Package 'rsimsum'

June 20, 2018

Version 0.3.3

Title Analysis of Simulation Studies Including Monte Carlo Error

Description Summarise results from simulation studies and compute Monte Carlo standard errors of commonly used summary statistics. This package is modelled on the 'simsum' user-written command in 'Stata' (See White I.R., 2010 <http://www.stata-journal.com/article.html?article=st0200>).

License GPL (>= 3)

Depends R (>= 2.10)

Imports checkmate, ggplot2, stats

Suggests covr, devtools, dplyr, eha, knitr, rmarkdown, rstpm2, survival, testthat

URL https://github.com/ellessenne/rsimsum

BugReports https://github.com/ellessenne/rsimsum/issues

VignetteBuilder knitr

RoxygenNote 6.0.1

LazyData true

ByteCompile true

NeedsCompilation no

Author Alessandro Gasparini [aut, cre] (<https://orcid.org/0000-0002-8319-7624>), Ian R. White [aut]

Maintainer Alessandro Gasparini <ag475@leicester.ac.uk>

Repository CRAN

Date/Publication 2018-06-20 14:03:01 UTC

R topics documented:

`bar` *Draw bar plots.*

Description

The S3 method `bar()` produces a bar plot to visually summarise the results of a simulation study analysed using `simsum()`.

Usage

```
bar(obj, ...)
```

Arguments

`obj`	An object used to select a method.
`...`	Further arguments passed to or from other methods.

Details

Coverage, bias corrected coverage, and power will be based on the α defined in the call to `simsum`.

Value

A `ggplot2::ggplot()` object that can be combined with additional `geom_*`, `scale_*`, `theme_*`, etc.

`bar.multisimsum` *forest method for multisimsum objects*

Description

`bar()` method for objects of class `multisimsum`.

Usage

```
## S3 method for class 'multisimsum'
bar(obj, sstat, par = NULL, by = NULL,
  target = NULL, level = 0.95, gpars = list(), ...)
```

Arguments

obj	An object of class multisimsum.
sstat	Summary statistic to plot. Possible choices are: nsim, number of replications without missing estimates / standard errors; thetamean, average estimated value; thetamedian, median estimated value; se2mean, average estimated standard error; se2median, median estimated standard error; bias, bias in point estimate; empse, empirical standard error; mse, mean squared error; relprec, percentage gain in precision relative to the reference method; modelse, model-based standard error; relerror, relative percentage error in standard error; cover, coverage of nominal $(1-\alpha)\%$ CI; bccover, bias corrected coverage of nominal $(1-\alpha)\%$ CI; power, power of $\alpha\%$ level test.
par	Estimand to plot. Defaults to NULL, in which case the par variable from multisimsum will be used for faceting.
by	Faceting factors passed to ggplot2::facet_wrap(). Defaults to NULL, i.e. no faceting.
target	Target value for the summary statistic of interest. If NULL (the default), the target value is inferred (except for sstat = nsim).
level	Specifies the confidence level for confidence intervals based on Monte Carlo standard errors, produced by default if the simsum or summary.simsum object passed to bar estimated Monte Carlo standard errors (e.g. with mcse = TRUE).
gpars	Graphical parameters. Must be a named list, with possible parameters: • target.shape, shape of the horizontal line at target value; • target.colour, colour of the horizontal line at target value; • bar.colour, colour of each bar; • bar.fill, fill of each bar; • width, the width of the end of each confidence interval. It is possible to redefine all the graphical parameters of a subset only; if not specified, sensible default values will be utilised.
...	Ignored.

Details

Coverage, bias corrected coverage, and power will be based on the α defined in the call to simsum.

Value

A ggplot2::ggplot() object that can be combined with additional geom_*, scale_*, theme_*, etc.

Examples

```
library(rsimsum)
library(ggplot2)
data("frailty", package = "rsimsum")
ms <- multisimsum(data = frailty, par = "par", true = c(trt = -0.50,
   fv = 0.75), estvarname = "b", se = "se", methodvar = "model",
```

```
   by = "fv_dist")
bar(ms, sstat = "bias", par = "trt", by = "fv_dist")
```

`bar.simsum` *bar method for simsum objects*

Description

`bar()` method for objects of class `simsum`.

Usage

```
## S3 method for class 'simsum'
bar(obj, sstat, by = NULL, target = NULL, level = 0.95,
  gpars = list(), ...)
```

Arguments

`obj`	An object of class `simsum`.
`sstat`	Summary statistic to plot. Possible choices are: `nsim`, number of replications without missing estimates / standard errors; `thetamean`, average estimated value; `thetamedian`, median estimated value; `se2mean`, average estimated standard error; `se2median`, median estimated standard error; `bias`, bias in point estimate; `empse`, empirical standard error; `mse`, mean squared error; `relprec`, percentage gain in precision relative to the reference method; `modelse`, model-based standard error; `relerror`, relative percentage error in standard error; `cover`, coverage of nominal $(1-\alpha)\%$ CI; `bccover`, bias corrected coverage of nominal $(1-\alpha)\%$ CI; `power`, power of $\alpha\%$ level test.
`by`	Faceting factors passed to `ggplot2::facet_wrap()`. Defaults to `NULL`, i.e. no faceting.
`target`	Target value for the summary statistic of interest. If `NULL` (the default), the target value is inferred (except for `sstat = nsim`).
`level`	Specifies the confidence level for confidence intervals based on Monte Carlo standard errors, produced by default if the `simsum` or `summary.simsum` object passed to `bar` estimated Monte Carlo standard errors (e.g. with `mcse = TRUE`).
`gpars`	Graphical parameters. Must be a named list, with possible parameters: • `target.shape`, shape of the horizontal line at `target` value; • `target.colour`, colour of the horizontal line at `target` value; • `bar.colour`, colour of each bar; • `bar.fill`, fill of each bar; • `width`, the width of the end of each confidence interval. It is possible to redefine all the graphical parameters of a subset only; if not specified, sensible default values will be utilised.
`...`	Ignored.

Details

Coverage, bias corrected coverage, and power will be based on the α defined in the call to `simsum`.

Value

A `ggplot2::ggplot()` object that can be combined with additional `geom_*`, `scale_*`, `theme_*`, etc.

Examples

```
library(rsimsum)
library(ggplot2)
data("relhaz", package = "rsimsum")
s <- simsum(data = relhaz, estvarname = "theta", true = -0.5, se = "se",
  methodvar = "model", by = c("n", "baseline"))
bar(s, sstat = "bias", by = c("n", "baseline"))
```

---

dropbig *Return observations dropped by a function*

Description

`dropbig` returns observations dropped by a function while performing some calculation - for any reason.

Usage

```
dropbig(x, ...)
```

Arguments

`x`	An object used to select a method.
`...`	Further arguments passed to or from other methods.

---

dropbig.multisimsum *Return observations dropped by simsum*

Description

`dropbig.multisimsum` returns observations dropped by simsum while computing summary statistics for a simulation study.

Usage

```
## S3 method for class 'multisimsum'
dropbig(x, ...)
```

Arguments

x	An object of class multisimsum.
...	Ignored.

Value

An object of class dropbig.multisimsum if dropbig was set to TRUE when calling multisimsum, NULL otherwise. It contains two slots: big_estvarname and big_se. Both slots consist in the portion of the original data that was identified to have estimated values or standard errors exceeding the limits defined by max and semax.

