

Package ‘cTMed’

July 21, 2026

Title Continuous-Time Mediation

Version 1.0.10

Description Computes effect sizes, standard errors, and confidence intervals for total, direct, and indirect effects in continuous-time mediation models as described in Pesigan, Russell, and Chow (2025) <[doi:10.1037/met0000779](https://doi.org/10.1037/met0000779)>.

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

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

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 3.5.0)

LinkingTo Rcpp (>= 1.0.12), RcppArmadillo (>= 15.0.2-2)

Imports Rcpp (>= 1.0.12), numDeriv, parallel, simStateSpace

Suggests knitr, rmarkdown, testthat, expm, dynr, betaDelta,
bootStateSpace

SystemRequirements GNU GSL (>= 2.5)

Config/roxygen2/version 8.0.0

NeedsCompilation yes

Author Ivan Jacob Agaloos Pesigan [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0003-4818-8420>>),
Michael A. Russell [ctb] (ORCID:
<<https://orcid.org/0000-0002-3956-604X>>),
Sy-Miin Chow [ctb] (ORCID: <<https://orcid.org/0000-0003-1938-027X>>)

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

Repository CRAN

Date/Publication 2026-07-21 13:40:02 UTC

Contents

BootBeta	3
BootBetaStd	7
BootDirectCentral	11
BootDirectCentralStd	15
BootIndirectCentral	19
BootIndirectCentralStd	23
BootMed	27
BootMedStd	31
BootTotalCentral	35
BootTotalCentralStd	39
confint.ctmedboot	43
confint.ctmeddelta	46
confint.ctmedmc	48
DeltaBeta	50
DeltaBetaStd	53
DeltaDirectCentral	58
DeltaDirectCentralStd	61
DeltaIndirectCentral	65
DeltaIndirectCentralStd	69
DeltaMed	73
DeltaMedStd	76
DeltaTotalCentral	81
DeltaTotalCentralStd	84
Direct	88
DirectCentral	91
DirectCentralStd	93
DirectStd	95
Indirect	97
IndirectCentral	100
IndirectCentralStd	102
IndirectStd	104
MCBeta	106
MCBetaStd	110
MCDirectCentral	114
MCDirectCentralStd	118
MCIndirectCentral	122
MCIndirectCentralStd	126
MCMed	130
MCMedStd	134
MCPhi	138
MCPhiSigma	140
MCTotalCentral	143
MCTotalCentralStd	146
Med	150
MedStd	153
plot.ctmedboot	156

plot.ctmeddelta	158
plot.ctmedmc	160
plot.ctmedmed	162
plot.ctmedtraj	163
PosteriorBeta	164
PosteriorBetaStd	167
PosteriorDirectCentral	170
PosteriorDirectCentralStd	173
PosteriorIndirectCentral	176
PosteriorIndirectCentralStd	179
PosteriorMed	182
PosteriorMedStd	185
PosteriorTotalCentral	188
PosteriorTotalCentralStd	191
print.ctmedboot	194
print.ctmeddelta	197
print.ctmedeffect	199
print.ctmedmc	200
print.ctmedmcphi	202
print.ctmedmed	203
print.ctmedtraj	204
summary.ctmedboot	205
summary.ctmeddelta	208
summary.ctmedmc	210
summary.ctmedmed	212
summary.ctmedposteriorphi	213
summary.ctmedtraj	214
Total	215
TotalCentral	217
TotalCentralStd	219
TotalStd	221
Trajectory	223

Index**226**

BootBeta

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

Description

This function generates a bootstrap 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

```
BootBeta(phi, phi_hat, delta_t, ncores = NULL, tol = 0.001)
```

Arguments

phi	List of numeric matrices. Each element of the list is a bootstrap estimate of the drift matrix (Φ).
phi_hat	Numeric matrix. The estimated drift matrix ($\hat{\Phi}$) from the original data set. phi_hat 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.
tol	Numeric. Smallest possible time interval to allow.

Details

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

Value

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

call Function call.

args Function arguments.

fun Function used ("BootBeta").

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

Each element in the output list has the following elements:

est Estimated elements of the matrix of lagged coefficients.

thetahatstar A matrix of bootstrap elements of the matrix of lagged coefficients.

Author(s)

Ivan Jacob Agaloos Pesigan

References

- Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. *Sociological Methodology*, 17, 37. [doi:10.2307/271028](#)
- Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. *Structural Equation Modeling: A Multidisciplinary Journal*, 23 (1), 61-75. [doi:10.1080/10705511.2014.973960](#)
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](#)
- 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: [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

Examples

```
## Not run:
library(bootStateSpace)
# prepare parameters
## number of individuals
n <- 50
## time points
time <- 100
delta_t <- 0.10
## dynamic structure
p <- 3
mu0 <- rep(x = 0, times = p)
sigma0 <- matrix(
  data = c(
    1.0,
    0.2,
    0.2,
    0.2,
    1.0,
    0.2,
    0.2,
    0.2,
    1.0
  ),
  nrow = p
)
sigma0_l <- t(chol(sigma0))
mu <- rep(x = 0, times = p)
phi <- matrix(
  data = c(
    -0.357,
    0.771,
    -0.450,
    0.0,
    -0.511,
    0.729,

```

```

      0,
      0,
      -0.693
    ),
    nrow = p
  )
  sigma <- matrix(
    data = c(
      0.24455556,
      0.02201587,
      -0.05004762,
      0.02201587,
      0.07067800,
      0.01539456,
      -0.05004762,
      0.01539456,
      0.07553061
    ),
    nrow = p
  )
  sigma_l <- t(chol(sigma))
  ## measurement model
  k <- 3
  nu <- rep(x = 0, times = k)
  lambda <- diag(k)
  theta <- 0.2 * diag(k)
  theta_l <- t(chol(theta))

  boot <- PBSSMOUFixed(
    R = 10L, # use at least 1000 in actual research
    path = getwd(),
    prefix = "ou",
    n = n,
    time = time,
    delta_t = delta_t,
    mu0 = mu0,
    sigma0_l = sigma0_l,
    mu = mu,
    phi = phi,
    sigma_l = sigma_l,
    nu = nu,
    lambda = lambda,
    theta_l = theta_l,
    ncores = NULL, # consider using multiple cores
    seed = 42
  )
  phi_hat <- phi
  colnames(phi_hat) <- rownames(phi_hat) <- c("x", "m", "y")
  phi <- extract(object = boot, what = "phi")

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

```

```

    phi_hat = phi_hat,
    delta_t = 1
)

# Range of time intervals -----
boot <- BootBeta(
  phi = phi,
  phi_hat = phi_hat,
  delta_t = 1:5
)
plot(boot)
plot(boot, type = "bc") # bias-corrected

# Methods -----
# BootBeta has a number of methods including
# print, summary, confint, and plot
print(boot)
summary(boot)
confint(boot, level = 0.95)
print(boot, type = "bc") # bias-corrected
summary(boot, type = "bc")
confint(boot, level = 0.95, type = "bc")

## End(Not run)

```

BootBetaStd

Bootstrap 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 bootstrap 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

```

BootBetaStd(
  phi,
  sigma,
  phi_hat,
  sigma_hat,
  delta_t,
  ncores = NULL,
  tol = 0.001
)

```

Arguments

phi	List of numeric matrices. Each element of the list is a bootstrap estimate of the drift matrix (Φ).
sigma	List of numeric matrices. Each element of the list is a bootstrap estimate of the process noise covariance matrix (Σ).
phi_hat	Numeric matrix. The estimated drift matrix ($\hat{\Phi}$) from the original data set. phi_hat should have row and column names pertaining to the variables in the system.
sigma_hat	Numeric matrix. The estimated process noise covariance matrix ($\hat{\Sigma}$) from the original data set.
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.
tol	Numeric. Smallest possible time interval to allow.

Details

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

Value

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

call Function call.

args Function arguments.

fun Function used ("BootBetaStd").

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

Each element in the output list has the following elements:

est Estimated elements of the standardized matrix of lagged coefficients.

thetahatstar A matrix of bootstrap elements of the standardized matrix of lagged coefficients.

Author(s)

Ivan Jacob Agaloos Pesigan

References

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

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

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](#)

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: [BootBeta\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

Examples

```
## Not run:
library(bootStateSpace)
# prepare parameters
## number of individuals
n <- 50
## time points
time <- 100
delta_t <- 0.10
## dynamic structure
p <- 3
mu0 <- rep(x = 0, times = p)
sigma0 <- matrix(
  data = c(
    1.0,
    0.2,
    0.2,
    0.2,
    1.0,
    0.2,
    0.2,
    0.2,
    1.0
  ),
  nrow = p
)
sigma0_l <- t(chol(sigma0))
mu <- rep(x = 0, times = p)
phi <- matrix(
  data = c(
    -0.357,
    0.771,
    -0.450,
```

```

    0.0,
    -0.511,
    0.729,
    0,
    0,
    -0.693
  ),
  nrow = p
)
sigma <- matrix(
  data = c(
    0.24455556,
    0.02201587,
    -0.05004762,
    0.02201587,
    0.07067800,
    0.01539456,
    -0.05004762,
    0.01539456,
    0.07553061
  ),
  nrow = p
)
sigma_l <- t(chol(sigma))
## measurement model
k <- 3
nu <- rep(x = 0, times = k)
lambda <- diag(k)
theta <- 0.2 * diag(k)
theta_l <- t(chol(theta))

boot <- PBSSMOUFixed(
  R = 10L, # use at least 1000 in actual research
  path = getwd(),
  prefix = "ou",
  n = n,
  time = time,
  delta_t = delta_t,
  mu0 = mu0,
  sigma0_l = sigma0_l,
  mu = mu,
  phi = phi,
  sigma_l = sigma_l,
  nu = nu,
  lambda = lambda,
  theta_l = theta_l,
  ncores = NULL, # consider using multiple cores
  seed = 42
)
phi_hat <- phi
colnames(phi_hat) <- rownames(phi_hat) <- c("x", "m", "y")
sigma_hat <- sigma
phi <- extract(object = boot, what = "phi")

```

```

sigma <- extract(object = boot, what = "sigma")

# Specific time interval -----
BootBetaStd(
  phi = phi,
  sigma = sigma,
  phi_hat = phi_hat,
  sigma_hat = sigma_hat,
  delta_t = 1
)

# Range of time intervals -----
boot <- BootBetaStd(
  phi = phi,
  sigma = sigma,
  phi_hat = phi_hat,
  sigma_hat = sigma_hat,
  delta_t = 1:5
)
plot(boot)
plot(boot, type = "bc") # bias-corrected

# Methods -----
# BootBetaStd has a number of methods including
# print, summary, confint, and plot
print(boot)
summary(boot)
confint(boot, level = 0.95)
print(boot, type = "bc") # bias-corrected
summary(boot, type = "bc")
confint(boot, level = 0.95, type = "bc")

## End(Not run)

```

BootDirectCentral

*Bootstrap Sampling Distribution for the Direct Effect Centrality Over
a Specific Time Interval or a Range of Time Intervals*

Description

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

Usage

```
BootDirectCentral(phi, phi_hat, delta_t, ncores = NULL, tol = 0.001)
```

Arguments

phi	List of numeric matrices. Each element of the list is a bootstrap estimate of the drift matrix (Φ).
phi_hat	Numeric matrix. The estimated drift matrix ($\hat{\Phi}$) from the original data set. phi_hat 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.
tol	Numeric. Smallest possible time interval to allow.

Details

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

Value

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

call Function call.

args Function arguments.

fun Function used ("BootDirectCentral").

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

Each element in the output list has the following elements:

est A vector of direct effect centrality.

thetahatstar A matrix of bootstrap direct 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](#)
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](#)
- 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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

Examples

```
## Not run:
library(bootStateSpace)
# prepare parameters
## number of individuals
n <- 50
## time points
time <- 100
delta_t <- 0.10
## dynamic structure
p <- 3
mu0 <- rep(x = 0, times = p)
sigma0 <- matrix(
  data = c(
    1.0,
    0.2,
    0.2,
    0.2,
    1.0,
    0.2,
    0.2,
    0.2,
    1.0
  ),
  nrow = p
)
sigma0_l <- t(chol(sigma0))
mu <- rep(x = 0, times = p)
phi <- matrix(
  data = c(
    -0.357,
    0.771,
    -0.450,
    0.0,
    -0.511,
    0.729,

```

```

      0,
      0,
      -0.693
    ),
    nrow = p
  )
sigma <- matrix(
  data = c(
    0.24455556,
    0.02201587,
    -0.05004762,
    0.02201587,
    0.07067800,
    0.01539456,
    -0.05004762,
    0.01539456,
    0.07553061
  ),
  nrow = p
)
sigma_l <- t(chol(sigma))
## measurement model
k <- 3
nu <- rep(x = 0, times = k)
lambda <- diag(k)
theta <- 0.2 * diag(k)
theta_l <- t(chol(theta))

boot <- PBSSMOUFixed(
  R = 10L, # use at least 1000 in actual research
  path = getwd(),
  prefix = "ou",
  n = n,
  time = time,
  delta_t = delta_t,
  mu0 = mu0,
  sigma0_l = sigma0_l,
  mu = mu,
  phi = phi,
  sigma_l = sigma_l,
  nu = nu,
  lambda = lambda,
  theta_l = theta_l,
  ncores = NULL, # consider using multiple cores
  seed = 42
)
phi_hat <- phi
colnames(phi_hat) <- rownames(phi_hat) <- c("x", "m", "y")
phi <- extract(object = boot, what = "phi")

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

```

```

    phi_hat = phi_hat,
    delta_t = 1
)

# Range of time intervals -----
boot <- BootDirectCentral(
  phi = phi,
  phi_hat = phi_hat,
  delta_t = 1:5
)
plot(boot)
plot(boot, type = "bc") # bias-corrected

# Methods -----
# BootDirectCentral has a number of methods including
# print, summary, confint, and plot
print(boot)
summary(boot)
confint(boot, level = 0.95)
print(boot, type = "bc") # bias-corrected
summary(boot, type = "bc")
confint(boot, level = 0.95, type = "bc")

## End(Not run)

```

BootDirectCentralStd	<i>Bootstrap Sampling Distribution for the Standardized Direct Effect Centrality Over a Specific Time Interval or a Range of Time Intervals</i>
----------------------	---

Description

This function generates a bootstrap method sampling distribution for the standardized direct effect centrality 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

```

BootDirectCentralStd(
  phi,
  sigma,
  phi_hat,
  sigma_hat,
  delta_t,
  ncores = NULL,
  tol = 0.001
)

```

Arguments

phi	List of numeric matrices. Each element of the list is a bootstrap estimate of the drift matrix (Φ).
sigma	List of numeric matrices. Each element of the list is a bootstrap estimate of the process noise covariance matrix (Σ).
phi_hat	Numeric matrix. The estimated drift matrix ($\hat{\Phi}$) from the original data set. phi_hat should have row and column names pertaining to the variables in the system.
sigma_hat	Numeric matrix. The estimated process noise covariance matrix ($\hat{\Sigma}$) from the original data set.
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.
tol	Numeric. Smallest possible time interval to allow.

Details

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

Value

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

call Function call.

args Function arguments.

fun Function used ("BootDirectCentralStd").

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

Each element in the output list has the following elements:

est A vector of standardized direct effect centrality.

thetahatstar A matrix of bootstrap standardized direct effect centrality.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Continuous-Time Mediation Functions: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#),

```

PosteriorDirectCentral(),PosteriorDirectCentralStd(),PosteriorIndirectCentral(),
PosteriorIndirectCentralStd(),PosteriorMed(),PosteriorMedStd(),PosteriorTotalCentral(),
PosteriorTotalCentralStd(), Total(), TotalCentral(), TotalCentralStd(), TotalStd(),
Trajectory()

```

Examples

```

## Not run:
library(bootStateSpace)
# prepare parameters
## number of individuals
n <- 50
## time points
time <- 100
delta_t <- 0.10
## dynamic structure
p <- 3
mu0 <- rep(x = 0, times = p)
sigma0 <- matrix(
  data = c(
    1.0,
    0.2,
    0.2,
    0.2,
    1.0,
    0.2,
    0.2,
    0.2,
    1.0
  ),
  nrow = p
)
sigma0_l <- t(chol(sigma0))
mu <- rep(x = 0, times = p)
phi <- matrix(
  data = c(
    -0.357,
    0.771,
    -0.450,
    0.0,
    -0.511,
    0.729,
    0,
    0,
    -0.693
  ),
  nrow = p
)
sigma <- matrix(
  data = c(
    0.24455556,
    0.02201587,

```

```

      -0.05004762,
      0.02201587,
      0.07067800,
      0.01539456,
      -0.05004762,
      0.01539456,
      0.07553061
    ),
    nrow = p
  )
  sigma_l <- t(chol(sigma))
  ## measurement model
  k <- 3
  nu <- rep(x = 0, times = k)
  lambda <- diag(k)
  theta <- 0.2 * diag(k)
  theta_l <- t(chol(theta))

  boot <- PBSSMOUFixed(
    R = 10L, # use at least 1000 in actual research
    path = getwd(),
    prefix = "ou",
    n = n,
    time = time,
    delta_t = delta_t,
    mu0 = mu0,
    sigma0_l = sigma0_l,
    mu = mu,
    phi = phi,
    sigma_l = sigma_l,
    nu = nu,
    lambda = lambda,
    theta_l = theta_l,
    ncores = NULL, # consider using multiple cores
    seed = 42
  )
  phi_hat <- phi
  sigma_hat <- sigma
  colnames(phi_hat) <- rownames(phi_hat) <- c("x", "m", "y")
  phi <- extract(object = boot, what = "phi")
  sigma <- extract(object = boot, what = "sigma")

  # Specific time interval -----
  BootDirectCentralStd(
    phi = phi,
    sigma = sigma,
    phi_hat = phi_hat,
    sigma_hat = sigma_hat,
    delta_t = 1
  )

  # Range of time intervals -----
  boot <- BootDirectCentralStd(

```

```

    phi = phi,
    sigma = sigma,
    phi_hat = phi_hat,
    sigma_hat = sigma_hat,
    delta_t = 1:5
  )
  plot(boot)
  plot(boot, type = "bc") # bias-corrected

# Methods -----
# BootDirectCentralStd has a number of methods including
# print, summary, confint, and plot
print(boot)
summary(boot)
confint(boot, level = 0.95)
print(boot, type = "bc") # bias-corrected
summary(boot, type = "bc")
confint(boot, level = 0.95, type = "bc")

## End(Not run)

```

BootIndirectCentral	<i>Bootstrap Sampling Distribution for the Indirect Effect Centrality Over a Specific Time Interval or a Range of Time Intervals</i>
---------------------	--

Description

This function generates a bootstrap method sampling distribution 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 drift matrix Φ .

Usage

```
BootIndirectCentral(phi, phi_hat, delta_t, ncores = NULL, tol = 0.001)
```

Arguments

phi	List of numeric matrices. Each element of the list is a bootstrap estimate of the drift matrix (Φ).
phi_hat	Numeric matrix. The estimated drift matrix ($\hat{\Phi}$) from the original data set. phi_hat 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.
tol	Numeric. Smallest possible time interval to allow.

Details

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

Value

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

call Function call.

args Function arguments.

fun Function used ("BootIndirectCentral").

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

Each element in the output list has the following elements:

est A vector of indirect effect centrality.

thetahatstar A matrix of bootstrap 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](#)

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](#)

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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

Examples

```

## Not run:
library(bootStateSpace)
# prepare parameters
## number of individuals
n <- 50
## time points
time <- 100
delta_t <- 0.10
## dynamic structure
p <- 3
mu0 <- rep(x = 0, times = p)
sigma0 <- matrix(
  data = c(
    1.0,
    0.2,
    0.2,
    0.2,
    1.0,
    0.2,
    0.2,
    0.2,
    1.0
  ),
  nrow = p
)
sigma0_l <- t(chol(sigma0))
mu <- rep(x = 0, times = p)
phi <- matrix(
  data = c(
    -0.357,
    0.771,
    -0.450,
    0.0,
    -0.511,
    0.729,
    0,
    0,
    -0.693
  ),
  nrow = p
)
sigma <- matrix(
  data = c(
    0.24455556,
    0.02201587,
    -0.05004762,
    0.02201587,
    0.07067800,
    0.01539456,
    -0.05004762,
    0.01539456,

```

```

    0.07553061
  ),
  nrow = p
)
sigma_l <- t(chol(sigma))
## measurement model
k <- 3
nu <- rep(x = 0, times = k)
lambda <- diag(k)
theta <- 0.2 * diag(k)
theta_l <- t(chol(theta))

boot <- PBSSMOUFixed(
  R = 10L, # use at least 1000 in actual research
  path = getwd(),
  prefix = "ou",
  n = n,
  time = time,
  delta_t = delta_t,
  mu0 = mu0,
  sigma0_l = sigma0_l,
  mu = mu,
  phi = phi,
  sigma_l = sigma_l,
  nu = nu,
  lambda = lambda,
  theta_l = theta_l,
  ncores = NULL, # consider using multiple cores
  seed = 42
)
phi_hat <- phi
colnames(phi_hat) <- rownames(phi_hat) <- c("x", "m", "y")
phi <- extract(object = boot, what = "phi")

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

# Range of time intervals -----
boot <- BootIndirectCentral(
  phi = phi,
  phi_hat = phi_hat,
  delta_t = 1:5
)
plot(boot)
plot(boot, type = "bc") # bias-corrected

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

```

```

print(boot)
summary(boot)
confint(boot, level = 0.95)
print(boot, type = "bc") # bias-corrected
summary(boot, type = "bc")
confint(boot, level = 0.95, type = "bc")

## End(Not run)

```

BootIndirectCentralStd

*Bootstrap Sampling Distribution for the Standardized Indirect Effect
Centrality Over a Specific Time Interval or a Range of Time Intervals*

Description

This function generates a bootstrap method sampling distribution for the standardized indirect effect centrality 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

```

BootIndirectCentralStd(
  phi,
  sigma,
  phi_hat,
  sigma_hat,
  delta_t,
  ncores = NULL,
  tol = 0.001
)

```

Arguments

phi	List of numeric matrices. Each element of the list is a bootstrap estimate of the drift matrix (Φ).
sigma	List of numeric matrices. Each element of the list is a bootstrap estimate of the process noise covariance matrix (Σ).
phi_hat	Numeric matrix. The estimated drift matrix ($\hat{\Phi}$) from the original data set. phi_hat should have row and column names pertaining to the variables in the system.
sigma_hat	Numeric matrix. The estimated process noise covariance matrix ($\hat{\Sigma}$) from the original data set.
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.
tol	Numeric. Smallest possible time interval to allow.

Details

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

Value

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

call Function call.

args Function arguments.

fun Function used ("BootIndirectCentralStd").

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

Each element in the output list has the following elements:

est A vector of standardized indirect effect centrality.

thetahatstar A matrix of bootstrap standardized indirect effect centrality.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Continuous-Time Mediation Functions: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

Examples

```
## Not run:
library(bootStateSpace)
# prepare parameters
## number of individuals
n <- 50
## time points
time <- 100
delta_t <- 0.10
## dynamic structure
p <- 3
mu0 <- rep(x = 0, times = p)
sigma0 <- matrix(
```

```

    data = c(
      1.0,
      0.2,
      0.2,
      0.2,
      1.0,
      0.2,
      0.2,
      0.2,
      1.0
    ),
    nrow = p
  )
  sigma0_l <- t(chol(sigma0))
  mu <- rep(x = 0, times = p)
  phi <- matrix(
    data = c(
      -0.357,
      0.771,
      -0.450,
      0.0,
      -0.511,
      0.729,
      0,
      0,
      -0.693
    ),
    nrow = p
  )
  sigma <- matrix(
    data = c(
      0.24455556,
      0.02201587,
      -0.05004762,
      0.02201587,
      0.07067800,
      0.01539456,
      -0.05004762,
      0.01539456,
      0.07553061
    ),
    nrow = p
  )
  sigma_l <- t(chol(sigma))
  ## measurement model
  k <- 3
  nu <- rep(x = 0, times = k)
  lambda <- diag(k)
  theta <- 0.2 * diag(k)
  theta_l <- t(chol(theta))

  boot <- PBSSMOUFixed(
    R = 10L, # use at least 1000 in actual research

```

```

path = getwd(),
prefix = "ou",
n = n,
time = time,
delta_t = delta_t,
mu0 = mu0,
sigma0_l = sigma0_l,
mu = mu,
phi = phi,
sigma_l = sigma_l,
nu = nu,
lambda = lambda,
theta_l = theta_l,
ncores = NULL, # consider using multiple cores
seed = 42
)
phi_hat <- phi
sigma_hat <- sigma
colnames(phi_hat) <- rownames(phi_hat) <- c("x", "m", "y")
phi <- extract(object = boot, what = "phi")
sigma <- extract(object = boot, what = "sigma")

# Specific time interval -----
BootIndirectCentralStd(
  phi = phi,
  sigma = sigma,
  phi_hat = phi_hat,
  sigma_hat = sigma_hat,
  delta_t = 1
)

# Range of time intervals -----
boot <- BootIndirectCentralStd(
  phi = phi,
  sigma = sigma,
  phi_hat = phi_hat,
  sigma_hat = sigma_hat,
  delta_t = 1:5
)
plot(boot)
plot(boot, type = "bc") # bias-corrected

# Methods -----
# BootIndirectCentralStd has a number of methods including
# print, summary, confint, and plot
print(boot)
summary(boot)
confint(boot, level = 0.95)
print(boot, type = "bc") # bias-corrected
summary(boot, type = "bc")
confint(boot, level = 0.95, type = "bc")

## End(Not run)

```

BootMed	<i>Bootstrap 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</i>
---------	---

Description

This function generates a bootstrap 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

```
BootMed(phi, phi_hat, delta_t, from, to, med, ncores = NULL, tol = 0.001)
```

Arguments

phi	List of numeric matrices. Each element of the list is a bootstrap estimate of the drift matrix (Φ).
phi_hat	Numeric matrix. The estimated drift matrix ($\hat{\Phi}$) from the original data set. phi_hat 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.
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.
tol	Numeric. Smallest possible time interval to allow.

