# Package 'manCTMed'

June 30, 2025

```
Title Continuous Time Mediation
Version 1.0.9
Description Research compendium for the manuscript
     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:10.0000/0000000000>.
URL https://github.com/jeksterslab/manCTMed,
     https://jeksterslab.github.io/manCTMed/, https://osf.io/qwnmf/,
     https://doi.org/10.0000/0000000000
BugReports https://github.com/jeksterslab/manCTMed/issues
License MIT + file LICENSE
Encoding UTF-8
LazyData true
LazyDataCompression xz
Roxygen list(markdown = TRUE)
Depends R (>= 4.0.0)
Imports stats, dynr, OpenMx, dynUtils, simStateSpace (== 1.2.9),
     bootStateSpace (== 1.0.2), cTMed (== 1.0.6), ggplot2
Suggests knitr, rmarkdown, testthat, DT
Remotes jeksterslab/dynUtils@dc3f47b
RoxygenNote 7.3.2
NeedsCompilation no
Author Ivan Jacob Agaloos Pesigan [aut, cre, cph]
      (<https://orcid.org/0000-0003-4818-8420>),
     Michael A. Russell [ctb] (<https://orcid.org/0000-0002-3956-604X>),
     Sy-Miin Chow [ctb] (<a href="https://orcid.org/0000-0003-1938-027X">https://orcid.org/0000-0003-1938-027X</a>)
Maintainer Ivan Jacob Agaloos Pesigan < r. jeksterslab@gmail.com>
```

2 Contents

# **Contents**

BootPara 4
BootParaStdXMY
BootParaStdXYM
BootParaStdYMX
BootParaXMY
BootParaXYM
BootParaYMX
Compress
DeltaStdXMY
DeltaStdXYM
DeltaStdYMX
DeltaXMY
DeltaXYM
DeltaYMX
FigPlotEffects
FigScatterPlotCoverage
FigScatterPlotPower
FigScatterPlotSeBias
FigScatterPlotType1
FitDynr
FitMx
GenData
IllustrationBootPara
IllustrationCompress
IllustrationFigPlotEffects
IllustrationFigScatterPlotCoverage
IllustrationFigScatterPlotPower
IllustrationFigScatterPlotSeBias
IllustrationFitDynr
IllustrationFitMx
IllustrationGenData
IllustrationMCPhiSigma
IllustrationPrepData
illustration_dist
illustration_dist_dt
illustration_dist_dt_mc
illustration_dist_mc
illustration_dist_med
illustration_dist_med_mc
illustration_dist_med_std
illustration dist med std mc
illustration results
MCStdXMY
MCStdXYM
MCStdYMX
MCXMY 40

Contents 3

MCXYM	
MCYMX	
params	43
PhiHat	43
RandomMeasurement	44
results	45
Sim	
SimDynrBootPara	47
SimDynrBootParaStdXMY	
SimDynrBootParaStdYMX	
SimDynrBootParaXMY	
SimDynrBootParaYMX	
SimDynrDeltaStdXMY	
SimDynrDeltaStdYMX	
SimDynrDeltaXMY	
SimDynrDeltaYMX	
SimDynrMCStdXMY	
SimDynrMCStdYMX	
SimDynrMCXMY	
SimDynrMCYMX	
SimFitDynr	
SimFitMx	
SimFN	
SimGenData	
SimIllustration	
SimIllustrationDynrBootPara	
SimIllustrationDynrBootParaStdXMY	
SimIllustrationDynrBootParaXMY	
SimIllustrationDynrDeltaStdXMY	
SimIllustrationDynrDeltaXMY	
SimIllustrationDynrMCPhiSigma	
SimIllustrationDynrMCStdXMY	
SimillustrationDynrMCXMY	
SimillustrationFitDynr	
SimilustrationFitMx	
SimIllustrationGenData	
SimIllustrationPara	
SimPara	
SimProj	
Sum	
SumDynrDeltaStdXMY	
SumDynrDeltaStdYMX	
SumDynrDeltaXMY	
SumDynrDeltaYMX	
SumDynrMCStdXMY	
SumDynrMCStdYMX	
SumDynrMCXMY	
SumDynrMCYMX	83

BootPara BootPara

83
84
85
86
87
88
88
89
90
91
92

BootPara

Parametric Bootstrap

# Description

The function generates simulated datasets based on a fitted model and refits the model to each generated dataset using the dynr package.

# Usage

```
BootPara(
  fit,
  path,
  prefix,
  taskid,
  B = 1000L,
  ncores = NULL,
  seed = NULL,
  clean = TRUE
)
```

# Arguments

fit	R object. Output of the FitDynr(), FitMx(), IllustrationFitDynr(), or IllustrationFitMx(), functions.
path	Path to a directory to store bootstrap samples and estimates.
prefix	Character string. Prefix used for the file names for the bootstrap samples and estimates.
taskid	Positive integer. Task ID.
В	Positive integer. Number of bootstrap samples.
ncores	Positive integer. Number of cores to use.
seed	Integer. Random seed.
clean	Logical. If clean = TRUE, delete intermediate files generated by the function.

BootParaStdXMY 5

### See Also

Other Confidence Interval Functions: BootParaStdXMY(), BootParaStdXYM(), BootParaStdYMX(), BootParaXYM(), BootParaXYM(), BootParaXYM(), DeltaStdXMY(), DeltaStdXMY(), DeltaStdXMY(), DeltaStdXMY(), DeltaXYM(), DeltaXYM(), IllustrationBootPara(), MCStdXMY(), MCStdXYM(), MCStdXMY(), MCXYM(), MCXYM(), MCYMX()

### **Examples**

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(taskid = 1)
data <- RandomMeasurement(sim)
fit <- FitDynr(data, taskid = 1)
BootPara(
  fit = fit,
  path = getwd(),
  prefix = "pb",
  taskid = 1,
  B = 1000L
)
## End(Not run)</pre>
```

BootParaStdXMY

Parametric Bootstrap Confidence Intervals for X-M-Y (Standardized)

### **Description**

The function generates parametric bootstrap method confidence intervals for the mediation model  $X \to M \to Y$  (Standardized).

### Usage

```
BootParaStdXMY(boot, theta_hat, delta_t = 1:30, ncores = NULL)
```

### **Arguments**

boot R object. Output of the BootPara() function.
theta\_hat R object. Output of the ThetaHat() function.
delta\_t Numeric vector. Vector of time intervals.
ncores Positive integer. Number of cores to use.

```
Other Confidence Interval Functions: BootPara(), BootParaStdXYM(), BootParaStdYMX(), BootParaXMY(), BootParaXYM(), BootParaYMX(), DeltaStdXMY(), DeltaStdXYM(), DeltaStdYMX(), DeltaXMY(), DeltaXMY(), DeltaXMY(), MCStdXYM(), MCStdXYM(), MCStdXYM(), MCStdXYM(), MCXMY(), MCXMY(), MCXMX()
```

6 BootParaStdXYM

### **Examples**

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(taskid = 1)</pre>
data <- RandomMeasurement(sim)</pre>
fit <- FitDynr(data, taskid = 1)</pre>
boot <- BootPara(</pre>
  fit = fit,
  path = getwd(),
  prefix = "pb",
  taskid = 1,
  B = 1000L
)
theta_hat <- ThetaHat(fit)</pre>
ci <- BootParaStdXMY(boot = boot, theta_hat = theta_hat)</pre>
plot(ci)
plot(ci, type = "bc")
## End(Not run)
```

BootParaStdXYM

Parametric Bootstrap Confidence Intervals for X-Y-M (Standardized)

### **Description**

The function generates parametric bootstrap method confidence intervals for the mediation model  $X \to Y \to M$  (Standardized).

### Usage

```
BootParaStdXYM(boot, theta_hat, delta_t = 1:30, ncores = NULL)
```

### **Arguments**

```
boot R object. Output of the BootPara() function.
theta_hat R object. Output of the ThetaHat() function.
delta_t Numeric vector. Vector of time intervals.
ncores Positive integer. Number of cores to use.
```

```
Other Confidence Interval Functions: BootPara(), BootParaStdXMY(), BootParaStdYMX(), BootParaXMY(), BootParaXYM(), BootParaYMX(), DeltaStdXMY(), DeltaStdXMY(), DeltaStdYMX(), DeltaXMY(), DeltaXMY(), DeltaXMY(), MCStdXMY(), MCStdXMY(), MCStdXMY(), MCStdXMX(), MCXMY(), MCXMY(), MCXMX()
```

BootParaStdYMX 7

### **Examples**

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(taskid = 1)
data <- RandomMeasurement(sim)</pre>
fit <- FitDynr(data, taskid = 1)</pre>
boot <- BootPara(</pre>
  fit = fit,
  path = getwd(),
  prefix = "pb",
  taskid = 1,
  B = 1000L
)
theta_hat <- ThetaHat(fit)</pre>
ci <- BootParaStdXYM(boot = boot, theta_hat = theta_hat)</pre>
plot(ci)
plot(ci, type = "bc")
## End(Not run)
```

BootParaStdYMX

Parametric Bootstrap Confidence Intervals for Y-M-X (Standardized)

### **Description**

The function generates parametric bootstrap method confidence intervals for the mediation model  $Y \to M \to X$  (Standardized).

### Usage

```
BootParaStdYMX(boot, theta_hat, delta_t = 1:30, ncores = NULL)
```

### **Arguments**

```
boot R object. Output of the BootPara() function.
theta_hat R object. Output of the ThetaHat() function.
delta_t Numeric vector. Vector of time intervals.
ncores Positive integer. Number of cores to use.
```

```
Other Confidence Interval Functions: BootPara(), BootParaStdXMY(), BootParaStdXYM(), BootParaXMY(), BootParaXYM(), BootParaYMX(), DeltaStdXMY(), DeltaStdXYM(), DeltaStdYMX(), DeltaXMY(), DeltaXMY(), DeltaXMY(), MCStdXYM(), MCStdXYM(), MCStdXYM(), MCStdXYM(), MCXMY(), MCXMY(), MCXMX()
```

8 BootParaXMY

### **Examples**

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(taskid = 1)</pre>
data <- RandomMeasurement(sim)</pre>
fit <- FitDynr(data, taskid = 1)</pre>
boot <- BootPara(</pre>
  fit = fit,
  path = getwd(),
  prefix = "pb",
  taskid = 1,
  B = 1000L
)
theta_hat <- ThetaHat(fit)</pre>
ci <- BootParaStdYMX(boot = boot, theta_hat = theta_hat)</pre>
plot(ci)
plot(ci, type = "bc")
## End(Not run)
```

BootParaXMY

Parametric Bootstrap Confidence Intervals for X-M-Y

### **Description**

The function generates parametric bootstrap method confidence intervals for the mediation model  $X \to M \to Y$ .

### Usage

```
BootParaXMY(boot, phi_hat, delta_t = 1:30, ncores = NULL)
```

### **Arguments**

```
boot R object. Output of the BootPara() function.

phi_hat R object. Output of the PhiHat() function.

delta_t Numeric vector. Vector of time intervals.

ncores Positive integer. Number of cores to use.
```

```
Other Confidence Interval Functions: BootPara(), BootParaStdXMY(), BootParaStdXYM(), BootParaStdYMX(), BootParaXYM(), BootParaYMX(), DeltaStdXMY(), DeltaStdXYM(), DeltaStdYMX(), DeltaXMY(), DeltaXMY(), DeltaXMY(), MCStdXYM(), MCStdXYM(), MCStdXYM(), MCStdXYM(), MCXMY(), MCXYM(), MCXYM(), MCXYM(), MCXYM()
```

BootParaXYM 9

### **Examples**

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(taskid = 1)</pre>
data <- RandomMeasurement(sim)</pre>
fit <- FitDynr(data, taskid = 1)</pre>
boot <- BootPara(</pre>
  fit = fit,
  path = getwd(),
  prefix = "pb",
  taskid = 1,
  B = 1000L
)
phi_hat <- PhiHat(fit)</pre>
ci <- BootParaXMY(boot = boot, phi_hat = phi_hat)</pre>
plot(ci)
plot(ci, type = "bc")
## End(Not run)
```

BootParaXYM

Parametric Bootstrap Confidence Intervals for X-Y-M

### **Description**

The function generates parametric bootstrap method confidence intervals for the mediation model  $X \to Y \to M$ .

### Usage

```
BootParaXYM(boot, phi_hat, delta_t = 1:30, ncores = NULL)
```

### **Arguments**

```
boot R object. Output of the BootPara() function.

phi_hat R object. Output of the PhiHat() function.

delta_t Numeric vector. Vector of time intervals.

ncores Positive integer. Number of cores to use.
```

```
Other Confidence Interval Functions: BootPara(), BootParaStdXMY(), BootParaStdXYM(), BootParaStdYMX(), BootParaXMY(), BootParaYMX(), DeltaStdXMY(), DeltaStdXYM(), DeltaStdYMX(), DeltaXMY(), DeltaXMY(), DeltaXMY(), MCStdXYM(), MCStdXYM(), MCStdXYM(), MCStdXYM(), MCXMY(), MCXYM(), MCXYM(), MCXYM(), MCXYM()
```

10 BootParaYMX

### **Examples**

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(taskid = 1)</pre>
data <- RandomMeasurement(sim)</pre>
fit <- FitDynr(data, taskid = 1)</pre>
boot <- BootPara(</pre>
  fit = fit,
  path = getwd(),
  prefix = "pb",
  taskid = 1,
  B = 1000L
)
phi_hat <- PhiHat(fit)</pre>
ci <- BootParaXYM(boot = boot, phi_hat = phi_hat)</pre>
plot(ci)
plot(ci, type = "bc")
## End(Not run)
```

BootParaYMX

Parametric Bootstrap Confidence Intervals for Y-M-X

### **Description**

The function generates parametric bootstrap method confidence intervals for the mediation model  $Y \to M \to X$ .