See Also

multisimsum(), print.dropbig.multisimsum()

Examples

```
data(frailty)
ms <- multisimsum(data = frailty, par = "par", true = c(trt = -0.50,
   fv = 0.75), estvarname = "b", se = "se", methodvar = "model",
   by = "fv_dist", dropbig = TRUE, max = 6, semax = 3)
dropbig(ms)
```

---

dropbig.simsum *Return observations dropped by simsum*

Description

dropbig.simsum returns observations dropped by simsum while computing summary statistics for a simulation study.

Usage

```
## S3 method for class 'simsum'
dropbig(x, ...)
```

Arguments

x	An object of class simsum.
...	Ignored.

Value

An object of class dropbig.simsum if dropbig was set to TRUE when calling simsum, NULL otherwise. It contains two slots: big_estvarname and big_se. Both slots consist in the portion of the original data that was identified to have estimated values or standard errors exceeding the limits defined by max and semax.

See Also

simsum(), print.dropbig.simsum()

Examples

```
data("MIsim")
x <- simsum(data = MIsim, estvarname = "b", true = 0.5, se = "se",
            methodvar = "method", mcse = TRUE,
dropbig = TRUE, max = 3, semax = 1.5)
d <- dropbig(x)
d
```

---

forest *Draw forest plots.*

Description

The S3 method forest() produces a forest plot to visually summarise the results of a simulation study analysed using simsum().

Usage

```
forest(obj, ...)
```

Arguments

obj	An object used to select a method.
...	Further arguments passed to or from other methods.

Details

Coverage, bias corrected coverage, and power will be based on the α defined in the call to simsum.

Value

A ggplot2::ggplot() object that can be combined with additional geom_*, scale_*, theme_*, etc.

forest.multisimsum *forest method for multisimsum objects*

Description

forest() method for objects of class multisimsum.

Usage

```
## S3 method for class 'multisimsum'
forest(obj, sstat, par = NULL, by = NULL,
  target = NULL, level = 0.95, gpars = list(), ...)
```

Arguments

obj	An object of class multisimsum.
sstat	Summary statistic to plot. Possible choices are: nsim, number of replications without missing estimates / standard errors; thetamean, average estimated value; thetamedian, median estimated value; se2mean, average estimated standard error; se2median, median estimated standard error; bias, bias in point estimate; empse, empirical standard error; mse, mean squared error; relprec, percentage gain in precision relative to the reference method; modelse, model-based standard error; relerror, relative percentage error in standard error; cover, coverage of nominal $(1-\alpha)\%$ CI; bccover, bias corrected coverage of nominal $(1-\alpha)\%$ CI; power, power of $\alpha\%$ level test.
par	Estimand to plot. Defaults to NULL, in which case the par variable from multisimsum will be used for faceting.
by	Faceting factors passed to ggplot2::facet_wrap(). Defaults to NULL, i.e. no faceting.
target	Target value for the summary statistic of interest. If NULL (the default), the target value is inferred (except for sstat = nsim).
level	Specifies the confidence level for confidence intervals based on Monte Carlo standard errors, produced by default if the simsum or summary.simsum object passed to forest estimated Monte Carlo standard errors (e.g. with mcse = TRUE).
gpars	Graphical parameters. Must be a named list, with possible parameters: • target.shape, shape of the horizontal line at target value; • target.colour, colour of the horizontal line at target value; • width, the width of the end of each confidence interval. It is possible to redefine all the graphical parameters of a subset only; if not specified, sensible default values will be utilised.
...	Ignored.

Details

Coverage, bias corrected coverage, and power will be based on the α defined in the call to simsum.

Value

A ggplot2::ggplot() object that can be combined with additional geom_*, scale_*, theme_*, etc.

Examples

```
library(rsimsum)
library(ggplot2)
data("frailty", package = "rsimsum")
ms <- multisimsum(data = frailty, par = "par", true = c(trt = -0.50,
   fv = 0.75), estvarname = "b", se = "se", methodvar = "model",
   by = "fv_dist")
forest(ms, sstat = "bias", par = "trt", by = "fv_dist")
```

---

forest.simsum *forest method for simsum objects*

Description

forest() method for objects of class simsum.

Usage

```
## S3 method for class 'simsum'
forest(obj, sstat, by = NULL, target = NULL,
  level = 0.95, gpars = list(), ...)
```

Arguments

obj	An object of class simsum.
sstat	Summary statistic to plot. Possible choices are: nsim, number of replications without missing estimates / standard errors; thetamean, average estimated value; thetamedian, median estimated value; se2mean, average estimated standard error; se2median, median estimated standard error; bias, bias in point estimate; empse, empirical standard error; mse, mean squared error; relprec, percentage gain in precision relative to the reference method; modelse, model-based standard error; relerror, relative percentage error in standard error; cover, coverage of nominal $(1-\alpha)\%$ CI; bccover, bias corrected coverage of nominal $(1-\alpha)\%$ CI; power, power of $\alpha\%$ level test.
by	Faceting factors passed to ggplot2::facet_wrap(). Defaults to NULL, i.e. no faceting.
target	Target value for the summary statistic of interest. If NULL (the default), the target value is inferred (except for sstat = nsim).
level	Specifies the confidence level for confidence intervals based on Monte Carlo standard errors, produced by default if the simsum or summary.simsum object passed to forest estimated Monte Carlo standard errors (e.g. with mcse = TRUE).

gpars	Graphical parameters. Must be a named list, with possible parameters: • target.shape, shape of the horizontal line at target value; • target.colour, colour of the horizontal line at target value; • width, the width of the end of each confidence interval. It is possible to redefine all the graphical parameters of a subset only; if not specified, sensible default values will be utilised.
...	Ignored.

Details

Coverage, bias corrected coverage, and power will be based on the α defined in the call to simsum.

Value

A ggplot2::ggplot() object that can be combined with additional geom_*, scale_*, theme_*, etc.

Examples

```
library(rsimsum)
library(ggplot2)
data("relhaz", package = "rsimsum")
s <- simsum(data = relhaz, estvarname = "theta", true = -0.5, se = "se",
  methodvar = "model", by = c("n", "baseline"))
forest(s, sstat = "bias", by = c("n", "baseline"))
```

---

format.multisimsum *Format multisimsum objects*

Description

Format an object of class multisimsum for pretty printing.

Usage

```
## S3 method for class 'multisimsum'
format(x, digits)
```

Arguments

x	An object of class multisimsum.
digits	Number of significant digits. Defaults to 4.

Value

An object of class multisimsum with its summ slot formatted for pretty printing.

See Also

multisimsum(), format.summary.multisimsum()

---

format.simsum *Format simsum objects*

Description

Format an object of class simsum for pretty printing.

Usage

```
## S3 method for class 'simsum'
format(x, digits)
```

Arguments

x	An object of class simsum.
digits	Number of significant digits. Defaults to 4.

Value

An object of class simsum with its summ slot formatted for pretty printing.

See Also

simsum(), format.summary.simsum()

---

format.summary.multisimsum
Format summary.multisimsum objects

Description

Format an object of class summary.multisimsum for pretty printing.

Usage

```
## S3 method for class 'summary.multisimsum'
format(x, digits)
```

Arguments

x	An object of class summary.multisimsum.
digits	Number of significant digits. Defaults to 4.

Details

format.summary.multisimsum

Value

An object of class `summary.multisimsum` with its `summ` slot formatted for pretty printing.

See Also

`multisimsum()`, `format.multisimsum()`

`format.summary.simsum` *Format summary.simsum objects*

Description

Format an object of class `summary.simsum` for pretty printing.

Usage

```
## S3 method for class 'summary.simsum'
format(x, digits)
```

Arguments

`x`	An object of class `summary.simsum`.
`digits`	Number of significant digits. Defaults to 4.

Value

An object of class `summary.simsum` with its `summ` slot formatted for pretty printing.