Details

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

Value

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

call Function call.

args Function arguments.

fun Function used ("BootMed").

output A list of length `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 bootstrap 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)

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:[10.1037/met0000779](https://doi.org/10.1037/met0000779)

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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

Examples

```
## Not run:
library(bootStateSpace)
# prepare parameters
## number of individuals
n <- 50
## time points
time <- 100
delta_t <- 0.10
## dynamic structure
p <- 3
```

```

mu0 <- rep(x = 0, times = p)
sigma0 <- matrix(
  data = c(
    1.0,
    0.2,
    0.2,
    0.2,
    1.0,
    0.2,
    0.2,
    0.2,
    1.0
  ),
  nrow = p
)
sigma0_l <- t(chol(sigma0))
mu <- rep(x = 0, times = p)
phi <- matrix(
  data = c(
    -0.357,
    0.771,
    -0.450,
    0.0,
    -0.511,
    0.729,
    0,
    0,
    -0.693
  ),
  nrow = p
)
sigma <- matrix(
  data = c(
    0.24455556,
    0.02201587,
    -0.05004762,
    0.02201587,
    0.07067800,
    0.01539456,
    -0.05004762,
    0.01539456,
    0.07553061
  ),
  nrow = p
)
sigma_l <- t(chol(sigma))
## measurement model
k <- 3
nu <- rep(x = 0, times = k)
lambda <- diag(k)
theta <- 0.2 * diag(k)
theta_l <- t(chol(theta))

```

```

boot <- PBSSMOUFixed(
  R = 10L, # use at least 1000 in actual research
  path = getwd(),
  prefix = "ou",
  n = n,
  time = time,
  delta_t = delta_t,
  mu0 = mu0,
  sigma0_l = sigma0_l,
  mu = mu,
  phi = phi,
  sigma_l = sigma_l,
  nu = nu,
  lambda = lambda,
  theta_l = theta_l,
  ncores = NULL, # consider using multiple cores
  seed = 42
)
phi_hat <- phi
colnames(phi_hat) <- rownames(phi_hat) <- c("x", "m", "y")
phi <- extract(object = boot, what = "phi")

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

# Range of time intervals -----
boot <- BootMed(
  phi = phi,
  phi_hat = phi_hat,
  delta_t = 1:5,
  from = "x",
  to = "y",
  med = "m"
)
plot(boot)
plot(boot, type = "bc") # bias-corrected

# Methods -----
# BootMed has a number of methods including
# print, summary, confint, and plot
print(boot)
summary(boot)
confint(boot, level = 0.95)
print(boot, type = "bc") # bias-corrected
summary(boot, type = "bc")
confint(boot, level = 0.95, type = "bc")

```

```
## End(Not run)
```

BootMedStd	<i>Bootstrap 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</i>
------------	---

Description

This function generates a bootstrap 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

```
BootMedStd(
  phi,
  sigma,
  phi_hat,
  sigma_hat,
  delta_t,
  from,
  to,
  med,
  ncores = NULL,
  tol = 0.001
)
```

Arguments

phi	List of numeric matrices. Each element of the list is a bootstrap estimate of the drift matrix (Φ).
sigma	List of numeric matrices. Each element of the list is a bootstrap estimate of the process noise covariance matrix (Σ).
phi_hat	Numeric matrix. The estimated drift matrix ($\hat{\Phi}$) from the original data set. phi_hat should have row and column names pertaining to the variables in the system.
sigma_hat	Numeric matrix. The estimated process noise covariance matrix ($\hat{\Sigma}$) from the original data set.
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.

<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 <code>R</code> is a large value.
<code>tol</code>	Numeric. Smallest possible time interval to allow.

Details

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

Value

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

call Function call.

args Function arguments.

fun Function used ("BootMedStd").

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

Each element in the output list has the following elements:

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

thetahatstar A matrix of bootstrap standardized 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](#)
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](#)
- 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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#),

```

MCDirectCentral(), MCDirectCentralStd(), MCIndirectCentral(), MCIndirectCentralStd(),
MCMed(), MCMedStd(), MCPHi(), MCPHiSigma(), MCTotalCentral(), MCTotalCentralStd(), Med(),
MedStd(), PosteriorBeta(), PosteriorBetaStd(), PosteriorDirectCentral(), PosteriorDirectCentralStd(),
PosteriorIndirectCentral(), PosteriorIndirectCentralStd(), PosteriorMed(), PosteriorMedStd(),
PosteriorTotalCentral(), PosteriorTotalCentralStd(), Total(), TotalCentral(), TotalCentralStd(),
TotalStd(), Trajectory()

```

Examples

```

## Not run:
library(bootStateSpace)
# prepare parameters
## number of individuals
n <- 50
## time points
time <- 100
delta_t <- 0.10
## dynamic structure
p <- 3
mu0 <- rep(x = 0, times = p)
sigma0 <- matrix(
  data = c(
    1.0,
    0.2,
    0.2,
    0.2,
    1.0,
    0.2,
    0.2,
    0.2,
    1.0
  ),
  nrow = p
)
sigma0_l <- t(chol(sigma0))
mu <- rep(x = 0, times = p)
phi <- matrix(
  data = c(
    -0.357,
    0.771,
    -0.450,
    0.0,
    -0.511,
    0.729,
    0,
    0,
    -0.693
  ),
  nrow = p
)
sigma <- matrix(
  data = c(

```

```

    0.24455556,
    0.02201587,
    -0.05004762,
    0.02201587,
    0.07067800,
    0.01539456,
    -0.05004762,
    0.01539456,
    0.07553061
  ),
  nrow = p
)
sigma_l <- t(chol(sigma))
## measurement model
k <- 3
nu <- rep(x = 0, times = k)
lambda <- diag(k)
theta <- 0.2 * diag(k)
theta_l <- t(chol(theta))

boot <- PBSSMOUFixed(
  R = 10L, # use at least 1000 in actual research
  path = getwd(),
  prefix = "ou",
  n = n,
  time = time,
  delta_t = delta_t,
  mu0 = mu0,
  sigma0_l = sigma0_l,
  mu = mu,
  phi = phi,
  sigma_l = sigma_l,
  nu = nu,
  lambda = lambda,
  theta_l = theta_l,
  ncores = NULL, # consider using multiple cores
  seed = 42
)
phi_hat <- phi
colnames(phi_hat) <- rownames(phi_hat) <- c("x", "m", "y")
sigma_hat <- sigma
phi <- extract(object = boot, what = "phi")
sigma <- extract(object = boot, what = "sigma")

# Specific time interval -----
BootMedStd(
  phi = phi,
  sigma = sigma,
  phi_hat = phi_hat,
  sigma_hat = sigma_hat,
  delta_t = 1,
  from = "x",
  to = "y",

```

```

    med = "m"
  )

  # Range of time intervals -----
  boot <- BootMedStd(
    phi = phi,
    sigma = sigma,
    phi_hat = phi_hat,
    sigma_hat = sigma_hat,
    delta_t = 1:5,
    from = "x",
    to = "y",
    med = "m"
  )
  plot(boot)
  plot(boot, type = "bc") # bias-corrected

  # Methods -----
  # BootMedStd has a number of methods including
  # print, summary, confint, and plot
  print(boot)
  summary(boot)
  confint(boot, level = 0.95)
  print(boot, type = "bc") # bias-corrected
  summary(boot, type = "bc")
  confint(boot, level = 0.95, type = "bc")

  ## End(Not run)

```

 BootTotalCentral

Bootstrap Sampling Distribution for the Total Effect Centrality Over a Specific Time Interval or a Range of Time Intervals

Description

This function generates a bootstrap method sampling distribution 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 drift matrix Φ .

Usage

```
BootTotalCentral(phi, phi_hat, delta_t, ncores = NULL, tol = 0.001)
```

Arguments

phi List of numeric matrices. Each element of the list is a bootstrap estimate of the drift matrix (Φ).

phi_hat	Numeric matrix. The estimated drift matrix ($\hat{\Phi}$) from the original data set. phi_hat 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.
tol	Numeric. Smallest possible time interval to allow.

Details

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

Value

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

call Function call.

args Function arguments.

fun Function used ("BootTotalCentral").

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

Each element in the output list has the following elements:

est A vector of total effect centrality.

thetahatstar A matrix of bootstrap 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](#)
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](#)
- 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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

Examples

```
## Not run:
library(bootStateSpace)
# prepare parameters
## number of individuals
n <- 50
## time points
time <- 100
delta_t <- 0.10
## dynamic structure
p <- 3
mu0 <- rep(x = 0, times = p)
sigma0 <- matrix(
  data = c(
    1.0,
    0.2,
    0.2,
    0.2,
    1.0,
    0.2,
    0.2,
    0.2,
    1.0
  ),
  nrow = p
)
sigma0_l <- t(chol(sigma0))
mu <- rep(x = 0, times = p)
phi <- matrix(
  data = c(
    -0.357,
    0.771,
    -0.450,
    0.0,
    -0.511,
    0.729,

```

```

      0,
      0,
      -0.693
    ),
    nrow = p
  )
sigma <- matrix(
  data = c(
    0.24455556,
    0.02201587,
    -0.05004762,
    0.02201587,
    0.07067800,
    0.01539456,
    -0.05004762,
    0.01539456,
    0.07553061
  ),
  nrow = p
)
sigma_l <- t(chol(sigma))
## measurement model
k <- 3
nu <- rep(x = 0, times = k)
lambda <- diag(k)
theta <- 0.2 * diag(k)
theta_l <- t(chol(theta))

boot <- PBSSMOUFixed(
  R = 10L, # use at least 1000 in actual research
  path = getwd(),
  prefix = "ou",
  n = n,
  time = time,
  delta_t = delta_t,
  mu0 = mu0,
  sigma0_l = sigma0_l,
  mu = mu,
  phi = phi,
  sigma_l = sigma_l,
  nu = nu,
  lambda = lambda,
  theta_l = theta_l,
  ncores = NULL, # consider using multiple cores
  seed = 42
)
phi_hat <- phi
colnames(phi_hat) <- rownames(phi_hat) <- c("x", "m", "y")
phi <- extract(object = boot, what = "phi")

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

```

```

    phi_hat = phi_hat,
    delta_t = 1
)

# Range of time intervals -----
boot <- BootTotalCentral(
  phi = phi,
  phi_hat = phi_hat,
  delta_t = 1:5
)
plot(boot)
plot(boot, type = "bc") # bias-corrected

# Methods -----
# BootTotalCentral has a number of methods including
# print, summary, confint, and plot
print(boot)
summary(boot)
confint(boot, level = 0.95)
print(boot, type = "bc") # bias-corrected
summary(boot, type = "bc")
confint(boot, level = 0.95, type = "bc")

## End(Not run)

```

BootTotalCentralStd	<i>Bootstrap Sampling Distribution for the Standardized Total Effect Centrality Over a Specific Time Interval or a Range of Time Intervals</i>
---------------------	--

Description

This function generates a bootstrap method sampling distribution for the standardized total effect centrality 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

```

BootTotalCentralStd(
  phi,
  sigma,
  phi_hat,
  sigma_hat,
  delta_t,
  ncores = NULL,
  tol = 0.001
)

```

Arguments

phi	List of numeric matrices. Each element of the list is a bootstrap estimate of the drift matrix (Φ).
sigma	List of numeric matrices. Each element of the list is a bootstrap estimate of the process noise covariance matrix (Σ).
phi_hat	Numeric matrix. The estimated drift matrix ($\hat{\Phi}$) from the original data set. phi_hat should have row and column names pertaining to the variables in the system.
sigma_hat	Numeric matrix. The estimated process noise covariance matrix ($\hat{\Sigma}$) from the original data set.
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.
tol	Numeric. Smallest possible time interval to allow.

Details

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

Value

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

call Function call.

args Function arguments.

fun Function used ("BootTotalCentralStd").

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

Each element in the output list has the following elements:

est A vector of standardized total effect centrality.

thetahatstar A matrix of bootstrap standardized 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](#)

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](#)

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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

Examples

```
## Not run:
library(bootStateSpace)
# prepare parameters
## number of individuals
n <- 50
## time points
time <- 100
delta_t <- 0.10
## dynamic structure
p <- 3
mu0 <- rep(x = 0, times = p)
sigma0 <- matrix(
  data = c(
    1.0,
    0.2,
    0.2,
    0.2,
    1.0,
    0.2,
    0.2,
    0.2,
    1.0
  ),
  nrow = p
)
sigma0_l <- t(chol(sigma0))
mu <- rep(x = 0, times = p)
phi <- matrix(
  data = c(
    -0.357,
    0.771,
    -0.450,
```

```

    0.0,
    -0.511,
    0.729,
    0,
    0,
    -0.693
  ),
  nrow = p
)
sigma <- matrix(
  data = c(
    0.24455556,
    0.02201587,
    -0.05004762,
    0.02201587,
    0.07067800,
    0.01539456,
    -0.05004762,
    0.01539456,
    0.07553061
  ),
  nrow = p
)
sigma_l <- t(chol(sigma))
## measurement model
k <- 3
nu <- rep(x = 0, times = k)
lambda <- diag(k)
theta <- 0.2 * diag(k)
theta_l <- t(chol(theta))

boot <- PBSSMOUFixed(
  R = 10L, # use at least 1000 in actual research
  path = getwd(),
  prefix = "ou",
  n = n,
  time = time,
  delta_t = delta_t,
  mu0 = mu0,
  sigma0_l = sigma0_l,
  mu = mu,
  phi = phi,
  sigma_l = sigma_l,
  nu = nu,
  lambda = lambda,
  theta_l = theta_l,
  ncores = NULL, # consider using multiple cores
  seed = 42
)
phi_hat <- phi
sigma_hat <- sigma
colnames(phi_hat) <- rownames(phi_hat) <- c("x", "m", "y")
phi <- extract(object = boot, what = "phi")

```

```

sigma <- extract(object = boot, what = "sigma")

# Specific time interval -----
BootTotalCentralStd(
  phi = phi,
  sigma = sigma,
  phi_hat = phi_hat,
  sigma_hat = sigma_hat,
  delta_t = 1
)

# Range of time intervals -----
boot <- BootTotalCentralStd(
  phi = phi,
  sigma = sigma,
  phi_hat = phi_hat,
  sigma_hat = sigma_hat,
  delta_t = 1:5
)
plot(boot)
plot(boot, type = "bc") # bias-corrected

# Methods -----
# BootTotalCentralStd has a number of methods including
# print, summary, confint, and plot
print(boot)
summary(boot)
confint(boot, level = 0.95)
print(boot, type = "bc") # bias-corrected
summary(boot, type = "bc")
confint(boot, level = 0.95, type = "bc")

## End(Not run)

```

confint.ctmedboot	<i>Bootstrap Method Confidence Intervals</i>
-------------------	--

Description

Bootstrap Method Confidence Intervals

Usage

```

## S3 method for class 'ctmedboot'
confint(object, parm = NULL, level = 0.95, type = "pc", ...)

```

Arguments

object	Object of class ctmedboot.
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.
type	Character string. Confidence interval type, that is, type = "pc" for percentile; type = "bc" for bias corrected.
...	additional arguments.

Value

Returns a data frame of confidence intervals.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
## Not run:
library(bootStateSpace)
# prepare parameters
## number of individuals
n <- 50
## time points
time <- 100
delta_t <- 0.10
## dynamic structure
p <- 3
mu0 <- rep(x = 0, times = p)
sigma0 <- matrix(
  data = c(
    1.0,
    0.2,
    0.2,
    0.2,
    1.0,
    0.2,
    0.2,
    0.2,
    1.0
  ),
  nrow = p
)
sigma0_1 <- t(chol(sigma0))
mu <- rep(x = 0, times = p)
phi <- matrix(
  data = c(
    -0.357,
```

```

      0.771,
      -0.450,
      0.0,
      -0.511,
      0.729,
      0,
      0,
      -0.693
    ),
    nrow = p
  )
sigma <- matrix(
  data = c(
    0.24455556,
    0.02201587,
    -0.05004762,
    0.02201587,
    0.07067800,
    0.01539456,
    -0.05004762,
    0.01539456,
    0.07553061
  ),
  nrow = p
)
sigma_l <- t(chol(sigma))
## measurement model
k <- 3
nu <- rep(x = 0, times = k)
lambda <- diag(k)
theta <- 0.2 * diag(k)
theta_l <- t(chol(theta))

boot <- PBSSMOUFixed(
  R = 1000L,
  path = getwd(),
  prefix = "ou",
  n = n,
  time = time,
  delta_t = delta_t,
  mu0 = mu0,
  sigma0_l = sigma0_l,
  mu = mu,
  phi = phi,
  sigma_l = sigma_l,
  nu = nu,
  lambda = lambda,
  theta_l = theta_l,
  ncores = parallel::detectCores() - 1,
  seed = 42
)
phi_hat <- phi
colnames(phi_hat) <- rownames(phi_hat) <- c("x", "m", "y")

```

```

phi <- extract(object = boot, what = "phi")

# Specific time interval -----
boot <- BootMed(
  phi = phi,
  phi_hat = phi_hat,
  delta_t = 1,
  from = "x",
  to = "y",
  med = "m"
)
confint(boot)
confint(boot, type = "bc") # bias-corrected

# Range of time intervals -----
boot <- BootMed(
  phi = phi,
  phi_hat = phi_hat,
  delta_t = 1:5,
  from = "x",
  to = "y",
  med = "m"
)
confint(boot)
confint(boot, type = "bc") # bias-corrected

## End(Not run)

```

confint.ctmeddelta *Delta Method Confidence Intervals*

Description

Delta Method Confidence Intervals

Usage

```

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

```

Arguments

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

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)

```

confint.ctmedmc

Monte Carlo Method Confidence Intervals

Description

Monte Carlo Method Confidence Intervals

Usage

```

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

```

Arguments

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

```

DeltaBeta

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

Description

This function computes the delta method sampling variance-covariance matrix for the elements of the matrix of lagged coefficients 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, tol = 0.001)
```

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.
tol	Numeric. Smallest possible time interval to allow.

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

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

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

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

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

The uncertainty associated with the estimator $\mathbf{g}(\hat{\theta})$ 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 $\mathbf{g}(\hat{\theta})$ is given by

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

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 of length `length(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)
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:[10.1037/met0000779](https://doi.org/10.1037/met0000779)
- 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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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,
  sigma,
  vcov_theta,
  delta_t,
  sigma_diag = FALSE,
  ncores = NULL,
  tol = 0.001
)
```

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).
vcov_theta	Numeric matrix. The sampling variance-covariance matrix of $\text{vec}(\Phi)$ and $\text{vech}(\Sigma)$
delta_t	Numeric. Time interval (Δt).
sigma_diag	Logical. If sigma_diag = TRUE, treat Σ as a diagonal matrix.
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.
tol	Numeric. Smallest possible time interval to allow.

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

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

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

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

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

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 of length `length(delta_t)`.

Each element in the output list has the following elements:

delta_t Time interval.

jacobian Jacobian matrix.

est Estimated elements of the standardized matrix of lagged coefficients.

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

Author(s)

Ivan Jacob Agaloos Pesigan

References

- Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. *Sociological Methodology*, 17, 37. doi:10.2307/271028
- Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. *Structural Equation Modeling: A Multidisciplinary Journal*, 23 (1), 61-75. doi:10.1080/10705511.2014.973960
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:10.1037/met0000779
- 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: `BootBeta()`, `BootBetaStd()`, `BootDirectCentral()`, `BootDirectCentralStd()`, `BootIndirectCentral()`, `BootIndirectCentralStd()`, `BootMed()`, `BootMedStd()`, `BootTotalCentral()`, `BootTotalCentralStd()`, `DeltaBeta()`, `DeltaDirectCentral()`, `DeltaDirectCentralStd()`, `DeltaIndirectCentral()`, `DeltaMed()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `DeltaTotalCentralStd()`, `Direct()`, `DirectCentral()`, `DirectCentralStd()`, `DirectStd()`, `Indirect()`, `IndirectCentral()`, `IndirectCentralStd()`, `IndirectStd()`, `MCBeta()`, `MCBetaStd()`, `MCDirectCentral()`, `MCDirectCentralStd()`, `MCIndirectCentral()`, `MCIndirectCentralStd()`, `MCMed()`, `MCMedStd()`, `MCPhi()`, `MCPhiSigma()`, `MCTotalCentral()`, `MCTotalCentralStd()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorBetaStd()`, `PosteriorDirectCentral()`, `PosteriorDirectCentralStd()`, `PosteriorIndirectCentral()`, `PosteriorIndirectCentralStd()`, `PosteriorMed()`, `PosteriorMedStd()`, `PosteriorTotalCentral()`, `PosteriorTotalCentralStd()`, `Total()`, `TotalCentral()`, `TotalCentralStd()`, `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(
    0.24455556, 0.02201587, -0.05004762,
    0.02201587, 0.07067800, 0.01539456,
    -0.05004762, 0.01539456, 0.07553061
  ),
  nrow = 3
)
vcov_theta <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151, -0.00600, -0.00033,
    0.00110, 0.00324, 0.00020, -0.00061, -0.00115,
    0.00011, 0.00015, 0.00001, -0.00002, -0.00001,
    0.00040, 0.00374, 0.00016, -0.00022, -0.00273,
    -0.00016, 0.00009, 0.00150, 0.00012, -0.00010,
    -0.00026, 0.00002, 0.00012, 0.00004, -0.00001,
    -0.00151, 0.00016, 0.00389, 0.00103, -0.00007,
    -0.00283, -0.00050, 0.00000, 0.00156, 0.00021,
    -0.00005, -0.00031, 0.00001, 0.00007, 0.00006,
    -0.00600, -0.00022, 0.00103, 0.00644, 0.00031,
    -0.00119, -0.00374, -0.00021, 0.00070, 0.00064,
    -0.00015, -0.00005, 0.00000, 0.00003, -0.00001,
    -0.00033, -0.00273, -0.00007, 0.00031, 0.00287,
    0.00013, -0.00014, -0.00170, -0.00012, 0.00006,
    0.00014, -0.00001, -0.00015, 0.00000, 0.00001,
    0.00110, -0.00016, -0.00283, -0.00119, 0.00013,
```

```

0.00297, 0.00063, -0.00004, -0.00177, -0.00013,
0.00005, 0.00017, -0.00002, -0.00008, 0.00001,
0.00324, 0.00009, -0.00050, -0.00374, -0.00014,
0.00063, 0.00495, 0.00024, -0.00093, -0.00020,
0.00006, -0.00010, 0.00000, -0.00001, 0.00004,
0.00020, 0.00150, 0.00000, -0.00021, -0.00170,
-0.00004, 0.00024, 0.00214, 0.00012, -0.00002,
-0.00004, 0.00000, 0.00006, -0.00005, -0.00001,
-0.00061, 0.00012, 0.00156, 0.00070, -0.00012,
-0.00177, -0.00093, 0.00012, 0.00223, 0.00004,
-0.00002, -0.00003, 0.00001, 0.00003, -0.00013,
-0.00115, -0.00010, 0.00021, 0.00064, 0.00006,
-0.00013, -0.00020, -0.00002, 0.00004, 0.00057,
0.00001, -0.00009, 0.00000, 0.00000, 0.00001,
0.00011, -0.00026, -0.00005, -0.00015, 0.00014,
0.00005, 0.00006, -0.00004, -0.00002, 0.00001,
0.00012, 0.00001, 0.00000, -0.00002, 0.00000,
0.00015, 0.00002, -0.00031, -0.00005, -0.00001,
0.00017, -0.00010, 0.00000, -0.00003, -0.00009,
0.00001, 0.00014, 0.00000, 0.00000, -0.00005,
0.00001, 0.00012, 0.00001, 0.00000, -0.00015,
-0.00002, 0.00000, 0.00006, 0.00001, 0.00000,
0.00000, 0.00000, 0.00010, 0.00001, 0.00000,
-0.00002, 0.00004, 0.00007, 0.00003, 0.00000,
-0.00008, -0.00001, -0.00005, 0.00003, 0.00000,
-0.00002, 0.00000, 0.00001, 0.00005, 0.00001,
-0.00001, -0.00001, 0.00006, -0.00001, 0.00001,
0.00001, 0.00004, -0.00001, -0.00013, 0.00001,
0.00000, -0.00005, 0.00000, 0.00001, 0.00012
),
nrow = 15
)

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

# Range of time intervals -----
delta <- DeltaBetaStd(
  phi = phi,
  sigma = sigma,
  vcov_theta = vcov_theta,
  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)
```

DeltaDirectCentral	<i>Delta Method Sampling Variance-Covariance Matrix for the Direct Effect Centrality Over a Specific Time Interval or a Range of Time Intervals</i>
--------------------	---

Description

This function computes the delta method sampling variance-covariance matrix for the direct 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

```
DeltaDirectCentral(phi, vcov_phi_vec, delta_t, ncores = NULL, tol = 0.01)
```

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.
tol	Numeric. Smallest possible time interval to allow.

