Package ‘simMetric’

October 31, 2022

Title Metrics (with Uncertainty) for Simulation Studies that Evaluate Statistical Methods

Version 0.1.1

Description Allows users to quickly apply individual or multiple metrics to evaluate Monte Carlo simulation studies.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.2

Imports stats, assertthat, dplyr

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Rex Parsons [aut, cre] (<`https://orcid.org/0000-0002-6053-8174`>)

Maintainer Rex Parsons <`Rex.Parsons94@gmail.com`>

Repository CRAN

Date/Publication 2022-10-31 17:42:27 UTC

R topics documented:

`bias` *Calculate the bias*

Description

Calculates the bias of the model estimates from the true value and the Monte Carlo standard error for this estimate.

Usage

```
bias(true_value, estimates, get = c("bias", "bias_mcse"), na.rm = FALSE, ...)
```

Arguments

`true_value`	The true value which is being estimated.
`estimates`	A numeric vector containing the estimates from the model(s).
`get`	A character vector containing the values returned by the function.
`na.rm`	A logical value indicating whether NA values for `estimates` should be removed before bias calculation.
`...`	Additional arguments to be ignored.

Value

A named vector containing the estimate and the Monte Carlo standard error for the bias.

Examples

```
bias(true_value=0, estimates=rnorm(100))
```

`biasEliminatedCoverage` *Calculate the bias-eliminated coverage*

Description

Estimate the bias-eliminated coverage and the Monte Carlo standard error of this estimate given a vector of confidence intervals and the true value.

Usage

```
biasEliminatedCoverage(
  estimates,
  ll,
  ul,
  get = c("biasEliminatedCoverage", "biasEliminatedCoverage_mcse"),
  na.rm = FALSE,
  ...
)
```

Arguments

`estimates`	A numeric vector containing the estimates from the model(s).
`ll`	A numeric vector containing the lower limits of the confidence intervals.
`ul`	A numeric vector containing the upper limits of the confidence intervals.
`get`	A character vector containing the values returned by the function.
`na.rm`	A logical value indicating whether NA values for ll and ul should be removed before coverage estimation.
`...`	Additional arguments to be ignored.

Value

A named vector containing the estimate and the Monte Carlo standard error for the coverage.

Examples

```
biasEliminatedCoverage(estimates=rnorm(4), ll=c(-1, -1, -1, -1), ul=c(1, 1, 1, -0.5))
```

`coverage` *Calculate the coverage*

Description

Estimate the coverage and the Monte Carlo standard error of this estimate given a vector of confidence intervals and the true value.

Usage

```
coverage(
  true_value,
  ll,
  ul,
  get = c("coverage", "coverage_mcse"),
  na.rm = FALSE,
  ...
)
```

Arguments

`true_value`	The true value which should be covered by the interval.
`ll`	A numeric vector containing the lower limits of the confidence intervals.
`ul`	A numeric vector containing the upper limits of the confidence intervals.
`get`	A character vector containing the values returned by the function.
`na.rm`	A logical value indicating whether NA values for ll and ul should be removed before coverage estimation.
`...`	Additional arguments to be ignored.

Value

A named vector containing the estimate and the Monte Carlo standard error for the coverage.

Examples

```
coverage(true_value=0, ll=c(-1, -1, -1, -1), ul=c(1, 1, 1, -0.5))
```

`empSE` *Calculate the empirical standard error*

Description

Calculates the empirical standard error of the model estimates and its Monte Carlo standard error.

Usage

```
empSE(estimates, get = c("empSE", "empSE_mcse"), na.rm = FALSE, ...)
```

Arguments

`estimates`	A numeric vector containing the estimates from the model(s).
`get`	A character vector containing the values returned by the function.
`na.rm`	A logical value indicating whether NA values for `estimates` should be removed before empSE calculation.
`...`	Additional arguments to be ignored.

Value

A named vector containing the estimate and the Monte Carlo standard error for the empirical standard error.

Examples

```
empSE(estimates=rnorm(100))
```

`join_metrics` *Join metrics*

Description

Calculate and join selected evaluation metrics given a `data.frame` of simulation study results Provides a fast way to add multiple metrics and their Monte Carlo standard errors.

Usage

```
join_metrics(
  data,
  id_cols,
  metrics = c("coverage", "mse", "modSE"),
  true_value = NULL,
  ll_col = NULL,
  ul_col = NULL,
  estimates_col = NULL,
  se_col = NULL,
  p_col = NULL,
  alpha = 0.05
)
```

Arguments

`data`	A `data.frame` that contains results from simulation study and the necessary columns to compute metrics.
`id_cols`	Column name(s) on which to group data and calculate metrics.
`metrics`	A vector of metrics to be calculated.
`true_value`	The true parameter to be estimated.
`ll_col`	Name of the column that contains the lower limit of the confidence intervals. (Required for calculating coverage.)
`ul_col`	Name of the column that contains the upper limit of the confidence intervals. (Required for calculating coverage.)
`estimates_col`	Name of the column that contains the parameter estimates. (Required for calculating bias, empSE, and mse.)
`se_col`	Name of the column that contains the standard errors. (Required for calculating modSE.)
`p_col`	Name of the column that contains the p-values. (Required for calculating rejection.)
`alpha`	The nominal significance level specified. (Required for calculating rejection.)