### Usage

```
BootParaYMX(boot, phi_hat, delta_t = 1:30, ncores = NULL)
```

### **Arguments**

```
boot R object. Output of the BootPara() function.

phi_hat R object. Output of the PhiHat() function.

delta_t Numeric vector. Vector of time intervals.

ncores Positive integer. Number of cores to use.
```

```
Other Confidence Interval Functions: BootPara(), BootParaStdXMY(), BootParaStdXYM(), BootParaStdYMX(), BootParaXYM(), BootParaXYM(), DeltaStdXMY(), DeltaStdXYM(), DeltaStdYMX(), DeltaXMY(), DeltaXMY(), DeltaXMY(), MCStdXYM(), MCStdXYM(), MCStdXYM(), MCStdXYM(), MCXMY(), MCXYM(), MCXYM(), MCXYM(), MCXYM()
```

Compress 11

### **Examples**

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(taskid = 1)</pre>
data <- RandomMeasurement(sim)</pre>
fit <- FitDynr(data, taskid = 1)</pre>
boot <- BootPara(</pre>
  fit = fit,
  path = getwd(),
  prefix = "pb",
  taskid = 1,
  B = 1000L
)
phi_hat <- PhiHat(fit)</pre>
ci <- BootParaYMX(boot = boot, phi_hat = phi_hat)</pre>
plot(ci, type = "bc")
## End(Not run)
```

Compress

Compress Replication

### **Description**

Compress Replication

### Usage

```
Compress(taskid, repid, output_folder)
```

### **Arguments**

taskid Positive integer. Task ID.
repid Positive integer. Replication ID.
output\_folder Character string. Output folder.

### Value

The output is saved as an external file in output\_folder.

# Author(s)

Ivan Jacob Agaloos Pesigan

### See Also

Other Compression Functions: IllustrationCompress()

12 DeltaStdXYM

DeltaStdXMY

Delta Method Confidence Intervals for X-M-Y (Standardized)

### **Description**

The function generates delta method confidence intervals for the mediation model  $X \to M \to Y$  (Standardized).

### Usage

```
DeltaStdXMY(theta_hat, delta_t = 1:30)
```

### **Arguments**

```
theta_hat R object. Output of the ThetaHat() function.

delta_t Numeric vector. Vector of time intervals.
```

### See Also

```
Other Confidence Interval Functions: BootPara(), BootParaStdXMY(), BootParaStdXYM(), BootParaStdYMX(), BootParaXYM(), BootParaXYM(), BootParaXYM(), DeltaStdXYM(), DeltaStdXYM(), DeltaXMY(), DeltaXMY(), DeltaXMY(), MCStdXYM(), MCStdXYM(), MCStdXYM(), MCStdXYM(), MCXYM(), MCXYM(), MCXYM(), MCXYM(), MCXYM()
```

### **Examples**

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(taskid = 1)
data <- RandomMeasurement(sim)
fit <- FitDynr(data, taskid = 1)
theta_hat <- ThetaHat(fit)
ci <- DeltaStdXMY(theta_hat)
plot(ci)
## End(Not run)</pre>
```

DeltaStdXYM

Delta Method Confidence Intervals for X-Y-M (Standardized)

### **Description**

The function generates delta method confidence intervals for the mediation model  $X \to Y \to M$  (Standardized).

DeltaStdYMX 13

### Usage

```
DeltaStdXYM(theta_hat, delta_t = 1:30)
```

# Arguments

theta\_hat R object. Output of the ThetaHat() function.

delta\_t Numeric vector. Vector of time intervals.

### See Also

```
Other Confidence Interval Functions: BootPara(), BootParaStdXMY(), BootParaStdXYM(), BootParaStdYMX(), BootParaXYM(), BootParaXYM(), BootParaYMX(), DeltaStdXMY(), DeltaStdYMX(), DeltaXMY(), DeltaXMY(), DeltaXMY(), MCStdXYM(), MCStdXYM(), MCStdXYM(), MCStdXYM(), MCXMY(), MCXYM(), MCXYM(), MCXYM(), MCXYM()
```

# Examples

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(taskid = 1)
data <- RandomMeasurement(sim)
fit <- FitDynr(data, taskid = 1)
theta_hat <- ThetaHat(fit)
ci <- DeltaStdXYM(theta_hat)
plot(ci)
## End(Not run)</pre>
```

DeltaStdYMX

Delta Method Confidence Intervals for Y-M-X (Standardized)

### **Description**

The function generates delta method confidence intervals for the mediation model  $Y \to M \to X$  (Standardized).

### Usage

```
DeltaStdYMX(theta_hat, delta_t = 1:30)
```

### **Arguments**

```
theta_hat R object. Output of the ThetaHat() function.

delta_t Numeric vector. Vector of time intervals.
```

14 DeltaXMY

### See Also

```
Other Confidence Interval Functions: BootPara(), BootParaStdXMY(), BootParaStdXYM(), BootParaStdYMX(), BootParaXYM(), BootParaXYM(), BootParaXYM(), DeltaStdXMY(), DeltaStdXYM(), DeltaXMY(), DeltaXMY(), DeltaXMY(), MCStdXYM(), MCStdXYM(), MCStdXYM(), MCStdXYM(), MCXYM(), MCXYM(), MCXYM(), MCXYM(), MCXYM()
```

### **Examples**

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(taskid = 1)
data <- RandomMeasurement(sim)
fit <- FitDynr(data, taskid = 1)
theta_hat <- ThetaHat(fit)
ci <- DeltaStdYMX(theta_hat)
plot(ci)
## End(Not run)</pre>
```

DeltaXMY

Delta Method Confidence Intervals for X-M-Y

### Description

The function generates delta method confidence intervals for the mediation model  $X \to M \to Y$ .

### Usage

```
DeltaXMY(phi_hat, delta_t = 1:30)
```

### **Arguments**

```
phi_hat R object. Output of the PhiHat() function.

delta_t Numeric vector. Vector of time intervals.
```

```
Other Confidence Interval Functions: BootPara(), BootParaStdXMY(), BootParaStdXYM(), BootParaStdYMX(), BootParaXYM(), BootParaXYM(), BootParaXYM(), DeltaStdXMY(), DeltaStdXYM(), DeltaStdYMX(), DeltaXYM(), DeltaYMX(), IllustrationBootPara(), MCStdXMY(), MCStdXYM(), MCStdYMX(), MCXMY(), MCXYM(), MCXYM(), MCXYM(), MCXYM()
```

DeltaXYM 15

### **Examples**

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(taskid = 1)
data <- RandomMeasurement(sim)
fit <- FitDynr(data, taskid = 1)
phi_hat <- PhiHat(fit)
ci <- DeltaXMY(phi_hat)
plot(ci)
## End(Not run)</pre>
```

DeltaXYM

Delta Method Confidence Intervals for X-Y-M

### **Description**

The function generates delta method confidence intervals for the mediation model  $X \to Y \to M$ .

#### Usage

```
DeltaXYM(phi_hat, delta_t = 1:30)
```

# Arguments

```
phi_hat R object. Output of the PhiHat() function.
delta_t Numeric vector. Vector of time intervals.
```

### See Also

```
Other Confidence Interval Functions: BootPara(), BootParaStdXMY(), BootParaStdXYM(), BootParaStdYMX(), BootParaXYM(), BootParaXYM(), BootParaXYM(), DeltaStdXMY(), DeltaStdXYM(), DeltaStdYMX(), DeltaXMY(), DeltaYMX(), IllustrationBootPara(), MCStdXMY(), MCStdXYM(), MCStdYMX(), MCXMY(), MCXYM(), MCXYM(), MCXYM(), MCXYM()
```

### **Examples**

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(taskid = 1)
data <- RandomMeasurement(sim)
fit <- FitDynr(data, taskid = 1)
phi_hat <- PhiHat(fit)
ci <- DeltaXYM(phi_hat)
plot(ci)
## End(Not run)</pre>
```

FigPlotEffects

DeltaYMX

Delta Method Confidence Intervals for Y-M-X

### **Description**

The function generates delta method confidence intervals for the mediation model  $Y \to M \to X$ .

### Usage

```
DeltaYMX(phi_hat, delta_t = 1:30)
```

### **Arguments**

```
phi_hat R object. Output of the PhiHat() function.

delta_t Numeric vector. Vector of time intervals.
```

### See Also

```
Other Confidence Interval Functions: BootPara(), BootParaStdXMY(), BootParaStdXYM(), BootParaStdYMX(), BootParaXYM(), BootParaXYM(), BootParaXYM(), DeltaStdXMY(), DeltaStdXYM(), DeltaStdYMX(), DeltaXMY(), DeltaXYM(), IllustrationBootPara(), MCStdXMY(), MCStdXYM(), MCStdYMX(), MCXMY(), MCXYM(), MCXYM(), MCXYM()
```

### **Examples**

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(taskid = 1)
data <- RandomMeasurement(sim)
fit <- FitDynr(data, taskid = 1)
phi_hat <- PhiHat(fit)
ci <- DeltaYMX(phi_hat)
plot(ci)
## End(Not run)</pre>
```

FigPlotEffects

Plot Total, Direct, and Indirect Effects

### **Description**

```
Effects for the model X \to M \to Y.
```

### Usage

```
FigPlotEffects(dynamics = 0, std = FALSE, max_delta_t = 30, xmy = TRUE)
```

### **Arguments**

dynamics	Integer dynamics =	0 for original drift matrix.	dynamics = -1 for near-neutral

dynamics, and dynamics = 1 for stronger damping.

std Logical. If std = TRUE, standardized total, direct, and indirect effects. If std =

FALSE, unstandardized total, direct, and indirect effects.

max\_delta\_t Numeric. Maximum time interval.

xmy Logical. If xmy = TRUE, plot the effects for the  $x \rightarrow m \rightarrow y$  mediation model. If xmy

= FALSE, plot the effects for the  $y \rightarrow m \rightarrow x$  mediation model.

### Author(s)

Ivan Jacob Agaloos Pesigan

### See Also

Other Figure Functions: FigScatterPlotCoverage(), FigScatterPlotPower(), FigScatterPlotSeBias(), FigScatterPlotType1(), IllustrationFigPlotEffects(), IllustrationFigScatterPlotCoverage(), IllustrationFigScatterPlotPower(), IllustrationFigScatterPlotSeBias()

### **Examples**

FigPlotEffects()

FigScatterPlotCoverage

Plot Coverage Probabilities

### **Description**

Coverage probabilities for the model  $X \to M \to Y$ .

### Usage

FigScatterPlotCoverage(results, delta\_t = NULL, dynamics = 0, std = FALSE)

### **Arguments**

results	Summary results data frame.
delta_t	Vector of time-interval value. If delta_t = NULL, use all available time-intervals
dynamics	Integer. dynamics = $0$ for original drift matrix, dynamics = $-1$ for near-neutral dynamics, and dynamics = $1$ for stronger damping.
std	Logical. If std = TRUE, standardized total, direct, and indirect effects. If std = FALSE, unstandardized total, direct, and indirect effects.

18 FigScatterPlotPower

#### Author(s)

Ivan Jacob Agaloos Pesigan

#### See Also

Other Figure Functions: FigPlotEffects(), FigScatterPlotPower(), FigScatterPlotSeBias(), FigScatterPlotType1(), IllustrationFigPlotEffects(), IllustrationFigScatterPlotCoverage(), IllustrationFigScatterPlotPower(), IllustrationFigScatterPlotSeBias()

### **Examples**

```
data(results, package = "manCTMed")
FigScatterPlotCoverage(results)
FigScatterPlotCoverage(results, delta_t = 1:14)
FigScatterPlotCoverage(results, delta_t = 15:30)
```

FigScatterPlotPower

Plot Statistical Power

### **Description**

Statistical power for the model  $X \to M \to Y$ .

### Usage

```
FigScatterPlotPower(results, delta_t = NULL, dynamics = 0, std = FALSE)
```

### **Arguments**

results	Summary results data frame.
delta_t	Vector of time-interval value. If delta_t = NULL, use all available time-intervals
dynamics	Integer. dynamics = $0$ for original drift matrix, dynamics = $-1$ for near-neutral dynamics, and dynamics = $1$ for stronger damping.
std	Logical. If std = TRUE, standardized total, direct, and indirect effects. If std = FALSE, unstandardized total, direct, and indirect effects.

### Author(s)

Ivan Jacob Agaloos Pesigan

```
Other Figure Functions: FigPlotEffects(), FigScatterPlotCoverage(), FigScatterPlotSeBias(),
FigScatterPlotType1(), IllustrationFigPlotEffects(), IllustrationFigScatterPlotCoverage(),
IllustrationFigScatterPlotPower(), IllustrationFigScatterPlotSeBias()
```

FigScatterPlotSeBias 19

### **Examples**

```
data(results, package = "manCTMed")
FigScatterPlotPower(results)
FigScatterPlotPower(results, delta_t = 1:14)
FigScatterPlotPower(results, delta_t = 15:30)
```

FigScatterPlotSeBias Plot Standard Error Bias

# Description

Standard Error Bias for the model  $X \to M \to Y$ .

### Usage