Details

See [DirectCentral\(\)](#) 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 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:

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

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 ("DeltaDirectCentral").

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

Each element in the output list has the following elements:

delta_t Time interval.

jacobian Jacobian matrix.

est Estimated direct effect centrality.

vcov Sampling variance-covariance matrix of estimated direct 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)
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](https://doi.org/10.1037/met0000779)
- 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: `BootBeta()`, `BootBetaStd()`, `BootDirectCentral()`, `BootDirectCentralStd()`, `BootIndirectCentral()`, `BootIndirectCentralStd()`, `BootMed()`, `BootMedStd()`, `BootTotalCentral()`, `BootTotalCentralStd()`, `DeltaBeta()`, `DeltaBetaStd()`, `DeltaDirectCentralStd()`, `DeltaIndirectCentral()`, `DeltaMed()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `DeltaTotalCentralStd()`, `Direct()`, `DirectCentral()`, `DirectCentralStd()`, `DirectStd()`, `Indirect()`, `IndirectCentral()`, `IndirectCentralStd()`, `IndirectStd()`, `MCBeta()`, `MCBetaStd()`, `MCDirectCentral()`, `MCDirectCentralStd()`, `MCIndirectCentral()`, `MCIndirectCentralStd()`, `MCMed()`, `MCMedStd()`, `MCPhi()`, `MCPhiSigma()`, `MCTotalCentral()`, `MCTotalCentralStd()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorBetaStd()`, `PosteriorDirectCentral()`, `PosteriorDirectCentralStd()`, `PosteriorIndirectCentral()`, `PosteriorIndirectCentralStd()`, `PosteriorMed()`, `PosteriorMedStd()`, `PosteriorTotalCentral()`, `PosteriorTotalCentralStd()`, `Total()`, `TotalCentral()`, `TotalCentralStd()`, `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 -----
DeltaDirectCentral(
  phi = phi,
  vcov_phi_vec = vcov_phi_vec,
  delta_t = 1
)

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

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

```

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

Usage

```

DeltaDirectCentralStd(
  phi,
  sigma,
  vcov_theta,

```

```

    delta_t,
    sigma_diag = FALSE,
    ncores = NULL,
    tol = 0.001
)

```

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).
vcov_theta	Numeric matrix. The sampling variance-covariance matrix of $\text{vec}(\Phi)$ and $\text{vech}(\Sigma)$
delta_t	Vector of positive numbers. Time interval (Δt).
sigma_diag	Logical. If sigma_diag = TRUE, treat Σ as a diagonal matrix.
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.
tol	Numeric. Smallest possible time interval to allow.

Details

See [DirectCentralStd\(\)](#) 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 $\text{vec}(\hat{\Phi})$ 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 ("DeltaDirectCentralStd").

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

Each element in the output list has the following elements:

delta_t Time interval.

jacobian Jacobian matrix.

est Estimated standardized direct effect centrality.

vcov Sampling variance-covariance matrix of estimated standardized direct 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)

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:[10.1037/met0000779](https://doi.org/10.1037/met0000779)

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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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(
    0.24455556, 0.02201587, -0.05004762,
    0.02201587, 0.07067800, 0.01539456,
    -0.05004762, 0.01539456, 0.07553061
  ),
  nrow = 3
)
vcov_theta <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151, -0.00600, -0.00033,
    0.00110, 0.00324, 0.00020, -0.00061, -0.00115,
    0.00011, 0.00015, 0.00001, -0.00002, -0.00001,
    0.00040, 0.00374, 0.00016, -0.00022, -0.00273,
    -0.00016, 0.00009, 0.00150, 0.00012, -0.00010,
    -0.00026, 0.00002, 0.00012, 0.00004, -0.00001,
    -0.00151, 0.00016, 0.00389, 0.00103, -0.00007,
    -0.00283, -0.00050, 0.00000, 0.00156, 0.00021,
    -0.00005, -0.00031, 0.00001, 0.00007, 0.00006,
    -0.00600, -0.00022, 0.00103, 0.00644, 0.00031,
    -0.00119, -0.00374, -0.00021, 0.00070, 0.00064,
    -0.00015, -0.00005, 0.00000, 0.00003, -0.00001,
    -0.00033, -0.00273, -0.00007, 0.00031, 0.00287,
    0.00013, -0.00014, -0.00170, -0.00012, 0.00006,
    0.00014, -0.00001, -0.00015, 0.00000, 0.00001,
    0.00110, -0.00016, -0.00283, -0.00119, 0.00013,
    0.00297, 0.00063, -0.00004, -0.00177, -0.00013,
    0.00005, 0.00017, -0.00002, -0.00008, 0.00001,
    0.00324, 0.00009, -0.00050, -0.00374, -0.00014,
    0.00063, 0.00495, 0.00024, -0.00093, -0.00020,
    0.00006, -0.00010, 0.00000, -0.00001, 0.00004,
    0.00020, 0.00150, 0.00000, -0.00021, -0.00170,
    -0.00004, 0.00024, 0.00214, 0.00012, -0.00002,
    -0.00004, 0.00000, 0.00006, -0.00005, -0.00001,
    -0.00061, 0.00012, 0.00156, 0.00070, -0.00012,
    -0.00177, -0.00093, 0.00012, 0.00223, 0.00004,
    -0.00002, -0.00003, 0.00001, 0.00003, -0.00013,
    -0.00115, -0.00010, 0.00021, 0.00064, 0.00006,
    -0.00013, -0.00020, -0.00002, 0.00004, 0.00057,
    0.00001, -0.00009, 0.00000, 0.00000, 0.00001,
    0.00011, -0.00026, -0.00005, -0.00015, 0.00014,
    0.00005, 0.00006, -0.00004, -0.00002, 0.00001,

```

```

0.00012, 0.00001, 0.00000, -0.00002, 0.00000,
0.00015, 0.00002, -0.00031, -0.00005, -0.00001,
0.00017, -0.00010, 0.00000, -0.00003, -0.00009,
0.00001, 0.00014, 0.00000, 0.00000, -0.00005,
0.00001, 0.00012, 0.00001, 0.00000, -0.00015,
-0.00002, 0.00000, 0.00006, 0.00001, 0.00000,
0.00000, 0.00000, 0.00010, 0.00001, 0.00000,
-0.00002, 0.00004, 0.00007, 0.00003, 0.00000,
-0.00008, -0.00001, -0.00005, 0.00003, 0.00000,
-0.00002, 0.00000, 0.00001, 0.00005, 0.00001,
-0.00001, -0.00001, 0.00006, -0.00001, 0.00001,
0.00001, 0.00004, -0.00001, -0.00013, 0.00001,
0.00000, -0.00005, 0.00000, 0.00001, 0.00012
),
nrow = 15
)

# Specific time interval -----
DeltaDirectCentralStd(
  phi = phi,
  sigma = sigma,
  vcov_theta = vcov_theta,
  delta_t = 1
)

# Range of time intervals -----
delta <- DeltaDirectCentralStd(
  phi = phi,
  sigma = sigma,
  vcov_theta = vcov_theta,
  delta_t = 1:5
)
plot(delta)

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

```

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

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, tol = 0.001)
```

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.
tol	Numeric. Smallest possible time interval to allow.

Details

See [IndirectCentral\(\)](#) 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 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 ("DeltaIndirectCentral").

output A list of length `length(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](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)

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:[10.1037/met0000779](https://doi.org/10.1037/met0000779)

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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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)

```

DeltaIndirectCentralStd

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

Usage

```

DeltaIndirectCentralStd(
  phi,
  sigma,
  vcov_theta,
  delta_t,
  sigma_diag = FALSE,
  ncores = NULL,
  tol = 0.001
)

```

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).
vcov_theta	Numeric matrix. The sampling variance-covariance matrix of vec (Φ) and vech (Σ)
delta_t	Vector of positive numbers. Time interval (Δt).

<code>sigma_diag</code>	Logical. If <code>sigma_diag = TRUE</code> , treat Σ as a diagonal matrix.
<code>ncores</code>	Positive integer. Number of cores to use. If <code>ncores = NULL</code> , use a single core. Consider using multiple cores when the length of <code>delta_t</code> is long.
<code>tol</code>	Numeric. Smallest possible time interval to allow.

Details

See [IndirectCentralStd\(\)](#) 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 $\text{vec}(\hat{\Phi})$ 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 ("DeltaIndirectCentralStd").

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

Each element in the output list has the following elements:

delta_t Time interval.

jacobian Jacobian matrix.

est Estimated standardized indirect effect centrality.

vcov Sampling variance-covariance matrix of estimated standardized 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
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:10.1037/met0000779
- 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

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(
    0.24455556, 0.02201587, -0.05004762,
    0.02201587, 0.07067800, 0.01539456,
    -0.05004762, 0.01539456, 0.07553061
  ),
  nrow = 3
)
vcov_theta <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151, -0.00600, -0.00033,
    0.00110, 0.00324, 0.00020, -0.00061, -0.00115,
    0.00011, 0.00015, 0.00001, -0.00002, -0.00001,
    0.00040, 0.00374, 0.00016, -0.00022, -0.00273,
    -0.00016, 0.00009, 0.00150, 0.00012, -0.00010,
    -0.00026, 0.00002, 0.00012, 0.00004, -0.00001,
    -0.00151, 0.00016, 0.00389, 0.00103, -0.00007,
    -0.00283, -0.00050, 0.00000, 0.00156, 0.00021,
    -0.00005, -0.00031, 0.00001, 0.00007, 0.00006,
    -0.00600, -0.00022, 0.00103, 0.00644, 0.00031,
    -0.00119, -0.00374, -0.00021, 0.00070, 0.00064,
    -0.00015, -0.00005, 0.00000, 0.00003, -0.00001,
    -0.00033, -0.00273, -0.00007, 0.00031, 0.00287,
```

```

0.00013, -0.00014, -0.00170, -0.00012, 0.00006,
0.00014, -0.00001, -0.00015, 0.00000, 0.00001,
0.00110, -0.00016, -0.00283, -0.00119, 0.00013,
0.00297, 0.00063, -0.00004, -0.00177, -0.00013,
0.00005, 0.00017, -0.00002, -0.00008, 0.00001,
0.00324, 0.00009, -0.00050, -0.00374, -0.00014,
0.00063, 0.00495, 0.00024, -0.00093, -0.00020,
0.00006, -0.00010, 0.00000, -0.00001, 0.00004,
0.00020, 0.00150, 0.00000, -0.00021, -0.00170,
-0.00004, 0.00024, 0.00214, 0.00012, -0.00002,
-0.00004, 0.00000, 0.00006, -0.00005, -0.00001,
-0.00061, 0.00012, 0.00156, 0.00070, -0.00012,
-0.00177, -0.00093, 0.00012, 0.00223, 0.00004,
-0.00002, -0.00003, 0.00001, 0.00003, -0.00013,
-0.00115, -0.00010, 0.00021, 0.00064, 0.00006,
-0.00013, -0.00020, -0.00002, 0.00004, 0.00057,
0.00001, -0.00009, 0.00000, 0.00000, 0.00001,
0.00011, -0.00026, -0.00005, -0.00015, 0.00014,
0.00005, 0.00006, -0.00004, -0.00002, 0.00001,
0.00012, 0.00001, 0.00000, -0.00002, 0.00000,
0.00015, 0.00002, -0.00031, -0.00005, -0.00001,
0.00017, -0.00010, 0.00000, -0.00003, -0.00009,
0.00001, 0.00014, 0.00000, 0.00000, -0.00005,
0.00001, 0.00012, 0.00001, 0.00000, -0.00015,
-0.00002, 0.00000, 0.00006, 0.00001, 0.00000,
0.00000, 0.00000, 0.00010, 0.00001, 0.00000,
-0.00002, 0.00004, 0.00007, 0.00003, 0.00000,
-0.00008, -0.00001, -0.00005, 0.00003, 0.00000,
-0.00002, 0.00000, 0.00001, 0.00005, 0.00001,
-0.00001, -0.00001, 0.00006, -0.00001, 0.00001,
0.00001, 0.00004, -0.00001, -0.00013, 0.00001,
0.00000, -0.00005, 0.00000, 0.00001, 0.00012
),
nrow = 15
)

# Specific time interval -----
DeltaIndirectCentralStd(
  phi = phi,
  sigma = sigma,
  vcov_theta = vcov_theta,
  delta_t = 1
)

# Range of time intervals -----
delta <- DeltaIndirectCentralStd(
  phi = phi,
  sigma = sigma,
  vcov_theta = vcov_theta,
  delta_t = 1:5
)
plot(delta)

```

```
# Methods -----
# DeltaIndirectCentralStd has a number of methods including
# print, summary, confint, and plot
print(delta)
summary(delta)
confint(delta, level = 0.95)
plot(delta)
#' @family Continuous-Time Mediation Functions
```

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, tol = 0.001)
```

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 <code>vec</code> (Φ).
<code>delta_t</code>	Vector of positive numbers. Time interval (Δt).
<code>from</code>	Character string. Name of the independent variable X in <code>phi</code> .
<code>to</code>	Character string. Name of the dependent variable Y in <code>phi</code> .
<code>med</code>	Character vector. Name/s of the mediator variable/s in <code>phi</code> .
<code>ncores</code>	Positive integer. Number of cores to use. If <code>ncores = NULL</code> , use a single core. Consider using multiple cores when the length of <code>delta_t</code> is long.
<code>tol</code>	Numeric. Smallest possible time interval to allow.

Details

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

Delta Method:

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

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

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 of length `length(delta_t)`.

Each element in the output list has the following elements:

delta_t Time interval.

jacobian Jacobian matrix.

est Estimated total, direct, and indirect effects.

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

Author(s)

Ivan Jacob Agaloos Pesigan

References

- Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. *Sociological Methodology*, 17, 37. [doi:10.2307/271028](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)
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](https://doi.org/10.1037/met0000779)
- 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: `BootBeta()`, `BootBetaStd()`, `BootDirectCentral()`, `BootDirectCentralStd()`, `BootIndirectCentral()`, `BootIndirectCentralStd()`, `BootMed()`, `BootMedStd()`, `BootTotalCentral()`, `BootTotalCentralStd()`, `DeltaBeta()`, `DeltaBetaStd()`, `DeltaDirectCentral()`, `DeltaDirectCentralStd()`, `DeltaIndirectCentral()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `DeltaTotalCentralStd()`, `Direct()`, `DirectCentral()`, `DirectCentralStd()`, `DirectStd()`, `Indirect()`, `IndirectCentral()`, `IndirectCentralStd()`, `IndirectStd()`, `MCBeta()`, `MCBetaStd()`, `MCDirectCentral()`, `MCDirectCentralStd()`, `MCIndirectCentral()`, `MCIndirectCentralStd()`, `MCMed()`, `MCMedStd()`, `MCPhi()`, `MCPhiSigma()`, `MCTotalCentral()`, `MCTotalCentralStd()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorBetaStd()`, `PosteriorDirectCentral()`, `PosteriorDirectCentralStd()`, `PosteriorIndirectCentral()`, `PosteriorIndirectCentralStd()`, `PosteriorMed()`, `PosteriorMedStd()`, `PosteriorTotalCentral()`, `PosteriorTotalCentralStd()`, `Total()`, `TotalCentral()`, `TotalCentralStd()`, `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,
  sigma,
  vcov_theta,
  delta_t,
  from,
  to,
  med,
  sigma_diag = FALSE,
  ncores = NULL,
  tol = 0.001
)
```

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).
vcov_theta	Numeric matrix. The sampling variance-covariance matrix of $\text{vec}(\Phi)$ and $\text{vech}(\Sigma)$
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.
sigma_diag	Logical. If sigma_diag = TRUE, treat Σ as a diagonal matrix.
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.
tol	Numeric. Smallest possible time interval to allow.

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 $\text{vec}(\hat{\Phi})$ and $\text{vech}(\hat{\Sigma})$. By the multivariate central limit theory, the function $\mathbf{g}$ using $\hat{\theta}$ as input can be expressed as:

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

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

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

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

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 ("DeltaMedStd").

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

Each element in the output list has the following elements:

delta_t Time interval.

jacobian Jacobian matrix.

est Estimated standardized total, direct, and indirect effects.

vcov Sampling variance-covariance matrix of the estimated standardized 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
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:10.1037/met0000779
- 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: `BootBeta()`, `BootBetaStd()`, `BootDirectCentral()`, `BootDirectCentralStd()`, `BootIndirectCentral()`, `BootIndirectCentralStd()`, `BootMed()`, `BootMedStd()`, `BootTotalCentral()`, `BootTotalCentralStd()`, `DeltaBeta()`, `DeltaBetaStd()`, `DeltaDirectCentral()`, `DeltaDirectCentralStd()`, `DeltaIndirectCentral()`, `DeltaMed()`, `DeltaTotalCentral()`, `DeltaTotalCentralStd()`, `Direct()`, `DirectCentral()`, `DirectCentralStd()`, `DirectStd()`, `Indirect()`, `IndirectCentral()`, `IndirectCentralStd()`, `IndirectStd()`, `MCBeta()`, `MCBetaStd()`, `MCDirectCentral()`, `MCDirectCentralStd()`, `MCIndirectCentral()`, `MCIndirectCentralStd()`, `MCMed()`, `MCMedStd()`, `MCPhi()`, `MCPhiSigma()`, `MCTotalCentral()`, `MCTotalCentralStd()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorBetaStd()`, `PosteriorDirectCentral()`, `PosteriorDirectCentralStd()`, `PosteriorIndirectCentral()`, `PosteriorIndirectCentralStd()`, `PosteriorMed()`, `PosteriorMedStd()`, `PosteriorTotalCentral()`, `PosteriorTotalCentralStd()`, `Total()`, `TotalCentral()`, `TotalCentralStd()`, `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(
    0.24455556, 0.02201587, -0.05004762,
    0.02201587, 0.07067800, 0.01539456,
    -0.05004762, 0.01539456, 0.07553061
  ),
  nrow = 3
)
vcov_theta <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151, -0.00600, -0.00033,
    0.00110, 0.00324, 0.00020, -0.00061, -0.00115,
    0.00011, 0.00015, 0.00001, -0.00002, -0.00001,
    0.00040, 0.00374, 0.00016, -0.00022, -0.00273,
    -0.00016, 0.00009, 0.00150, 0.00012, -0.00010,
    -0.00026, 0.00002, 0.00012, 0.00004, -0.00001,
    -0.00151, 0.00016, 0.00389, 0.00103, -0.00007,
    -0.00283, -0.00050, 0.00000, 0.00156, 0.00021,
    -0.00005, -0.00031, 0.00001, 0.00007, 0.00006,
    -0.00600, -0.00022, 0.00103, 0.00644, 0.00031,
    -0.00119, -0.00374, -0.00021, 0.00070, 0.00064,
    -0.00015, -0.00005, 0.00000, 0.00003, -0.00001,
    -0.00033, -0.00273, -0.00007, 0.00031, 0.00287,
    0.00013, -0.00014, -0.00170, -0.00012, 0.00006,
    0.00014, -0.00001, -0.00015, 0.00000, 0.00001,
    0.00110, -0.00016, -0.00283, -0.00119, 0.00013,
  ),
  nrow = 3
)
```

```

0.00297, 0.00063, -0.00004, -0.00177, -0.00013,
0.00005, 0.00017, -0.00002, -0.00008, 0.00001,
0.00324, 0.00009, -0.00050, -0.00374, -0.00014,
0.00063, 0.00495, 0.00024, -0.00093, -0.00020,
0.00006, -0.00010, 0.00000, -0.00001, 0.00004,
0.00020, 0.00150, 0.00000, -0.00021, -0.00170,
-0.00004, 0.00024, 0.00214, 0.00012, -0.00002,
-0.00004, 0.00000, 0.00006, -0.00005, -0.00001,
-0.00061, 0.00012, 0.00156, 0.00070, -0.00012,
-0.00177, -0.00093, 0.00012, 0.00223, 0.00004,
-0.00002, -0.00003, 0.00001, 0.00003, -0.00013,
-0.00115, -0.00010, 0.00021, 0.00064, 0.00006,
-0.00013, -0.00020, -0.00002, 0.00004, 0.00057,
0.00001, -0.00009, 0.00000, 0.00000, 0.00001,
0.00011, -0.00026, -0.00005, -0.00015, 0.00014,
0.00005, 0.00006, -0.00004, -0.00002, 0.00001,
0.00012, 0.00001, 0.00000, -0.00002, 0.00000,
0.00015, 0.00002, -0.00031, -0.00005, -0.00001,
0.00017, -0.00010, 0.00000, -0.00003, -0.00009,
0.00001, 0.00014, 0.00000, 0.00000, -0.00005,
0.00001, 0.00012, 0.00001, 0.00000, -0.00015,
-0.00002, 0.00000, 0.00006, 0.00001, 0.00000,
0.00000, 0.00000, 0.00010, 0.00001, 0.00000,
-0.00002, 0.00004, 0.00007, 0.00003, 0.00000,
-0.00008, -0.00001, -0.00005, 0.00003, 0.00000,
-0.00002, 0.00000, 0.00001, 0.00005, 0.00001,
-0.00001, -0.00001, 0.00006, -0.00001, 0.00001,
0.00001, 0.00004, -0.00001, -0.00013, 0.00001,
0.00000, -0.00005, 0.00000, 0.00001, 0.00012
),
nrow = 15
)

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

# Range of time intervals -----
delta <- DeltaMedStd(
  phi = phi,
  sigma = sigma,
  vcov_theta = vcov_theta,
  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	<i>Delta Method Sampling Variance-Covariance Matrix for the Total Effect Centrality Over a Specific Time Interval or a Range of Time Intervals</i>
-------------------	--

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, tol = 0.001)
```

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.
tol	Numeric. Smallest possible time interval to allow.