Value

`data.frame` containing metrics and `id_cols`

Examples

```
simulations_df <- data.frame(
  idx=rep(1:10, 100),
  idx2=sample(c("a", "b"), size=1000, replace=TRUE),
  p_value=runif(1000),
  est=rnorm(n=1000),
  conf.ll= rnorm(n=1000, mean=-20),
  conf.ul= rnorm(n=1000, mean=20)
)
res <- join_metrics(
  data=simulations_df,
  id_cols=c("idx", "idx2"),
  metrics=c("rejection", "coverage", "mse"),
  true_value=0,
  ll_col="conf.ll",
  ul_col="conf.ul",
  estimates_col="est",
  p_col="p_value",
)
```

`modSE` *Calculates the average model standard error*

Description

Calculates the average model standard error and the Monte Carlo standard error of this estimate.

Usage

```
modSE(se, get = c("modSE", "modSE_mcse"), na.rm = FALSE, ...)
```

Arguments

`se`	A numeric vector containing the standard errors from the model(s).
`get`	A character vector containing the values returned by the function.
`na.rm`	A logical value indicating whether NA values for `se` should be removed before modSE calculation.
`...`	Additional arguments to be ignored.

Value

A named vector containing the estimate and the Monte Carlo standard error for the average model standard error.

Examples

```
modSE(se=runif(n=20, min=1, max=1.5))
```

`mse` *Calculate the Mean Squared Error*

Description

Calculates the Mean Squared Error of the model estimates from the true value and the Monte Carlo standard error for this estimate.

Usage

```
mse(true_value, estimates, get = c("mse", "mse_mcse"), na.rm = FALSE, ...)
```

Arguments

`true_value`	The true value which is being estimated.
`estimates`	A numeric vector containing the estimates from the model(s).
`get`	A character vector containing the values returned by the function.
`na.rm`	A logical value indicating whether NA values for `estimates` should be removed before MSE calculation.
`...`	Additional arguments to be ignored.

Value

A named vector containing the estimate and the Monte Carlo standard error for the bias.

Examples

```
mse(true_value=0, estimates=rnorm(100))
```

`rejection` *Calculate the rejection*

Description

Calculates the rejection (%) of the model p-values, according to the specified alpha, and the Monte Carlo standard error for this estimate.

Usage

```
rejection(
  p,
  alpha = 0.05,
  get = c("rejection", "rejection_mcse"),
  na.rm = FALSE,
  ...
)
```

Arguments

`p`	P-values from the models.
`alpha`	The nominal significance level specified. The default is `0.05`.
`get`	A character vector containing the values returned by the function.
`na.rm`	A logical value indicating whether NA values for `p` should be removed before rejection calculation.
`...`	Additional arguments to be ignored.

Value

A named vector containing the estimate and the Monte Carlo standard error for the rejection.

Examples

```
rejection(p=runif(200, min=0, max=1))
```

`relativeErrorModSE` *Calculates the relative (%) error in model standard error*

Description

Calculates the relative (%) error in model standard error and the (approximate) Monte Carlo standard error of this estimate.

Usage

```
relativeErrorModSE(
  se,
  estimates,
  get = c("relativeErrorModSE", "relativeErrorModSE_mcse"),
  na.rm = FALSE,
  ...
)
```

Arguments

`se`	A numeric vector containing the standard errors from the model(s).
`estimates`	A numeric vector containing the estimates from the model(s).
`get`	A character vector containing the values returned by the function.
`na.rm`	A logical value indicating whether NA values for `se` and `estimates` should be removed before modSE and empSE calculation.
`...`	Additional arguments to be ignored.

Value

A named vector containing the estimate and the Monte Carlo standard error for the relative (%) error in model standard error.

Examples

```
relativeErrorModSE(se=rnorm(n=1000, mean=10, sd=0.5), estimates=rnorm(n=1000))
```

`relativePrecision` *Calculates the relative (%) increase in precision between two methods*

Description

Calculates the relative (%) increase in precision between two competing methods (B vs A). As this metric compares two methods directly, it cannot be used in `join_metrics()`.

Usage

```
relativePrecision(
  estimates_A,
  estimates_B,
  get = c("relativePrecision", "relativePrecision_mcse"),
  na.rm = FALSE
)
```

Arguments

`estimates_A`	A numeric vector containing the estimates from model A.
`estimates_B`	A numeric vector containing the estimates from model B.
`get`	A character vector containing the values returned by the function.
`na.rm`	A logical value indicating whether NA values for `estimates` should be removed before empSE calculation.

Value

A named vector containing the estimate and the Monte Carlo standard error for the relative (%) increase in precision of method B versus method A.

Examples

```
relativePrecision(estimates_A=rnorm(n=1000), estimates_B=rnorm(n=1000))
```

Index