See Also

`simsum()`, `format.simsum()`

frailty *Example of a simulation study on frailty survival models*

Description

A dataset from a simulation study comparing frailty flexible parametric models fitted using penalised likelihood to semiparametric frailty models. Both models are fitted assuming a Gamma and a log-Normal frailty. One thousand datasets were simulated, each containing a binary treatment variable with a log-hazard ratio of -0.50. Clustered survival data was simulated assuming 50 clusters of 50 individuals each, with a mixture Weibull baseline hazard function and a frailty following either a Gamma or a Log-Normal distribution. The comparison involves estimates of the log-treatment effect, and estimates of heterogeneity (i.e. the estimated frailty variance).

Usage

```
frailty
```

Format

A data frame with 16,000 rows and 6 variables:

- `i` Simulated dataset number.
- `b` Point estimate.
- `se` Standard error of the point estimate.
- `par` The estimand. `trt` is the log-treatment effect, `fv` is the variance of the frailty.
- `fv_dist` The true frailty distribution.
- `model` Method used (`Cox, Gamma`, `Cox, Log-Normal`, `RP(P), Gamma`, or `RP(P), Log-Normal`).

Examples

```
data("frailty", package = "rsimsum")
```

get_data *Extract data slot from objects*

Description

`get_data` returns a data set contained in an S3 object.

Usage

```
get_data(x, ...)
```

Arguments

`x`	An object used to select a method.
`...`	Further arguments passed to or from other methods.

get_data.miss *get_data.miss*

Description

Extract data from a miss object

Usage

```
## S3 method for class 'miss'
get_data(x, ...)
```

Arguments

x	An object of class miss.
...	Ignored.

Value

A data.frame with estimates and standard errors substituted for a missingness boolean value.

See Also

miss(), get_data()

Examples

```
data("frailty", package = "rsimsum")
m <- miss(data = frailty, estvarname = "b", se = "se", par = "par",
            methodvar = "model", by = "fv_dist")
get_data(m)
```

get_data.multisimsum *get_data.multisimsum*

Description

Extract data from a multisimsum object

Usage

```
## S3 method for class 'multisimsum'
get_data(x, sstat = "all", description = FALSE, ...)
```

Arguments

x	An object of class multisimsum.
sstat	Summary statistics to include; can be a scalar value or a vector. Possible choices are: • all, all the summary statistics are returned This is the default option. • nsim, the number of replications with non-missing point estimates and standard error. • thetamean, average point estimate. • thetamedian, median point estimate. • se2mean, average standard error. • se2median, median standard error. • bias, bias in point estimate. • empse, empirical standard error. • mse, mean squared error. • relprec, percentage gain in precision relative to the reference method. • modelse, model-based standard error. • relerror, relative percentage error in standard error. • cover, coverage of a nominal level% confidence interval. • bccover, bias corrected coverage of a nominal level% confidence interval. • power, power of a (1 - level)% level test.#'
description	Append a column to the returned data.frame with a description of each summary statistic? Defaults to FALSE.
...	Ignored.

Value

A data.frame containing summary statistics from a simulation study.

See Also

multisimsum(), get_data()

Examples

```
data(frailty)
ms <- multisimsum(data = frailty, par = "par", true = c(trt = -0.50,
   fv = 0.75), estvarname = "b", se = "se", methodvar = "model",
   by = "fv_dist")
get_data(ms)
get_data(ms, description = TRUE)
```

get_data.simsum *get_data.simsum*

Description

Extract data from a simsum object

Usage

```
## S3 method for class 'simsum'
get_data(x, sstat = "all", description = FALSE, ...)
```

Arguments

x	An object of class simsum.
sstat	Summary statistics to include; can be a scalar value or a vector. Possible choices are: • all, all the summary statistics are returned This is the default option. • nsim, the number of replications with non-missing point estimates and standard error. • thetamean, average point estimate. • thetamedian, median point estimate. • se2mean, average standard error. • se2median, median standard error. • bias, bias in point estimate. • empse, empirical standard error. • mse, mean squared error. • relprec, percentage gain in precision relative to the reference method. • modelse, model-based standard error. • relerror, relative percentage error in standard error. • cover, coverage of a nominal level% confidence interval. • bccover, bias corrected coverage of a nominal level% confidence interval. • power, power of a (1 - level)% level test.#'
description	Append a column to the returned data.frame with a description of each summary statistic? Defaults to FALSE.
...	Ignored.

Value

A data.frame containing summary statistics from a simulation study.

See Also

simsum(), get_data()

Examples

```
data(MIsim)
x <- simsum(data = MIsim, estvarname = "b", true = 0.5, se = "se",
            methodvar = "method", mcse = TRUE)
get_data(x)

# Exporting only bias and coverage:
get_data(x, ssta = c("bias", "cover"))

# Including a description of the summary statistics being exported:
get_data(x, ssta = c("bias", "cover"), description = TRUE)
```

`get_data.summary.multisimsum`
get_data.summary.multisimsum

Description

Extract data from a summary.simsum object

Usage

```
## S3 method for class 'summary.multisimsum'
get_data(x, sstat = "all",
  description = FALSE, ...)
```

Arguments

`x`	An object of class `summary.multisimsum`.
`sstat`	Summary statistics to include; can be a scalar value or a vector. Possible choices are:

- `all`, all the summary statistics are returned This is the default option.
- `nsim`, the number of replications with non-missing point estimates and standard error.
- `thetamean`, average point estimate.
- `thetamedian`, median point estimate.
- `se2mean`, average standard error.
- `se2median`, median standard error.
- `bias`, bias in point estimate.
- `empse`, empirical standard error.
- `mse`, mean squared error.
- `relprec`, percentage gain in precision relative to the reference method.
- `modelse`, model-based standard error.
- `relerror`, relative percentage error in standard error.
- `cover`, coverage of a nominal `level`% confidence interval.

- bccover, bias corrected coverage of a nominal level% confidence interval.
- power, power of a (1 - level)% level test.#'

description — Append a column to the returned data.frame with a description of each summary statistic? Defaults to FALSE.

... — Ignored.

Value

A data.frame containing summary statistics from a simulation study.

See Also

summary.multisimsum(), get_data()

Examples

```
data(frailty)
ms <- multisimsum(data = frailty, par = "par", true = c(trt = -0.50,
   fv = 0.75), estvarname = "b", se = "se", methodvar = "model",
   by = "fv_dist")
sms <- summary(ms)
get_data(sms)
get_data(sms, description = TRUE)
```

---

get_data.summary.simsum

get_data.summary.simsum

Description

Extract data from a summary.simsum object

Usage

```
## S3 method for class 'summary.simsum'
get_data(x, sstat = "all", description = FALSE,
  ...)
```

Arguments

x — An object of class summary.simsum.

sstat — Summary statistics to include; can be a scalar value or a vector. Possible choices are:

- all, all the summary statistics are returned This is the default option.
- nsim, the number of replications with non-missing point estimates and standard error.
- thetamean, average point estimate.

- thetamedian, median point estimate.
- se2mean, average standard error.
- se2median, median standard error.
- bias, bias in point estimate.
- empse, empirical standard error.
- mse, mean squared error.
- relprec, percentage gain in precision relative to the reference method.
- modelse, model-based standard error.
- relerror, relative percentage error in standard error.
- cover, coverage of a nominal level% confidence interval.
- bccover, bias corrected coverage of a nominal level% confidence interval.
- power, power of a (1 - level)% level test.#'

description — Append a column to the returned data.frame with a description of each summary statistic? Defaults to FALSE.

... — Ignored.

Value

A data.frame containing summary statistics from a simulation study.