Details

See [TotalCentral\(\)](#) 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 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 $\mathbf{g}$ 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 ("DeltaTotalCentral").

output A list of length `length(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)
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](https://doi.org/10.1037/met0000779)
- 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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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)

```

DeltaTotalCentralStd	<i>Delta Method Sampling Variance-Covariance Matrix for the Standardized Total Effect Centrality Over a Specific Time Interval or a Range of Time Intervals</i>
----------------------	---

Description

This function computes the delta method sampling variance-covariance matrix for the standardized 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 and process noise covariance matrix Σ .

Usage

```

DeltaTotalCentralStd(
  phi,
  sigma,
  vcov_theta,
  delta_t,

```

```

    sigma_diag = FALSE,
    ncores = NULL,
    tol = 0.001
)

```

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).
vcov_theta	Numeric matrix. The sampling variance-covariance matrix of $\text{vec}(\Phi)$ and $\text{vech}(\Sigma)$
delta_t	Vector of positive numbers. Time interval (Δt).
sigma_diag	Logical. If sigma_diag = TRUE, treat Σ as a diagonal matrix.
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.
tol	Numeric. Smallest possible time interval to allow.

Details

See [TotalCentralStd\(\)](#) 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 $\text{vec}(\hat{\Phi})$ 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 ("DeltaTotalCentralStd").

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

Each element in the output list has the following elements:

delta_t Time interval.

jacobian Jacobian matrix.

est Estimated standardized total effect centrality.

vcov Sampling variance-covariance matrix of estimated standardized 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)

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:[10.1037/met0000779](https://doi.org/10.1037/met0000779)

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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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(
    0.24455556, 0.02201587, -0.05004762,
    0.02201587, 0.07067800, 0.01539456,
    -0.05004762, 0.01539456, 0.07553061
  ),
  nrow = 3
)
vcov_theta <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151, -0.00600, -0.00033,
    0.00110, 0.00324, 0.00020, -0.00061, -0.00115,
    0.00011, 0.00015, 0.00001, -0.00002, -0.00001,
    0.00040, 0.00374, 0.00016, -0.00022, -0.00273,
    -0.00016, 0.00009, 0.00150, 0.00012, -0.00010,
    -0.00026, 0.00002, 0.00012, 0.00004, -0.00001,
    -0.00151, 0.00016, 0.00389, 0.00103, -0.00007,
    -0.00283, -0.00050, 0.00000, 0.00156, 0.00021,
    -0.00005, -0.00031, 0.00001, 0.00007, 0.00006,
    -0.00600, -0.00022, 0.00103, 0.00644, 0.00031,
    -0.00119, -0.00374, -0.00021, 0.00070, 0.00064,
    -0.00015, -0.00005, 0.00000, 0.00003, -0.00001,
    -0.00033, -0.00273, -0.00007, 0.00031, 0.00287,
    0.00013, -0.00014, -0.00170, -0.00012, 0.00006,
    0.00014, -0.00001, -0.00015, 0.00000, 0.00001,
    0.00110, -0.00016, -0.00283, -0.00119, 0.00013,
    0.00297, 0.00063, -0.00004, -0.00177, -0.00013,
    0.00005, 0.00017, -0.00002, -0.00008, 0.00001,
    0.00324, 0.00009, -0.00050, -0.00374, -0.00014,
    0.00063, 0.00495, 0.00024, -0.00093, -0.00020,
    0.00006, -0.00010, 0.00000, -0.00001, 0.00004,
    0.00020, 0.00150, 0.00000, -0.00021, -0.00170,
    -0.00004, 0.00024, 0.00214, 0.00012, -0.00002,
    -0.00004, 0.00000, 0.00006, -0.00005, -0.00001,
    -0.00061, 0.00012, 0.00156, 0.00070, -0.00012,
    -0.00177, -0.00093, 0.00012, 0.00223, 0.00004,
    -0.00002, -0.00003, 0.00001, 0.00003, -0.00013,
    -0.00115, -0.00010, 0.00021, 0.00064, 0.00006,
    -0.00013, -0.00020, -0.00002, 0.00004, 0.00057,
    0.00001, -0.00009, 0.00000, 0.00000, 0.00001,
    0.00011, -0.00026, -0.00005, -0.00015, 0.00014,
    0.00005, 0.00006, -0.00004, -0.00002, 0.00001,

```

```

0.00012, 0.00001, 0.00000, -0.00002, 0.00000,
0.00015, 0.00002, -0.00031, -0.00005, -0.00001,
0.00017, -0.00010, 0.00000, -0.00003, -0.00009,
0.00001, 0.00014, 0.00000, 0.00000, -0.00005,
0.00001, 0.00012, 0.00001, 0.00000, -0.00015,
-0.00002, 0.00000, 0.00006, 0.00001, 0.00000,
0.00000, 0.00000, 0.00010, 0.00001, 0.00000,
-0.00002, 0.00004, 0.00007, 0.00003, 0.00000,
-0.00008, -0.00001, -0.00005, 0.00003, 0.00000,
-0.00002, 0.00000, 0.00001, 0.00005, 0.00001,
-0.00001, -0.00001, 0.00006, -0.00001, 0.00001,
0.00001, 0.00004, -0.00001, -0.00013, 0.00001,
0.00000, -0.00005, 0.00000, 0.00001, 0.00012
),
nrow = 15
)

# Specific time interval -----
DeltaTotalCentralStd(
  phi = phi,
  sigma = sigma,
  vcov_theta = vcov_theta,
  delta_t = 1
)

# Range of time intervals -----
delta <- DeltaTotalCentralStd(
  phi = phi,
  sigma = sigma,
  vcov_theta = vcov_theta,
  delta_t = 1:5
)
plot(delta)

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

```

Direct

Direct Effect of X on Y Over a Specific Time Interval

Description

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

Usage

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

Arguments

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

Details

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

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

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)

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](https://doi.org/10.1037/met0000779)

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: `BootBeta()`, `BootBetaStd()`, `BootDirectCentral()`, `BootDirectCentralStd()`, `BootIndirectCentral()`, `BootIndirectCentralStd()`, `BootMed()`, `BootMedStd()`, `BootTotalCentral()`, `BootTotalCentralStd()`, `DeltaBeta()`, `DeltaBetaStd()`, `DeltaDirectCentral()`, `DeltaDirectCentralStd()`, `DeltaIndirectCentral()`, `DeltaMed()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `DeltaTotalCentralStd()`, `DirectCentral()`, `DirectCentralStd()`, `DirectStd()`, `Indirect()`, `IndirectCentral()`, `IndirectCentralStd()`, `IndirectStd()`, `MCBeta()`, `MCBetaStd()`, `MCDirectCentral()`, `MCDirectCentralStd()`, `MCIndirectCentral()`, `MCIndirectCentralStd()`, `MCMed()`, `MCMedStd()`, `MCPhi()`, `MCPhiSigma()`, `MCTotalCentral()`, `MCTotalCentralStd()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorBetaStd()`, `PosteriorDirectCentral()`, `PosteriorDirectCentralStd()`, `PosteriorIndirectCentral()`, `PosteriorIndirectCentralStd()`, `PosteriorMed()`, `PosteriorMedStd()`, `PosteriorTotalCentral()`, `PosteriorTotalCentralStd()`, `Total()`, `TotalCentral()`, `TotalCentralStd()`, `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")
)
```

DirectCentral	<i>Direct Effect Centrality</i>
---------------	---------------------------------

Description

Direct Effect Centrality

Usage

```
DirectCentral(phi, delta_t, tol = 0.001)
```

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).
tol	Numeric. Smallest possible time interval to allow.

Details

Direct effect centrality is the sum of all possible direct 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 ("DirectCentral").

output A matrix of direct 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)

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](https://doi.org/10.1037/met0000779)

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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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 -----
DirectCentral(
  phi = phi,
  delta_t = 1
)

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

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

```

DirectCentralStd

Standardized Direct Effect Centrality

Description

Standardized Direct Effect Centrality

Usage

```
DirectCentralStd(phi, sigma, delta_t, tol = 0.001)
```

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).
tol	Numeric. Smallest possible time interval to allow.

Details

Standardized direct effect centrality is the sum of all possible standardized direct 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 ("DirectCentralStd").

output A matrix of standardized direct effect centrality.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Continuous-Time Mediation Functions: `BootBeta()`, `BootBetaStd()`, `BootDirectCentral()`, `BootDirectCentralStd()`, `BootIndirectCentral()`, `BootIndirectCentralStd()`, `BootMed()`, `BootMedStd()`, `BootTotalCentral()`, `BootTotalCentralStd()`, `DeltaBeta()`, `DeltaBetaStd()`, `DeltaDirectCentral()`, `DeltaDirectCentralStd()`, `DeltaIndirectCentral()`, `DeltaMed()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `DeltaTotalCentralStd()`, `Direct()`, `DirectCentral()`, `DirectStd()`, `Indirect()`, `IndirectCentral()`, `IndirectCentralStd()`, `IndirectStd()`, `MCBeta()`, `MCBetaStd()`, `MCDirectCentral()`, `MCDirectCentralStd()`, `MCIndirectCentral()`, `MCIndirectCentralStd()`, `MCMed()`, `MCMedStd()`, `MCPhi()`, `MCPhiSigma()`, `MCTotalCentral()`, `MCTotalCentralStd()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorBetaStd()`, `PosteriorDirectCentral()`, `PosteriorDirectCentralStd()`, `PosteriorIndirectCentral()`, `PosteriorIndirectCentralStd()`, `PosteriorMed()`, `PosteriorMedStd()`, `PosteriorTotalCentral()`, `PosteriorTotalCentralStd()`, `Total()`, `TotalCentral()`, `TotalCentralStd()`, `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(
      0.24455556, 0.02201587, -0.05004762,
      0.02201587, 0.07067800, 0.01539456,
      -0.05004762, 0.01539456, 0.07553061
    ),
    nrow = 3
  )

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

  # Range of time intervals -----
  direct_central_std <- DirectCentralStd(
    phi = phi,
    sigma = sigma,
    delta_t = 1:30
  )
  plot(direct_central_std)

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

```

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

$$\text{Direct}_{\Delta t_{i,j}}^* = \text{Direct}_{\Delta t_{i,j}} \left(\frac{\sigma_{x_j}}{\sigma_{y_i}} \right)$$

where Φ denotes the drift matrix, σ_{x_j} and σ_{y_i} are the steady-state model-implied standard deviations of the state independent and dependent variables, respectively, 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 standardized 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)
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](https://doi.org/10.1037/met0000779)
- 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: `BootBeta()`, `BootBetaStd()`, `BootDirectCentral()`, `BootDirectCentralStd()`, `BootIndirectCentral()`, `BootIndirectCentralStd()`, `BootMed()`, `BootMedStd()`, `BootTotalCentral()`, `BootTotalCentralStd()`, `DeltaBeta()`, `DeltaBetaStd()`, `DeltaDirectCentral()`, `DeltaDirectCentralStd()`, `DeltaIndirectCentral()`, `DeltaMed()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `DeltaTotalCentralStd()`, `Direct()`, `DirectCentral()`, `DirectCentralStd()`, `Indirect()`, `IndirectCentral()`, `IndirectCentralStd()`, `IndirectStd()`, `MCBeta()`, `MCBetaStd()`, `MCDirectCentral()`, `MCDirectCentralStd()`, `MCIndirectCentral()`, `MCIndirectCentralStd()`, `MCMed()`, `MCMedStd()`, `MCPhi()`, `MCPhiSigma()`, `MCTotalCentral()`, `MCTotalCentralStd()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorBetaStd()`, `PosteriorDirectCentral()`, `PosteriorDirectCentralStd()`, `PosteriorIndirectCentral()`, `PosteriorIndirectCentralStd()`, `PosteriorMed()`, `PosteriorMedStd()`, `PosteriorTotalCentral()`, `PosteriorTotalCentralStd()`, `Total()`, `TotalCentral()`, `TotalCentralStd()`, `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(
    0.24455556, 0.02201587, -0.05004762,
    0.02201587, 0.07067800, 0.01539456,
    -0.05004762, 0.01539456, 0.07553061
  ),
  nrow = 3
)
delta_t <- 1
DirectStd(
  phi = phi,
  sigma = sigma,
  delta_t = delta_t,
  from = "x",
  to = "y",
  med = "m"
)
```

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

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

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

Linear Stochastic Differential Equation Model:

The measurement model is given by

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

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

An alternative representation of the measurement error is given by

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

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

The dynamic structure is given by

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

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

Value

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

call Function call.

args Function arguments.

fun Function used ("Indirect").

output The indirect effect.

Author(s)

Ivan Jacob Agaloos Pesigan

References

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

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

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:[10.1037/met0000779](https://doi.org/10.1037/met0000779)

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: `BootBeta()`, `BootBetaStd()`, `BootDirectCentral()`, `BootDirectCentralStd()`, `BootIndirectCentral()`, `BootIndirectCentralStd()`, `BootMed()`, `BootMedStd()`, `BootTotalCentral()`, `BootTotalCentralStd()`, `DeltaBeta()`, `DeltaBetaStd()`, `DeltaDirectCentral()`, `DeltaDirectCentralStd()`, `DeltaIndirectCentral()`, `DeltaMed()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `DeltaTotalCentralStd()`, `Direct()`, `DirectCentral()`, `DirectCentralStd()`, `DirectStd()`, `IndirectCentral()`, `IndirectCentralStd()`, `IndirectStd()`, `MCBeta()`, `MCBetaStd()`, `MCDirectCentral()`, `MCDirectCentralStd()`, `MCIndirectCentral()`, `MCIndirectCentralStd()`, `MCMed()`, `MCMedStd()`, `MCPPhi()`, `MCPPhiSigma()`, `MCTotalCentral()`, `MCTotalCentralStd()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorBetaStd()`, `PosteriorDirectCentral()`, `PosteriorDirectCentralStd()`, `PosteriorIndirectCentral()`, `PosteriorIndirectCentralStd()`, `PosteriorMed()`, `PosteriorMedStd()`, `PosteriorTotalCentral()`, `PosteriorTotalCentralStd()`, `Total()`, `TotalCentral()`, `TotalCentralStd()`, `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, tol = 0.001)
```

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).
tol	Numeric. Smallest possible time interval to allow.

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)

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](https://doi.org/10.1037/met0000779)

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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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)

```

IndirectCentralStd	<i>Standardized Indirect Effect Centrality</i>
--------------------	--

Description

Standardized Indirect Effect Centrality

Usage

```
IndirectCentralStd(phi, sigma, delta_t, tol = 0.001)
```

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).
tol	Numeric. Smallest possible time interval to allow.

Details

Standardized indirect effect centrality is the sum of all possible standardized 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 ("IndirectCentralStd").

output A matrix of standardized indirect effect centrality.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Continuous-Time Mediation Functions: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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(
      0.24455556, 0.02201587, -0.05004762,
      0.02201587, 0.07067800, 0.01539456,
      -0.05004762, 0.01539456, 0.07553061
    ),
    nrow = 3
  )

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

# Range of time intervals -----
indirect_central_std <- IndirectCentralStd(
  phi = phi,
  sigma = sigma,
  delta_t = 1:30
)
plot(indirect_central_std)

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

```

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

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

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 standardized 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
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:10.1037/met0000779
- 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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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(
    0.24455556, 0.02201587, -0.05004762,
    0.02201587, 0.07067800, 0.01539456,
    -0.05004762, 0.01539456, 0.07553061
  ),
  nrow = 3
)
delta_t <- 1
IndirectStd(
  phi = phi,
  sigma = sigma,
  delta_t = delta_t,
  from = "x",
  to = "y",
  med = "m"
)
```

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,
  tol = 0.001
)
```

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 vec(Φ).
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.
tol	Numeric. Smallest possible time interval to allow.

Details

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

Monte Carlo Method:

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

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

Each element in the output list has the following elements:

est Estimated elements of the matrix of lagged coefficients.

thetahatstar A matrix of Monte Carlo elements of the matrix of lagged coefficients.

Author(s)

Ivan Jacob Agaloos Pesigan

References

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

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

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:10.1037/met0000779

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: `BootBeta()`, `BootBetaStd()`, `BootDirectCentral()`, `BootDirectCentralStd()`, `BootIndirectCentral()`, `BootIndirectCentralStd()`, `BootMed()`, `BootMedStd()`, `BootTotalCentral()`, `BootTotalCentralStd()`, `DeltaBeta()`, `DeltaBetaStd()`, `DeltaDirectCentral()`, `DeltaDirectCentralStd()`, `DeltaIndirectCentral()`, `DeltaMed()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `DeltaTotalCentralStd()`, `Direct()`, `DirectCentral()`, `DirectCentralStd()`, `DirectStd()`, `Indirect()`, `IndirectCentral()`, `IndirectCentralStd()`, `IndirectStd()`, `MCBetaStd()`, `MCDirectCentral()`, `MCDirectCentralStd()`, `MCIndirectCentral()`, `MCIndirectCentralStd()`, `MCMed()`, `MCMedStd()`, `MCPhi()`, `MCPhiSigma()`, `MCTotalCentral()`, `MCTotalCentralStd()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorBetaStd()`, `PosteriorDirectCentral()`,

PosteriorDirectCentralStd(), PosteriorIndirectCentral(), PosteriorIndirectCentralStd(),
 PosteriorMed(), PosteriorMedStd(), PosteriorTotalCentral(), PosteriorTotalCentralStd(),
 Total(), TotalCentral(), TotalCentralStd(), 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,
  sigma,
  vcov_theta,
  delta_t,
  R,
  test_phi = TRUE,
  sigma_diag = FALSE,
  ncores = NULL,
  seed = NULL,
  tol = 0.001
)

```

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).
vcov_theta	Numeric matrix. The sampling variance-covariance matrix of $\text{vec}(\Phi)$ and $\text{vech}(\Sigma)$
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.
sigma_diag	Logical. If sigma_diag = TRUE, treat Σ as a diagonal matrix.
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.
tol	Numeric. Smallest possible time interval to allow.

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 $\text{vec}(\hat{\Phi})$ 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 ("MCBetaStd").

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

Each element in the output list has the following elements:

est Estimated elements of the standardized matrix of lagged coefficients.

thetahatstar A matrix of Monte Carlo elements of the standardized 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)

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:[10.1037/met0000779](https://doi.org/10.1037/met0000779)

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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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(
    0.24455556, 0.02201587, -0.05004762,
    0.02201587, 0.07067800, 0.01539456,
    -0.05004762, 0.01539456, 0.07553061
  ),
  nrow = 3
)
vcov_theta <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151, -0.00600, -0.00033,
    0.00110, 0.00324, 0.00020, -0.00061, -0.00115,
    0.00011, 0.00015, 0.00001, -0.00002, -0.00001,
    0.00040, 0.00374, 0.00016, -0.00022, -0.00273,
    -0.00016, 0.00009, 0.00150, 0.00012, -0.00010,
    -0.00026, 0.00002, 0.00012, 0.00004, -0.00001,
    -0.00151, 0.00016, 0.00389, 0.00103, -0.00007,
    -0.00283, -0.00050, 0.00000, 0.00156, 0.00021,
    -0.00005, -0.00031, 0.00001, 0.00007, 0.00006,
    -0.00600, -0.00022, 0.00103, 0.00644, 0.00031,
    -0.00119, -0.00374, -0.00021, 0.00070, 0.00064,
    -0.00015, -0.00005, 0.00000, 0.00003, -0.00001,
    -0.00033, -0.00273, -0.00007, 0.00031, 0.00287,
    0.00013, -0.00014, -0.00170, -0.00012, 0.00006,
    0.00014, -0.00001, -0.00015, 0.00000, 0.00001,
    0.00110, -0.00016, -0.00283, -0.00119, 0.00013,
    0.00297, 0.00063, -0.00004, -0.00177, -0.00013,
    0.00005, 0.00017, -0.00002, -0.00008, 0.00001,
    0.00324, 0.00009, -0.00050, -0.00374, -0.00014,
    0.00063, 0.00495, 0.00024, -0.00093, -0.00020,
    0.00006, -0.00010, 0.00000, -0.00001, 0.00004,
    0.00020, 0.00150, 0.00000, -0.00021, -0.00170,
    -0.00004, 0.00024, 0.00214, 0.00012, -0.00002,
    -0.00004, 0.00000, 0.00006, -0.00005, -0.00001,
    -0.00061, 0.00012, 0.00156, 0.00070, -0.00012,
    -0.00177, -0.00093, 0.00012, 0.00223, 0.00004,
    -0.00002, -0.00003, 0.00001, 0.00003, -0.00013,
    -0.00115, -0.00010, 0.00021, 0.00064, 0.00006,
    -0.00013, -0.00020, -0.00002, 0.00004, 0.00057,
    0.00001, -0.00009, 0.00000, 0.00000, 0.00001,
    0.00011, -0.00026, -0.00005, -0.00015, 0.00014,
    0.00005, 0.00006, -0.00004, -0.00002, 0.00001,
    0.00012, 0.00001, 0.00000, -0.00002, 0.00000,
    0.00015, 0.00002, -0.00031, -0.00005, -0.00001,
    0.00017, -0.00010, 0.00000, -0.00003, -0.00009,
    0.00001, 0.00014, 0.00000, 0.00000, -0.00005,
    0.00001, 0.00012, 0.00001, 0.00000, -0.00015,
    -0.00002, 0.00000, 0.00006, 0.00001, 0.00000,
    0.00000, 0.00000, 0.00010, 0.00001, 0.00000,
    -0.00002, 0.00004, 0.00007, 0.00003, 0.00000,

```

```

-0.00008, -0.00001, -0.00005, 0.00003, 0.00000,
-0.00002, 0.00000, 0.00001, 0.00005, 0.00001,
-0.00001, -0.00001, 0.00006, -0.00001, 0.00001,
0.00001, 0.00004, -0.00001, -0.00013, 0.00001,
0.00000, -0.00005, 0.00000, 0.00001, 0.00012
),
nrow = 15
)

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

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

```

MCDirectCentral

Monte Carlo Sampling Distribution of Direct 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 direct effect centrality at a particular time interval Δt using the first-order stochastic differential equation model drift matrix Φ .

Usage

```
MCDirectCentral(
```

```

    phi,
    vcov_phi_vec,
    delta_t,
    R,
    test_phi = TRUE,
    ncores = NULL,
    seed = NULL,
    tol = 0.001
)

```

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.
tol	Numeric. Smallest possible time interval to allow.

Details

See [DirectCentral\(\)](#) 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 $\mathbf{g}(\hat{\theta})$ 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 ("MCDirectCentral").

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

Each element in the output list has the following elements:

est A vector of direct effect centrality.

thetahatstar A matrix of Monte Carlo direct 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)

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](https://doi.org/10.1037/met0000779)

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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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 -----
MCDirectCentral(
  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 <- MCDirectCentral(
  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 -----
# MCDirectCentral has a number of methods including
# print, summary, confint, and plot
print(mc)
summary(mc)
confint(mc, level = 0.95)
plot(mc)

```

MCDirectCentralStd	<i>Monte Carlo Sampling Distribution of Standardized Direct Effect Centrality Over a Specific Time Interval or a Range of Time Intervals</i>
--------------------	--

Description

This function generates a Monte Carlo method sampling distribution of the standardized direct effect centrality at a particular time interval Δt using the first-order stochastic differential equation model drift matrix Φ and process noise covariance matrix Σ .

Usage

```

MCDirectCentralStd(
  phi,
  sigma,
  vcov_theta,
  delta_t,
  R,
  test_phi = TRUE,
  sigma_diag = FALSE,
  ncores = NULL,
  seed = NULL,
  tol = 0.001
)

```

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).

vcov_theta	Numeric matrix. The sampling variance-covariance matrix of $\text{vec}(\Phi)$ and $\text{vech}(\Sigma)$
delta_t	Numeric. Time interval (Δt).
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> .
sigma_diag	Logical. If <code>sigma_diag = TRUE</code> , treat Σ as a diagonal matrix.
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.
tol	Numeric. Smallest possible time interval to allow.

Details

See [DirectCentralStd\(\)](#) 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 $g(\hat{\theta})$ 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 ("MCDirectCentralStd").