```
FigScatterPlotSeBias(results, delta_t = NULL, dynamics = 0, std = FALSE)
```

# Arguments

results	Summary results data frame.
delta_t	Vector of time-interval value. If delta_t = NULL, use all available time-intervals
dynamics	Integer. dynamics = $0$ for original drift matrix, dynamics = $-1$ for near-neutral dynamics, and dynamics = $1$ for stronger damping.
std	Logical. If std = TRUE, standardized total, direct, and indirect effects. If std = FALSE, unstandardized total, direct, and indirect effects.

### Author(s)

Ivan Jacob Agaloos Pesigan

#### See Also

```
Other Figure Functions: FigPlotEffects(), FigScatterPlotCoverage(), FigScatterPlotPower(), FigScatterPlotType1(), IllustrationFigPlotEffects(), IllustrationFigScatterPlotCoverage(), IllustrationFigScatterPlotPower(), IllustrationFigScatterPlotSeBias()
```

### **Examples**

```
data(results, package = "manCTMed")
FigScatterPlotSeBias(results)
```

FigScatterPlotType1

FigScatterPlotType1 Plot Type I Error

# Description

Type I error for the model  $Y \to M \to X$ .

# Usage

```
FigScatterPlotType1(results, delta_t = NULL, dynamics = 0, std = FALSE)
```

### **Arguments**

results	Summary results data frame.
delta_t	Vector of time-interval value. If delta_t = NULL, use all available time-intervals
dynamics	Integer. dynamics = $0$ for original drift matrix, dynamics = $-1$ for near-neutral dynamics, and dynamics = $1$ for stronger damping.
std	Logical. If std = TRUE, standardized total, direct, and indirect effects. If std = FALSE, unstandardized total, direct, and indirect effects.

### Author(s)

Ivan Jacob Agaloos Pesigan

### See Also

```
Other Figure Functions: FigPlotEffects(), FigScatterPlotCoverage(), FigScatterPlotPower(), FigScatterPlotSeBias(), IllustrationFigPlotEffects(), IllustrationFigScatterPlotCoverage(), IllustrationFigScatterPlotPower(), IllustrationFigScatterPlotSeBias()
```

### **Examples**

```
data(results, package = "manCTMed")
FigScatterPlotType1(results)
FigScatterPlotType1(results, delta_t = 1:14)
FigScatterPlotType1(results, delta_t = 15:30)
```

FitDynr 21

FitDynr

Fit the Model using the dynr Package

### **Description**

The function fits the model using the dynr package.

### Usage

```
FitDynr(data, taskid)
```

# Arguments

data R object. Output of the RandomMeasurement() function.

taskid Positive integer. Task ID.

### See Also

Other Model Fitting Functions: FitMx(), IllustrationFitDynr(), IllustrationFitMx(), IllustrationMCPhiSigma() IllustrationPrepData(), PhiHat(), ThetaHat()

# **Examples**

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(taskid = 1)
data <- RandomMeasurement(sim)
fit <- FitDynr(data, taskid = 1)
summary(fit)
## End(Not run)</pre>
```

FitMx

Fit the Model using the OpenMx Package

# Description

The function fits the model using the OpenMx package.

### Usage

```
FitMx(data, taskid)
```

22 GenData

### **Arguments**

data R object. Output of the RandomMeasurement() function.

taskid Positive integer. Task ID.

### See Also

```
Other Model Fitting Functions: FitDynr(), IllustrationFitDynr(), IllustrationFitMx(), IllustrationMCPhiSigma(), IllustrationPrepData(), PhiHat(), ThetaHat()
```

### **Examples**

```
## Not run:
set.seed(42)
library(OpenMx)
sim <- GenData(taskid = 1)
data <- RandomMeasurement(sim)
fit <- FitMx(data, taskid = 1)
summary(fit)
## End(Not run)</pre>
```

GenData

Simulate Data

# Description

The function simulates data using the simStateSpace::SimSSMOUFixed() function.

### Usage

```
GenData(taskid)
```

### **Arguments**

taskid Positive integer. Task ID.

### See Also

Other Data Generation Functions: IllustrationGenData(), RandomMeasurement()

# **Examples**

```
## Not run:
set.seed(42)
sim <- GenData(taskid = 1)
plot(sim)
## End(Not run)</pre>
```

IllustrationBootPara 23

IllustrationBootPara Parametric Bootstrap (Illustration)

### **Description**

The function generates simulated datasets based on a fitted model and refits the model to each generated dataset using the dynr package.

### Usage

```
IllustrationBootPara(
  fit,
  path,
  prefix,
  taskid,
  B = 1000L,
  ncores = NULL,
  seed = NULL
)
```

### **Arguments**

```
fit R object. Fitted CT-VAR model.

path Path to a directory to store bootstrap samples and estimates.

prefix Character string. Prefix used for the file names for the bootstrap samples and estimates.

taskid Positive integer. Task ID.

B Positive integer. Number of bootstrap samples.

ncores Positive integer. Number of cores to use.

seed Integer. Random seed.
```

### See Also

```
Other Confidence Interval Functions: BootPara(), BootParaStdXMY(), BootParaStdXYM(), BootParaStdYMX(), BootParaXYM(), BootParaXYM(), BootParaXYM(), DeltaStdXMY(), DeltaStdXMY(), DeltaStdXMY(), DeltaStdXMY(), MCStdXMY(), MCStdXMY(), MCXMY(), MCXMY(), MCXMY(), MCXMY(), MCXMY(), MCXMY(), MCXMX()
```

# Examples

```
## Not run:
library(dynr)
sim <- IllustrationGenData(seed = 42)
data <- IllustrationPrepData(sim)
fit <- IllustrationFitDynr(data)
summary(fit)</pre>
```

24 IllustrationCompress

```
IllustrationBootPara(
  fit = fit,
  path = getwd(),
  prefix = "pb",
  taskid = 1,
  B = 1000L,
  seed = 42
)

## End(Not run)
```

IllustrationCompress Compress Replication (Illustration)

# Description

Compress Replication (Illustration)

### Usage

```
IllustrationCompress(taskid, repid, output_folder)
```

# Arguments

taskid Positive integer. Task ID.

repid Positive integer. Replication ID. output\_folder Character string. Output folder.

### Value

The output is saved as an external file in output\_folder.

# Author(s)

Ivan Jacob Agaloos Pesigan

#### See Also

Other Compression Functions: Compress()

 $Illustration {\tt FigPlotEffects}$ 

Plot Total, Direct, and Indirect Effects (Illustration)

### **Description**

Effects for the model  $X \to M \to Y$ .

### Usage

```
IllustrationFigPlotEffects(std = FALSE, max_delta_t = 30)
```

### **Arguments**

std Logical. If std = TRUE, standardized total, direct, and indirect effects. If std =

FALSE, unstandardized total, direct, and indirect effects.

max\_delta\_t Numeric. Maximum time interval.

### Author(s)

Ivan Jacob Agaloos Pesigan

### See Also

```
Other Figure Functions: FigPlotEffects(), FigScatterPlotCoverage(), FigScatterPlotPower(), FigScatterPlotSeBias(), FigScatterPlotType1(), IllustrationFigScatterPlotCoverage(), IllustrationFigScatterPlotPower(), IllustrationFigScatterPlotSeBias()
```

### **Examples**

```
IllustrationFigPlotEffects(std = FALSE)
IllustrationFigPlotEffects(std = TRUE)
```

IllustrationFigScatterPlotCoverage

Illustration Plot Coverage Probabilities

### **Description**

Coverage probabilities for the model  $X \to M \to Y$ .

### Usage

IllustrationFigScatterPlotCoverage(illustration\_results)

#### **Arguments**

```
illustration_results
```

Summary results data frame.

### Author(s)

Ivan Jacob Agaloos Pesigan

#### See Also

```
Other Figure Functions: FigPlotEffects(), FigScatterPlotCoverage(), FigScatterPlotPower(), FigScatterPlotSeBias(), FigScatterPlotType1(), IllustrationFigPlotEffects(), IllustrationFigScatterPlotIllustrationFigScatterPlotSeBias()
```

### **Examples**

```
data(illustration_results, package = "manCTMed")
IllustrationFigScatterPlotCoverage(illustration_results)
```

 $Illustration {\tt FigScatterPlotPower}$ 

Illustration Plot Statistical Power

### Description

Statistical Power for the model  $X \to M \to Y$ .

#### **Usage**

IllustrationFigScatterPlotPower(illustration\_results)

### **Arguments**

```
illustration_results
```

Summary results data frame.

### Author(s)

Ivan Jacob Agaloos Pesigan

```
Other Figure Functions: FigPlotEffects(), FigScatterPlotCoverage(), FigScatterPlotPower(), FigScatterPlotSeBias(), FigScatterPlotType1(), IllustrationFigPlotEffects(), IllustrationFigScatterPlotIllustrationFigScatterPlotSeBias()
```

### **Examples**

```
data(illustration_results, package = "manCTMed")
IllustrationFigScatterPlotPower(illustration_results)
```

IllustrationFigScatterPlotSeBias

Illustration Plot Standard Error Bias

### **Description**

Standard Error Bias for the model  $X \to M \to Y$ .

# Usage

IllustrationFigScatterPlotSeBias(illustration\_results)

### Arguments

illustration\_results

Summary results data frame.

### Author(s)

Ivan Jacob Agaloos Pesigan

#### See Also

```
Other Figure Functions: FigPlotEffects(), FigScatterPlotCoverage(), FigScatterPlotPower(), FigScatterPlotSeBias(), FigScatterPlotType1(), IllustrationFigPlotEffects(), IllustrationFigScatterPlotIllustrationFigScatterPlotPower()
```

### **Examples**

```
data(illustration_results, package = "manCTMed")
IllustrationFigScatterPlotSeBias(illustration_results)
```

28 IllustrationFitMx

IllustrationFitDynr Fit the Model using the dynr Package (Illustration)

### Description

The function fits the model using the dynr package.

### Usage

```
IllustrationFitDynr(data)
```

### **Arguments**

data

R object. Output of the IllustrationPrepData() function.

### See Also

```
Other Model Fitting Functions: FitDynr(), FitMx(), IllustrationFitMx(), IllustrationMCPhiSigma(), IllustrationPrepData(), PhiHat(), ThetaHat()
```

# **Examples**

```
## Not run:
library(dynr)
sim <- IllustrationGenData(seed = 42)
data <- IllustrationPrepData(sim)
fit <- IllustrationFitDynr(data)
summary(fit)
## End(Not run)</pre>
```

Illustration Fit Mx

Fit the Model using the OpenMx Package (Illustration)

### **Description**

The function fits the model using the OpenMx package.

# Usage

```
IllustrationFitMx(data)
```

### **Arguments**

data

R object. Output of the IllustrationPrepData() function.

IllustrationGenData 29

### See Also

```
Other Model Fitting Functions: FitDynr(), FitMx(), IllustrationFitDynr(), IllustrationMCPhiSigma(), IllustrationPrepData(), PhiHat(), ThetaHat()
```

# **Examples**

```
## Not run:
library(OpenMx)
sim <- IllustrationGenData(seed = 42)
data <- IllustrationPrepData(sim)
fit <- IllustrationFitMx(data)
summary(fit)
## End(Not run)</pre>
```

IllustrationGenData Simulate Data (Illustration)

### **Description**

The function simulates data using the simStateSpace::SimSSMOUFixed() function.

### Usage

```
IllustrationGenData(seed = NULL, n = 133, m = 101, delta_t_gen = 0.1)
```

# **Arguments**

```
seed Integer. Random seed.

n Positive integer. Sample size.

m Positive integer. Measurement occasions.

delta_t_gen Numeric. Time interval used to generate data.
```

#### See Also

Other Data Generation Functions: GenData(), RandomMeasurement()

# Examples

```
## Not run:
sim <- IllustrationGenData(seed = 42)
plot(sim)
## End(Not run)</pre>
```

30 IllustrationPrepData

```
IllustrationMCPhiSigma
```

Generate a Sampling Distribution of Drift Matrices and Process Noise Covariance Matrices (Illustration)

### **Description**

The function generates a sampling distribution of drift matrices and process noise covariance matrices using te Monte Carlo method.

### Usage

```
IllustrationMCPhiSigma(fit, R = 20000L, seed = NULL)
```

### **Arguments**

fit R object. Fitted CT-VAR model.

R Positive integer. Number of Monte Carlo replications.

seed Integer. Random seed.

### See Also

```
Other Model Fitting Functions: FitDynr(), FitMx(), IllustrationFitDynr(), IllustrationFitMx(), IllustrationPrepData(), PhiHat(), ThetaHat()
```

# Examples

```
## Not run:
library(dynr)
sim <- IllustrationGenData(seed = 42)
data <- IllustrationPrepData(sim)
fit <- IllustrationFitDynr(data)
IllustrationMCPhiSigma(fit, seed = 42)
## End(Not run)</pre>
```

IllustrationPrepData Prepare Data Before Model Fitting (Illustration)

### **Description**

The function converts the output of IllustrationGenData() into a data frame.

### Usage

```
IllustrationPrepData(sim)
```

illustration\_dist 31

### **Arguments**

sim

R object. Output of the IllustrationGenData() function.