See Also

summary.simsum(), get_data()

Examples

```
data(MIsim)
x <- simsum(data = MIsim, estvarname = "b", true = 0.5, se = "se",
            methodvar = "method", mcse = TRUE)
xs <- summary(x)
get_data(xs)

# Exporting only bias and coverage:
get_data(xs, sstat = c("bias", "cover"))

# Append a column with a description of each summary statistic exported:
get_data(xs, sstat = c("bias", "cover"), description = TRUE)
```

---

heat *Draw heat plots.*

Description

The S3 method heat() produces a heat plot to visually summarise the results of a simulation study analysed using simsum().

Usage

```
heat(obj, ...)
```

Arguments

obj	An object used to select a method.
...	Further arguments passed to or from other methods.

Details

Coverage, bias corrected coverage, and power will be based on the α defined in the call to simsum.

Value

A ggplot2::ggplot() object that can be combined with additional geom_*, scale_*, theme_*, etc.

---

heat.multisimsum *heat method for simsum objects*

Description

heat() method for objects of class simsum.

Usage

```
## S3 method for class 'multisimsum'
heat(obj, par, sstat, y, target = NULL, text = FALSE,
  gpars = list(), ...)
```

Arguments

obj	An object of class simsum.
par	Estimand to plot.
sstat	Summary statistic to plot. Possible choices are: nsim, number of replications without missing estimates / standard errors; thetamean, average estimated value; thetamedian, median estimated value; se2mean, average estimated standard error; se2median, median estimated standard error; bias, bias in point estimate; empse, empirical standard error; mse, mean squared error; relprec, percentage gain in precision relative to the reference method; modelse, model-based standard error; relerror, relative percentage error in standard error; cover, coverage of nominal $(1-\alpha)\%$ CI; bccover, bias corrected coverage of nominal $(1-\alpha)\%$ CI; power, power of $\alpha\%$ level test.
y	Factor on the y-axis of the plot.
target	Target value for the summary statistic of interest. If NULL (the default), the target value is inferred (except for sstat = nsim).

`text` Including estimates and Monte Carlo standard errors in the plot? Defaults to `FALSE`.

`gpars` Graphical parameters. Must be a named list, with possible parameters:

- `target.colour`, colour representing the target value;
- `low.colour`, colour representing the lowest value;
- `high.colour`, colour representing the highest value;
- `fmt`, format string passed to `base::sprintf()` to format values and Monte Carlo standard errors.

It is possible to redefine all the graphical parameters of a subset only; if not specified, sensible default values will be utilised. The default colours are colorblind-friendly.

`...` Ignored.

Details

Coverage, bias corrected coverage, and power will be based on the α defined in the call to `simsum`.

Value

A `ggplot2::ggplot()` object that can be combined with additional `geom_*`, `scale_*`, `theme_*`, etc.

Examples

```
library(rsimsum)
library(ggplot2)
data("relhaz", package = "rsimsum")
s <- simsum(
  data = relhaz, estvarname = "theta", true = -0.5, se = "se",
  methodvar = "model", by = c("n", "baseline")
)
heat(s, sstat = "bias", y = "baseline")
library(rsimsum)
library(ggplot2)
data("frailty", package = "rsimsum")
ms <- multisimsum(data = frailty, par = "par", true = c(trt = -0.50,
   fv = 0.75), estvarname = "b", se = "se", methodvar = "model",
   by = "fv_dist")
heat(ms, sstat = "bias", par = "trt", y = "fv_dist", x = TRUE)
```

`heat.simsum` *heat method for simsum objects*

Description

`heat()` method for objects of class `simsum`.

Usage

```
## S3 method for class 'simsum'
heat(obj, sstat, y, target = NULL, text = FALSE,
  gpars = list(), ...)
```

Arguments

obj	An object of class simsum.
sstat	Summary statistic to plot. Possible choices are: nsim, number of replications without missing estimates / standard errors; thetamean, average estimated value; thetamedian, median estimated value; se2mean, average estimated standard error; se2median, median estimated standard error; bias, bias in point estimate; empse, empirical standard error; mse, mean squared error; relprec, percentage gain in precision relative to the reference method; modelse, model-based standard error; relerror, relative percentage error in standard error; cover, coverage of nominal $(1-\alpha)\%$ CI; bccover, bias corrected coverage of nominal $(1-\alpha)\%$ CI; power, power of $\alpha\%$ level test.
y	Factor on the y-axis of the plot.
target	Target value for the summary statistic of interest. If NULL (the default), the target value is inferred (except for sstat = nsim).
text	Including estimates and Monte Carlo standard errors in the plot? Defaults to FALSE.
gpars	Graphical parameters. Must be a named list, with possible parameters: • target.colour, colour representing the target value; • low.colour, colour representing the lowest value; • high.colour, colour representing the highest value; • fmt, format string passed to base::sprintf() to format values and Monte Carlo standard errors. It is possible to redefine all the graphical parameters of a subset only; if not specified, sensible default values will be utilised. The default colours are colorblind-friendly.
...	Ignored.

Details

Coverage, bias corrected coverage, and power will be based on the α defined in the call to simsum.

Value

A ggplot2::ggplot() object that can be combined with additional geom_*, scale_*, theme_*, etc.

Examples

```
library(rsimsum)
library(ggplot2)
data("relhaz", package = "rsimsum")
s <- simsum(
  data = relhaz, estvarname = "theta", true = -0.5, se = "se",
  methodvar = "model", by = c("n", "baseline")
)
heat(s, sstat = "bias", y = "baseline")
heat(s, sstat = "bias", y = "n", text = TRUE)
```

`is.multisimsum` *is.multisimsum*

Description

Reports whether x is a multisimsum object

Usage

```
is.multisimsum(x)
```

Arguments

x	An object to test.

`is.simsum` *is.simsum*

Description

Reports whether x is a simsum object

Usage

```
is.simsum(x)
```

Arguments

x	An object to test.

lolly *Draw lolly plots.*

Description

The S3 method `lolly()` produces a lolly plot to visually summarise the results of a simulation study analysed using `simsum()`.

Usage

```
lolly(obj, ...)
```

Arguments

`obj`	An object used to select a method.
`...`	Further arguments passed to or from other methods.

Details

Coverage, bias corrected coverage, and power will be based on the α defined in the call to `simsum`.

Value

A `ggplot2::ggplot()` object that can be combined with additional `geom_*`, `scale_*`, `theme_*`, etc.

lolly.multisimsum *lolly method for multisimsum objects*

Description

`lolly()` method for objects of class `multisimsum`.

Usage

```
## S3 method for class 'multisimsum'
lolly(obj, sstat, par = NULL, by = NULL,
  target = NULL, level = 0.95, gpars = list(), ...)
```

Arguments

obj	An object of class multisimsum.
sstat	Summary statistic to plot. Possible choices are: nsim, number of replications without missing estimates / standard errors; thetamean, average estimated value; thetamedian, median estimated value; se2mean, average estimated standard error; se2median, median estimated standard error; bias, bias in point estimate; empse, empirical standard error; mse, mean squared error; relprec, percentage gain in precision relative to the reference method; modelse, model-based standard error; relerror, relative percentage error in standard error; cover, coverage of nominal $(1-\alpha)\%$ CI; bccover, bias corrected coverage of nominal $(1-\alpha)\%$ CI; power, power of $\alpha\%$ level test.
par	Estimand to plot. Defaults to NULL, in which case the par variable from multisimsum will be used for faceting.
by	Faceting factors passed to ggplot2::facet_wrap(). Defaults to NULL, i.e. no faceting.
target	Target value for the summary statistic of interest. If NULL (the default), the target value is inferred (except for sstat = nsim).
level	Specifies the confidence level for confidence intervals based on Monte Carlo standard errors, produced by default if the simsum or summary.simsum object passed to lolly estimated Monte Carlo standard errors (e.g. with mcse = TRUE).
gpars	Graphical parameters. Must be a named list, with possible parameters: • target.shape, shape of the vertical line at target value; • target.colour, colour of the vertical line at target value; • segment.shape, shape of the horizontal segment between target and estimated value; • segment.colour, colour of the horizontal segment between target and estimated value; • low.end, shape of the lower end of the confidence interval; • upp.end, shape of the upper end of the confidence interval. It is possible to redefine all the graphical parameters of a subset only; if not specified, sensible default values will be utilised.
...	Ignored.