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

Each element in the output list has the following elements:

est A vector of standardized direct effect centrality.

thetahatstar A matrix of Monte Carlo standardized direct 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
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:10.1037/met0000779
- 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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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(
    0.24455556, 0.02201587, -0.05004762,
    0.02201587, 0.07067800, 0.01539456,
    -0.05004762, 0.01539456, 0.07553061
  ),
  nrow = 3
)
```

```

)
vcov_theta <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151, -0.00600, -0.00033,
    0.00110, 0.00324, 0.00020, -0.00061, -0.00115,
    0.00011, 0.00015, 0.00001, -0.00002, -0.00001,
    0.00040, 0.00374, 0.00016, -0.00022, -0.00273,
    -0.00016, 0.00009, 0.00150, 0.00012, -0.00010,
    -0.00026, 0.00002, 0.00012, 0.00004, -0.00001,
    -0.00151, 0.00016, 0.00389, 0.00103, -0.00007,
    -0.00283, -0.00050, 0.00000, 0.00156, 0.00021,
    -0.00005, -0.00031, 0.00001, 0.00007, 0.00006,
    -0.00600, -0.00022, 0.00103, 0.00644, 0.00031,
    -0.00119, -0.00374, -0.00021, 0.00070, 0.00064,
    -0.00015, -0.00005, 0.00000, 0.00003, -0.00001,
    -0.00033, -0.00273, -0.00007, 0.00031, 0.00287,
    0.00013, -0.00014, -0.00170, -0.00012, 0.00006,
    0.00014, -0.00001, -0.00015, 0.00000, 0.00001,
    0.00110, -0.00016, -0.00283, -0.00119, 0.00013,
    0.00297, 0.00063, -0.00004, -0.00177, -0.00013,
    0.00005, 0.00017, -0.00002, -0.00008, 0.00001,
    0.00324, 0.00009, -0.00050, -0.00374, -0.00014,
    0.00063, 0.00495, 0.00024, -0.00093, -0.00020,
    0.00006, -0.00010, 0.00000, -0.00001, 0.00004,
    0.00020, 0.00150, 0.00000, -0.00021, -0.00170,
    -0.00004, 0.00024, 0.00214, 0.00012, -0.00002,
    -0.00004, 0.00000, 0.00006, -0.00005, -0.00001,
    -0.00061, 0.00012, 0.00156, 0.00070, -0.00012,
    -0.00177, -0.00093, 0.00012, 0.00223, 0.00004,
    -0.00002, -0.00003, 0.00001, 0.00003, -0.00013,
    -0.00115, -0.00010, 0.00021, 0.00064, 0.00006,
    -0.00013, -0.00020, -0.00002, 0.00004, 0.00057,
    0.00001, -0.00009, 0.00000, 0.00000, 0.00001,
    0.00011, -0.00026, -0.00005, -0.00015, 0.00014,
    0.00005, 0.00006, -0.00004, -0.00002, 0.00001,
    0.00012, 0.00001, 0.00000, -0.00002, 0.00000,
    0.00015, 0.00002, -0.00031, -0.00005, -0.00001,
    0.00017, -0.00010, 0.00000, -0.00003, -0.00009,
    0.00001, 0.00014, 0.00000, 0.00000, -0.00005,
    0.00001, 0.00012, 0.00001, 0.00000, -0.00015,
    -0.00002, 0.00000, 0.00006, 0.00001, 0.00000,
    0.00000, 0.00000, 0.00010, 0.00001, 0.00000,
    -0.00002, 0.00004, 0.00007, 0.00003, 0.00000,
    -0.00008, -0.00001, -0.00005, 0.00003, 0.00000,
    -0.00002, 0.00000, 0.00001, 0.00005, 0.00001,
    -0.00001, -0.00001, 0.00006, -0.00001, 0.00001,
    0.00001, 0.00004, -0.00001, -0.00013, 0.00001,
    0.00000, -0.00005, 0.00000, 0.00001, 0.00012
  ),
  nrow = 15
)

# Specific time interval -----

```

```

MCDirectCentralStd(
  phi = phi,
  sigma = sigma,
  vcov_theta = vcov_theta,
  delta_t = 1,
  R = 100L # use a large value for R in actual research
)

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

# Methods -----
# MCDirectCentralStd 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,
  tol = 0.001
)

```

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.
tol	Numeric. Smallest possible time interval to allow.

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 of length `length(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
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:10.1037/met0000779
- 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: `BootBeta()`, `BootBetaStd()`, `BootDirectCentral()`, `BootDirectCentralStd()`, `BootIndirectCentral()`, `BootIndirectCentralStd()`, `BootMed()`, `BootMedStd()`, `BootTotalCentral()`, `BootTotalCentralStd()`, `DeltaBeta()`, `DeltaBetaStd()`, `DeltaDirectCentral()`, `DeltaDirectCentralStd()`, `DeltaIndirectCentral()`, `DeltaMed()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `DeltaTotalCentralStd()`, `Direct()`, `DirectCentral()`, `DirectCentralStd()`, `DirectStd()`, `Indirect()`, `IndirectCentral()`, `IndirectCentralStd()`, `IndirectStd()`, `MCBeta()`, `MCBetaStd()`, `MCDirectCentral()`, `MCDirectCentralStd()`, `MCIndirectCentralStd()`, `MCMed()`, `MCMedStd()`, `MCPhi()`, `MCPhiSigma()`, `MCTotalCentral()`, `MCTotalCentralStd()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorBetaStd()`, `PosteriorDirectCentral()`, `PosteriorDirectCentralStd()`, `PosteriorIndirectCentral()`, `PosteriorIndirectCentralStd()`, `PosteriorMed()`, `PosteriorMedStd()`, `PosteriorTotalCentral()`, `PosteriorTotalCentralStd()`, `Total()`, `TotalCentral()`, `TotalCentralStd()`, `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)

```

MCIndirectCentralStd *Monte Carlo Sampling Distribution of Standardized 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 standardized indirect effect centrality at a particular time interval Δt using the first-order stochastic differential equation model drift matrix Φ and process noise covariance matrix Σ .

Usage

```
MCIndirectCentralStd(
  phi,
  sigma,
  vcov_theta,
  delta_t,
  R,
  test_phi = TRUE,
  sigma_diag = FALSE,
  ncores = NULL,
  seed = NULL,
  tol = 0.001
)
```

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).
vcov_theta	Numeric matrix. The sampling variance-covariance matrix of vec (Φ) and vech (Σ)
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.
sigma_diag	Logical. If sigma_diag = TRUE, treat Σ as a diagonal matrix.
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.
tol	Numeric. Smallest possible time interval to allow.

Details

See `IndirectCentralStd()` 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 $g(\hat{\theta})$ 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 ("MCIndirectCentralStd").

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

Each element in the output list has the following elements:

est A vector of standardized indirect effect centrality.

thetahatstar A matrix of Monte Carlo standardized 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

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:10.1037/met0000779

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: `BootBeta()`, `BootBetaStd()`, `BootDirectCentral()`, `BootDirectCentralStd()`, `BootIndirectCentral()`, `BootIndirectCentralStd()`, `BootMed()`, `BootMedStd()`, `BootTotalCentral()`, `BootTotalCentralStd()`, `DeltaBeta()`, `DeltaBetaStd()`, `DeltaDirectCentral()`, `DeltaDirectCentralStd()`, `DeltaIndirectCentral()`, `DeltaMed()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `DeltaTotalCentralStd()`, `Direct()`, `DirectCentral()`, `DirectCentralStd()`, `DirectStd()`, `Indirect()`, `IndirectCentral()`, `IndirectCentralStd()`, `IndirectStd()`, `MCBeta()`, `MCBetaStd()`, `MCDirectCentral()`, `MCDirectCentralStd()`, `MCIndirectCentral()`, `MCIndirectStd()`, `MCMed()`, `MCMedStd()`, `MCPPhi()`, `MCPPhiSigma()`, `MCTotalCentral()`, `MCTotalCentralStd()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorBetaStd()`, `PosteriorDirectCentral()`, `PosteriorDirectCentralStd()`, `PosteriorIndirectCentral()`, `PosteriorIndirectCentralStd()`, `PosteriorMed()`, `PosteriorMedStd()`, `PosteriorTotalCentral()`, `PosteriorTotalCentralStd()`, `Total()`, `TotalCentral()`, `TotalCentralStd()`, `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(
    0.24455556, 0.02201587, -0.05004762,
    0.02201587, 0.07067800, 0.01539456,
    -0.05004762, 0.01539456, 0.07553061
  ),
  nrow = 3
)
vcov_theta <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151, -0.00600, -0.00033,
    0.00110, 0.00324, 0.00020, -0.00061, -0.00115,
    0.00011, 0.00015, 0.00001, -0.00002, -0.00001,
    0.00040, 0.00374, 0.00016, -0.00022, -0.00273,
    -0.00016, 0.00009, 0.00150, 0.00012, -0.00010,
    -0.00026, 0.00002, 0.00012, 0.00004, -0.00001,
    -0.00151, 0.00016, 0.00389, 0.00103, -0.00007,
    -0.00283, -0.00050, 0.00000, 0.00156, 0.00021,
    -0.00005, -0.00031, 0.00001, 0.00007, 0.00006,
```

```

-0.00600, -0.00022, 0.00103, 0.00644, 0.00031,
-0.00119, -0.00374, -0.00021, 0.00070, 0.00064,
-0.00015, -0.00005, 0.00000, 0.00003, -0.00001,
-0.00033, -0.00273, -0.00007, 0.00031, 0.00287,
0.00013, -0.00014, -0.00170, -0.00012, 0.00006,
0.00014, -0.00001, -0.00015, 0.00000, 0.00001,
0.00110, -0.00016, -0.00283, -0.00119, 0.00013,
0.00297, 0.00063, -0.00004, -0.00177, -0.00013,
0.00005, 0.00017, -0.00002, -0.00008, 0.00001,
0.00324, 0.00009, -0.00050, -0.00374, -0.00014,
0.00063, 0.00495, 0.00024, -0.00093, -0.00020,
0.00006, -0.00010, 0.00000, -0.00001, 0.00004,
0.00020, 0.00150, 0.00000, -0.00021, -0.00170,
-0.00004, 0.00024, 0.00214, 0.00012, -0.00002,
-0.00004, 0.00000, 0.00006, -0.00005, -0.00001,
-0.00061, 0.00012, 0.00156, 0.00070, -0.00012,
-0.00177, -0.00093, 0.00012, 0.00223, 0.00004,
-0.00002, -0.00003, 0.00001, 0.00003, -0.00013,
-0.00115, -0.00010, 0.00021, 0.00064, 0.00006,
-0.00013, -0.00020, -0.00002, 0.00004, 0.00057,
0.00001, -0.00009, 0.00000, 0.00000, 0.00001,
0.00011, -0.00026, -0.00005, -0.00015, 0.00014,
0.00005, 0.00006, -0.00004, -0.00002, 0.00001,
0.00012, 0.00001, 0.00000, -0.00002, 0.00000,
0.00015, 0.00002, -0.00031, -0.00005, -0.00001,
0.00017, -0.00010, 0.00000, -0.00003, -0.00009,
0.00001, 0.00014, 0.00000, 0.00000, -0.00005,
0.00001, 0.00012, 0.00001, 0.00000, -0.00015,
-0.00002, 0.00000, 0.00006, 0.00001, 0.00000,
0.00000, 0.00000, 0.00010, 0.00001, 0.00000,
-0.00002, 0.00004, 0.00007, 0.00003, 0.00000,
-0.00008, -0.00001, -0.00005, 0.00003, 0.00000,
-0.00002, 0.00000, 0.00001, 0.00005, 0.00001,
-0.00001, -0.00001, 0.00006, -0.00001, 0.00001,
0.00001, 0.00004, -0.00001, -0.00013, 0.00001,
0.00000, -0.00005, 0.00000, 0.00001, 0.00012
),
nrow = 15
)

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

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

```

```

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

# Methods -----
# MCIndirectCentralStd 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,
  tol = 0.001
)

```

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 <code>vec(Φ)</code> .

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.
tol	Numeric. Smallest possible time interval to allow.

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 of length `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
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:10.1037/met0000779
- 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: `BootBeta()`, `BootBetaStd()`, `BootDirectCentral()`, `BootDirectCentralStd()`, `BootIndirectCentral()`, `BootIndirectCentralStd()`, `BootMed()`, `BootMedStd()`, `BootTotalCentral()`, `BootTotalCentralStd()`, `DeltaBeta()`, `DeltaBetaStd()`, `DeltaDirectCentral()`, `DeltaDirectCentralStd()`, `DeltaIndirectCentral()`, `DeltaMed()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `DeltaTotalCentralStd()`, `Direct()`, `DirectCentral()`, `DirectCentralStd()`, `DirectStd()`, `Indirect()`, `IndirectCentral()`, `IndirectCentralStd()`, `IndirectStd()`, `MCBeta()`, `MCBetaStd()`, `MCDirectCentral()`, `MCDirectCentralStd()`, `MCIndirectCentral()`, `MCIndirectCentralStd()`, `MCMedStd()`, `MCPhi()`, `MCPhiSigma()`, `MCTotalCentral()`, `MCTotalCentralStd()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorBetaStd()`, `PosteriorDirectCentral()`, `PosteriorDirectCentralStd()`, `PosteriorIndirectCentral()`, `PosteriorIndirectCentralStd()`, `PosteriorMed()`, `PosteriorMedStd()`, `PosteriorTotalCentral()`, `PosteriorTotalCentralStd()`, `Total()`, `TotalCentral()`, `TotalCentralStd()`, `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,
  sigma,
  vcov_theta,
  delta_t,
  from,
  to,
  med,
  R,
  test_phi = TRUE,
  sigma_diag = FALSE,
  ncores = NULL,
  seed = NULL,
  tol = 0.001
)
```

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).
vcov_theta	Numeric matrix. The sampling variance-covariance matrix of vec (Φ) and vech (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.

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.
sigma_diag	Logical. If sigma_diag = TRUE, treat Σ as a diagonal matrix.
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.
tol	Numeric. Smallest possible time interval to allow.

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 $g(\hat{\theta})$ 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 ("MCMedStd").

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

Each element in the output list has the following elements:

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

thetahatstar A matrix of Monte Carlo standardized 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
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:10.1037/met0000779
- 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: `BootBeta()`, `BootBetaStd()`, `BootDirectCentral()`, `BootDirectCentralStd()`, `BootIndirectCentral()`, `BootIndirectCentralStd()`, `BootMed()`, `BootMedStd()`, `BootTotalCentral()`, `BootTotalCentralStd()`, `DeltaBeta()`, `DeltaBetaStd()`, `DeltaDirectCentral()`, `DeltaDirectCentralStd()`, `DeltaIndirectCentral()`, `DeltaMed()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `DeltaTotalCentralStd()`, `Direct()`, `DirectCentral()`, `DirectCentralStd()`, `DirectStd()`, `Indirect()`, `IndirectCentral()`, `IndirectCentralStd()`, `IndirectStd()`, `MCBeta()`, `MCBetaStd()`, `MCDirectCentral()`, `MCDirectCentralStd()`, `MCIndirectCentral()`, `MCIndirectCentralStd()`, `MCMed()`, `MCPhi()`, `MCPhiSigma()`, `MCTotalCentral()`, `MCTotalCentralStd()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorBetaStd()`, `PosteriorDirectCentral()`, `PosteriorDirectCentralStd()`, `PosteriorIndirectCentral()`, `PosteriorIndirectCentralStd()`, `PosteriorMed()`, `PosteriorMedStd()`, `PosteriorTotalCentral()`, `PosteriorTotalCentralStd()`, `Total()`, `TotalCentral()`, `TotalCentralStd()`, `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(
    0.24455556, 0.02201587, -0.05004762,
    0.02201587, 0.07067800, 0.01539456,
    -0.05004762, 0.01539456, 0.07553061
  ),
  nrow = 3
)
```

```

)
vcov_theta <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151, -0.00600, -0.00033,
    0.00110, 0.00324, 0.00020, -0.00061, -0.00115,
    0.00011, 0.00015, 0.00001, -0.00002, -0.00001,
    0.00040, 0.00374, 0.00016, -0.00022, -0.00273,
    -0.00016, 0.00009, 0.00150, 0.00012, -0.00010,
    -0.00026, 0.00002, 0.00012, 0.00004, -0.00001,
    -0.00151, 0.00016, 0.00389, 0.00103, -0.00007,
    -0.00283, -0.00050, 0.00000, 0.00156, 0.00021,
    -0.00005, -0.00031, 0.00001, 0.00007, 0.00006,
    -0.00600, -0.00022, 0.00103, 0.00644, 0.00031,
    -0.00119, -0.00374, -0.00021, 0.00070, 0.00064,
    -0.00015, -0.00005, 0.00000, 0.00003, -0.00001,
    -0.00033, -0.00273, -0.00007, 0.00031, 0.00287,
    0.00013, -0.00014, -0.00170, -0.00012, 0.00006,
    0.00014, -0.00001, -0.00015, 0.00000, 0.00001,
    0.00110, -0.00016, -0.00283, -0.00119, 0.00013,
    0.00297, 0.00063, -0.00004, -0.00177, -0.00013,
    0.00005, 0.00017, -0.00002, -0.00008, 0.00001,
    0.00324, 0.00009, -0.00050, -0.00374, -0.00014,
    0.00063, 0.00495, 0.00024, -0.00093, -0.00020,
    0.00006, -0.00010, 0.00000, -0.00001, 0.00004,
    0.00020, 0.00150, 0.00000, -0.00021, -0.00170,
    -0.00004, 0.00024, 0.00214, 0.00012, -0.00002,
    -0.00004, 0.00000, 0.00006, -0.00005, -0.00001,
    -0.00061, 0.00012, 0.00156, 0.00070, -0.00012,
    -0.00177, -0.00093, 0.00012, 0.00223, 0.00004,
    -0.00002, -0.00003, 0.00001, 0.00003, -0.00013,
    -0.00115, -0.00010, 0.00021, 0.00064, 0.00006,
    -0.00013, -0.00020, -0.00002, 0.00004, 0.00057,
    0.00001, -0.00009, 0.00000, 0.00000, 0.00001,
    0.00011, -0.00026, -0.00005, -0.00015, 0.00014,
    0.00005, 0.00006, -0.00004, -0.00002, 0.00001,
    0.00012, 0.00001, 0.00000, -0.00002, 0.00000,
    0.00015, 0.00002, -0.00031, -0.00005, -0.00001,
    0.00017, -0.00010, 0.00000, -0.00003, -0.00009,
    0.00001, 0.00014, 0.00000, 0.00000, -0.00005,
    0.00001, 0.00012, 0.00001, 0.00000, -0.00015,
    -0.00002, 0.00000, 0.00006, 0.00001, 0.00000,
    0.00000, 0.00000, 0.00010, 0.00001, 0.00000,
    -0.00002, 0.00004, 0.00007, 0.00003, 0.00000,
    -0.00008, -0.00001, -0.00005, 0.00003, 0.00000,
    -0.00002, 0.00000, 0.00001, 0.00005, 0.00001,
    -0.00001, -0.00001, 0.00006, -0.00001, 0.00001,
    0.00001, 0.00004, -0.00001, -0.00013, 0.00001,
    0.00000, -0.00005, 0.00000, 0.00001, 0.00012
  ),
  nrow = 15
)

# Specific time interval -----

```

```

MCMedStd(
  phi = phi,
  sigma = sigma,
  vcov_theta = vcov_theta,
  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,
  sigma = sigma,
  vcov_theta = vcov_theta,
  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 `ctmedmcpHi` which is a list with the following elements:

call Function call.

args Function arguments.

fun Function used ("MCPHi").

output A list of simulated drift matrices.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Continuous-Time Mediation Functions: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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
)

```

MCPhiSigma

Generate Random Drift Matrices and Process Noise Covariance Matrices Using the Monte Carlo Method

Description

This function generates random drift matrices Φ and process noise covariance matrices Σ using the Monte Carlo method.

Usage

```

MCPhiSigma(
  phi,
  sigma,
  vcov_theta,

```

```

R,
test_phi = TRUE,
sigma_diag = FALSE,
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.
sigma	Numeric matrix. The process noise covariance matrix (Σ).
vcov_theta	Numeric matrix. The sampling variance-covariance matrix of $\text{vec}(\Phi)$ and $\text{vech}(\Sigma)$
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.
sigma_diag	Logical. If sigma_diag = TRUE, treat Σ as a diagonal matrix.
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 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,

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

Value

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

call Function call.

args Function arguments.

fun Function used ("MCPhiSigma").

output A list of simulated draws. Each element contains a drift matrix and a process noise covariance matrix.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Continuous-Time Mediation Functions: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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")
sigma <- matrix(
  data = c(
    0.24455556, 0.02201587, -0.05004762,
    0.02201587, 0.07067800, 0.01539456,
    -0.05004762, 0.01539456, 0.07553061
  ),
  nrow = 3
)
MCPhiSigma(
  phi = phi,
  sigma = sigma,
  vcov_theta = 0.1 * diag(15),
  R = 100L # use a large value for R in actual research
)
```

MCTotalCentral	<i>Monte Carlo Sampling Distribution of Total Effect Centrality Over a Specific Time Interval or a Range of Time Intervals</i>
----------------	--

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,
  tol = 0.001
)
```

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.
tol	Numeric. Smallest possible time interval to allow.

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 of length `length(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](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)
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](https://doi.org/10.1037/met0000779)
- 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: `BootBeta()`, `BootBetaStd()`, `BootDirectCentral()`, `BootDirectCentralStd()`, `BootIndirectCentral()`, `BootIndirectCentralStd()`, `BootMed()`, `BootMedStd()`, `BootTotalCentral()`, `BootTotalCentralStd()`, `DeltaBeta()`, `DeltaBetaStd()`, `DeltaDirectCentral()`, `DeltaDirectCentralStd()`, `DeltaIndirectCentral()`, `DeltaMed()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `DeltaTotalCentralStd()`, `Direct()`, `DirectCentral()`, `DirectCentralStd()`, `DirectStd()`, `Indirect()`, `IndirectCentral()`, `IndirectCentralStd()`, `IndirectStd()`, `MCBeta()`, `MCBetaStd()`, `MCDirectCentral()`, `MCDirectCentralStd()`, `MCIndirectCentral()`, `MCIndirectCentralStd()`, `MCMed()`, `MCMedStd()`, `MCPPhi()`, `MCPPhiSigma()`, `MCTotalCentralStd()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorBetaStd()`, `PosteriorDirectCentral()`, `PosteriorDirectCentralStd()`, `PosteriorIndirectCentral()`, `PosteriorIndirectCentralStd()`, `PosteriorMed()`, `PosteriorMedStd()`, `PosteriorTotalCentral()`, `PosteriorTotalCentralStd()`, `Total()`, `TotalCentral()`, `TotalCentralStd()`, `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)

```

MCTotalCentralStd

Monte Carlo Sampling Distribution of Standardized 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 standardized total effect centrality at a particular time interval Δt using the first-order stochastic differential equation model drift matrix Φ and process noise covariance matrix Σ .

Usage

```

MCTotalCentralStd(
  phi,
  sigma,
  vcov_theta,

```

```

    delta_t,
    R,
    test_phi = TRUE,
    sigma_diag = FALSE,
    ncores = NULL,
    seed = NULL,
    tol = 0.001
)

```

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).
vcov_theta	Numeric matrix. The sampling variance-covariance matrix of vec (Φ) and vech (Σ)
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.
sigma_diag	Logical. If sigma_diag = TRUE, treat Σ as a diagonal matrix.
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.
tol	Numeric. Smallest possible time interval to allow.

Details

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

Monte Carlo Method:

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

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

Each element in the output list has the following elements:

est A vector of standardized total effect centrality.

thetahatstar A matrix of Monte Carlo standardized 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

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:10.1037/met0000779

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: `BootBeta()`, `BootBetaStd()`, `BootDirectCentral()`, `BootDirectCentralStd()`, `BootIndirectCentral()`, `BootIndirectCentralStd()`, `BootMed()`, `BootMedStd()`, `BootTotalCentral()`, `BootTotalCentralStd()`, `DeltaBeta()`, `DeltaBetaStd()`, `DeltaDirectCentral()`, `DeltaDirectCentralStd()`, `DeltaIndirectCentral()`, `DeltaMed()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `DeltaTotalCentralStd()`, `Direct()`, `DirectCentral()`, `DirectCentralStd()`, `DirectStd()`, `Indirect()`, `IndirectCentral()`, `IndirectCentralStd()`, `IndirectStd()`, `MCBeta()`, `MCBetaStd()`, `MCDirectCentral()`, `MCDirectCentralStd()`, `MCIndirectCentral()`, `MCIndirectCentralStd()`, `MCMed()`, `MCMedStd()`, `MCPHi()`, `MCPHiSigma()`, `MCTotalCentral()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorBetaStd()`, `PosteriorDirectCentral()`, `PosteriorDirectCentralStd()`, `PosteriorIndirectCentral()`, `PosteriorIndirectCentralStd()`, `PosteriorMed()`, `PosteriorMedStd()`, `PosteriorTotalCentral()`, `PosteriorTotalCentralStd()`, `Total()`, `TotalCentral()`, `TotalCentralStd()`, `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(
    0.24455556, 0.02201587, -0.05004762,
    0.02201587, 0.07067800, 0.01539456,
    -0.05004762, 0.01539456, 0.07553061
  ),
  nrow = 3
)
vcov_theta <- matrix(
  data = c(
    0.00843, 0.00040, -0.00151, -0.00600, -0.00033,
    0.00110, 0.00324, 0.00020, -0.00061, -0.00115,
    0.00011, 0.00015, 0.00001, -0.00002, -0.00001,
    0.00040, 0.00374, 0.00016, -0.00022, -0.00273,
    -0.00016, 0.00009, 0.00150, 0.00012, -0.00010,
    -0.00026, 0.00002, 0.00012, 0.00004, -0.00001,
    -0.00151, 0.00016, 0.00389, 0.00103, -0.00007,
    -0.00283, -0.00050, 0.00000, 0.00156, 0.00021,
    -0.00005, -0.00031, 0.00001, 0.00007, 0.00006,
    -0.00600, -0.00022, 0.00103, 0.00644, 0.00031,
    -0.00119, -0.00374, -0.00021, 0.00070, 0.00064,
    -0.00015, -0.00005, 0.00000, 0.00003, -0.00001,
    -0.00033, -0.00273, -0.00007, 0.00031, 0.00287,
    0.00013, -0.00014, -0.00170, -0.00012, 0.00006,
    0.00014, -0.00001, -0.00015, 0.00000, 0.00001,
    0.00110, -0.00016, -0.00283, -0.00119, 0.00013,
    0.00297, 0.00063, -0.00004, -0.00177, -0.00013,
    0.00005, 0.00017, -0.00002, -0.00008, 0.00001,
    0.00324, 0.00009, -0.00050, -0.00374, -0.00014,
    0.00063, 0.00495, 0.00024, -0.00093, -0.00020,
    0.00006, -0.00010, 0.00000, -0.00001, 0.00004,
    0.00020, 0.00150, 0.00000, -0.00021, -0.00170,
    -0.00004, 0.00024, 0.00214, 0.00012, -0.00002,
    -0.00004, 0.00000, 0.00006, -0.00005, -0.00001,
    -0.00061, 0.00012, 0.00156, 0.00070, -0.00012,
    -0.00177, -0.00093, 0.00012, 0.00223, 0.00004,
    -0.00002, -0.00003, 0.00001, 0.00003, -0.00013,
    -0.00115, -0.00010, 0.00021, 0.00064, 0.00006,
    -0.00013, -0.00020, -0.00002, 0.00004, 0.00057,
    0.00001, -0.00009, 0.00000, 0.00000, 0.00001,
    0.00011, -0.00026, -0.00005, -0.00015, 0.00014,
    0.00005, 0.00006, -0.00004, -0.00002, 0.00001,