#### See Also

```
Other Model Fitting Functions: FitDynr(), FitMx(), IllustrationFitDynr(), IllustrationFitMx(), IllustrationMCPhiSigma(), PhiHat(), ThetaHat()
```

# **Examples**

```
## Not run:
sim <- IllustrationGenData(seed = 42)
data <- IllustrationPrepData(sim)
head(data)
dim(data)
## End(Not run)</pre>
```

illustration\_dist

Illustration Sampling Distribution

### **Description**

Illustration Sampling Distribution

# Usage

```
data(illustration_dist)
```

### **Format**

A matrix with 1000 rows and 27 columns:

phi\_xx Elements of the drift matrix.

sigma\_xx Elements of the process noise covariance matrix.

theta\_xx Elements of the measurement error covariance matrix.

mu0\_x Elements of the initial condition mean vector.

sigma0\_xx Elements of the initial condition covariance matrix.

### Author(s)

### **Description**

Illustration Sampling Distribution Discrete Time - Time Interval of 1

### Usage

```
data(illustration_dist_dt)
```

#### **Format**

A matrix with 1000 rows and 15 columns:

beta\_xx Elements of the matrix of lagged coefficients.

psi\_xx Elements of the process noise covariance matrix.

### Author(s)

Ivan Jacob Agaloos Pesigan

```
illustration_dist_dt_mc
```

Illustration Sampling Distribution Discrete Time - Time Interval of 1 (Monte Carlo Method)

### Description

Illustration Sampling Distribution Discrete Time - Time Interval of 1 (Monte Carlo Method)

### Usage

```
data(illustration_dist_dt_mc)
```

#### **Format**

A matrix with 20000 rows and 15 columns:

beta\_xx Elements of the matrix of lagged coefficients.

psi\_xx Elements of the process noise covariance matrix.

#### Author(s)

illustration\_dist\_mc 33

# Description

Illustration Sampling Distribution (Monte Carlo Method)

### Usage

```
data(illustration_dist_mc)
```

#### **Format**

A matrix with 20000 rows and 15 columns:

phi\_xx Elements of the drift matrix.

**sigma\_xx** Elements of the process noise covariance matrix.

### Author(s)

Ivan Jacob Agaloos Pesigan

### **Description**

Illustration Sampling Distribution Total, Direct, and Indirect Effects - Time Interval of 1

# Usage

```
data(illustration_dist_med)
```

### **Format**

A matrix with 1000 rows and 27 columns:

total Total effect.

direct Direct effect.

indirect Indirect effect.

#### Author(s)

```
illustration_dist_med_mc
```

Illustration Sampling Distribution Total, Direct, and Indirect Effects -Time Interval of 1 (Monte Carlo Method)

### **Description**

Illustration Sampling Distribution Total, Direct, and Indirect Effects - Time Interval of 1 (Monte Carlo Method)

### Usage

```
data(illustration_dist_med_mc)
```

### **Format**

A matrix with 20000 rows and 27 columns:

total Total effect.

direct Direct effect.

indirect Indirect effect.

### Author(s)

Ivan Jacob Agaloos Pesigan

```
illustration\_dist\_med\_std
```

Illustration Sampling Distribution Standardized Total, Direct, and Indirect Effects - Time Interval of  $\it 1$ 

### **Description**

Illustration Sampling Distribution Standardized Total, Direct, and Indirect Effects - Time Interval of  $\mathbf{1}$ 

### Usage

```
data(illustration_dist_med_std)
```

### **Format**

A matrix with 1000 rows and 27 columns:

total Total effect.

direct Direct effect.

indirect Indirect effect.

### Author(s)

Ivan Jacob Agaloos Pesigan

illustration\_dist\_med\_std\_mc

Illustration Sampling Distribution Standardized Total, Direct, and Indirect Effects - Time Interval of 1 (Monte Carlo Method)

# Description

Illustration Sampling Distribution Standardized Total, Direct, and Indirect Effects - Time Interval of 1 (Monte Carlo Method)

### Usage

```
data(illustration_dist_med_std_mc)
```

#### **Format**

A matrix with 20000 rows and 27 columns:

total Total effect.

direct Direct effect.

indirect Indirect effect.

### Author(s)

Ivan Jacob Agaloos Pesigan

 $illustration\_results \quad \textit{Illustration Small Scale Simulation Results}$ 

### **Description**

Illustration Small Scale Simulation Results

### Usage

```
data(illustration_results)
```

36 illustration\_results

### **Format**

```
A with 22 columns:
```

taskid Task ID.

replications Number of replications.

effect Total, direct, or indirect effect.

interval Time interval.

parameter Population parameter.

method Method used to generate confidence intervals.

**xmy** Logical. TRUE for x to m to y path.

std Logical. TRUE for standardized. FALSE for unstandardized.

est Mean parameter estimate.

se Mean standard error.

**z** Mean z statistic.

**p** Mean p-value.

**R** Number of Monte Carlo or bootstrap replications.

Il Mean lower limit of the 95% confidence interval.

ul Mean upper limit of the 95% confidence interval.

sig Proportion of statistically significant results.

zero\_hit Proportion of replications where the confidence intervals included zero.

**theta\_hit** Proportion of replications where the confidence intervals included the population parameter.

sq\_error Mean squared error.

se\_bias Bias in standard error estimate.

coverage Coverage probability.

power Statistical power.

### Author(s)

MCStdXMY 37

MCStdXMY

Monte Carlo Method Confidence Intervals for X-M-Y (Standardized)

### **Description**

The function generates Monte Carlo method confidence intervals for the mediation model  $X \to M \to Y$  (Standardized).

### Usage

```
MCStdXMY(theta_hat, delta_t = 1:30, R = 20000L, seed = NULL)
```

### **Arguments**

theta\_hat R object. Output of the ThetaHat() function.

delta\_t Numeric vector. Vector of time intervals.

R Positive integer. Number of Monte Carlo replications.

seed Integer. Random seed.

#### See Also

```
Other Confidence Interval Functions: BootPara(), BootParaStdXMY(), BootParaStdXYM(), BootParaStdYMX(), BootParaXYM(), BootParaXYM(), BootParaXYM(), DeltaStdXMY(), DeltaStdXYM(), DeltaStdYMX(), DeltaXMY(), DeltaXYM(), DeltaXYM(), IllustrationBootPara(), MCStdXYM(), MCStdYMX(), MCXMY(), MCXYM(), MCXYM(), MCXYM()
```

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(taskid = 1)
data <- RandomMeasurement(sim)
fit <- FitDynr(data, taskid = 1)
theta_hat <- ThetaHat(fit)
ci <- MCStdXMY(theta_hat, seed = 42)
plot(ci)
## End(Not run)</pre>
```

38 MCStdXYM

MCStdXYM

Monte Carlo Method Confidence Intervals for X-Y-M (Standardized)

### **Description**

The function generates Monte Carlo method confidence intervals for the mediation model  $X \to Y \to M$  (Standardized).

### Usage

```
MCStdXYM(theta_hat, delta_t = 1:30, R = 20000L, seed = NULL)
```

### **Arguments**

theta\_hat R object. Output of the ThetaHat() function.

delta\_t Numeric vector. Vector of time intervals.

R Positive integer. Number of Monte Carlo replications.

seed Integer. Random seed.

#### See Also

```
Other Confidence Interval Functions: BootPara(), BootParaStdXMY(), BootParaStdXYM(), BootParaStdYMX(), BootParaXYM(), BootParaXYM(), BootParaXYM(), DeltaStdXMY(), DeltaStdXMY(), DeltaStdXMY(), DeltaStdYMX(), DeltaXMY(), DeltaXYM(), DeltaXYM(), IllustrationBootPara(), MCStdXMY(), MCStdYMX(), MCXMY(), MCXYM(), MCXYM(), MCXYM()
```

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(taskid = 1)
data <- RandomMeasurement(sim)
fit <- FitDynr(data, taskid = 1)
theta_hat <- ThetaHat(fit)
ci <- MCStdXYM(theta_hat, seed = 42)
plot(ci)
## End(Not run)</pre>
```

MCStdYMX 39

MCStdYMX

Monte Carlo Method Confidence Intervals for Y-M-X (Standardized)

### **Description**

The function generates Monte Carlo method confidence intervals for the mediation model  $Y \to M \to X$  (Standardized).

### Usage

```
MCStdYMX(theta_hat, delta_t = 1:30, R = 20000L, seed = NULL)
```

### **Arguments**

theta\_hat R object. Output of the ThetaHat() function.

delta\_t Numeric vector. Vector of time intervals.

R Positive integer. Number of Monte Carlo replications.

seed Integer. Random seed.

#### See Also

```
Other Confidence Interval Functions: BootPara(), BootParaStdXMY(), BootParaStdXYM(), BootParaStdYMX(), BootParaXYM(), BootParaXYM(), BootParaXYM(), DeltaStdXMY(), DeltaStdXYM(), DeltaStdYMX(), DeltaXMY(), DeltaXYM(), DeltaXYM(), IllustrationBootPara(), MCStdXMY(), MCStdXYM(), MCXMY(), MCXYM(), MCXYM(), MCXYM()
```

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(taskid = 1)
data <- RandomMeasurement(sim)
fit <- FitDynr(data, taskid = 1)
theta_hat <- ThetaHat(fit)
ci <- MCStdYMX(phi_hat, seed = 42)
plot(ci)
## End(Not run)</pre>
```

40 MCXMY

**MCXMY** 

Monte Carlo Method Confidence Intervals for X-M-Y

### **Description**

The function generates Monte Carlo method confidence intervals for the mediation model  $X \to M \to Y$ .

### Usage

```
MCXMY(phi_hat, delta_t = 1:30, R = 20000L, seed = NULL)
```

### **Arguments**

phi\_hat R object. Output of the PhiHat() function.

delta\_t Numeric vector. Vector of time intervals.

R Positive integer. Number of Monte Carlo replications.

seed Integer. Random seed.

#### See Also

```
Other Confidence Interval Functions: BootPara(), BootParaStdXMY(), BootParaStdXYM(), BootParaStdYMX(), BootParaXYM(), BootParaXYM(), BootParaXYM(), DeltaStdXMY(), DeltaStdXMY(), DeltaStdXMY(), DeltaStdXMY(), DeltaXYM(), DeltaXYM(), MCStdXMY(), MCStdXMY(), MCStdXMY(), MCStdXMY(), MCStdXMY(), MCXYM(), MCYMX()
```

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(taskid = 1)
data <- RandomMeasurement(sim)
fit <- FitDynr(data, taskid = 1)
phi_hat <- PhiHat(fit)
ci <- MCXMY(phi_hat, seed = 42)
plot(ci)
## End(Not run)</pre>
```

MCXYM 41

**MCXYM** 

Monte Carlo Method Confidence Intervals for X-Y-M

### **Description**

The function generates Monte Carlo method confidence intervals for the mediation model  $X \to Y \to M$ .

### Usage

```
MCXYM(phi_hat, delta_t = 1:30, R = 20000L, seed = NULL)
```

### **Arguments**

phi\_hat R object. Output of the PhiHat() function.

delta\_t Numeric vector. Vector of time intervals.

R Positive integer. Number of Monte Carlo replications.

seed Integer. Random seed.

#### See Also

```
Other Confidence Interval Functions: BootPara(), BootParaStdXMY(), BootParaStdXYM(), BootParaStdYMX(), BootParaXYM(), BootParaXYM(), BootParaXYM(), DeltaStdXMY(), DeltaStdXYM(), DeltaStdYMX(), DeltaXMY(), DeltaXYM(), DeltaXYM(), IllustrationBootPara(), MCStdXMY(), MCStdXYM(), MCStdXYM(), MCStdXMY(), MCXMY(), MCYMX()
```

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(taskid = 1)
data <- RandomMeasurement(sim)
fit <- FitDynr(data, taskid = 1)
phi_hat <- PhiHat(fit)
ci <- MCXYM(phi_hat, seed = 42)
plot(ci)
## End(Not run)</pre>
```

42 MCYMX

**MCYMX** 

Monte Carlo Method Confidence Intervals for Y-M-X

### **Description**

The function generates Monte Carlo method confidence intervals for the mediation model  $Y \to M \to X$ .

### Usage

```
MCYMX(phi_hat, delta_t = 1:30, R = 20000L, seed = NULL)
```

### **Arguments**

phi\_hat R object. Output of the PhiHat() function.

delta\_t Numeric vector. Vector of time intervals.

R Positive integer. Number of Monte Carlo replications.

seed Integer. Random seed.

#### See Also

```
Other Confidence Interval Functions: BootPara(), BootParaStdXMY(), BootParaStdXYM(), BootParaStdYMX(), BootParaXYM(), BootParaXYM(), BootParaXYM(), DeltaStdXMY(), DeltaStdXMY(), DeltaStdXMY(), DeltaStdXMY(), DeltaXYM(), DeltaXYM(), MCStdXMY(), MCStdXMY(), MCStdXMY(), MCStdXMY(), MCXMY(), MCXYM()
```

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(taskid = 1)
data <- RandomMeasurement(sim)
fit <- FitDynr(data, taskid = 1)
phi_hat <- PhiHat(fit)
ci <- MCYMX(phi_hat, seed = 42)
plot(ci)
## End(Not run)</pre>
```

params 43

params

Simulation Parameters

## **Description**

Simulation Parameters

## Usage

data(params)

#### **Format**

A dataframe with 30 rows and 3 columns:

taskid Simulation Task ID.

n Sample size.

**dynamics** Dynamics. 0 for original drift matrix, -1 for near-neutral dynamics, and 1 for stronger damping.

## Author(s)

Ivan Jacob Agaloos Pesigan

PhiHat

Estimated Drift Matrix

## **Description**

The function extracts the estimated drift matrix from the fitted model.