Details

Coverage, bias corrected coverage, and power will be based on the α defined in the call to simsum.

Value

A ggplot2::ggplot() object that can be combined with additional geom_*, scale_*, theme_*, etc.

Examples

```
library(rsimsum)
library(ggplot2)
data("frailty", package = "rsimsum")
ms <- multisimsum(data = frailty, par = "par", true = c(trt = -0.50,
   fv = 0.75), estvarname = "b", se = "se", methodvar = "model",
   by = "fv_dist")
lolly(ms, sstat = "bias", par = "trt", by = "fv_dist")
```

`lolly.simsum` *lolly method for simsum objects*

Description

`lolly()` method for objects of class `simsum`.

Usage

```
## S3 method for class 'simsum'
lolly(obj, sstat, by = NULL, target = NULL, level = 0.95,
  gpars = list(), ...)
```

Arguments

`obj`	An object of class `simsum`.
`sstat`	Summary statistic to plot. Possible choices are: `nsim`, number of replications without missing estimates / standard errors; `thetamean`, average estimated value; `thetamedian`, median estimated value; `se2mean`, average estimated standard error; `se2median`, median estimated standard error; `bias`, bias in point estimate; `empse`, empirical standard error; `mse`, mean squared error; `relprec`, percentage gain in precision relative to the reference method; `modelse`, model-based standard error; `relerror`, relative percentage error in standard error; `cover`, coverage of nominal $(1-\alpha)\%$ CI; `bccover`, bias corrected coverage of nominal $(1-\alpha)\%$ CI; `power`, power of $\alpha\%$ level test.
`by`	Faceting factors passed to `ggplot2::facet_wrap()`. Defaults to `NULL`, i.e. no faceting.
`target`	Target value for the summary statistic of interest. If `NULL` (the default), the target value is inferred (except for `sstat = nsim`).
`level`	Specifies the confidence level for confidence intervals based on Monte Carlo standard errors, produced by default if the `simsum` or `summary.simsum` object passed to `lolly` estimated Monte Carlo standard errors (e.g. with `mcse = TRUE`).
`gpars`	Graphical parameters. Must be a named list, with possible parameters: • `target.shape`, shape of the vertical line at `target` value; • `target.colour`, colour of the vertical line at `target` value;

- segment.shape, shape of the horizontal segment between target and estimated value;
- segment.colour, colour of the horizontal segment between target and estimated value;
- low.end, shape of the lower end of the confidence interval;
- upp.end, shape of the upper end of the confidence interval. It is possible to redefine all the graphical parameters of a subset only; if not specified, sensible default values will be utilised.

... Ignored.

Details

Coverage, bias corrected coverage, and power will be based on the α defined in the call to simsum.

Value

A ggplot2::ggplot() object that can be combined with additional geom_*, scale_*, theme_*, etc.

Examples

```
library(rsimsum)
library(ggplot2)
data("relhaz", package = "rsimsum")
s <- simsum(data = relhaz, estvarname = "theta", true = -0.5, se = "se",
  methodvar = "model", by = c("n", "baseline"))
lolly(s, sstat = "bias", by = c("n", "baseline"))
```

MIsim *Example of a simulation study on missing data*

Description

A dataset from a simulation study comparing different ways to handle missing covariates when fitting a Cox model (White and Royston, 2009). One thousand datasets were simulated, each containing normally distributed covariates x and z and time-to-event outcome. Both covariates has 20% of their values deleted independently of all other variables so the data became missing completely at random (Little and Rubin, 2002). Each simulated dataset was analysed in three ways. A Cox model was fit to the complete cases (CC). Then two methods of multiple imputation using chained equations (van Buuren, Boshuizen, and Knook, 1999) were used. The MI_LOGT method multiply imputes the missing values of x and z with the outcome included as $\log(t)$ and d, where t is the survival time and d is the event indicator. The MI_T method is the same except that $\log(t)$ is replaced by t in the imputation model. The results are stored in long format.

Usage

```
MIsim
```

Format

A data frame with 3,000 rows and 4 variables:

- `dataset` Simulated dataset number.
- `method` Method used (`CC`, `MI_LOGT` or `MI_T`).
- `b` Point estimate.
- `se` Standard error of the point estimate.

References

White, I.R., and P. Royston. 2009. Imputing missing covariate values for the Cox model. Statistics in Medicine 28(15):1982-1998 doi: 10.1002/sim.3618

Little, R.J.A., and D.B. Rubin. 2002. Statistical analysis with missing data. 2nd ed. Hoboken, NJ: Wiley doi: 10.1002/9781119013563

Examples

```
data("MIsim", package = "rsimsum")
```

`miss` *miss*

Description

Obtain basic information on the proportion of missing data within the results of a simulation study.

Usage

```
miss(data, estvarname, se, par = NULL, methodvar = NULL, by = NULL)
```

Arguments

`data`	A `data.frame` in which variable names are interpreted. It has to be in tidy format, e.g. each variable forms a column and each observation forms a row.
`estvarname`	The name of the variable containing the point estimates.
`se`	The name of the variable containing the standard errors of the point estimates.
`par`	The name of the variable containing the different estimands from the simulation study. Can be `NULL`.
`methodvar`	The name of the variable containing the methods to compare. Can be `NULL`.
`by`	A vector of variable names to compute performance measures by a list of factors. Can be `NULL`.

Value

An object of class `miss`.

Examples

```
library(rsimsum)
data("frailty", package = "rsimsum")
miss(data = frailty, estvarname = "b", se = "se", par = "par",
       methodvar = "model", by = "fv_dist")
```

`multisimsum` *Analyses of simulation studies with multiple estimands at once, including Monte Carlo error*

Description

`multisimsum` is an extension of `simsum()` that can handle multiple estimated parameters at once. `multisimsum` calls `simsum()` internally, each estimands at once. There is only one new argument that must be set when calling `multisimsum`: `par`, a string representing the column of `data` that identifies the different estimands.