```

```

0.00012, 0.00001, 0.00000, -0.00002, 0.00000,
0.00015, 0.00002, -0.00031, -0.00005, -0.00001,
0.00017, -0.00010, 0.00000, -0.00003, -0.00009,
0.00001, 0.00014, 0.00000, 0.00000, -0.00005,
0.00001, 0.00012, 0.00001, 0.00000, -0.00015,
-0.00002, 0.00000, 0.00006, 0.00001, 0.00000,
0.00000, 0.00000, 0.00010, 0.00001, 0.00000,
-0.00002, 0.00004, 0.00007, 0.00003, 0.00000,
-0.00008, -0.00001, -0.00005, 0.00003, 0.00000,
-0.00002, 0.00000, 0.00001, 0.00005, 0.00001,
-0.00001, -0.00001, 0.00006, -0.00001, 0.00001,
0.00001, 0.00004, -0.00001, -0.00013, 0.00001,
0.00000, -0.00005, 0.00000, 0.00001, 0.00012
),
nrow = 15
)

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

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

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

```

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, tol = 0.001)
```

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.
tol	Numeric. Smallest possible time interval to allow.

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

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

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, Σ is the matrix of volatility or randomness in the process, and $d\mathbf{W}$ is a Wiener process or Brownian motion, which represents random fluctuations.

Value

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

call Function call.

args Function arguments.

fun Function used ("Med").

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

Author(s)

Ivan Jacob Agaloos Pesigan

References

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

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

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:[10.1037/met0000779](https://doi.org/10.1037/met0000779)

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: `BootBeta()`, `BootBetaStd()`, `BootDirectCentral()`, `BootDirectCentralStd()`, `BootIndirectCentral()`, `BootIndirectCentralStd()`, `BootMed()`, `BootMedStd()`, `BootTotalCentral()`, `BootTotalCentralStd()`, `DeltaBeta()`, `DeltaBetaStd()`, `DeltaDirectCentral()`, `DeltaDirectCentralStd()`, `DeltaIndirectCentral()`, `DeltaMed()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `DeltaTotalCentralStd()`, `Direct()`, `DirectCentral()`, `DirectCentralStd()`, `DirectStd()`, `Indirect()`, `IndirectCentral()`, `IndirectCentralStd()`, `IndirectStd()`, `MCBeta()`, `MCBetaStd()`, `MCDirectCentral()`, `MCDirectCentralStd()`, `MCIndirectCentral()`, `MCIndirectCentralStd()`, `MCMed()`, `MCMedStd()`, `MCPhi()`, `MCPhiSigma()`, `MCTotalCentral()`, `MCTotalCentralStd()`, `MedStd()`, `PosteriorBeta()`, `PosteriorBetaStd()`, `PosteriorDirectCentral()`, `PosteriorDirectCentralStd()`, `PosteriorIndirectCentral()`, `PosteriorIndirectCentralStd()`, `PosteriorMed()`, `PosteriorMedStd()`, `PosteriorTotalCentral()`, `PosteriorTotalCentralStd()`, `Total()`, `TotalCentral()`, `TotalCentralStd()`, `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

*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 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, tol = 0.001)
```

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.
tol	Numeric. Smallest possible time interval to allow.

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 standardized 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](#)
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](#)
- 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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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(
    0.24455556, 0.02201587, -0.05004762,
    0.02201587, 0.07067800, 0.01539456,
    -0.05004762, 0.01539456, 0.07553061
  ),
  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.ctmedboot

Plot Method for an Object of Class ctmedboot

Description

Plot Method for an Object of Class ctmedboot

Usage

```

## S3 method for class 'ctmedboot'
plot(x, alpha = 0.05, col = NULL, type = "pc", ...)

```

Arguments

x	Object of class ctmedboot.
alpha	Numeric. Significance level
col	Character vector. Optional argument. Character vector of colors.
type	Character string. Confidence interval type, that is, type = "pc" for percentile; type = "bc" for bias corrected.
...	Additional arguments.

Value

Displays plots of point estimates and confidence intervals.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```

## Not run:
library(bootStateSpace)
# prepare parameters
## number of individuals
n <- 50
## time points
time <- 100
delta_t <- 0.10
## dynamic structure
p <- 3
mu0 <- rep(x = 0, times = p)
sigma0 <- matrix(
  data = c(
    1.0,
    0.2,
    0.2,
    0.2,
    1.0,
    0.2,
    0.2,
    0.2,
    1.0
  ),
  nrow = p
)
sigma0_l <- t(chol(sigma0))
mu <- rep(x = 0, times = p)
phi <- matrix(
  data = c(
    -0.357,
    0.771,
    -0.450,
    0.0,
    -0.511,
    0.729,
    0,
    0,
    -0.693
  ),
  nrow = p
)
sigma <- matrix(
  data = c(
    0.24455556,
    0.02201587,
    -0.05004762,
    0.02201587,
    0.07067800,
    0.01539456,
    -0.05004762,
    0.01539456,

```

```

      0.07553061
    ),
    nrow = p
  )
  sigma_l <- t(chol(sigma))
  ## measurement model
  k <- 3
  nu <- rep(x = 0, times = k)
  lambda <- diag(k)
  theta <- 0.2 * diag(k)
  theta_l <- t(chol(theta))

  boot <- PBSSMOUFixed(
    R = 1000L,
    path = getwd(),
    prefix = "ou",
    n = n,
    time = time,
    delta_t = delta_t,
    mu0 = mu0,
    sigma0_l = sigma0_l,
    mu = mu,
    phi = phi,
    sigma_l = sigma_l,
    nu = nu,
    lambda = lambda,
    theta_l = theta_l,
    ncores = parallel::detectCores() - 1,
    seed = 42
  )
  phi_hat <- phi
  colnames(phi_hat) <- rownames(phi_hat) <- c("x", "m", "y")
  phi <- extract(object = boot, what = "phi")

  # Range of time intervals -----
  boot <- BootMed(
    phi = phi,
    phi_hat = phi_hat,
    delta_t = 1:5,
    from = "x",
    to = "y",
    med = "m"
  )
  confint(boot)
  confint(boot, type = "bc") # bias-corrected

  ## End(Not run)

```

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	<i>Plot Method for an Object of Class ctmedmed</i>
---------------	--

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	<i>Plot Method for an Object of Class ctmedtraj</i>
----------------	---

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, tol = 0.001)
```

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.
tol	Numeric. Smallest possible time interval to allow.

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 of length `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)

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](https://doi.org/10.1037/met0000779)

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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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)

```

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

Description

This function generates a posterior 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 posterior distribution of the first-order stochastic differential equation model drift matrix Φ and process noise covariance matrix Σ .

Usage

```
PosteriorBetaStd(phi, sigma, delta_t, ncores = NULL, tol = 0.001)
```

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.
sigma	List of numeric matrices. Each element of the list is a sample from the posterior distribution of the process noise covariance matrix (Σ). 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.
tol	Numeric. Smallest possible time interval to allow.

Details

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

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

Each element in the output list has the following elements:

est Mean of the posterior distribution of the elements of the standardized matrix of lagged coefficients.

thetahatstar Posterior distribution of the elements of the standardized matrix of lagged coefficients.

Author(s)

Ivan Jacob Agaloos Pesigan

References

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

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

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](#)

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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#),

PosteriorTotalCentral(),PosteriorTotalCentralStd(),Total(),TotalCentral(),TotalCentralStd(),
TotalStd(),Trajectory()

Examples

```
set.seed(42)
phi <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.000, -0.511, 0.729,
    0.000, 0.000, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")
sigma <- matrix(
  data = c(
    0.24455556, 0.02201587, -0.05004762,
    0.02201587, 0.07067800, 0.01539456,
    -0.05004762, 0.01539456, 0.07553061
  ),
  nrow = 3
)
colnames(sigma) <- rownames(sigma) <- c("x", "m", "y")
input <- MCPHiSigma(
  phi = phi,
  sigma = sigma,
  vcov_theta = 0.001 * diag(15),
  R = 100L,
  seed = 42
)$output
phi <- lapply(
  X = input,
  FUN = function(x) {
    x[[1]]
  }
)
sigma <- lapply(
  X = input,
  FUN = function(x) {
    x[[2]]
  }
)

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

# Range of time intervals -----
posterior <- PosteriorBetaStd(
```

```

    phi = phi,
    sigma = sigma,
    delta_t = 1:5
  )
  plot(posterior)

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

```

PosteriorDirectCentral

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

Description

This function generates a posterior distribution of the direct 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

```
PosteriorDirectCentral(phi, delta_t, ncores = NULL, tol = 0.001)
```

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.
tol	Numeric. Smallest possible time interval to allow.

Details

See [DirectCentral\(\)](#) 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 ("PosteriorDirectCentral").

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

Each element in the output list has the following elements:

est Mean of the posterior distribution of the direct effect centrality.

thetahatstar Posterior distribution of the direct effect centrality measure.

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

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:10.1037/met0000779

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: `BootBeta()`, `BootBetaStd()`, `BootDirectCentral()`, `BootDirectCentralStd()`, `BootIndirectCentral()`, `BootIndirectCentralStd()`, `BootMed()`, `BootMedStd()`, `BootTotalCentral()`, `BootTotalCentralStd()`, `DeltaBeta()`, `DeltaBetaStd()`, `DeltaDirectCentral()`, `DeltaDirectCentralStd()`, `DeltaIndirectCentral()`, `DeltaMed()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `DeltaTotalCentralStd()`, `Direct()`, `DirectCentral()`, `DirectCentralStd()`, `DirectStd()`, `Indirect()`, `IndirectCentral()`, `IndirectCentralStd()`, `IndirectStd()`, `MCBeta()`, `MCBetaStd()`, `MCDirectCentral()`, `MCDirectCentralStd()`, `MCIndirectCentral()`, `MCIndirectCentralStd()`, `MCMed()`, `MCMedStd()`, `MCPHi()`, `MCPHiSigma()`, `MCTotalCentral()`, `MCTotalCentralStd()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorBetaStd()`, `PosteriorDirectCentralStd()`, `PosteriorIndirectCentral()`, `PosteriorIndirectCentralStd()`, `PosteriorMed()`, `PosteriorMedStd()`, `PosteriorTotalCentral()`, `PosteriorTotalCentralStd()`, `Total()`, `TotalCentral()`, `TotalCentralStd()`, `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 -----
PosteriorDirectCentral(
  phi = phi,

```

```

    delta_t = 1
  )

  # Range of time intervals -----
  posterior <- PosteriorDirectCentral(
    phi = phi,
    delta_t = 1:5
  )

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

```

PosteriorDirectCentralStd

*Posterior Distribution of the Standardized Direct Effect Centrality
Over a Specific Time Interval or a Range of Time Intervals*

Description

This function generates a posterior distribution of the standardized direct 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 and process noise covariance matrix Σ .

Usage

```
PosteriorDirectCentralStd(phi, sigma, delta_t, ncores = NULL, tol = 0.001)
```

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.
sigma	List of numeric matrices. Each element is a posterior draw of the diffusion covariance matrix.
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.
tol	Numeric. Smallest possible time interval to allow.

Details

See [DirectCentralStd\(\)](#) 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 ("PosteriorDirectCentralStd").

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

Each element in the output list has the following elements:

est Mean of the posterior distribution of the standardized direct effect centrality.

thetahatstar Posterior distribution of the standardized direct effect centrality measure.

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](#)

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](#)

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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

Examples

```

set.seed(42)
phi <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.000, -0.511, 0.729,
    0.000, 0.000, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")

sigma <- matrix(
  data = c(
    0.24455556, 0.02201587, -0.05004762,
    0.02201587, 0.07067800, 0.01539456,
    -0.05004762, 0.01539456, 0.07553061
  ),
  nrow = 3
)
colnames(sigma) <- rownames(sigma) <- c("x", "m", "y")

input <- MCPhiSigma(
  phi = phi,
  sigma = sigma,
  vcov_theta = 0.001 * diag(15),
  R = 100L,
  seed = 42
)$output

phi <- lapply(
  X = input,
  FUN = function(x) {
    x[[1]]
  }
)

sigma <- lapply(
  X = input,
  FUN = function(x) {
    x[[2]]
  }
)

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

# Range of time intervals -----

```

```

posterior <- PosteriorDirectCentralStd(
  phi = phi,
  sigma = sigma,
  delta_t = 1:5
)

# Methods -----
# PosteriorDirectCentralStd 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, tol = 0.001)
```

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.
tol	Numeric. Smallest possible time interval to allow.

Details

See [IndirectCentral\(\)](#) 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 of length `length(delta_t)`.

Each element in the output list has the following elements:

est Mean of the posterior distribution of the indirect effect centrality.

thetahatstar Posterior distribution of the indirect effect centrality measure.

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

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:10.1037/met0000779

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: `BootBeta()`, `BootBetaStd()`, `BootDirectCentral()`, `BootDirectCentralStd()`, `BootIndirectCentral()`, `BootIndirectCentralStd()`, `BootMed()`, `BootMedStd()`, `BootTotalCentral()`, `BootTotalCentralStd()`, `DeltaBeta()`, `DeltaBetaStd()`, `DeltaDirectCentral()`, `DeltaDirectCentralStd()`, `DeltaIndirectCentral()`, `DeltaMed()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `DeltaTotalCentralStd()`, `Direct()`, `DirectCentral()`, `DirectCentralStd()`, `DirectStd()`, `Indirect()`, `IndirectCentral()`, `IndirectCentralStd()`, `IndirectStd()`, `MCBeta()`, `MCBetaStd()`, `MCDirectCentral()`, `MCDirectCentralStd()`, `MCIndirectCentral()`, `MCIndirectCentralStd()`, `MCMed()`, `MCMedStd()`, `MCPHi()`, `MCPHiSigma()`, `MCTotalCentral()`, `MCTotalCentralStd()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorBetaStd()`, `PosteriorDirectCentral()`, `PosteriorDirectCentralStd()`, `PosteriorIndirectCentralStd()`, `PosteriorMed()`, `PosteriorMedStd()`, `PosteriorTotalCentral()`, `PosteriorTotalCentralStd()`, `Total()`, `TotalCentral()`, `TotalCentralStd()`, `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)

```

PosteriorIndirectCentralStd

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

Description

This function generates a posterior distribution of the standardized 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 and process noise covariance matrix Σ .

Usage

```
PosteriorIndirectCentralStd(phi, sigma, delta_t, ncores = NULL, tol = 0.001)
```

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.
sigma	List of numeric matrices. Each element is a posterior draw of the diffusion covariance matrix.
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.
tol	Numeric. Smallest possible time interval to allow.

Details

See [IndirectCentralStd\(\)](#) 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 ("PosteriorIndirectCentralStd").

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

Each element in the output list has the following elements:

est Mean of the posterior distribution of the standardized indirect effect centrality.

thetahatstar Posterior distribution of the standardized indirect effect centrality measure.

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

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. doi:10.1037/met0000779

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: `BootBeta()`, `BootBetaStd()`, `BootDirectCentral()`, `BootDirectCentralStd()`, `BootIndirectCentral()`, `BootIndirectCentralStd()`, `BootMed()`, `BootMedStd()`, `BootTotalCentral()`, `BootTotalCentralStd()`, `DeltaBeta()`, `DeltaBetaStd()`, `DeltaDirectCentral()`, `DeltaDirectCentralStd()`, `DeltaIndirectCentral()`, `DeltaMed()`, `DeltaMedStd()`, `DeltaTotalCentral()`, `DeltaTotalCentralStd()`, `Direct()`, `DirectCentral()`, `DirectCentralStd()`, `DirectStd()`, `Indirect()`, `IndirectCentral()`, `IndirectCentralStd()`, `IndirectStd()`, `MCBeta()`, `MCBetaStd()`, `MCDirectCentral()`, `MCDirectCentralStd()`, `MCIndirectCentral()`, `MCIndirectCentralStd()`, `MCMed()`, `MCMedStd()`, `MCPHi()`, `MCPHiSigma()`, `MCTotalCentral()`, `MCTotalCentralStd()`, `Med()`, `MedStd()`, `PosteriorBeta()`, `PosteriorBetaStd()`, `PosteriorDirectCentral()`, `PosteriorDirectCentralStd()`, `PosteriorIndirectCentral()`, `PosteriorMed()`, `PosteriorMedStd()`, `PosteriorTotalCentral()`, `PosteriorTotalCentralStd()`, `Total()`, `TotalCentral()`, `TotalCentralStd()`, `TotalStd()`, `Trajectory()`

Examples

```

set.seed(42)
phi <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.000, -0.511, 0.729,
    0.000, 0.000, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")

sigma <- matrix(
  data = c(
    0.24455556, 0.02201587, -0.05004762,
    0.02201587, 0.07067800, 0.01539456,
    -0.05004762, 0.01539456, 0.07553061
  ),
  nrow = 3
)
colnames(sigma) <- rownames(sigma) <- c("x", "m", "y")

input <- MCPhiSigma(
  phi = phi,
  sigma = sigma,
  vcov_theta = 0.001 * diag(15),
  R = 100L,
  seed = 42
)$output

phi <- lapply(
  X = input,
  FUN = function(x) {
    x[[1]]
  }
)

sigma <- lapply(
  X = input,
  FUN = function(x) {
    x[[2]]
  }
)

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

# Range of time intervals -----

```

```

posterior <- PosteriorIndirectCentralStd(
  phi = phi,
  sigma = sigma,
  delta_t = 1:5
)

# Methods -----
# PosteriorIndirectCentralStd 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 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, tol = 0.001)
```

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.
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.
tol	Numeric. Smallest possible time interval to allow.

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 of length `length(delta_t)`.

Each element in the output list has the following elements:

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

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

Author(s)

Ivan Jacob Agaloos Pesigan

References

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

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](https://doi.org/10.1037/met0000779)

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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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)
```

PosteriorMedStd	<i>Posterior 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</i>
-----------------	--

Description

This function generates a posterior 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 posterior distribution of the first-order stochastic differential equation model drift matrix Φ and process noise covariance matrix Σ .

Usage

```
PosteriorMedStd(phi, sigma, delta_t, from, to, med, ncores = NULL, tol = 0.001)
```

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.
sigma	List of numeric matrices. Each element of the list is a sample from the posterior distribution of the process noise covariance matrix (Σ). 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 ϕ .
to	Character string. Name of the dependent variable Y in ϕ .
med	Character vector. Name/s of the mediator variable/s in ϕ .
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.
tol	Numeric. Smallest possible time interval to allow.

Details

See [TotalStd\(\)](#), [DirectStd\(\)](#), and [IndirectStd\(\)](#) 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 ("PosteriorMedStd").

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

Each element in the output list has the following elements:

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

thetahatstar Posterior distribution of the standardized 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](#)
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](#)
- 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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

Examples

```
set.seed(42)
phi <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.000, -0.511, 0.729,
    0.000, 0.000, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")
sigma <- matrix(
  data = c(
    0.24455556, 0.02201587, -0.05004762,
    0.02201587, 0.07067800, 0.01539456,
    -0.05004762, 0.01539456, 0.07553061
  ),
  nrow = 3
)
colnames(sigma) <- rownames(sigma) <- c("x", "m", "y")
input <- MCPhiSigma(
  phi = phi,
  sigma = sigma,
  vcov_theta = 0.001 * diag(15),
  R = 100L,
  seed = 42
)$output
phi <- lapply(
  X = input,
  FUN = function(x) {
    x[[1]]
  }
)
sigma <- lapply(
  X = input,
  FUN = function(x) {
```

```

        x[[2]]
    }
)

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

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

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

```

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, tol = 0.001)
```

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 <code>R</code> is a large value.
<code>tol</code>	Numeric. Smallest possible time interval to allow.

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 of length `length(delta_t)`.

Each element in the output list has the following elements:

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

thetahatstar Posterior distribution of the total effect centrality measure.

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](#)
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](#)
- 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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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)

```

PosteriorTotalCentralStd

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

Description

This function generates a posterior distribution of the standardized 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 and process noise covariance matrix Σ .

Usage

```
PosteriorTotalCentralStd(phi, sigma, delta_t, ncores = NULL, tol = 0.001)
```

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>sigma</code>	List of numeric matrices. Each element is a posterior draw of the diffusion covariance matrix.
<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 <code>R</code> is a large value.
<code>tol</code>	Numeric. Smallest possible time interval to allow.

Details

See [TotalCentralStd\(\)](#) 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 ("PosteriorTotalCentralStd").

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

Each element in the output list has the following elements:

est Mean of the posterior distribution of the standardized total effect centrality.

thetahatstar Posterior distribution of the standardized total effect centrality measure.

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](#)
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](#)
- 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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [TotalStd\(\)](#), [Trajectory\(\)](#)

Examples

```
set.seed(42)
phi <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.000, -0.511, 0.729,
    0.000, 0.000, -0.693
  ),
  nrow = 3
)
colnames(phi) <- rownames(phi) <- c("x", "m", "y")

sigma <- matrix(
  data = c(
    0.24455556, 0.02201587, -0.05004762,
    0.02201587, 0.07067800, 0.01539456,
    -0.05004762, 0.01539456, 0.07553061
  ),
  nrow = 3
)
colnames(sigma) <- rownames(sigma) <- c("x", "m", "y")

input <- MCPhiSigma(
  phi = phi,
  sigma = sigma,
  vcov_theta = 0.001 * diag(15),
  R = 100L,
  seed = 42
)$output

phi <- lapply(
  X = input,
  FUN = function(x) {
    x[[1]]
  }
)
```

```

sigma <- lapply(
  X = input,
  FUN = function(x) {
    x[[2]]
  }
)

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

# Range of time intervals -----
posterior <- PosteriorTotalCentralStd(
  phi = phi,
  sigma = sigma,
  delta_t = 1:5
)

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

```

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

Description

Print Method for Object of Class ctmedboot

Usage

```

## S3 method for class 'ctmedboot'
print(x, alpha = 0.05, digits = 4, type = "pc", ...)

```

Arguments

x	an object of class ctmedboot.
alpha	Numeric vector. Significance level α .
digits	Integer indicating the number of decimal places to display.

type	Character string. Confidence interval type, that is, type = "pc" for percentile; type = "bc" for bias corrected.
...	further arguments.

Value

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

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
## Not run:
library(bootStateSpace)
# prepare parameters
## number of individuals
n <- 50
## time points
time <- 100
delta_t <- 0.10
## dynamic structure
p <- 3
mu0 <- rep(x = 0, times = p)
sigma0 <- matrix(
  data = c(
    1.0,
    0.2,
    0.2,
    0.2,
    1.0,
    0.2,
    0.2,
    0.2,
    1.0
  ),
  nrow = p
)
sigma0_l <- t(chol(sigma0))
mu <- rep(x = 0, times = p)
phi <- matrix(
  data = c(
    -0.357,
    0.771,
    -0.450,
    0.0,
    -0.511,
    0.729,
    0,
    0,
    0
  )
```

```

      -0.693
    ),
    nrow = p
  )
  sigma <- matrix(
    data = c(
      0.24455556,
      0.02201587,
      -0.05004762,
      0.02201587,
      0.07067800,
      0.01539456,
      -0.05004762,
      0.01539456,
      0.07553061
    ),
    nrow = p
  )
  sigma_l <- t(chol(sigma))
  ## measurement model
  k <- 3
  nu <- rep(x = 0, times = k)
  lambda <- diag(k)
  theta <- 0.2 * diag(k)
  theta_l <- t(chol(theta))

  boot <- PBSSMOUFixed(
    R = 1000L,
    path = getwd(),
    prefix = "ou",
    n = n,
    time = time,
    delta_t = delta_t,
    mu0 = mu0,
    sigma0_l = sigma0_l,
    mu = mu,
    phi = phi,
    sigma_l = sigma_l,
    nu = nu,
    lambda = lambda,
    theta_l = theta_l,
    ncores = parallel::detectCores() - 1,
    seed = 42
  )
  phi_hat <- phi
  colnames(phi_hat) <- rownames(phi_hat) <- c("x", "m", "y")
  phi <- extract(object = boot, what = "phi")

  # Specific time interval -----
  boot <- BootMed(
    phi = phi,
    phi_hat = phi_hat,
    delta_t = 1,

```

```

    from = "x",
    to = "y",
    med = "m"
  )
  print(boot)
  print(boot, type = "bc") # bias-corrected

# Range of time intervals -----
boot <- BootMed(
  phi = phi,
  phi_hat = phi_hat,
  delta_t = 1:5,
  from = "x",
  to = "y",
  med = "m"
)
print(boot)
print(boot, type = "bc") # bias-corrected

## End(Not run)

```

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

Description

Print Method for Object of Class ctmeddelta

Usage

```

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

```

Arguments

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

Value

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

Print Method for Object of Class ctmedmc

Description

Print Method for Object of Class ctmedmc

Usage

```

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

```

Arguments

x	an object of class ctmedmc.
alpha	Numeric vector. Significance level 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, digits = 4, ...)