### Usage

PhiHat(fit)

## **Arguments**

fit

R object. Output of the FitDynr(), FitMx(), IllustrationFitDynr(), or IllustrationFitMx(), functions.

### See Also

```
Other Model Fitting Functions: FitDynr(), FitMx(), IllustrationFitDynr(), IllustrationFitMx(), IllustrationMCPhiSigma(), IllustrationPrepData(), ThetaHat()
```

44 RandomMeasurement

## **Examples**

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(n = 50)
data <- RandomMeasurement(sim)
fit <- FitDynr(data)
PhiHat(fit)
## End(Not run)</pre>
```

RandomMeasurement

Simulate Random Measurement

# Description

The function randomly selects 100 observations from the generated data and replaces the unselected observations with NA.

# Usage

RandomMeasurement(sim)

# Arguments

sim

R object. Output of the GenData() function.

## See Also

Other Data Generation Functions: GenData(), IllustrationGenData()

```
## Not run:
set.seed(42)
sim <- GenData(taskid = 1)
RandomMeasurement(sim)
## End(Not run)</pre>
```

results 45

results

Simulation Results

## Description

Simulation Results

### Usage

data(results)

#### **Format**

A dataframe with 24 columns:

taskid Task ID.

replications Number of replications.

effect Total, direct, or indirect effect.

interval Time interval.

**dynamics** Dynamics. 0 for original drift matrix, -1 for near-neutral dynamics, and 1 for stronger damping.

parameter Population parameter.

**method** Method used to generate confidence intervals.

**xmy** If TRUE, the mediation model is  $X \to M \to Y$ . If FALSE, the mediation model is  $Y \to M \to X$ .

std If TRUE, standardized total, direct, and indirect effects. If FALSE, unstandardized total, direct, and indirect effects.

n Sample size.

est Mean parameter estimate.

- se Mean standard error.
- **z** Mean z statistic.
- **p** Mean *p*-value.
- R Number of Monte Carlo replications.
- **ll** Mean lower limit of the 95% confidence interval.
- ul Mean upper limit of the 95% confidence interval.
- sig Proportion of statistically significant results.

**zero\_hit** Proportion of replications where the confidence intervals contained zero.

**theta\_hit** Proportion of replications where the confidence intervals contained the population parameter.

sq\_error Mean squared error.

se\_bias Bias in standard error estimate.

coverage Coverage probability.

power Statistical power.

46 Sim

### Author(s)

Ivan Jacob Agaloos Pesigan

Sim Simulation Replication

#### **Description**

Simulation Replication

## Usage

```
Sim(
  taskid,
  repid,
  output_folder,
  overwrite,
  integrity,
  seed,
  ci,
  pb,
  delta_t,
  R,
  B
)
```

## Arguments

Positive integer. Task ID. taskid repid Positive integer. Replication ID. Character string. Output folder. output\_folder overwrite Logical. Overwrite existing output in output\_folder. Logical. If integrity = TRUE, check for the output file integrity when overwrite integrity = FALSE. Integer. Random seed. seed Logical. Run simulations for confidence intervals. ci pb Logical. Run simulations for parametric bootstrap confidence intervals. delta\_t Numeric vector. Vector of time intervals. R Positive integer. Number of Monte Carlo replications. В Positive integer. Number of bootstrap samples.

### Value

The output is saved as an external file in output\_folder.

SimDynrBootPara 47

### Author(s)

Ivan Jacob Agaloos Pesigan

SimDynrBootPara Simulation Replication - BootPara

# Description

Simulation Replication - BootPara

## Usage

```
SimDynrBootPara(
   taskid,
   repid,
   output_folder,
   seed,
   suffix,
   overwrite,
   integrity,
   B,
   ncores = NULL
)
```

## **Arguments**

Positive integer. Task ID. taskid repid Positive integer. Replication ID. output\_folder Character string. Output folder. seed Integer. Random seed. suffix Character string. Output of manCTMed:::.SimSuffix(). Logical. Overwrite existing output in output\_folder. overwrite Logical. If integrity = TRUE, check for the output file integrity when overwrite integrity = FALSE. В Positive integer. Number of bootstrap samples. Positive integer. Number of cores to use. ncores

### **Details**

This function is executed via the Sim function.

## Value

The output is saved as an external file in output\_folder.

### Author(s)

Ivan Jacob Agaloos Pesigan

SimDynrBootParaStdXMY Simulation Replication - BootParaStdXMY

## **Description**

Simulation Replication - BootParaStdXMY

# Usage

```
SimDynrBootParaStdXMY(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  delta_t
)
```

## Arguments

taskid Positive integer. Task ID.
repid Positive integer. Replication ID.
output\_folder Character string. Output folder.
seed Integer. Random seed.

suffix Character string. Output of manCTMed:::.SimSuffix().
overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

delta\_t Numeric vector. Vector of time intervals.

## **Details**

This function is executed via the Sim function.

### Value

The output is saved as an external file in output\_folder.

## Author(s)

SimDynrBootParaStdYMX Simulation Replication - BootParaStdYMX

### **Description**

Simulation Replication - BootParaStdYMX

## Usage

```
SimDynrBootParaStdYMX(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  delta_t
)
```

## **Arguments**

taskid Positive integer. Task ID.

repid Positive integer. Replication ID. output\_folder Character string. Output folder.

seed Integer. Random seed.

suffix Character string. Output of manCTMed:::.SimSuffix().

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

delta\_t Numeric vector. Vector of time intervals.

#### **Details**

This function is executed via the Sim function.

### Value

The output is saved as an external file in output\_folder.

## Author(s)

SimDynrBootParaXMY

Simulation Replication - BootParaXMY

## **Description**

Simulation Replication - BootParaXMY

## Usage

```
SimDynrBootParaXMY(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  delta_t
)
```

## **Arguments**

taskid Positive integer. Task ID.

repid Positive integer. Replication ID. output\_folder Character string. Output folder.

seed Integer. Random seed.

suffix Character string. Output of manCTMed:::.SimSuffix().

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

delta\_t Numeric vector. Vector of time intervals.

#### **Details**

This function is executed via the Sim function.

### Value

The output is saved as an external file in output\_folder.

## Author(s)

SimDynrBootParaYMX Sim

Simulation Replication - BootParaYMX

## **Description**

Simulation Replication - BootParaYMX

## Usage

```
SimDynrBootParaYMX(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  delta_t
)
```

## **Arguments**

taskid Positive integer. Task ID.

repid Positive integer. Replication ID. output\_folder Character string. Output folder.

seed Integer. Random seed.

suffix Character string. Output of manCTMed:::.SimSuffix().

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

delta\_t Numeric vector. Vector of time intervals.

#### **Details**

This function is executed via the Sim function.

### Value

The output is saved as an external file in output\_folder.

## Author(s)

SimDynrDeltaStdXMY

Simulation Replication - DynrDeltaStdXMY

## **Description**

Simulation Replication - DynrDeltaStdXMY

## Usage

```
SimDynrDeltaStdXMY(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  delta_t
)
```

## **Arguments**

taskid Positive integer. Task ID.

repid Positive integer. Replication ID. output\_folder Character string. Output folder.

seed Integer. Random seed.

suffix Character string. Output of manCTMed:::.SimSuffix().

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

delta\_t Numeric vector. Vector of time intervals.

#### **Details**

This function is executed via the Sim function.

### Value

The output is saved as an external file in output\_folder.

## Author(s)

SimDynrDeltaStdYMX Simulation Replication - DynrDeltaStdYMX

### **Description**

Simulation Replication - DynrDeltaStdYMX

## Usage

```
SimDynrDeltaStdYMX(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  delta_t
)
```

## **Arguments**

taskid Positive integer. Task ID.

repid Positive integer. Replication ID. output\_folder Character string. Output folder.

seed Integer. Random seed.

suffix Character string. Output of manCTMed:::.SimSuffix().

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

delta\_t Numeric vector. Vector of time intervals.

#### **Details**

This function is executed via the Sim function.

### Value

The output is saved as an external file in output\_folder.

## Author(s)

54 SimDynrDeltaXMY

SimDynrDeltaXMY

Simulation Replication - DynrDeltaXMY

## **Description**

Simulation Replication - DynrDeltaXMY

## Usage

```
SimDynrDeltaXMY(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  delta_t
)
```

## **Arguments**

taskid Positive integer. Task ID.

repid Positive integer. Replication ID. output\_folder Character string. Output folder.

seed Integer. Random seed.

suffix Character string. Output of manCTMed:::.SimSuffix().

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

delta\_t Numeric vector. Vector of time intervals.

# **Details**

This function is executed via the Sim function.

### Value

The output is saved as an external file in output\_folder.

## Author(s)

SimDynrDeltaYMX 55

SimDynrDeltaYMX Simulation Replication - DynrDeltaYMX	SimDynrDeltaYMX	Simulation Replication - DynrDeltaYMX	
---	-----------------	---------------------------------------	--

## **Description**

Simulation Replication - DynrDeltaYMX

# Usage

```
SimDynrDeltaYMX(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  delta_t
)
```

## **Arguments**

taskid Positive integer. Task ID. repid Positive integer. Replication ID. output\_folder Character string. Output folder. seed Integer. Random seed. suffix Character string. Output of manCTMed:::.SimSuffix(). Logical. Overwrite existing output in output\_folder. overwrite integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite = FALSE. Numeric vector. Vector of time intervals. delta\_t

#### **Details**

This function is executed via the Sim function.

### Value

The output is saved as an external file in output\_folder.

## Author(s)

SimDynrMCStdXMY

Simulation Replication - DynrMCStdXMY

### **Description**

Simulation Replication - DynrMCStdXMY

# Usage

```
SimDynrMCStdXMY(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
 overwrite,
  integrity,
  delta_t,
 R
)
```

## **Arguments**

taskid Positive integer. Task ID. Positive integer. Replication ID. repid output\_folder Character string. Output folder. Integer. Random seed. seed Character string. Output of manCTMed:::.SimSuffix(). suffix overwrite Logical. Overwrite existing output in output\_folder. Logical. If integrity = TRUE, check for the output file integrity when overwrite integrity = FALSE. Numeric vector. Vector of time intervals.

delta\_t

R Positive integer. Number of Monte Carlo replications.

### **Details**

This function is executed via the Sim function.

## Value

The output is saved as an external file in output\_folder.

# Author(s)

SimDynrMCStdYMX 57

SimDynrMCStdYMX Simulation Replication - DynrMCStdYMX

Description

Simulation Replication - DynrMCStdYMX

# Usage

```
SimDynrMCStdYMX(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  delta_t,
  R
)
```

## Arguments

taskid Positive integer. Task ID. repid Positive integer. Replication ID. output\_folder Character string. Output folder. Integer. Random seed. seed Character string. Output of manCTMed:::.SimSuffix(). suffix overwrite Logical. Overwrite existing output in output\_folder. Logical. If integrity = TRUE, check for the output file integrity when overwrite integrity = FALSE. Numeric vector. Vector of time intervals. delta\_t R Positive integer. Number of Monte Carlo replications.

### **Details**

This function is executed via the Sim function.

## Value

The output is saved as an external file in output\_folder.

# Author(s)

58 SimDynrMCXMY

SimDynrMCXMY

Simulation Replication - DynrMCXMY

### **Description**

Simulation Replication - DynrMCXMY

# Usage

```
SimDynrMCXMY(
   taskid,
   repid,
   output_folder,
   seed,
   suffix,
   overwrite,
   integrity,
   delta_t,
   R
)
```

## Arguments

taskid Positive integer. Task ID.

repid Positive integer. Replication ID.

output\_folder Character string. Output folder.

seed Integer. Random seed.

Suffix Character string. Output of many

suffix Character string. Output of manCTMed:::.SimSuffix().
overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

delta\_t Numeric vector. Vector of time intervals.

R Positive integer. Number of Monte Carlo replications.

### **Details**

This function is executed via the Sim function.

## Value

The output is saved as an external file in output\_folder.

# Author(s)

SimDynrMCYMX 59

SimDynrMCYMX

Simulation Replication - DynrMCYMX

### **Description**

Simulation Replication - DynrMCYMX

# Usage

```
SimDynrMCYMX(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  delta_t,
  R
)
```

## Arguments

taskid Positive integer. Task ID. Positive integer. Replication ID. repid output\_folder Character string. Output folder. Integer. Random seed. seed Character string. Output of manCTMed:::.SimSuffix(). suffix overwrite Logical. Overwrite existing output in output\_folder. Logical. If integrity = TRUE, check for the output file integrity when overwrite integrity = FALSE. Numeric vector. Vector of time intervals. delta\_t

Positive integer. Number of Monte Carlo replications.

### **Details**

R

This function is executed via the Sim function.

## Value

The output is saved as an external file in output\_folder.

# Author(s)

60 SimFitDynr

# Description

Simulation Replication - FitDynr

## Usage

```
SimFitDynr(taskid, repid, output_folder, seed, suffix, overwrite, integrity)
```

# Arguments

taskid Positive integer. Task ID.

repid Positive integer. Replication ID.
output\_folder Character string. Output folder.

seed Integer. Random seed.

suffix Character string. Output of manCTMed:::.SimSuffix().

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

### **Details**

This function is executed via the Sim function.

# Value

The output is saved as an external file in output\_folder.

### Author(s)

SimFitMx 61

SimFitMx	Simulation Replication - FitMx

# Description

Simulation Replication - FitMx

## Usage

