [66](#), [74](#),
[77](#), [79](#), [82](#), [83](#), [99](#), [102](#), [104](#), [106](#)
- Total(), [7](#), [17](#), [39](#), [50](#), [63](#), [73](#), [79](#)
- TotalCentral, [8](#), [12](#), [15](#), [19](#), [22](#), [26](#), [29](#), [31](#),
[34](#), [36](#), [38](#), [40](#), [44](#), [48](#), [51](#), [55](#), [58](#), [61](#),
[64](#), [66](#), [74](#), [77](#), [79](#), [82](#), [83](#), [101](#), [101](#),
[104](#), [106](#)
- TotalCentral(), [25](#), [60](#), [76](#), [82](#)
- TotalStd, [8](#), [12](#), [15](#), [19](#), [22](#), [26](#), [29](#), [31](#), [34](#), [36](#),
[38](#), [40](#), [44](#), [48](#), [51](#), [55](#), [58](#), [61](#), [64](#), [66](#),
[74](#), [77](#), [79](#), [82](#), [83](#), [101](#), [102](#), [103](#), [106](#)
- TotalStd(), [10](#), [21](#), [43](#), [54](#), [65](#)
- Trajectory, [8](#), [12](#), [15](#), [19](#), [22](#), [26](#), [29](#), [31](#), [34](#),
[36](#), [38](#), [40](#), [44](#), [48](#), [51](#), [55](#), [58](#), [61](#), [64](#),
[66](#), [74](#), [77](#), [79](#), [82](#), [83](#), [101](#), [102](#), [104](#),
[105](#)