```

Arguments

`x` an object of class `ctmedtraj`.
`digits` Integer indicating the number of decimal places to display.
`...` 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.ctmedboot	<i>Summary Method for an Object of Class ctmedboot</i>
-------------------	--

Description

Summary Method for an Object of Class `ctmedboot`

Usage

```
## S3 method for class 'ctmedboot'
summary(object, alpha = 0.05, type = "pc", digits = 4, ...)
```

Arguments

object	Object of class ctmedboot.
alpha	Numeric vector. Significance level α .
type	Character string. Confidence interval type, that is, type = "pc" for percentile; type = "bc" for bias corrected.
digits	Integer indicating the number of decimal places to display.
...	additional arguments.

Value

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

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
## Not run:
library(bootStateSpace)
# prepare parameters
## number of individuals
n <- 50
## time points
time <- 100
delta_t <- 0.10
## dynamic structure
p <- 3
mu0 <- rep(x = 0, times = p)
sigma0 <- matrix(
  data = c(
    1.0,
    0.2,
    0.2,
    0.2,
    1.0,
    0.2,
    0.2,
    0.2,
    1.0
  ),
  nrow = p
)
sigma0_l <- t(chol(sigma0))
mu <- rep(x = 0, times = p)
phi <- matrix(
  data = c(
    -0.357,
    0.771,
```

```

      -0.450,
      0.0,
      -0.511,
      0.729,
      0,
      0,
      -0.693
    ),
    nrow = p
  )
  sigma <- matrix(
    data = c(
      0.24455556,
      0.02201587,
      -0.05004762,
      0.02201587,
      0.07067800,
      0.01539456,
      -0.05004762,
      0.01539456,
      0.07553061
    ),
    nrow = p
  )
  sigma_l <- t(chol(sigma))
  ## measurement model
  k <- 3
  nu <- rep(x = 0, times = k)
  lambda <- diag(k)
  theta <- 0.2 * diag(k)
  theta_l <- t(chol(theta))

  boot <- PBSSMOUFixed(
    R = 1000L,
    path = getwd(),
    prefix = "ou",
    n = n,
    time = time,
    delta_t = delta_t,
    mu0 = mu0,
    sigma0_l = sigma0_l,
    mu = mu,
    phi = phi,
    sigma_l = sigma_l,
    nu = nu,
    lambda = lambda,
    theta_l = theta_l,
    ncores = parallel::detectCores() - 1,
    seed = 42
  )
  phi_hat <- phi
  colnames(phi_hat) <- rownames(phi_hat) <- c("x", "m", "y")
  phi <- extract(object = boot, what = "phi")

```

```

# Specific time interval -----
boot <- BootMed(
  phi = phi,
  phi_hat = phi_hat,
  delta_t = 1,
  from = "x",
  to = "y",
  med = "m"
)
summary(boot)
summary(boot, type = "bc") # bias-corrected

# Range of time intervals -----
boot <- BootMed(
  phi = phi,
  phi_hat = phi_hat,
  delta_t = 1:5,
  from = "x",
  to = "y",
  med = "m"
)
summary(boot)
summary(boot, type = "bc") # bias-corrected

## End(Not run)

```

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, digits = 4, ...)

```

Arguments

object	Object of class ctmeddelta.
alpha	Numeric vector. Significance level α .
digits	Integer indicating the number of decimal places to display.
...	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

*Summary Method for an Object of Class ctmedmc***Description**

Summary Method for an Object of Class ctmedmc

Usage

```

## S3 method for class 'ctmedmc'
summary(object, alpha = 0.05, digits = 4, ...)

```

Arguments

object	Object of class ctmedmc.
alpha	Numeric vector. Significance level α .
digits	Integer indicating the number of decimal places to display.
...	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)

```

summary.ctmedmed

Summary Method for an Object of Class ctmedmed

Description

Summary Method for an Object of Class ctmedmed

Usage

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

Arguments

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

Value

Returns a matrix of effects.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```

phi <- matrix(
  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	<i>Summary Method for an Object of Class ctmedtraj</i>
-------------------	--

Description

Summary Method for an Object of Class ctmedtraj

Usage

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

Arguments

object	an object of class ctmedtraj.
digits	Integer indicating the number of decimal places to display.
...	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

*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 .

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

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

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

Value

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

call Function call.

args Function arguments.

fun Function used ("Total").

output The matrix of total effects.

Author(s)

Ivan Jacob Agaloos Pesigan

References

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

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

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](https://doi.org/10.1037/met0000779)

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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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, tol = 0.001)
```

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).
- tol Numeric. Smallest possible time interval to allow.

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)

Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](https://doi.org/10.1037/met0000779)

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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentralStd\(\)](#), [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)

```

TotalCentralStd

Standardized Total Effect Centrality

Description

Standardized Total Effect Centrality

Usage

```
TotalCentralStd(phi, sigma, delta_t, tol = 0.001)
```

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).
tol	Numeric. Smallest possible time interval to allow.

Details

The standardized total effect centrality of a variable is the sum of the standardized 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 ("TotalCentralStd").

output A matrix of standardized total effect centrality.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Continuous-Time Mediation Functions: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [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(
      0.24455556, 0.02201587, -0.05004762,
      0.02201587, 0.07067800, 0.01539456,
      -0.05004762, 0.01539456, 0.07553061
    ),
    nrow = 3
  )

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

  # Range of time intervals -----
  total_central_std <- TotalCentralStd(
    phi = phi,
    sigma = sigma,
    delta_t = 1:30
  )
  plot(total_central_std)

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

```

TotalStd

Standardized Total Effect Matrix Over a Specific Time Interval

Description

This function computes the standardized 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

$$\text{Total}_{\Delta t, i, j}^* = \text{Total}_{\Delta t, i, j} \left(\frac{\sigma_{x_j}}{\sigma_{y_i}} \right)$$

where Φ denotes the drift matrix, σ_{x_j} and σ_{y_i} are the steady-state model-implied standard deviations of the state independent and dependent variables, respectively, 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 standardized matrix of total effects.

Author(s)