```
SimFitMx(taskid, repid, output_folder, seed, suffix, overwrite, integrity)
```

# Arguments

taskid Positive integer. Task ID.

repid Positive integer. Replication ID.

output\_folder Character string. Output folder.

seed Integer. Random seed.

suffix Character string. Output of manCTMed:::.SimSuffix().

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

### **Details**

This function is executed via the Sim function.

# Value

The output is saved as an external file in output\_folder.

### Author(s)

62 SimGenData

SimFN	Simulation File Name	

## **Description**

Simulation File Name

### Usage

```
SimFN(output_type, output_folder, suffix)
```

# Arguments

output\_type Character string. Output type.
output\_folder Character string. Output folder.

suffix Character string. Output of manCTMed:::.SimSuffix().

#### Value

Returns a character string file name with the output\_folder in the OS-specific format.

|--|

## **Description**

Simulation Replication - GenData

### Usage

```
SimGenData(taskid, repid, output_folder, seed, suffix, overwrite, integrity)
```

#### **Arguments**

taskid Positive integer. Task ID.

repid Positive integer. Replication ID. output\_folder Character string. Output folder.

seed Integer. Random seed.

suffix Character string. Output of manCTMed:::.SimSuffix().

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

SimIllustration 63

## **Details**

This function is executed via the Sim function.

#### Value

The output is saved as an external file in output\_folder.

### Author(s)

Ivan Jacob Agaloos Pesigan

SimIllustration

Simulation Replication (Illustration)

# Description

Simulation Replication (Illustration)

## Usage

```
SimIllustration(
  taskid,
  repid,
  output_folder,
  overwrite,
  integrity,
  seed,
  ci,
  pb,
  delta_t,
  R,
  B
)
```

### **Arguments**

```
taskid
                  Positive integer. Task ID.
                  Positive integer. Replication ID.
repid
                  Character string. Output folder.
output_folder
                  Logical. Overwrite existing output in output_folder.
overwrite
                  Logical. If integrity = TRUE, check for the output file integrity when overwrite
integrity
                  = FALSE.
                  Integer. Random seed.
seed
                  Logical. Run simulations for confidence intervals.
ci
                  Logical. Run simulations for parametric bootstrap confidence intervals.
pb
```

delta\_t Numeric vector. Vector of time intervals.

R Positive integer. Number of Monte Carlo replications.

B Positive integer. Number of bootstrap samples.

### Value

The output is saved as an external file in output\_folder.

#### Author(s)

Ivan Jacob Agaloos Pesigan

SimIllustrationDynrBootPara

Simulation Replication - BootPara

## **Description**

Simulation Replication - BootPara

# Usage

```
SimIllustrationDynrBootPara(
   taskid,
   repid,
   output_folder,
   seed,
   suffix,
   overwrite,
   integrity,
   B,
   ncores = NULL
)
```

## **Arguments**

taskid Positive integer. Task ID.

repid Positive integer. Replication ID. output\_folder Character string. Output folder.

seed Integer. Random seed.

suffix Character string. Output of manCTMed:::.SimSuffix().
overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

B Positive integer. Number of bootstrap samples.

ncores Positive integer. Number of cores to use.

This function is executed via the IllustrationSim function.

#### Value

The output is saved as an external file in output\_folder.

### Author(s)

Ivan Jacob Agaloos Pesigan

```
SimIllustrationDynrBootParaStdXMY
```

Simulation Replication - BootParaStdXMY

# Description

Simulation Replication - BootParaStdXMY

### Usage

```
SimIllustrationDynrBootParaStdXMY(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  delta_t
)
```

# **Arguments**

delta\_t

repid Positive integer. Task ID.

repid Positive integer. Replication ID.

output\_folder Character string. Output folder.

seed Integer. Random seed.

suffix Character string. Output of manCTMed:::.SimSuffix().

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite = FALSE.

Numeric vector. Vector of time intervals.

This function is executed via the IllustrationSim function.

### Value

The output is saved as an external file in output\_folder.

### Author(s)

Ivan Jacob Agaloos Pesigan

```
SimIllustrationDynrBootParaXMY
```

Simulation Replication - BootParaXMY

# Description

Simulation Replication - BootParaXMY

## Usage

```
SimIllustrationDynrBootParaXMY(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  delta_t
)
```

# **Arguments**

taskid Positive integer. Task ID.

repid Positive integer. Replication ID. output\_folder Character string. Output folder.

seed Integer. Random seed.

suffix Character string. Output of manCTMed:::.SimSuffix().

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

delta\_t Numeric vector. Vector of time intervals.

This function is executed via the IllustrationSim function.

#### Value

The output is saved as an external file in output\_folder.

### Author(s)

Ivan Jacob Agaloos Pesigan

```
SimIllustrationDynrDeltaStdXMY
```

Simulation Replication - Illustration (DynrDeltaStdXMY)

# Description

Simulation Replication - Illustration (DynrDeltaStdXMY)

### Usage

```
SimIllustrationDynrDeltaStdXMY(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  delta_t
)
```

## **Arguments**

```
taskid
                  Positive integer. Task ID.
repid
                  Positive integer. Replication ID.
                  Character string. Output folder.
output_folder
                  Integer. Random seed.
seed
suffix
                  Character string. Output of manCTMed:::.SimSuffix().
overwrite
                  Logical. Overwrite existing output in output_folder.
integrity
                  Logical. If integrity = TRUE, check for the output file integrity when overwrite
                  = FALSE.
                  Numeric vector. Vector of time intervals.
delta_t
```

This function is executed via the IllustrationSim function.

#### Value

The output is saved as an external file in output\_folder.

### Author(s)

Ivan Jacob Agaloos Pesigan

```
SimIllustrationDynrDeltaXMY
```

Simulation Replication - Illustration (DynrDeltaXMY)

# Description

Simulation Replication - Illustration (DynrDeltaXMY)

### Usage

```
SimIllustrationDynrDeltaXMY(
   taskid,
   repid,
   output_folder,
   seed,
   suffix,
   overwrite,
   integrity,
   delta_t
)
```

## **Arguments**

taskid Positive integer. Task ID.

repid Positive integer. Replication ID. output\_folder Character string. Output folder.

seed Integer. Random seed.

suffix Character string. Output of manCTMed:::.SimSuffix().

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

delta\_t Numeric vector. Vector of time intervals.

This function is executed via the IllustrationSim function.

#### Value

The output is saved as an external file in output\_folder.

### Author(s)

Ivan Jacob Agaloos Pesigan

```
SimIllustrationDynrMCPhiSigma
```

Simulation Replication - Illustration (MCPhiSigma)

# Description

Simulation Replication - Illustration (MCPhiSigma)

### Usage

```
SimIllustrationDynrMCPhiSigma(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  R
)
```

## **Arguments**

taskid Positive integer. Task ID.
repid Positive integer. Replication ID.
output\_folder Character string. Output folder.

seed Integer. Random seed.

suffix Character string. Output of manCTMed:::.SimSuffix().

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

R Positive integer. Number of Monte Carlo replications.

This function is executed via the IllustrationSim function.

#### Value

The output is saved as an external file in output\_folder.

### Author(s)

Ivan Jacob Agaloos Pesigan

SimIllustrationDynrMCStdXMY

Simulation Replication - Illustration (DynrMCStdXMY)

## **Description**

Simulation Replication - Illustration (DynrMCStdXMY)

# Usage

```
SimIllustrationDynrMCStdXMY(
   taskid,
   repid,
   output_folder,
   seed,
   suffix,
   overwrite,
   integrity,
   delta_t,
   R
)
```

### **Arguments**

taskid Positive integer. Task ID.
repid Positive integer. Replication ID.
output\_folder Character string. Output folder.
seed Integer. Random seed.

suffix Character string. Output of manCTMed:::.SimSuffix().
overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

delta\_t Numeric vector. Vector of time intervals.

R Positive integer. Number of Monte Carlo replications.

This function is executed via the IllustrationSim function.

#### Value

The output is saved as an external file in output\_folder.

### Author(s)

Ivan Jacob Agaloos Pesigan

SimIllustrationDynrMCXMY

Simulation Replication - Illustration (DynrMCXMY)

## **Description**

Simulation Replication - Illustration (DynrMCXMY)

# Usage

```
SimIllustrationDynrMCXMY(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity,
  delta_t,
  R
)
```

### **Arguments**

taskid Positive integer. Task ID. repid Positive integer. Replication ID. output\_folder Character string. Output folder. seed Integer. Random seed. suffix Character string. Output of manCTMed:::.SimSuffix(). overwrite Logical. Overwrite existing output in output\_folder. integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite = FALSE. Numeric vector. Vector of time intervals. delta\_t Positive integer. Number of Monte Carlo replications. R

This function is executed via the IllustrationSim function.

#### Value

The output is saved as an external file in output\_folder.

### Author(s)

Ivan Jacob Agaloos Pesigan

SimIllustrationFitDynr

Simulation Replication - IllustrationFitDynr

# Description

Simulation Replication - IllustrationFitDynr

## Usage

```
SimIllustrationFitDynr(
  taskid,
  repid,
 output_folder,
  seed,
  suffix,
 overwrite,
  integrity
)
```

### **Arguments**

taskid Positive integer. Task ID. Positive integer. Replication ID. repid output\_folder Character string. Output folder.

Integer. Random seed. seed

suffix Character string. Output of manCTMed:::.SimSuffix(). overwrite Logical. Overwrite existing output in output\_folder.

Logical. If integrity = TRUE, check for the output file integrity when overwrite integrity

= FALSE.

### **Details**

This function is executed via the IllustrationSim function.

SimIllustrationFitMx 73

# Value

The output is saved as an external file in output\_folder.

## Author(s)

Ivan Jacob Agaloos Pesigan

SimIllustrationFitMx Simulation Replication - IllustrationFitMx

# **Description**

Simulation Replication - IllustrationFitMx

## Usage

```
SimIllustrationFitMx(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity
)
```

#### **Arguments**

taskid Positive integer. Task ID.

repid Positive integer. Replication ID. output\_folder Character string. Output folder.

seed Integer. Random seed.

suffix Character string. Output of manCTMed:::.SimSuffix().
overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

# **Details**

This function is executed via the IllustrationSim function.

#### Value

The output is saved as an external file in output\_folder.

74 SimIllustrationGenData

#### Author(s)

Ivan Jacob Agaloos Pesigan

SimIllustrationGenData

Simulation Replication - IllustrationGenData

# Description

Simulation Replication - IllustrationGenData

#### Usage

```
SimIllustrationGenData(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity
)
```

# Arguments

taskid Positive integer. Task ID.
repid Positive integer. Replication ID.
output\_folder Character string. Output folder.
seed Integer. Random seed.

suffix Character string. Output of manCTMed:::.SimSuffix().

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

#### **Details**

This function is executed via the IllustrationSim function.

## Value

The output is saved as an external file in output\_folder.

#### Author(s)

SimIllustrationPara 75

SimIllustrationPara Simulation Replication Parametric Bootstrap (Parallel)

# Description

Simulation Replication Parametric Bootstrap (Parallel)

# Usage

```
SimIllustrationPara(
  taskid,
  repid,
  output_folder,
  overwrite,
  integrity,
  seed,
  B
)
```

# **Arguments**

taskid Positive integer. Task ID.

repid Positive integer. Replication ID.

output\_folder Character string. Output folder.

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

seed Integer. Random seed.

B Positive integer. Number of bootstrap samples.

## Value

The output is saved as an external file in output\_folder.

#### Author(s)

76 SimProj

SimPara	Simulation Replication Parametric Bootstrap (Parallel)

# Description

Simulation Replication Parametric Bootstrap (Parallel)

# Usage

```
SimPara(taskid, repid, output_folder, overwrite, integrity, seed, B)
```

# Arguments

taskid Positive integer. Task ID.

repid Positive integer. Replication ID.

output\_folder Character string. Output folder.

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite = FALSE.

seed Integer. Random seed.

B Positive integer. Number of bootstrap samples.

#### Value

The output is saved as an external file in output\_folder.

# Author(s)

Ivan Jacob Agaloos Pesigan

SimProj	Simulation Project Name	

# Description

Simulation Project Name

# Usage

SimProj()

# Value

Returns the project name as a character string.

Sum 77

#### Author(s)

Ivan Jacob Agaloos Pesigan

Sum Summary

# **Description**

Summary

# Usage

Sum(taskid, reps, output\_folder, overwrite, integrity)

# **Arguments**

taskid Positive integer. Task ID.

reps Positive integer. Number of replications.

output\_folder Character string. Output folder.

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

## Value

The output is saved as an external file in output\_folder.

# Author(s)

Ivan Jacob Agaloos Pesigan

SumDynrDeltaStdXMY Summary (DynrDeltaStdXMY)

# Description

Summary (DynrDeltaStdXMY)

# Usage

SumDynrDeltaStdXMY(taskid, reps, output\_folder, overwrite, integrity)

## **Arguments**

taskid Positive integer. Task ID.

reps Positive integer. Number of replications.

output\_folder Character string. Output folder.

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

## **Details**

This function is executed via the Sum function.

#### Value

The output is saved as an external file in output\_folder.

## Author(s)

Ivan Jacob Agaloos Pesigan

SumDynrDeltaStdYMX Summary (DynrDeltaStdYMX)

# **Description**

Summary (DynrDeltaStdYMX)

#### Usage

SumDynrDeltaStdYMX(taskid, reps, output\_folder, overwrite, integrity)

# **Arguments**

taskid Positive integer. Task ID.

reps Positive integer. Number of replications.

output\_folder Character string. Output folder.

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

#### **Details**

This function is executed via the Sum function.

SumDynrDeltaXMY 79

# Value

The output is saved as an external file in output\_folder.

#### Author(s)

Ivan Jacob Agaloos Pesigan

SumDynrDeltaXMY Summary (DynrDeltaXMY)

# **Description**

Summary (DynrDeltaXMY)

#### Usage

SumDynrDeltaXMY(taskid, reps, output\_folder, overwrite, integrity)

# **Arguments**

taskid Positive integer. Task ID.

reps Positive integer. Number of replications.

output\_folder Character string. Output folder.

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

## Details

This function is executed via the Sum function.

# Value

The output is saved as an external file in output\_folder.

#### Author(s)

SumDynrDeltaYMX	Summary (DynrDeltaYMX)
Juliby III DCI Call IX	Summary (Bym Benarmar)

# Description

Summary (DynrDeltaYMX)

#### Usage