Usage

```
multisimsum(data, par, true, estvarname, se, methodvar = NULL, ref = NULL,
  df = NULL, dropbig = FALSE, max = 10, semax = 100, level = 0.95,
  by = NULL, mcse = TRUE, sanitise = TRUE, na.rm = TRUE,
  na.pair = TRUE, x = FALSE)
```

Arguments

`data`	A `data.frame` in which variable names are interpreted. It has to be in tidy format, e.g. each variable forms a column and each observation forms a row.
`par`	The name of the variable containing the methods to compare. Can be `NULL`.
`true`	The true value of the parameter. This is used in calculations of bias and coverage.
`estvarname`	The name of the variable containing the point estimates.
`se`	The name of the variable containing the standard errors of the point estimates.
`methodvar`	The name of the variable containing the methods to compare. For instance, methods could be the models compared within a simulation study. Can be `NULL`.
`ref`	Specifies the reference method against which relative precision will be calculated. Only useful if `methodvar` is specified.
`df`	If specified, a t distribution with `df` degrees of freedom is used when calculating coverage and power.
`dropbig`	Specifies that point estimates or standard errors beyond the maximum acceptable values should be dropped.
`max`	Specifies the maximum acceptable absolute value of the point estimates, standardised to mean 0 and SD 1. Defaults to `10`.

semax	Specifies the maximum acceptable value of the standard error, as a multiple of the mean standard error. Defaults to 100.
level	Specifies the confidence level for coverage and power. Defaults to 0.95.
by	A vector of variable names to compute performance measures by a list of factors. Factors listed here are the (potentially several) data-generating mechanisms used to simulate data under different scenarios (e.g. sample size, true distribution of a variable, etc.). Can be NULL.
mcse	Reports Monte Carlo standard errors for all performance measures. Defaults to TRUE.
sanitise	Sanitise column names passed to simsum by removing all dot characters (.), which could cause problems. Defaults to TRUE.
na.rm	A logical value indicating whether missing values (NA) should be removed before the computation proceeds. Defaults to TRUE.
na.pair	Removes estimates that have a missing standard error (and vice versa). Defaults to TRUE.
x	Set to TRUE to include the data argument (as utilised to compute summary statistics, i.e. applying dropbig, na.rm, na.pair) as a slot. Defaults to FALSE.

Details

The following names are not allowed for estvarname, se, methodvar, by: stat, est, mcse, lower, upper. Calling the function with x = TRUE is required to produce zip plots (e.g. via the zip() method). The downside is that the size of the returned object increases considerably, therefore it is set to FALSE by default. Please note that the data slot returned when x = TRUE is obtained according to the value of the arguments dropbig, na.rm, na.pair; all rows with missing values are removed via a call to stats::na.omit().

Value

An object of class multisimsum.

Examples

```
data("frailty", package = "rsimsum")
ms <- multisimsum(data = frailty, par = "par", true = c(trt = -0.50,
   fv = 0.75), estvarname = "b", se = "se", methodvar = "model",
   by = "fv_dist")
ms
```

---

nsim *Compute number of simulations required*

Description

The function `nsim` computes the number of simulations B to perform based on the accuracy of an estimate of interest, using the following equation:

$$B = \left(\frac{(Z_{1-\alpha/2} + Z_{1-theta})\sigma}{\delta} \right)^2,$$

where δ is the specified level of accuracy of the estimate of interest you are willing to accept (i.e. the permissible difference from the true value β), $Z_{1-\alpha/2}$ is the $(1-\alpha/2)$ quantile of the standard normal distribution, $Z_{1-\theta}$ is the $(1-\theta)$ quantile of the standard normal distribution with $(1-\theta)$ being the power to detect a specific difference from the true value as significant, and σ^2 is the variance of the parameter of interest.

Usage

```
nsim(alpha, sigma, delta, power = 0.5)
```

Arguments

`alpha`	Significance level. Must be a value between 0 and 1.
`sigma`	Variance for the parameter of interest. Must be greater than 0.
`delta`	Specified level of accuracy of the estimate of interest you are willing to accept. Must be greater than 0.
`power`	Power to detect a specific difference from the true value as significant. Must be a value between 0 and 1. Defaults to 0.5, e.g. a power of 50%.

Value

A scalar value B representing the number of simulations to perform based on the accuracy required.

References

Burton, A., Douglas G. Altman, P. Royston. et al. 2006. The design of simulation studies in medical statistics. Statistics in Medicine 25: 4279-4292 doi: 10.1002/sim.2673

Examples

```
# Number of simulations required to produce an estimate to within 5%
# accuracy of the true coefficient of 0.349 with a 5% significance level,
# assuming the variance of the estimate is 0.0166 and 50% power:
nsim(alpha = 0.05, sigma = sqrt(0.0166), delta = 0.349 * 5 / 100, power = 0.5)
# Number of simulations required to produce an estimate to within 1%
# accuracy of the true coefficient of 0.349 with a 5% significance level,
# assuming the variance of the estimate is 0.0166 and 50% power:
nsim(alpha = 0.05, sigma = sqrt(0.0166), delta = 0.349 * 1 / 100, power = 0.5)
```

pattern *Draw pattern plots.*

Description

The S3 method `pattern()` produces a pattern plot to visualise estimates against standard errors for different methods.

Usage

```
pattern(obj, ...)
```

Arguments

`obj`	An object used to select a method.
`...`	Further arguments passed to or from other methods.

Details

`pattern()` requires `simsum` and `multisimsum` objects top be fit with the `x = TRUE` option.

Value

A `ggplot2::ggplot()` object that can be combined with additional `geom_*`, `scale_*`, `theme_*`, etc.

pattern.multisimsum *pattern method for multisimsum objects*

Description

`pattern()` method for objects of class `multisimsum`.

Usage

```
## S3 method for class 'multisimsum'
pattern(obj, par = NULL, gpars = list(), ...)
```

Arguments

obj	An object of class multisimsum.
par	Estimand to plot.
gpars	Graphical parameters. Must be a named list, with possible parameters: • alpha, alpha value of each point on the scatterplot; • scales, scale of x and y axis of each facet. It is possible to redefine all the graphical parameters or a subset only; if not specified, sensible default values will be utilised. Good practice would be adding a colorblind-safe palette, e.g. using ggthemes::scale_color_colorblind().
...	Ignored.

Details

If par = NULL (the default), all estimands are plotted and included as faceting variables.

Value

A ggplot2::ggplot() object that can be combined with additional geom_*, scale_*, theme_*, etc.

Examples

```
library(rsimsum)
library(ggplot2)
data("frailty", package = "rsimsum")
ms <- multisimsum(data = frailty, par = "par", true = c(trt = -0.50, fv = 0.75),
                  estvarname = "b", se = "se", methodvar = "model",
                  by = "fv_dist", x = TRUE)
pattern(ms)
pattern(ms, par = "trt")
```

---

pattern.simsum *pattern method for simsum objects*

Description

pattern() method for objects of class simsum.

Usage

```
## S3 method for class 'simsum'
pattern(obj, gpars = list(), ...)
```

Arguments

obj	An object of class simsum.
gpars	Graphical parameters. Must be a named list, with possible parameters: • alpha, alpha value of each point on the scatterplot; • scales, scale of x and y axis of each facet. It is possible to redefine all the graphical parameters or a subset only; if not specified, sensible default values will be utilised. Good practice would be adding a colorblind-safe palette, e.g. using ggthemes::scale_color_colorblind().
...	Ignored.

Details

pattern() requires simsum and multisimsum objects top be fit with the x = TRUE option.

Value

A ggplot2::ggplot() object that can be combined with additional geom_*, scale_*, theme_*, etc.

Note

pattern() automatically factorises the variable representing methods if present and if not already a factor; this allows ggplot2 to appropriately pick a discrete colour scale rather than a continuous one.

Examples

```
library(rsimsum)
library(ggplot2)
data("relhaz", package = "rsimsum")
s <- simsum(data = relhaz, estvarname = "theta", true = -0.5, se = "se",
  methodvar = "model", by = c("n", "baseline"), x = TRUE)
pattern(s)
```

---

print.dropbig.multisimsum *print.dropbig.multisimsum*

Description

Print method for dropbig.simsum objects

Usage

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

Arguments

x	An object of class dropbig.multisimsum.
...	Ignored.

See Also

dropbig.multisimsum()

---

print.dropbig.simsum *print.dropbig.simsum*

Description

Print method for dropbig.simsum objects

Usage

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

Arguments

x	An object of class dropbig.simsum.
...	Ignored.