Ivan Jacob Agaloos Pesigan

References

- Bollen, K. A. (1987). Total, direct, and indirect effects in structural equation models. *Sociological Methodology*, 17, 37. [doi:10.2307/271028](https://doi.org/10.2307/271028)
- Deboeck, P. R., & Preacher, K. J. (2015). No need to be discrete: A method for continuous time mediation analysis. *Structural Equation Modeling: A Multidisciplinary Journal*, 23 (1), 61-75. [doi:10.1080/10705511.2014.973960](https://doi.org/10.1080/10705511.2014.973960)
- Pesigan, I. J. A., Russell, M. A., & Chow, S.-M. (2025). Inferences and effect sizes for direct, indirect, and total effects in continuous-time mediation models. *Psychological Methods*. [doi:10.1037/met0000779](https://doi.org/10.1037/met0000779)
- 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: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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(
    0.24455556, 0.02201587, -0.05004762,
    0.02201587, 0.07067800, 0.01539456,
    -0.05004762, 0.01539456, 0.07553061
  ),
  nrow = 3
)
delta_t <- 1
TotalStd(
  phi = phi,
  sigma = sigma,
  delta_t = delta_t
)
```

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 list containing simulated trajectories for total, direct, and indirect effects.

See Also

Other Continuous-Time Mediation Functions: [BootBeta\(\)](#), [BootBetaStd\(\)](#), [BootDirectCentral\(\)](#), [BootDirectCentralStd\(\)](#), [BootIndirectCentral\(\)](#), [BootIndirectCentralStd\(\)](#), [BootMed\(\)](#), [BootMedStd\(\)](#), [BootTotalCentral\(\)](#), [BootTotalCentralStd\(\)](#), [DeltaBeta\(\)](#), [DeltaBetaStd\(\)](#), [DeltaDirectCentral\(\)](#), [DeltaDirectCentralStd\(\)](#), [DeltaIndirectCentral\(\)](#), [DeltaMed\(\)](#), [DeltaMedStd\(\)](#), [DeltaTotalCentral\(\)](#), [DeltaTotalCentralStd\(\)](#), [Direct\(\)](#), [DirectCentral\(\)](#), [DirectCentralStd\(\)](#), [DirectStd\(\)](#), [Indirect\(\)](#), [IndirectCentral\(\)](#), [IndirectCentralStd\(\)](#), [IndirectStd\(\)](#), [MCBeta\(\)](#), [MCBetaStd\(\)](#), [MCDirectCentral\(\)](#), [MCDirectCentralStd\(\)](#), [MCIndirectCentral\(\)](#), [MCIndirectCentralStd\(\)](#), [MCMed\(\)](#), [MCMedStd\(\)](#), [MCPhi\(\)](#), [MCPhiSigma\(\)](#), [MCTotalCentral\(\)](#), [MCTotalCentralStd\(\)](#), [Med\(\)](#), [MedStd\(\)](#), [PosteriorBeta\(\)](#), [PosteriorBetaStd\(\)](#), [PosteriorDirectCentral\(\)](#), [PosteriorDirectCentralStd\(\)](#), [PosteriorIndirectCentral\(\)](#), [PosteriorIndirectCentralStd\(\)](#), [PosteriorMed\(\)](#), [PosteriorMedStd\(\)](#), [PosteriorTotalCentral\(\)](#), [PosteriorTotalCentralStd\(\)](#), [Total\(\)](#), [TotalCentral\(\)](#), [TotalCentralStd\(\)](#), [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

BootBeta, [3](#)
BootBetaStd, [7](#)
BootDirectCentral, [11](#)
BootDirectCentralStd, [15](#)
BootIndirectCentral, [19](#)
BootIndirectCentralStd, [23](#)
BootMed, [27](#)
BootMedStd, [31](#)
BootTotalCentral, [35](#)
BootTotalCentralStd, [39](#)
DeltaBeta, [50](#)
DeltaBetaStd, [53](#)
DeltaDirectCentral, [58](#)
DeltaDirectCentralStd, [61](#)
DeltaIndirectCentral, [65](#)
DeltaMed, [73](#)
DeltaMedStd, [76](#)
DeltaTotalCentral, [81](#)
DeltaTotalCentralStd, [84](#)
Direct, [88](#)
DirectCentral, [91](#)
DirectCentralStd, [93](#)
DirectStd, [95](#)
Indirect, [97](#)
IndirectCentral, [100](#)
IndirectCentralStd, [102](#)
IndirectStd, [104](#)
MCBETA, [106](#)
MCBETAStd, [110](#)
MCDirectCentral, [114](#)
MCDirectCentralStd, [118](#)
MCIndirectCentral, [122](#)
MCIndirectCentralStd, [126](#)
MCMed, [130](#)
MCMedStd, [134](#)
MCPhi, [138](#)
MCPhiSigma, [140](#)
MCTotalCentral, [143](#)

MCTotalCentralStd, [146](#)
Med, [150](#)
MedStd, [153](#)
PosteriorBeta, [164](#)
PosteriorBetaStd, [167](#)
PosteriorDirectCentral, [170](#)
PosteriorDirectCentralStd, [173](#)
PosteriorIndirectCentral, [176](#)
PosteriorIndirectCentralStd, [179](#)
PosteriorMed, [182](#)
PosteriorMedStd, [185](#)
PosteriorTotalCentral, [188](#)
PosteriorTotalCentralStd, [191](#)
Total, [215](#)
TotalCentral, [217](#)
TotalCentralStd, [219](#)
TotalStd, [221](#)
Trajectory, [223](#)

* beta

DeltaBeta, [50](#)
DeltaBetaStd, [53](#)
MCBETA, [106](#)
MCBETAStd, [110](#)
PosteriorBeta, [164](#)
PosteriorBetaStd, [167](#)

* boot

BootBeta, [3](#)
BootBetaStd, [7](#)
BootDirectCentral, [11](#)
BootDirectCentralStd, [15](#)
BootIndirectCentral, [19](#)
BootIndirectCentralStd, [23](#)
BootMed, [27](#)
BootMedStd, [31](#)
BootTotalCentral, [35](#)
BootTotalCentralStd, [39](#)

* cTMed

BootBeta, [3](#)
BootBetaStd, [7](#)

BootDirectCentral, 11
 BootDirectCentralStd, 15
 BootIndirectCentral, 19
 BootIndirectCentralStd, 23
 BootMed, 27
 BootMedStd, 31
 BootTotalCentral, 35
 BootTotalCentralStd, 39
 DeltaBeta, 50
 DeltaBetaStd, 53
 DeltaDirectCentral, 58
 DeltaDirectCentralStd, 61
 DeltaIndirectCentral, 65
 DeltaIndirectCentralStd, 69
 DeltaMed, 73
 DeltaMedStd, 76
 DeltaTotalCentral, 81
 DeltaTotalCentralStd, 84
 Direct, 88
 DirectCentral, 91
 DirectCentralStd, 93
 DirectStd, 95
 Indirect, 97
 IndirectCentral, 100
 IndirectCentralStd, 102
 IndirectStd, 104
 MCBeta, 106
 MCBetaStd, 110
 MCDirectCentral, 114
 MCDirectCentralStd, 118
 MCIndirectCentral, 122
 MCIndirectCentralStd, 126
 MCMed, 130
 MCMedStd, 134
 MCPhi, 138
 MCPhiSigma, 140
 MCTotalCentral, 143
 MCTotalCentralStd, 146
 Med, 150
 MedStd, 153
 PosteriorBeta, 164
 PosteriorBetaStd, 167
 PosteriorDirectCentral, 170
 PosteriorDirectCentralStd, 173
 PosteriorIndirectCentral, 176
 PosteriorIndirectCentralStd, 179
 PosteriorMed, 182
 PosteriorMedStd, 185

PosteriorTotalCentral, 188
 PosteriorTotalCentralStd, 191
 Total, 215
 TotalCentral, 217
 TotalCentralStd, 219
 TotalStd, 221
 Trajectory, 223
*** delta**
 DeltaBeta, 50
 DeltaBetaStd, 53
 DeltaDirectCentral, 58
 DeltaDirectCentralStd, 61
 DeltaIndirectCentral, 65
 DeltaIndirectCentralStd, 69
 DeltaMed, 73
 DeltaMedStd, 76
 DeltaTotalCentral, 81
 DeltaTotalCentralStd, 84
*** effects**
 Direct, 88
 DirectCentral, 91
 DirectCentralStd, 93
 DirectStd, 95
 Indirect, 97
 IndirectCentral, 100
 IndirectCentralStd, 102
 IndirectStd, 104
 Med, 150
 MedStd, 153
 Total, 215
 TotalCentral, 217
 TotalCentralStd, 219
 TotalStd, 221
 Trajectory, 223
*** mc**
 MCBeta, 106
 MCBetaStd, 110
 MCDirectCentral, 114
 MCDirectCentralStd, 118
 MCIndirectCentral, 122
 MCIndirectCentralStd, 126
 MCMed, 130
 MCMedStd, 134
 MCPhi, 138
 MCPhiSigma, 140
 MCTotalCentral, 143
 MCTotalCentralStd, 146
*** methods**

- confint.ctmedboot, 43
- confint.ctmeddelta, 46
- confint.ctmedmc, 48
- plot.ctmedboot, 156
- plot.ctmeddelta, 158
- plot.ctmedmc, 160
- plot.ctmedmed, 162
- plot.ctmedtraj, 163
- print.ctmedboot, 194
- print.ctmeddelta, 197
- print.ctmedeffect, 199
- print.ctmedmc, 200
- print.ctmedmcphi, 202
- print.ctmedmed, 203
- print.ctmedtraj, 204
- summary.ctmedboot, 205
- summary.ctmeddelta, 208
- summary.ctmedmc, 210
- summary.ctmedmed, 212
- summary.ctmedposteriorphi, 213
- summary.ctmedtraj, 214
- * network**
 - BootDirectCentral, 11
 - BootDirectCentralStd, 15
 - BootIndirectCentral, 19
 - BootIndirectCentralStd, 23
 - BootTotalCentral, 35
 - BootTotalCentralStd, 39
 - DeltaDirectCentral, 58
 - DeltaDirectCentralStd, 61
 - DeltaIndirectCentral, 65
 - DeltaIndirectCentralStd, 69
 - DeltaTotalCentral, 81
 - DeltaTotalCentralStd, 84
 - DirectCentral, 91
 - DirectCentralStd, 93
 - IndirectCentral, 100
 - IndirectCentralStd, 102
 - MCDirectCentral, 114
 - MCDirectCentralStd, 118
 - MCIndirectCentral, 122
 - MCIndirectCentralStd, 126
 - MCTotalCentral, 143
 - MCTotalCentralStd, 146
 - PosteriorDirectCentral, 170
 - PosteriorDirectCentralStd, 173
 - PosteriorIndirectCentral, 176
 - PosteriorIndirectCentralStd, 179
 - PosteriorTotalCentral, 188
 - PosteriorTotalCentralStd, 191
 - TotalCentral, 217
 - TotalCentralStd, 219
- * path**
 - BootBeta, 3
 - BootBetaStd, 7
 - BootMed, 27
 - BootMedStd, 31
 - DeltaMed, 73
 - DeltaMedStd, 76
 - MCMed, 130
 - MCMedStd, 134
 - Med, 150
 - MedStd, 153
 - PosteriorMed, 182
 - PosteriorMedStd, 185
 - Trajectory, 223
- * posterior**
 - PosteriorBeta, 164
 - PosteriorBetaStd, 167
 - PosteriorDirectCentral, 170
 - PosteriorDirectCentralStd, 173
 - PosteriorIndirectCentral, 176
 - PosteriorIndirectCentralStd, 179
 - PosteriorMed, 182
 - PosteriorMedStd, 185
 - PosteriorTotalCentral, 188
 - PosteriorTotalCentralStd, 191
- BootBeta, 3
- BootBeta(), 9, 13, 16, 20, 24, 28, 32, 37, 41, 52, 56, 60, 63, 67, 75, 79, 83, 86, 90, 92, 94, 97, 99, 101, 103, 106, 108, 112, 116, 120, 124, 128, 132, 136, 139, 142, 145, 148, 152, 155, 165, 168, 171, 174, 177, 180, 183, 187, 190, 193, 216, 218, 220, 223, 224
- BootBetaStd, 7
- BootBetaStd(), 5, 13, 16, 20, 24, 28, 32, 37, 41, 52, 56, 60, 63, 67, 75, 79, 83, 86, 90, 92, 94, 97, 99, 101, 103, 106, 108, 112, 116, 120, 124, 128, 132, 136, 139, 142, 145, 148, 152, 155, 165, 168, 171, 174, 177, 180, 183, 187, 190, 193, 216, 218, 220, 223, 224
- BootDirectCentral, 11

- [BootDirectCentral\(\)](#), [5](#), [9](#), [16](#), [20](#), [24](#), [28](#),
[32](#), [37](#), [41](#), [52](#), [56](#), [60](#), [63](#), [67](#), [75](#), [79](#),
[83](#), [86](#), [90](#), [92](#), [94](#), [97](#), [99](#), [101](#), [103](#),
[106](#), [108](#), [112](#), [116](#), [120](#), [124](#), [128](#),
[132](#), [136](#), [139](#), [142](#), [145](#), [148](#), [152](#),
[155](#), [165](#), [168](#), [171](#), [174](#), [177](#), [180](#),
[183](#), [187](#), [190](#), [193](#), [216](#), [218](#), [220](#),
[223](#), [224](#)
- [BootDirectCentralStd](#), [15](#)
- [BootDirectCentralStd\(\)](#), [5](#), [9](#), [13](#), [20](#), [24](#),
[28](#), [32](#), [37](#), [41](#), [52](#), [56](#), [60](#), [63](#), [67](#), [75](#),
[79](#), [83](#), [86](#), [90](#), [92](#), [94](#), [97](#), [99](#), [101](#),
[103](#), [106](#), [108](#), [112](#), [116](#), [120](#), [124](#),
[128](#), [132](#), [136](#), [139](#), [142](#), [145](#), [148](#),
[152](#), [155](#), [165](#), [168](#), [171](#), [174](#), [177](#),
[180](#), [183](#), [187](#), [190](#), [193](#), [216](#), [218](#),
[220](#), [223](#), [224](#)
- [BootIndirectCentral](#), [19](#)
- [BootIndirectCentral\(\)](#), [5](#), [9](#), [13](#), [16](#), [24](#), [28](#),
[32](#), [37](#), [41](#), [52](#), [56](#), [60](#), [63](#), [67](#), [75](#), [79](#),
[83](#), [86](#), [90](#), [92](#), [94](#), [97](#), [99](#), [101](#), [103](#),
[106](#), [108](#), [112](#), [116](#), [120](#), [124](#), [128](#),
[132](#), [136](#), [139](#), [142](#), [145](#), [148](#), [152](#),
[155](#), [165](#), [168](#), [171](#), [174](#), [177](#), [180](#),
[183](#), [187](#), [190](#), [193](#), [216](#), [218](#), [220](#),
[223](#), [224](#)
- [BootIndirectCentralStd](#), [23](#)
- [BootIndirectCentralStd\(\)](#), [5](#), [9](#), [13](#), [16](#), [20](#),
[28](#), [32](#), [37](#), [41](#), [52](#), [56](#), [60](#), [63](#), [67](#), [75](#),
[79](#), [83](#), [86](#), [90](#), [92](#), [94](#), [97](#), [99](#), [101](#),
[103](#), [106](#), [108](#), [112](#), [116](#), [120](#), [124](#),
[128](#), [132](#), [136](#), [139](#), [142](#), [145](#), [148](#),
[152](#), [155](#), [165](#), [168](#), [171](#), [174](#), [177](#),
[180](#), [183](#), [187](#), [190](#), [193](#), [216](#), [218](#),
[220](#), [223](#), [224](#)
- [BootMed](#), [27](#)
- [BootMed\(\)](#), [5](#), [9](#), [13](#), [16](#), [20](#), [24](#), [32](#), [37](#), [41](#), [52](#),
[56](#), [60](#), [63](#), [67](#), [75](#), [79](#), [83](#), [86](#), [90](#), [92](#),
[94](#), [97](#), [99](#), [101](#), [103](#), [106](#), [108](#), [112](#),
[116](#), [120](#), [124](#), [128](#), [132](#), [136](#), [139](#),
[142](#), [145](#), [148](#), [152](#), [155](#), [165](#), [168](#),
[171](#), [174](#), [177](#), [180](#), [183](#), [187](#), [190](#),
[193](#), [216](#), [218](#), [220](#), [223](#), [224](#)
- [BootMedStd](#), [31](#)
- [BootMedStd\(\)](#), [5](#), [9](#), [13](#), [16](#), [20](#), [24](#), [28](#), [37](#), [41](#),
[52](#), [56](#), [60](#), [63](#), [67](#), [75](#), [79](#), [83](#), [86](#), [90](#),
[92](#), [94](#), [97](#), [99](#), [101](#), [103](#), [106](#), [108](#),
[112](#), [116](#), [120](#), [124](#), [128](#), [132](#), [136](#),
[139](#), [142](#), [145](#), [148](#), [152](#), [155](#), [165](#),
[168](#), [171](#), [174](#), [177](#), [180](#), [183](#), [187](#),
[190](#), [193](#), [216](#), [218](#), [220](#), [223](#), [224](#)
- [BootTotalCentral](#), [35](#)
- [BootTotalCentral\(\)](#), [5](#), [9](#), [13](#), [16](#), [20](#), [24](#), [28](#),
[32](#), [41](#), [52](#), [56](#), [60](#), [63](#), [67](#), [75](#), [79](#), [83](#),
[86](#), [90](#), [92](#), [94](#), [97](#), [99](#), [101](#), [103](#), [106](#),
[108](#), [112](#), [116](#), [120](#), [124](#), [128](#), [132](#),
[136](#), [139](#), [142](#), [145](#), [148](#), [152](#), [155](#),
[165](#), [168](#), [171](#), [174](#), [177](#), [180](#), [183](#),
[187](#), [190](#), [193](#), [216](#), [218](#), [220](#), [223](#),
[224](#)
- [BootTotalCentralStd](#), [39](#)
- [BootTotalCentralStd\(\)](#), [5](#), [9](#), [13](#), [16](#), [20](#), [24](#),
[28](#), [32](#), [37](#), [52](#), [56](#), [60](#), [63](#), [67](#), [75](#), [79](#),
[83](#), [86](#), [90](#), [92](#), [94](#), [97](#), [99](#), [101](#), [103](#),
[106](#), [108](#), [112](#), [116](#), [120](#), [124](#), [128](#),
[132](#), [136](#), [139](#), [142](#), [145](#), [148](#), [152](#),
[155](#), [165](#), [168](#), [171](#), [174](#), [177](#), [180](#),
[183](#), [187](#), [190](#), [193](#), [216](#), [218](#), [220](#),
[223](#), [224](#)
- [confint.ctmedboot](#), [43](#)
- [confint.ctmeddelta](#), [46](#)
- [confint.ctmedmc](#), [48](#)
- [DeltaBeta](#), [50](#)
- [DeltaBeta\(\)](#), [5](#), [9](#), [13](#), [16](#), [20](#), [24](#), [28](#), [32](#), [37](#),
[41](#), [56](#), [60](#), [63](#), [67](#), [75](#), [79](#), [83](#), [86](#), [90](#),
[92](#), [94](#), [97](#), [99](#), [101](#), [103](#), [106](#), [108](#),
[112](#), [116](#), [120](#), [124](#), [128](#), [132](#), [136](#),
[139](#), [142](#), [145](#), [148](#), [152](#), [155](#), [165](#),
[168](#), [171](#), [174](#), [177](#), [180](#), [183](#), [187](#),
[190](#), [193](#), [216](#), [218](#), [220](#), [223](#), [224](#)
- [DeltaBetaStd](#), [53](#)
- [DeltaBetaStd\(\)](#), [5](#), [9](#), [13](#), [16](#), [20](#), [24](#), [28](#), [32](#),
[37](#), [41](#), [52](#), [60](#), [63](#), [67](#), [75](#), [79](#), [83](#), [86](#),
[90](#), [92](#), [94](#), [97](#), [99](#), [101](#), [103](#), [106](#),
[108](#), [112](#), [116](#), [120](#), [124](#), [128](#), [132](#),
[136](#), [139](#), [142](#), [145](#), [148](#), [152](#), [155](#),
[165](#), [168](#), [171](#), [174](#), [177](#), [180](#), [183](#),
[187](#), [190](#), [193](#), [216](#), [218](#), [220](#), [223](#),
[224](#)
- [DeltaDirectCentral](#), [58](#)
- [DeltaDirectCentral\(\)](#), [5](#), [9](#), [13](#), [16](#), [20](#), [24](#),
[28](#), [32](#), [37](#), [41](#), [52](#), [56](#), [63](#), [67](#), [75](#), [79](#),
[83](#), [86](#), [90](#), [92](#), [94](#), [97](#), [99](#), [101](#), [103](#),
[106](#), [108](#), [112](#), [116](#), [120](#), [124](#), [128](#),
[132](#), [136](#), [139](#), [142](#), [145](#), [148](#), [152](#),

- 155, 165, 168, 171, 174, 177, 180,
183, 187, 190, 193, 216, 218, 220,
223, 224
- DeltaDirectCentralStd, 61
- DeltaDirectCentralStd(), 5, 9, 13, 16, 20,
24, 28, 32, 37, 41, 52, 56, 60, 67, 75,
79, 83, 86, 90, 92, 94, 97, 99, 101,
103, 106, 108, 112, 116, 120, 124,
128, 132, 136, 139, 142, 145, 148,
152, 155, 165, 168, 171, 174, 177,
180, 183, 187, 190, 193, 216, 218,
220, 223, 224
- DeltaIndirectCentral, 65
- DeltaIndirectCentral(), 5, 9, 13, 16, 20,
24, 28, 32, 37, 41, 52, 56, 60, 63, 75,
79, 83, 86, 90, 92, 94, 97, 99, 101,
103, 106, 108, 112, 116, 120, 124,
128, 132, 136, 139, 142, 145, 148,
152, 155, 165, 168, 171, 174, 177,
180, 183, 187, 190, 193, 216, 218,
220, 223, 224
- DeltaIndirectCentralStd, 69
- DeltaMed, 73
- DeltaMed(), 5, 9, 13, 16, 20, 24, 28, 32, 37,
41, 52, 56, 60, 63, 67, 79, 83, 86, 90,
92, 94, 97, 99, 101, 103, 106, 108,
112, 116, 120, 124, 128, 132, 136,
139, 142, 145, 148, 152, 155, 165,
168, 171, 174, 177, 180, 183, 187,
190, 193, 216, 218, 220, 223, 224
- DeltaMedStd, 76
- DeltaMedStd(), 5, 9, 13, 16, 20, 24, 28, 32,
37, 41, 52, 56, 60, 63, 67, 75, 83, 86,
90, 92, 94, 97, 99, 101, 103, 106,
108, 112, 116, 120, 124, 128, 132,
136, 139, 142, 145, 148, 152, 155,
165, 168, 171, 174, 177, 180, 183,
187, 190, 193, 216, 218, 220, 223,
224
- DeltaTotalCentral, 81
- DeltaTotalCentral(), 5, 9, 13, 16, 20, 24,
28, 32, 37, 41, 52, 56, 60, 63, 67, 75,
79, 86, 90, 92, 94, 97, 99, 101, 103,
106, 108, 112, 116, 120, 124, 128,
132, 136, 139, 142, 145, 148, 152,
155, 165, 168, 171, 174, 177, 180,
183, 187, 190, 193, 216, 218, 220,
223, 224
- DeltaTotalCentralStd, 84
- DeltaTotalCentralStd(), 5, 9, 13, 16, 20,
24, 28, 32, 37, 41, 52, 56, 60, 63, 67,
75, 79, 83, 90, 92, 94, 97, 99, 101,
103, 106, 108, 112, 116, 120, 124,
128, 132, 136, 139, 142, 145, 148,
152, 155, 165, 168, 171, 174, 177,
180, 183, 187, 190, 193, 216, 218,
220, 223, 224
- Direct, 88
- Direct(), 5, 9, 13, 16, 20, 24, 27, 28, 32, 37,
41, 52, 56, 60, 63, 67, 73, 75, 79, 83,
86, 92, 94, 97, 99, 101, 103, 106,
108, 112, 116, 120, 124, 128, 131,
132, 136, 139, 142, 145, 148, 151,
152, 155, 165, 168, 171, 174, 177,
180, 182, 183, 187, 190, 193, 216,
218, 220, 223, 224
- DirectCentral, 91
- DirectCentral(), 5, 9, 12, 13, 16, 20, 24, 28,
32, 37, 41, 52, 56, 58, 60, 63, 67, 75,
79, 83, 86, 90, 94, 97, 99, 101, 103,
106, 108, 112, 115, 116, 120, 124,
128, 132, 136, 139, 142, 145, 148,
152, 155, 165, 168, 170, 171, 174,
177, 180, 183, 187, 190, 193, 216,
218, 220, 223, 224
- DirectCentralStd, 93
- DirectCentralStd(), 5, 9, 13, 16, 20, 24, 28,
32, 37, 41, 52, 56, 60, 62, 63, 67, 75,
79, 83, 86, 90, 92, 97, 99, 101, 103,
106, 108, 112, 116, 119, 120, 124,
128, 132, 136, 139, 142, 145, 148,
152, 155, 165, 168, 171, 174, 177,
180, 183, 187, 190, 193, 216, 218,
220, 223, 224
- DirectStd, 95
- DirectStd(), 5, 9, 13, 16, 20, 24, 28, 32, 37,
41, 52, 56, 60, 63, 67, 75, 77, 79, 83,
86, 90, 92, 94, 99, 101, 103, 106,
108, 112, 116, 120, 124, 128, 132,
135, 136, 139, 142, 145, 148, 152,
154, 155, 165, 168, 171, 174, 177,
180, 183, 186, 187, 190, 193, 216,
218, 220, 223, 224
- Indirect, 97
- Indirect(), 5, 9, 13, 16, 20, 24, 27, 28, 32,
37, 41, 52, 56, 60, 63, 67, 73, 75, 79,

- 83, 86, 90, 92, 94, 97, 101, 103, 106,
108, 112, 116, 120, 124, 128, 131,
132, 136, 139, 142, 145, 148, 151,
152, 155, 165, 168, 171, 174, 177,
180, 182, 183, 187, 190, 193, 216,
218, 220, 223, 224
- IndirectCentral, 100
- IndirectCentral(), 5, 9, 13, 16, 20, 24, 28,
32, 37, 41, 52, 56, 60, 63, 66, 67, 75,
79, 83, 86, 90, 92, 94, 97, 99, 103,
106, 108, 112, 116, 120, 123, 124,
128, 132, 136, 139, 142, 145, 148,
152, 155, 165, 168, 171, 174, 176,
177, 180, 183, 187, 190, 193, 216,
218, 220, 223, 224
- IndirectCentralStd, 102
- IndirectCentralStd(), 5, 9, 13, 16, 20, 24,
28, 32, 37, 41, 52, 56, 60, 63, 67, 70,
75, 79, 83, 86, 90, 92, 94, 97, 99,
101, 106, 108, 112, 116, 120, 124,
127, 128, 132, 136, 139, 142, 145,
148, 152, 155, 165, 168, 171, 174,
177, 179, 180, 183, 187, 190, 193,
216, 218, 220, 223, 224
- IndirectStd, 104
- IndirectStd(), 5, 9, 13, 16, 20, 24, 28, 32,
37, 41, 52, 56, 60, 63, 67, 75, 77, 79,
83, 86, 90, 92, 94, 97, 99, 101, 103,
108, 112, 116, 120, 124, 128, 132,
135, 136, 139, 142, 145, 148, 152,
154, 155, 165, 168, 171, 174, 177,
180, 183, 186, 187, 190, 193, 216,
218, 220, 223, 224
- MCBeta, 106
- MCBeta(), 5, 9, 13, 16, 20, 24, 28, 32, 37, 41,
52, 56, 60, 63, 67, 75, 79, 83, 86, 90,
92, 94, 97, 99, 101, 103, 106, 112,
116, 120, 124, 128, 132, 136, 139,
142, 145, 148, 152, 155, 165, 168,
171, 174, 177, 180, 183, 187, 190,
193, 216, 218, 220, 223, 224
- MCBetaStd, 110
- MCBetaStd(), 5, 9, 13, 16, 20, 24, 28, 32, 37,
41, 52, 56, 60, 63, 67, 75, 79, 83, 86,
90, 92, 94, 97, 99, 101, 103, 106,
108, 116, 120, 124, 128, 132, 136,
139, 142, 145, 148, 152, 155, 165,
168, 171, 174, 177, 180, 183, 187,
190, 193, 216, 218, 220, 223, 224
- MCDirectCentral, 114
- MCDirectCentral(), 5, 9, 13, 16, 20, 24, 28,
33, 37, 41, 52, 56, 60, 63, 67, 75, 79,
83, 86, 90, 92, 94, 97, 99, 101, 103,
106, 108, 112, 120, 124, 128, 132,
136, 139, 142, 145, 148, 152, 155,
165, 168, 171, 174, 177, 180, 183,
187, 190, 193, 216, 218, 220, 223,
224
- MCDirectCentralStd, 118
- MCDirectCentralStd(), 5, 9, 13, 16, 20, 24,
28, 33, 37, 41, 52, 56, 60, 63, 67, 75,
79, 83, 86, 90, 92, 94, 97, 99, 101,
103, 106, 108, 112, 116, 124, 128,
132, 136, 139, 142, 145, 148, 152,
155, 165, 168, 171, 174, 177, 180,
183, 187, 190, 193, 216, 218, 220,
223, 224
- MCIndirectCentral, 122
- MCIndirectCentral(), 5, 9, 13, 16, 20, 24,
28, 33, 37, 41, 52, 56, 60, 63, 67, 75,
79, 83, 86, 90, 92, 94, 97, 99, 101,
103, 106, 108, 112, 116, 120, 128,
132, 136, 139, 142, 145, 148, 152,
155, 165, 168, 171, 174, 177, 180,
183, 187, 190, 193, 216, 218, 220,
223, 224
- MCIndirectCentralStd, 126
- MCIndirectCentralStd(), 5, 9, 13, 16, 20,
24, 28, 33, 37, 41, 52, 56, 60, 63, 67,
75, 79, 83, 86, 90, 92, 94, 97, 99,
101, 103, 106, 108, 112, 116, 120,
124, 132, 136, 139, 142, 145, 148,
152, 155, 165, 168, 171, 174, 177,
180, 183, 187, 190, 193, 216, 218,
220, 223, 224
- MCMed, 130
- MCMed(), 5, 9, 13, 16, 20, 24, 28, 33, 37, 41,
52, 56, 60, 63, 67, 75, 79, 83, 86, 90,
92, 94, 97, 99, 101, 103, 106, 108,
112, 116, 120, 124, 128, 136, 139,
142, 145, 148, 152, 155, 165, 168,
171, 174, 177, 180, 183, 187, 190,
193, 216, 218, 220, 223, 224
- MCMedStd, 134
- MCMedStd(), 5, 9, 13, 16, 20, 24, 28, 33, 37,
41, 52, 56, 60, 63, 67, 75, 79, 83, 86,

- 90, 92, 94, 97, 99, 101, 103, 106, 108, 112, 116, 120, 124, 128, 132, 139, 142, 145, 148, 152, 155, 165, 168, 171, 174, 177, 180, 183, 187, 190, 193, 216, 218, 220, 223, 224
- MCP ϕ i, 138
- MCP ϕ i(), 5, 9, 13, 16, 20, 24, 28, 33, 37, 41, 52, 56, 60, 63, 67, 75, 79, 83, 86, 90, 92, 94, 97, 99, 101, 103, 106, 108, 112, 116, 120, 124, 128, 132, 136, 142, 145, 148, 152, 155, 165, 168, 171, 174, 177, 180, 183, 187, 190, 193, 216, 218, 220, 223, 224
- MCP ϕ iSigma, 140
- MCP ϕ iSigma(), 5, 9, 13, 16, 20, 24, 28, 33, 37, 41, 52, 56, 60, 63, 67, 75, 79, 83, 86, 90, 92, 94, 97, 99, 101, 103, 106, 108, 112, 116, 120, 124, 128, 132, 136, 139, 145, 148, 152, 155, 165, 168, 171, 174, 177, 180, 183, 187, 190, 193, 216, 218, 220, 223, 224
- MCTotalCentral, 143
- MCTotalCentral(), 5, 9, 13, 16, 20, 24, 28, 33, 37, 41, 52, 56, 60, 63, 67, 75, 79, 83, 86, 90, 92, 94, 97, 99, 101, 103, 106, 108, 112, 116, 120, 124, 128, 132, 136, 139, 142, 148, 152, 155, 165, 168, 171, 174, 177, 180, 183, 187, 190, 193, 216, 218, 220, 223, 224
- MCTotalCentralStd, 146
- MCTotalCentralStd(), 5, 9, 13, 16, 20, 24, 28, 33, 37, 41, 52, 56, 60, 63, 67, 75, 79, 83, 86, 90, 92, 94, 97, 99, 101, 103, 106, 108, 112, 116, 120, 124, 128, 132, 136, 139, 142, 145, 152, 155, 165, 168, 171, 174, 177, 180, 183, 187, 190, 193, 216, 218, 220, 223, 224
- Med, 150
- Med(), 5, 9, 13, 16, 20, 24, 28, 33, 37, 41, 52, 56, 60, 63, 67, 75, 79, 83, 86, 90, 92, 94, 97, 99, 101, 103, 106, 108, 112, 116, 120, 124, 128, 132, 136, 139, 142, 145, 148, 155, 165, 168, 171, 174, 177, 180, 183, 187, 190, 193, 216, 218, 220, 223, 224
- MedStd, 153
- MedStd(), 5, 9, 13, 16, 20, 24, 28, 33, 37, 41, 52, 56, 60, 63, 67, 75, 79, 83, 86, 90, 92, 94, 97, 99, 101, 103, 106, 108, 112, 116, 120, 124, 128, 132, 136, 139, 142, 145, 148, 152, 165, 168, 171, 174, 177, 180, 183, 187, 190, 193, 216, 218, 220, 223, 224
- plot.ctmedboot, 156
- plot.ctmeddelta, 158
- plot.ctmedmc, 160
- plot.ctmedmed, 162
- plot.ctmedtraj, 163
- PosteriorBeta, 164
- PosteriorBeta(), 5, 9, 13, 16, 20, 24, 28, 33, 37, 41, 52, 56, 60, 63, 67, 75, 79, 83, 86, 90, 92, 94, 97, 99, 101, 103, 106, 108, 112, 116, 120, 124, 128, 132, 136, 139, 142, 145, 148, 152, 155, 168, 171, 174, 177, 180, 183, 187, 190, 193, 216, 218, 220, 223, 224
- PosteriorBetaStd, 167
- PosteriorBetaStd(), 5, 9, 13, 16, 20, 24, 28, 33, 37, 41, 52, 56, 60, 63, 67, 75, 79, 83, 86, 90, 92, 94, 97, 99, 101, 103, 106, 108, 112, 116, 120, 124, 128, 132, 136, 139, 142, 145, 148, 152, 155, 165, 171, 174, 177, 180, 183, 187, 190, 193, 216, 218, 220, 223, 224
- PosteriorDirectCentral, 170
- PosteriorDirectCentral(), 5, 9, 13, 17, 20, 24, 28, 33, 37, 41, 52, 56, 60, 63, 67, 75, 79, 83, 86, 90, 92, 94, 97, 99, 101, 103, 106, 108, 112, 116, 120, 124, 128, 132, 136, 139, 142, 145, 148, 152, 155, 165, 168, 174, 177, 180, 183, 187, 190, 193, 216, 218, 220, 223, 224
- PosteriorDirectCentralStd, 173
- PosteriorDirectCentralStd(), 5, 9, 13, 17, 20, 24, 28, 33, 37, 41, 52, 56, 60, 63, 67, 75, 79, 83, 86, 90, 92, 94, 97, 99, 101, 103, 106, 109, 112, 116, 120, 124, 128, 132, 136, 139, 142, 145, 148, 152, 155, 165, 168, 171, 177, 180, 183, 187, 190, 193, 216, 218, 220, 223, 224
- PosteriorIndirectCentral, 176

- PosteriorIndirectCentral(), 5, 9, 13, 17,
 20, 24, 28, 33, 37, 41, 52, 56, 60, 63,
 67, 75, 79, 83, 86, 90, 92, 94, 97, 99,
 101, 103, 106, 109, 112, 116, 120,
 124, 128, 132, 136, 139, 142, 145,
 148, 152, 155, 165, 168, 171, 174,
 180, 183, 187, 190, 193, 216, 218,
 220, 223, 224
- PosteriorIndirectCentralStd, 179
- PosteriorIndirectCentralStd(), 5, 9, 13,
 17, 20, 24, 28, 33, 37, 41, 52, 56, 60,
 63, 67, 75, 79, 83, 86, 90, 92, 94, 97,
 99, 101, 103, 106, 109, 112, 116,
 120, 124, 128, 132, 136, 139, 142,
 145, 148, 152, 155, 165, 168, 171,
 174, 177, 183, 187, 190, 193, 216,
 218, 220, 223, 224
- PosteriorMed, 182
- PosteriorMed(), 5, 9, 13, 17, 20, 24, 28, 33,
 37, 41, 52, 56, 60, 63, 67, 75, 79, 83,
 86, 90, 92, 94, 97, 99, 101, 103, 106,
 109, 112, 116, 120, 124, 128, 132,
 136, 139, 142, 145, 148, 152, 155,
 165, 168, 171, 174, 177, 180, 187,
 190, 193, 216, 218, 220, 223, 224
- PosteriorMedStd, 185
- PosteriorMedStd(), 5, 9, 13, 17, 20, 24, 28,
 33, 37, 41, 52, 56, 60, 63, 67, 75, 79,
 83, 86, 90, 92, 94, 97, 99, 101, 103,
 106, 109, 112, 116, 120, 124, 128,
 132, 136, 139, 142, 145, 148, 152,
 155, 165, 168, 171, 174, 177, 180,
 183, 190, 193, 216, 218, 220, 223,
 224
- PosteriorTotalCentral, 188
- PosteriorTotalCentral(), 5, 9, 13, 17, 20,
 24, 28, 33, 37, 41, 52, 56, 60, 63, 67,
 75, 79, 83, 86, 90, 92, 94, 97, 99,
 101, 103, 106, 109, 112, 116, 120,
 124, 128, 132, 136, 139, 142, 145,
 148, 152, 155, 165, 169, 171, 174,
 177, 180, 183, 187, 193, 216, 218,
 220, 223, 224
- PosteriorTotalCentralStd, 191
- PosteriorTotalCentralStd(), 5, 9, 13, 17,
 20, 24, 28, 33, 37, 41, 52, 56, 60, 63,
 67, 75, 79, 83, 86, 90, 92, 94, 97, 99,
 101, 103, 106, 109, 112, 116, 120,
 124, 128, 132, 136, 139, 142, 145,
 148, 152, 155, 165, 169, 171, 174,
 177, 180, 183, 187, 190, 192,
 193, 216, 218, 223, 224
- print.ctmedboot, 194
- print.ctmeddelta, 197
- print.ctmedeffect, 199
- print.ctmedmc, 200
- print.ctmedmcphi, 202
- print.ctmedmed, 203
- print.ctmedtraj, 204
- summary.ctmedboot, 205
- summary.ctmeddelta, 208
- summary.ctmedmc, 210
- summary.ctmedmed, 212
- summary.ctmedposteriorphi, 213
- summary.ctmedtraj, 214
- Total, 215
- Total(), 4, 5, 9, 13, 17, 20, 24, 27, 28, 33, 37,
 41, 51, 52, 56, 60, 63, 67, 73, 75, 79,
 83, 86, 90, 92, 94, 97, 99, 101, 103,
 106, 107, 109, 112, 116, 120, 124,
 128, 131, 132, 136, 139, 142, 145,
 148, 151, 152, 155, 164, 165, 169,
 171, 174, 177, 180, 182, 183, 187,
 190, 193, 218, 220, 223, 224
- TotalCentral, 217
- TotalCentral(), 5, 9, 13, 17, 20, 24, 28, 33,
 36, 37, 41, 52, 56, 60, 63, 67, 75, 79,
 81, 83, 86, 90, 92, 94, 97, 99, 101,
 103, 106, 109, 112, 116, 120, 124,
 128, 132, 136, 139, 142, 143, 145,
 148, 152, 155, 165, 169, 171, 174,
 177, 180, 183, 187, 189, 190, 193,
 216, 220, 223, 224
- TotalCentralStd, 219
- TotalCentralStd(), 5, 9, 13, 17, 20, 24, 28,
 33, 37, 40, 41, 52, 56, 60, 63, 67, 75,
 79, 83, 85, 86, 90, 92, 94, 97, 99,
 101, 103, 106, 109, 112, 116, 120,
 124, 128, 132, 136, 139, 142, 145,
 147, 148, 152, 155, 165, 169, 171,
 174, 177, 180, 183, 187, 190, 192,
 193, 216, 218, 223, 224
- TotalStd, 221
- TotalStd(), 5, 8, 9, 13, 17, 20, 24, 28, 32, 33,
 37, 41, 52, 54, 56, 60, 63, 67, 75, 77,

*79, 83, 86, 90, 92, 94, 97, 99, 101,
103, 106, 109, 111, 112, 116, 120,
124, 128, 132, 135, 136, 139, 142,
145, 148, 152, 154, 155, 165, 168,
169, 171, 174, 177, 180, 183, 186,
187, 190, 193, 216, 218, 220, 224*

Trajectory, *223*

Trajectory(), *5, 9, 13, 17, 20, 24, 28, 33, 37,
41, 52, 56, 60, 63, 67, 75, 79, 83, 86,
90, 92, 94, 97, 99, 101, 103, 106,
109, 112, 116, 120, 124, 128, 132,
136, 139, 142, 145, 148, 152, 155,
165, 169, 171, 174, 177, 180, 183,
187, 190, 193, 216, 218, 220, 223*