```
SumDynrDeltaYMX(taskid, reps, output_folder, overwrite, integrity)
```

#### **Arguments**

taskid Positive integer. Task ID.

reps Positive integer. Number of replications.

output\_folder Character string. Output folder.

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

#### **Details**

This function is executed via the Sum function.

# Value

The output is saved as an external file in output\_folder.

#### Author(s)

Ivan Jacob Agaloos Pesigan

SumDynrMCStdXMY Summary (DynrMCStdXMY)

# Description

Summary (DynrMCStdXMY)

# Usage

SumDynrMCStdXMY(taskid, reps, output\_folder, overwrite, integrity)

SumDynrMCStdYMX 81

## **Arguments**

taskid Positive integer. Task ID.

reps Positive integer. Number of replications.

output\_folder Character string. Output folder.

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

#### **Details**

This function is executed via the Sum function.

#### Value

The output is saved as an external file in output\_folder.

## Author(s)

Ivan Jacob Agaloos Pesigan

SumDynrMCStdYMX Summary (DynrMCStdYMX)

# **Description**

Summary (DynrMCStdYMX)

#### Usage

SumDynrMCStdYMX(taskid, reps, output\_folder, overwrite, integrity)

# **Arguments**

taskid Positive integer. Task ID.

reps Positive integer. Number of replications.

output\_folder Character string. Output folder.

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

#### **Details**

This function is executed via the Sum function.

82 SumDynrMCXMY

# Value

The output is saved as an external file in output\_folder.

# Author(s)

Ivan Jacob Agaloos Pesigan

SumDynrMCXMY Summary (DynrMCXMY)

# Description

Summary (DynrMCXMY)

#### Usage

SumDynrMCXMY(taskid, reps, output\_folder, overwrite, integrity)

# **Arguments**

taskid Positive integer. Task ID.

reps Positive integer. Number of replications.

output\_folder Character string. Output folder.

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

## Details

This function is executed via the Sum function.

# Value

The output is saved as an external file in output\_folder.

# Author(s)

SumDynrMCYMX 83

ary (DynrMCYMX)
iry (DynimCim

# Description

Summary (DynrMCYMX)

#### Usage

```
SumDynrMCYMX(taskid, reps, output_folder, overwrite, integrity)
```

## **Arguments**

taskid Positive integer. Task ID.

reps Positive integer. Number of replications.

output\_folder Character string. Output folder.

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

#### **Details**

This function is executed via the Sum function.

# Value

The output is saved as an external file in output\_folder.

# Author(s)

Ivan Jacob Agaloos Pesigan

SumFitDynr Summary (FitDynr)

# Description

Summary (FitDynr)

# Usage

SumFitDynr(taskid, reps, output\_folder, overwrite, integrity)

84 SumIllustration

#### **Arguments**

taskid Positive integer. Task ID.

reps Positive integer. Number of replications.

output\_folder Character string. Output folder.

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

#### **Details**

This function is executed via the Sum function.

#### Value

The output is saved as an external file in output\_folder.

#### Author(s)

Ivan Jacob Agaloos Pesigan

SumIllustration Summary (Illustration)

# Description

Summary (Illustration)

# Usage

SumIllustration(taskid, reps, output\_folder, overwrite, integrity)

# **Arguments**

taskid Positive integer. Task ID.

reps Positive integer. Number of replications.

output\_folder Character string. Output folder.

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

#### Value

The output is saved as an external file in output\_folder.

#### Author(s)

 ${\tt SumIllustrationDynrBootParaStdXMY}$ 

Summary - Illustration (DynrBootParaStdXMY)

# Description

Summary - Illustration (DynrBootParaStdXMY)

# Usage

```
SumIllustrationDynrBootParaStdXMY(
  taskid,
  reps,
  output_folder,
  overwrite,
  integrity,
  type = "pc"
)
```

# **Arguments**

taskid Positive integer. Task ID.

reps Positive integer. Number of replications.

output\_folder Character string. Output folder.

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

type Character string. Confidence interval type.

# **Details**

This function is executed via the IllustrationSum function.

#### Value

The output is saved as an external file in output\_folder.

#### Author(s)

 ${\tt SumIllustrationDynrBootParaXMY}$ 

Summary - Illustration (DynrBootParaXMY)

# Description

Summary - Illustration (DynrBootParaXMY)

# Usage

```
SumIllustrationDynrBootParaXMY(
   taskid,
   reps,
   output_folder,
   overwrite,
   integrity,
   type = "pc"
)
```

# **Arguments**

taskid Positive integer. Task ID.

reps Positive integer. Number of replications.

output\_folder Character string. Output folder.

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

type Character string. Confidence interval type.

# **Details**

This function is executed via the IllustrationSum function.

#### Value

The output is saved as an external file in output\_folder.

#### Author(s)

 ${\tt SumIllustrationDynrDeltaStdXMY}$ 

Summary - Illustration (DynrDeltaStdXMY)

# Description

Summary - Illustration (DynrDeltaStdXMY)

# Usage

```
SumIllustrationDynrDeltaStdXMY(
  taskid,
  reps,
  output_folder,
  overwrite,
  integrity
)
```

# Arguments

taskid Positive integer. Task ID.

reps Positive integer. Number of replications.

output\_folder Character string. Output folder.

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

## **Details**

This function is executed via the IllustrationSum function.

#### Value

The output is saved as an external file in output\_folder.

# Author(s)

SumIllustrationDynrDeltaXMY

Summary - Illustration (DynrDeltaXMY)

# Description

Summary - Illustration (DynrDeltaXMY)

# Usage

SumIllustrationDynrDeltaXMY(taskid, reps, output\_folder, overwrite, integrity)

# **Arguments**

taskid Positive integer. Task ID.

reps Positive integer. Number of replications.

output\_folder Character string. Output folder.

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

#### **Details**

This function is executed via the IllustrationSum function.

## Value

The output is saved as an external file in output\_folder.

# Author(s)

Ivan Jacob Agaloos Pesigan

 ${\tt SumIllustrationDynrMCStdXMY}$ 

Summary - Illustration (DynrMCStdXMY)

#### **Description**

Summary - Illustration (DynrMCStdXMY)

#### Usage

SumIllustrationDynrMCStdXMY(taskid, reps, output\_folder, overwrite, integrity)

#### **Arguments**

taskid Positive integer. Task ID.

reps Positive integer. Number of replications.

output\_folder Character string. Output folder.

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

#### **Details**

This function is executed via the IllustrationSum function.

#### Value

The output is saved as an external file in output\_folder.

## Author(s)

Ivan Jacob Agaloos Pesigan

SumIllustrationDynrMCXMY

Summary - Illustration (DynrMCXMY)

# **Description**

Summary - Illustration (DynrMCXMY)

# Usage

SumIllustrationDynrMCXMY(taskid, reps, output\_folder, overwrite, integrity)

#### **Arguments**

taskid Positive integer. Task ID.

reps Positive integer. Number of replications.

output\_folder Character string. Output folder.

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

#### **Details**

This function is executed via the IllustrationSum function.

# Value

The output is saved as an external file in output\_folder.

#### Author(s)

Ivan Jacob Agaloos Pesigan

 ${\tt SumIllustrationFitDynr}$ 

Summary - Illustration (FitDynr)

# Description

Summary - Illustration (FitDynr)

# Usage

SumIllustrationFitDynr(taskid, reps, output\_folder, overwrite, integrity)

# Arguments

taskid Positive integer. Task ID.

reps Positive integer. Number of replications.

 $output\_folder \quad Character\ string.\ Output\ folder.$ 

overwrite Logical. Overwrite existing output in output\_folder.

integrity Logical. If integrity = TRUE, check for the output file integrity when overwrite

= FALSE.

#### **Details**

This function is executed via the IllustrationSum function.

#### Value

The output is saved as an external file in output\_folder.

#### Author(s)

ThetaHat 91

ThetaHat

Estimated Drift Matrix and Process Noise

# Description

The function extracts the estimated drift matrix and process noise from the fitted model.

# Usage

```
ThetaHat(fit)
```

# **Arguments**

fit

R object. Output of the FitDynr(), FitMx(), IllustrationFitDynr(), or IllustrationFitMx(), functions.

# See Also

```
Other Model Fitting Functions: FitDynr(), FitMx(), IllustrationFitDynr(), IllustrationFitMx(), IllustrationMCPhiSigma(), IllustrationPrepData(), PhiHat()
```

# **Examples**