See Also

dropbig.simsum()

---

print.miss *print.miss*

Description

Print method for miss objects

Usage

```
## S3 method for class 'miss'
print(x, fmt = "%.4f", ...)
```

Arguments

x	An object of class miss.
fmt	Format string passed to base::sprintf() to format proportions of missing values when printed.
...	Ignored.

Examples

```
library(rsimsum)
data("frailty", package = "rsimsum")
m <- miss(data = frailty, estvarname = "b", se = "se", par = "par",
            methodvar = "model", by = "fv_dist")
print(m)
print(m, fmt = "%.2f")
```

`print.multisimsum` *print.multisimsum*

Description

Print method for multisimsum objects

Usage

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

Arguments

x	An object of class multisimsum.
...	Ignored.

Examples

```
data(frailty)
ms <- multisimsum(data = frailty, par = "par", true = c(trt = -0.50,
   fv = 0.75), estvarname = "b", se = "se", methodvar = "model",
   by = "fv_dist")
ms
```

print.simsum *print.simsum*

Description

Print method for simsum objects

Usage

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

Arguments

x	An object of class simsum.
...	Ignored.

Examples

```
data("MIsim")
x <- simsum(data = MIsim, estvarname = "b", true = 0.5, se = "se",
            methodvar = "method", mcse = TRUE)
x
```

print.summary.multisimsum *print.summary.multisimsum*

Description

Print method for summary.multisimsum objects

Usage

```
## S3 method for class 'summary.multisimsum'
print(x, digits = 4, sstat = "all", ...)
```

Arguments

x	An object of class summary.multisimsum.
digits	Number of significant digits used for printing. Defaults to 4.
sstat	Summary statistics to print; can be a scalar value or a vector (for printing multiple summary statistics at once). Possible choices are: • all, all the summary statistics are printed. This is the default option.

- nsim, the number of replications with non-missing point estimates and standard error.
- thetamean, average point estimate.
- thetamedian, median point estimate.
- se2mean, average standard error.
- se2median, median standard error.
- bias, bias in point estimate.
- empse, empirical standard error.
- mse, mean squared error.
- relprec, percentage gain in precision relative to the reference method.
- modelse, model-based standard error.
- relerror, relative percentage error in standard error.
- cover, coverage of a nominal level% confidence interval.
- bccover, bias corrected coverage of a nominal level% confidence interval.
- power, power of a (1 - level)% level test.

... Ignored.

Note

If sstat is a vector that contains all, all summary statistics are printed by default.

Examples

```
data(frailty)
ms <- multisimsum(data = frailty, par = "par", true = c(trt = -0.50,
   fv = 0.75), estvarname = "b", se = "se", methodvar = "model",
   by = "fv_dist")
sms <- summary(ms)
sms
```

---

print.summary.simsum *print.summary.simsum*

Description

Print method for summary.simsum objects

Usage

```
## S3 method for class 'summary.simsum'
print(x, digits = 4, sstat = "all", ...)
```

Arguments

x	An object of class summary.simsum.
digits	Number of significant digits used for printing. Defaults to 4.
sstat	Summary statistics to print; can be a scalar value or a vector (for printing multiple summary statistics at once). Possible choices are: • all, all the summary statistics are printed. This is the default option. • nsim, the number of replications with non-missing point estimates and standard error. • thetamean, average point estimate. • thetamedian, median point estimate. • se2mean, average standard error. • se2median, median standard error. • bias, bias in point estimate. • empse, empirical standard error. • mse, mean squared error. • relprec, percentage gain in precision relative to the reference method. • modelse, model-based standard error. • relerror, relative percentage error in standard error. • cover, coverage of a nominal level% confidence interval. • bccover, bias corrected coverage of a nominal level% confidence interval. • power, power of a (1 - level)% level test.
...	Ignored.

Note

If sstat is a vector that contains all, all summary statistics are printed by default.

Examples

```
data("MIsim")
x <- simsum(data = MIsim, estvarname = "b", true = 0.5, se = "se",
            methodvar = "method", mcse = TRUE)
xs <- summary(x)
xs

# Printing only bias and coverage:
print(xs, sstat = c("bias", "cover"))
```

relhaz *Example of a simulation study on survival modelling*

Description

A dataset from a simulation study assessing the impact of misspecifying the baseline hazard in survival models on regression coefficients. One thousand datasets were simulated, each containing a binary treatment variable with a log-hazard ratio of -0.50. Survival data was simulated for two different sample sizes, 50 and 250 individuals, and under two different baseline hazard functions, exponential and Weibull. Consequently, a Cox model (Cox, 1972), a fully parametric exponential model, and a Royston-Parmar (Royston and Parmar, 2002) model with two degrees of freedom were fit to each simulated dataset.

Usage

```
relhaz
```

Format

A data frame with 12,000 rows and 6 variables:

- `dataset` Simulated dataset number.
- `n` Sample size of the simulate dataset.
- `baseline` Baseline hazard function of the simulated dataset.
- `model` Method used (`Cox`, `Exp`, or `RP(2)`).
- `theta` Point estimate for the log-hazard ratio.
- `se` Standard error of the point estimate.

References

Cox D.R. 1972. Regression models and life-tables. Journal of the Royal Statistical Society, Series B (Methodological) 34(2):187-220. http://www.jstor.org/stable/2985181

Royston, P. and Parmar, M.K. 2002. Flexible parametric proportional-hazards and proportional-odds models for censored survival data, with application to prognostic modelling and estimation of treatment effects. Statistics in Medicine 21(15):2175-2197 doi: 10.1002/sim.1203

Examples

```
data("relhaz", package = "rsimsum")
```

rsimsum *Analysis of Simulation Studies Including Monte Carlo Error*

Description

Summarise results from simulation studies and compute Monte Carlo standard errors of commonly used summary statistics. This package is modelled on the 'simsum' user-written command in 'Stata' (See White I.R., 2010 http://www.stata-journal.com/article.html?article=st0200).

Author(s)

Alessandro Gasparini (ag475@leicester.ac.uk)

simsum *Analyses of simulation studies including Monte Carlo error*

Description

simsum computes performance measures for simulation studies in which each simulated data set yields point estimates by one or more analysis methods. Bias, empirical standard error and precision relative to a reference method can be computed for each method. If, in addition, model-based standard errors are available then simsum can compute the average model-based standard error, the relative error in the model-based standard error, the coverage of nominal confidence intervals, and the power to reject a null hypothesis. Monte Carlo errors are available for all estimated quantities.

Usage

```
simsum(data, estvarname, true, se, methodvar = NULL, ref = NULL,
  df = NULL, dropbig = FALSE, max = 10, semax = 100, level = 0.95,
  by = NULL, mcse = TRUE, sanitise = TRUE, na.rm = TRUE,
  na.pair = TRUE, x = FALSE)
```

Arguments

data	A data.frame in which variable names are interpreted. It has to be in tidy format, e.g. each variable forms a column and each observation forms a row.
estvarname	The name of the variable containing the point estimates.
true	The true value of the parameter. This is used in calculations of bias and coverage.
se	The name of the variable containing the standard errors of the point estimates.
methodvar	The name of the variable containing the methods to compare. For instance, methods could be the models compared within a simulation study. Can be NULL.
ref	Specifies the reference method against which relative precision will be calculated. Only useful if methodvar is specified.

df	If specified, a t distribution with df degrees of freedom is used when calculating coverage and power.
dropbig	Specifies that point estimates or standard errors beyond the maximum acceptable values should be dropped.
max	Specifies the maximum acceptable absolute value of the point estimates, standardised to mean 0 and SD 1. Defaults to 10.
semax	Specifies the maximum acceptable value of the standard error, as a multiple of the mean standard error. Defaults to 100.
level	Specifies the confidence level for coverage and power. Defaults to 0.95.
by	A vector of variable names to compute performance measures by a list of factors. Factors listed here are the (potentially several) data-generating mechanisms used to simulate data under different scenarios (e.g. sample size, true distribution of a variable, etc.). Can be NULL.
mcse	Reports Monte Carlo standard errors for all performance measures. Defaults to TRUE.
sanitise	Sanitise column names passed to simsum by removing all dot characters (.), which could cause problems. Defaults to TRUE.
na.rm	A logical value indicating whether missing values (NA) should be removed before the computation proceeds. Defaults to TRUE.
na.pair	Removes estimates that have a missing standard error (and vice versa). Defaults to TRUE.
x	Set to TRUE to include the data argument (as utilised to compute summary statistics, i.e. applying dropbig, na.rm, na.pair) as a slot. Defaults to FALSE.

Details

The following names are not allowed for estvarname, se, methodvar, by: stat, est, mcse, lower, upper. Calling the function with x = TRUE is required to produce zip plots (e.g. via the zip() method). The downside is that the size of the returned object increases considerably, therefore it is set to FALSE by default. Please note that the data slot returned when x = TRUE is obtained according to the value of the arguments dropbig, na.rm, na.pair; all rows with missing values are removed via a call to stats::na.omit().

Value

An object of class simsum.

References

White, I.R. 2010. simsum: Analyses of simulation studies including Monte Carlo error. The Stata Journal 10(3): 369-385. http://www.stata-journal.com/article.html?article=st0200

Morris, T.P, White, I.R. and Crowther, M.J. 2017. Using simulation studies to evaluate statistical methods. arXiv:1712.03198

Examples

```
data("MIsim")
s <- simsum(data = MIsim, estvarname = "b", true = 0.5, se = "se", methodvar = "method", ref = "CC")
# If `ref` is not specified, the reference method is inferred
s <- simsum(data = MIsim, estvarname = "b", true = 0.5, se = "se", methodvar = "method")
```

`summary.multisimsum` *Summarising multisimsum objects*

Description

The `summary()` method for objects of class `multisimsum` returns confidence intervals for performance measures based on Monte Carlo standard errors.

Usage

```
## S3 method for class 'multisimsum'
summary(object, ci_level = 0.95, ...)
```

Arguments

`object`	An object of class `multisimsum`.
`ci_level`	Significance level for confidence intervals based on Monte Carlo standard errors. Ignored if a `multisimsum` object is obtained with `mcse = FALSE`.
`...`	Ignored.

Value

An object of class `summary.multisimsum`.

Note

Confidence intervals are constructed using quantiles from a normal distribution with the specified `ci_level` confidence level, and are therefore symmetric by definition.

See Also

`multisimsum()`, `print.summary.multisimsum()`

Examples

```
data(frailty)
ms <- multisimsum(data = frailty, par = "par", true = c(trt = -0.50,
   fv = 0.75), estvarname = "b", se = "se", methodvar = "model",
   by = "fv_dist")
sms <- summary(ms)
sms
```

`summary.simsum` *Summarising simsum objects*

Description

The `summary()` method for objects of class `simsum` returns confidence intervals for performance measures based on Monte Carlo standard errors.

Usage

```
## S3 method for class 'simsum'
summary(object, ci_level = 0.95, ...)
```

Arguments

`object`	An object of class `simsum`.
`ci_level`	Significance level for confidence intervals based on Monte Carlo standard errors. Ignored if a `simsum` object with `mcse = FALSE` is passed.
`...`	Ignored.

Value

An object of class `summary.simsum`.

Note

Confidence intervals are constructed using quantiles from a normal distribution with the specified `ci_level` confidence level, and are therefore symmetric by definition.

See Also

`simsum()`, `print.summary.simsum()`

Examples

```
data("MIsim")
object <- simsum(data = MIsim, estvarname = "b", true = 0.5, se = "se",
methodvar = "method", mcse = TRUE)
xs <- summary(object)
xs
```

zip *Draw zip plots.*

Description

The S3 method `zip()` produces a zip plot to visualise coverage. Zip plots are introduced in Morris *et al*., 2017.

Usage

```
zip(obj, ...)
```

Arguments

`obj`	An object used to select a method.
`...`	Further arguments passed to or from other methods.

Value

A `ggplot2::ggplot()` object that can be combined with additional `geom_*`, `scale_*`, `theme_*`, etc.

References

Morris, T.P, White, I.R. and Crowther, M.J. 2017. Using simulation studies to evaluate statistical methods. arXiv:1712.03198

zip.multisimsum *zip method for multisimsum objects*

Description

`zip()` method for objects of class `multisimsum`.

Usage

```
## S3 method for class 'multisimsum'
zip(obj, par, wald.level = 0.95, gpars = list(), ...)
```

Arguments

obj	An object of class multisimsum.
par	Estimand to plot.
wald.level	Confidence level of the Wald test used to compute p-values for sorting each confidence interval. Defaults to 0.95.
gpars	Graphical parameters. Must be a named list, with possible parameters: • ci.alpha, alpha value of each individual confidence interval; • true.colour, colour of the vertical line at true value; • true.shape, shape of the vertical line at true value; • ci.colour, colour of the horizontal lines representing confidence intervals for estimated coverage based on Monte Carlo standard errors; • ci.shape, shape of the horizontal lines representing confidence intervals for estimated coverage based on Monte Carlo standard errors. It is possible to redefine all the graphical parameters of a subset only; if not specified, sensible default values will be utilised.
...	Ignored.

Value

A ggplot2::ggplot() object that can be combined with additional geom_*, scale_*, theme_*, etc.

Examples

```
library(rsimsum)
library(ggplot2)
data("frailty", package = "rsimsum")
ms <- multisimsum(data = frailty, par = "par", true = c(trt = -0.50,
   fv = 0.75), estvarname = "b", se = "se", methodvar = "model",
   by = "fv_dist", x = TRUE)
zip(ms, par = "trt")
```

---

zip.simsum *zip method for simsum objects*

Description

zip() method for objects of class simsum.

Usage

```
## S3 method for class 'simsum'
zip(obj, wald.level = 0.95, gpars = list(), ...)
```

Arguments

`obj`	An object of class `simsum`.
`wald.level`	Confidence level of the Wald test used to compute p-values for sorting each confidence interval. Defaults to `0.95`.
`gpars`	Graphical parameters. Must be a named list, with possible parameters: • `ci.alpha`, alpha value of each individual confidence interval; • `true.colour`, colour of the vertical line at `true` value; • `true.shape`, shape of the vertical line at `true` value; • `ci.colour`, colour of the horizontal lines representing confidence intervals for estimated coverage based on Monte Carlo standard errors; • `ci.shape`, shape of the horizontal lines representing confidence intervals for estimated coverage based on Monte Carlo standard errors. It is possible to redefine all the graphical parameters of a subset only; if not specified, sensible default values will be utilised.
`...`	Ignored.

Value

A `ggplot2::ggplot()` object that can be combined with additional `geom_*`, `scale_*`, `theme_*`, etc.

Examples

```
library(rsimsum)
library(ggplot2)
data("relhaz", package = "rsimsum")
s <- simsum(data = relhaz, estvarname = "theta", true = -0.5, se = "se",
  methodvar = "model", by = c("n", "baseline"), x = TRUE)
zip(s)
data("MIsim", package = "rsimsum")
s2 <- simsum(data = MIsim, estvarname = "b", true = 0.5, se = "se",
  methodvar = "method", x = TRUE)
zip(s2)
```

Index