```
## Not run:
set.seed(42)
library(dynr)
sim <- GenData(n = 50)
data <- RandomMeasurement(sim)
fit <- FitDynr(data)
ThetaHat(fit)
## End(Not run)</pre>
```

# **Index**

* Compression Functions	$Illustration {\tt FigScatterPlotSeBias},$
Compress, 11	27
IllustrationCompress, 24	* Model Fitting Functions
* Confidence Interval Functions	FitDynr, 21
BootPara, 4	FitMx, 21
BootParaStdXMY, 5	IllustrationFitDynr,28
BootParaStdXYM, 6	IllustrationFitMx, 28
BootParaStdYMX, 7	IllustrationMCPhiSigma, 30
BootParaXMY, 8	IllustrationPrepData, 30
BootParaXYM, 9	PhiHat, 43
BootParaYMX, 10	ThetaHat, 91
DeltaStdXMY, 12	* <b>ci</b>
DeltaStdXYM, 12	BootPara, 4
DeltaStdYMX, 13	BootParaStdXMY, 5
DeltaXMY, 14	BootParaStdXYM, 6
DeltaXYM, 15	BootParaStdYMX, 7
DeltaYMX, 16	BootParaXMY, 8
IllustrationBootPara, 23	BootParaXYM, 9
MCStdXMY, 37	BootParaYMX, 10
MCStdXYM, 38	DeltaStdXMY, 12
MCStdYMX, 39	DeltaStdXYM, 12
MCXMY, 40	DeltaStdYMX, 13
MCXYM, 41	DeltaXMY, 14
MCYMX, 42	DeltaXYM, 15
* Data Generation Functions	DeltaYMX, 16
GenData, 22	IllustrationBootPara, 23
IllustrationGenData, 29	IllustrationMCPhiSigma, 30
RandomMeasurement, 44	MCStdXMY, 37
* Figure Functions	MCStdXYM, 38
FigPlotEffects, 16	MCStdYMX, 39
FigScatterPlotCoverage, 17	MCXMY, 40
FigScatterPlotPower, 18	MCXYM, 41
FigScatterPlotSeBias, 19	MCYMX, 42
FigScatterPlotType1, 20	PhiHat, 43
IllustrationFigPlotEffects, 25	SimDynrBootPara, 47
IllustrationFigScatterPlotCoverage,	SimDynrBootParaStdXMY, 48
25	SimDynrBootParaStdYMX, 49
IllustrationFigScatterPlotPower,	SimDynrBootParaXMY, 50
26	SimDynrBootParaYMX, 51

SimDynrDeltaStdXMY, 52	FitMx, 21
SimDynrDeltaStdYMX, 53	IllustrationFitDynr, 28
SimDynrDeltaXMY, 54	IllustrationFitMx, 28
SimDynrDeltaYMX, 55	SimFitDynr, 60
SimDynrMCStdXMY, 56	SimFitMx, 61
SimDynrMCStdYMX, 57	SimIllustrationFitDynr, 72
SimDynrMCXMY, 58	SimIllustrationFitMx, 73
SimDynrMCYMX, 59	* gendata
SimIllustrationDynrBootPara,64	GenData, 22
SimIllustrationDynrBootParaStdXMY,	IllustrationGenData, 29
65	IllustrationPrepData, 30
SimIllustrationDynrBootParaXMY,66	RandomMeasurement, 44
SimIllustrationDynrDeltaStdXMY, 67	SimGenData, 62
SimIllustrationDynrDeltaXMY, 68	SimIllustrationGenData, 74
SimIllustrationDynrMCPhiSigma, 69	* illustration
SimIllustrationDynrMCStdXMY, 70	illustration_dist, 31
SimIllustrationDynrMCXMY, 71	illustration_dist_dt, 32
ThetaHat, 91	illustration_dist_dt_mc, 32
* compress	illustration_dist_mc, 33
Compress, 11	illustration_dist_med, 33
IllustrationCompress, 24	illustration_dist_med_mc, 34
* data	illustration_dist_med_std, 34
illustration_dist, 31	illustration_dist_med_std_mc, 35
illustration_dist_dt, 32	illustration_results, 35
illustration_dist_dt_mc, 32	IllustrationBootPara, 23
illustration_dist_mc, 33	IllustrationFigScatterPlotCoverage,
illustration_dist_med, 33	25
illustration_dist_med_mc, 34	IllustrationFigScatterPlotPower,
illustration_dist_med_std, 34	26
illustration_dist_med_std_mc, 35	IllustrationFigScatterPlotSeBias,
illustration_results, 35	27
params, 43	IllustrationFitDynr, 28
results, 45	IllustrationFitMx, 28
* figure	IllustrationGenData, 29
FigPlotEffects, 16	IllustrationMCPhiSigma, 30
FigScatterPlotCoverage, 17	IllustrationPrepData, 30
FigScatterPlotPower, 18	SimIllustration, 63
FigScatterPlotSeBias, 19	SimIllustrationDynrBootPara, 64
FigScatterPlotType1, 20	SimIllustrationDynrBootParaStdXMY,
IllustrationFigPlotEffects, 25	65
IllustrationFigScatterPlotCoverage,	SimIllustrationDynrBootParaXMY,66
25	SimIllustrationDynrDeltaStdXMY, 67
IllustrationFigScatterPlotPower,	SimIllustrationDynrDeltaXMY, 68
26	SimIllustrationDynrMCPhiSigma, 69
IllustrationFigScatterPlotSeBias,	SimIllustrationDynrMCStdXMY, 70
27	SimIllustrationDynrMCXMY, 71
* fit	SimIllustrationFitDynr, 72
* M FitDynr, 21	SimIllustrationFitMx, 73
i i coyiii , 41	SIMITITUS CI ACTOMI I CINA, 13

SimIllustrationGenData, 74	MCStdXMY, 37
SimIllustrationPara, 75	MCStdXYM, 38
SumIllustration, 84	MCStdYMX, 39
SumIllustrationDynrBootParaStdXMY,	MCXMY, 40
85	MCXYM, 41
SumIllustrationDynrBootParaXMY, 86	MCYMX, 42
SumIllustrationDynrDeltaStdXMY, 87	PhiHat, 43
SumIllustrationDynrDeltaXMY,88	RandomMeasurement, 44
SumIllustrationDynrMCStdXMY,88	Sim, 46
SumIllustrationDynrMCXMY,89	SimDynrBootPara, 47
SumIllustrationFitDynr, 90	SimDynrBootParaStdXMY, 48
* manCTMed	SimDynrBootParaStdYMX, 49
BootPara, 4	SimDynrBootParaXMY, 50
BootParaStdXMY, 5	SimDynrBootParaYMX, 51
BootParaStdXYM, 6	SimDynrDeltaStdXMY, 52
BootParaStdYMX, 7	SimDynrDeltaStdYMX, 53
BootParaXMY, 8	SimDynrDeltaXMY, 54
BootParaXYM, 9	SimDynrDeltaYMX, 55
BootParaYMX, 10	SimDynrMCStdXMY, 56
Compress, 11	SimDynrMCStdYMX, 57
DeltaStdXMY, 12	SimDynrMCXMY, 58
DeltaStdXYM, 12	SimDynrMCYMX, 59
DeltaStdYMX, 13	SimFitDynr, 60
DeltaXMY, 14	SimFitMx, 61
DeltaXYM, 15	SimFN, 62
DeltaYMX, 16	SimGenData, 62
FigPlotEffects, 16	SimIllustration, 63
FigScatterPlotCoverage, 17	SimIllustrationDynrBootPara, 64
FigScatterPlotPower, 18	SimIllustrationDynrBootParaStdXMY
FigScatterPlotSeBias, 19	65
FigScatterPlotType1, 20	SimIllustrationDynrBootParaXMY, 66
FitDynr, 21	SimIllustrationDynrDeltaStdXMY, 67
FitMx, 21	SimIllustrationDynrDeltaXMY, $68$
GenData, 22	SimIllustrationDynrMCPhiSigma, 69
IllustrationBootPara, 23	${\sf SimIllustrationDynrMCStdXMY}, 70$
IllustrationCompress, 24	SimIllustrationDynrMCXMY, 71
IllustrationFigPlotEffects, 25	SimIllustrationFitDynr, $72$
${\tt Illustration Fig Scatter Plot Coverage},$	SimIllustrationFitMx, 73
25	${\sf SimIllustrationGenData}, 74$
$Illustration {\tt FigScatterPlotPower},$	SimIllustrationPara, 75
26	SimPara, 76
$Illustration {\tt FigScatterPlotSeBias},$	SimProj, 76
27	Sum, 77
IllustrationFitDynr, 28	SumDynrDeltaStdXMY, 77
IllustrationFitMx, 28	SumDynrDeltaStdYMX, 78
IllustrationGenData, 29	SumDynrDeltaXMY, 79
IllustrationMCPhiSigma, 30	SumDynrDeltaYMX, 80
IllustrationPrepData, 30	SumDynrMCStdXMY, 80

	SumDynrMCStdYMX, 81	* summary
	SumDynrMCXMY, 82	Sum, 77
	SumDynrMCYMX, 83	SumDynrDeltaStdXMY, 77
	SumFitDynr, 83	SumDynrDeltaStdYMX, 78
	SumIllustration, 84	SumDynrDeltaXMY, 79
	SumIllustrationDynrBootParaStdXMY,	SumDynrDeltaYMX, 80
	85	SumDynrMCStdXMY, 80
	SumIllustrationDynrBootParaXMY, 86	SumDynrMCStdYMX, 81
	SumIllustrationDynrDeltaStdXMY, 87	SumDynrMCXMY, 82
	SumIllustrationDynrDeltaXMY, 88	SumDynrMCYMX, 83
	SumIllustrationDynrMCStdXMY, 88	SumFitDynr, 83
	SumIllustrationDynrMCXMY, 89	SumIllustration, 84
	SumIllustrationFitDynr, 90	SumIllustrationDynrBootParaStdXMY,
	ThetaHat, 91	85
* n	arameters	SumIllustrationDynrBootParaXMY, 86
P	params, 43	SumIllustrationDynrDeltaStdXMY, 87
	results, 45	SumIllustrationDynrDeltaXMY, 88
∗ ci	mulation	SumIllustrationDynrMCStdXMY, 88
. 31	Sim, 46	SumIllustrationDynrMCXMY, 89
	SimDynrBootPara, 47	SumIllustrationFitDynr, 90
	SimDynrBootParaStdXMY, 48	•
	SimDynrBootParaStdYMX, 49	BootPara, 4, 5–10, 12–16, 23, 37–42
	SimDynrBootParaXMY, 50	BootPara(), $5-10$
	SimDynrBootParaYMX, 51	BootParaStdXMY, 5, 5, 6–10, 12–16, 23, 37–42
	SimDynrDeltaStdXMY, 52	BootParaStdXYM, 5, 6, 7–10, 12–16, 23, 37–42
	SimDynrDeltaStdYMX, 53	BootParaStdYMX, 5, 6, 7, 8–10, 12–16, 23,
	SimDynrDeltaXMY, 54	37–42
	SimDynrDeltaYMX, 55	BootParaXMY, 5-7, 8, 9, 10, 12-16, 23, 37-42
	SimDynrMCStdXMY, 56	BootParaXYM, 5-8, 9, 10, 12-16, 23, 37-42
	SimDynrMCStdYMX, 57	BootParaYMX, 5-9, 10, 12-16, 23, 37-42
	SimDynrMCXMY, 58	
	SimDynrMCYMX, 59	Compress, 11, 24
	SimFitDynr, 60	
	SimFitMx, 61	DeltaStdXMY, 5-10, 12, 13-16, 23, 37-42
	SimFN, 62	DeltaStdXYM, 5-10, 12, 12, 14-16, 23, 37-42
	SimGenData, 62	DeltaStdYMX, 5-10, 12, 13, 13, 14-16, 23,
	SimPara, 76	37–42
	SimProj, 76	DeltaXMY, 5-10, 12-14, 14, 15, 16, 23, 37-42
	Sum, 77	DeltaXYM, 5-10, 12-14, 15, 16, 23, 37-42
	SumDynrDeltaStdXMY, 77	DeltaYMX, 5-10, 12-15, 16, 23, 37-42
	SumDynrDeltaStdYMX, 78	dynr, 21, 28
	SumDynrDeltaXMY, 79	F: -D] -+Fff+- 16 10 20 25 27
		FigPlotEffects, 16, 18–20, 25–27
	SumDynrDeltaYMX, 80	FigScatterPlotCoverage, 17, 17, 18–20,
	SumDynrMCStdXMY, 80	25-27
	SumDynrMCStdYMX, 81	FigScatterPlotPower, 17, 18, 18, 19, 20,
	SumDynrMCVMV, 82	25-27  FigScattorPlatSoPice 17 18 10 20
	SumDynrMCYMX, 83	FigScatterPlotSeBias, 17, 18, 19, 20,
	SumFitDynr, 83	25–27

FigScatterPlotType1, <i>17–19</i> , 20, <i>25–27</i> FitDynr, 21, <i>22</i> , <i>28–31</i> , <i>43</i> , <i>91</i>	OpenMx, 21, 28
FitDynr(), 4, 43, 91	params, 43
FitMx, 21, 21, 28–31, 43, 91	PhiHat, 21, 22, 28-31, 43, 91
FitMx(), 4, 43, 91	PhiHat(), 8-10, 14-16, 40-42
GenData, 22, 29, 44	RandomMeasurement, 22, 29, 44
GenData(), 44	RandomMeasurement(), 21, 22
	results, 45
illustration_dist, 31	
illustration_dist_dt, 32	Sim, 46
<pre>illustration_dist_dt_mc, 32</pre>	SimDynrBootPara,47
illustration_dist_mc, 33	SimDynrBootParaStdXMY,48
illustration_dist_med, 33	SimDynrBootParaStdYMX, 49
illustration_dist_med_mc, 34	SimDynrBootParaXMY, 50
illustration_dist_med_std, 34	SimDynrBootParaYMX, 51
illustration_dist_med_std_mc, 35	SimDynrDeltaStdXMY, 52
illustration_results, 35	SimDynrDeltaStdYMX, 53
IllustrationBootPara, <i>5–10</i> , <i>12–16</i> , 23,	SimDynrDeltaXMY, 54
37–42	SimDynrDeltaYMX, 55
IllustrationCompress, 11, 24	SimDynrMCStdXMY, 56
IllustrationFigPlotEffects, 17–20, 25,	SimDynrMCStdYMX, 57
26, 27	SimDynrMCXMY, 58
IllustrationFigScatterPlotCoverage,	SimDynrMCYMX, 59
17–20, 25, 25, 26, 27	SimFitDynr, 60
IllustrationFigScatterPlotPower, 17–20,	SimFitMx, 61
25, 26, 26, 27	SimFN, 62
IllustrationFigScatterPlotSeBias,	SimGenData, 62
17–20, 25, 26, 27	SimIllustration, 63
IllustrationFitDynr, 21, 22, 28, 29–31, 43,	SimIllustrationDynrBootPara, 64
91	SimIllustrationDynrBootParaStdXMY, 65
IllustrationFitDynr(), 4, 43, 91	SimIllustrationDynrBootParaXMY, 66
	SimIllustrationDynrDeltaStdXMY, 67
IllustrationFitMx, 21, 22, 28, 28, 30, 31,	SimIllustrationDynrDeltaXMY, 68
43,91 IllustrationFitMx(),4,43,91	SimIllustrationDynrMCPhiSigma, 69
** * * *	SimIllustrationDynrMCStdXMY, 70
IllustrationGenData, 22, 29, 44	SimIllustrationDynrMCXMY, 71
IllustrationGenData(), 30, 31	SimillustrationFitDynr, 72
IllustrationMCPhiSigma, 21, 22, 28, 29, 30,	SimIllustrationFitMx, 73
31, 43, 91	SimillustrationFithx, 73 SimIllustrationGenData, 74
IllustrationPrepData, 21, 22, 28–30, 30,	SimillustrationPara, 75
43, 91	· · · · · · · · · · · · · · · · · · ·
IllustrationPrepData(), 28	SimPara, 76
MCC+JVMV 5 10 12 16 22 27 20 42	SimProj, 76
MCStdXMY, 5–10, 12–16, 23, 37, 38–42	simStateSpace::SimSSMOUFixed(), 22, 29
MCStdXYM, 5–10, 12–16, 23, 37, 38, 39–42	Sum, 77
MCStdYMX, 5–10, 12–16, 23, 37, 38, 39, 40–42	SumDynrDeltaStdXMY, 77
MCXMY, 5-10, 12-16, 23, 37-39, 40, 41, 42	SumDynrDeltaStdYMX, 78
MCXYM, 5–10, 12–16, 23, 37–40, 41, 42	SumDynrDeltaXMY, 79
MCYMX, 5–10, 12–16, 23, 37–41, 42	SumDynrDeltaYMX, 80

```
SumDynrMCStdXMY, 80
SumDynrMCStdYMX, 81
SumDynrMCXMY, 82
SumDynrMCYMX, 83
SumFitDynr, 83
SumIllustration, 84
SumIllustrationDynrBootParaStdXMY, 85
SumIllustrationDynrBootParaXMY, 86
SumIllustrationDynrDeltaStdXMY, 87
SumIllustrationDynrDeltaXMY, 88
SumIllustrationDynrMCStdXMY, 88
SumIllustrationDynrMCStdXMY, 88
SumIllustrationDynrMCXMY, 89
SumIllustrationFitDynr, 90
ThetaHat, 21, 22, 28-31, 43, 91
ThetaHat(), 5-7, 12, 13, 37-39
```