Package ‘simhelpers’

February 14, 2021

Type Package

Title Helper Functions for Simulation Studies

Version 0.1.1

Maintainer Megha Joshi <megha.j456@gmail.com>

Description Calculates performance criteria measures and associated Monte Carlo standard errors for simulation results. Includes functions to help run simulation studies. Our derivation and explanation of formulas and our general simulation workflow is closely aligned with the approach described by Morris, White, and Crowther (2019) <DOI: 10.1002/sim.8086>.


BugReports https://github.com/meghapsimatrix/simhelpers/issues

Depends R (>= 2.10)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

SystemRequirements RStudio

Imports rlang, stats, dplyr, furrr, future, magrittr, utils, purrr, tibble, rstudioapi, Rdpack, tidyr

Suggests plyr, knitr, rmarkdown, pkgdown, covr, testthat, kableExtra, ggplot2, broom

RdMacros Rdpack

VignetteBuilder knitr

NeedsCompilation no

Author Megha Joshi [aut, cre],
James Pustejovsky [aut] (<https://orcid.org/0000-0003-0591-9465>)

Repository CRAN

Date/Publication 2021-02-14 17:50:02 UTC
R topics documented:

- `alpha_res` .......................... 2
- `calc_absolute` .......................... 2
- `calc_coverage` .......................... 3
- `calc_rejection` .......................... 4
- `calc_relative` .......................... 5
- `calc_relative_var` .......................... 6
- `create_skeleton` .......................... 6
- `evaluate_by_row` .......................... 7
- `Tipton_Pusto` .......................... 8
- `t_res` .......................... 9
- `welch_res` .......................... 9

Index 11

---

**alpha_res**

Cronbach’s alpha simulation results

**Description**

A dataset containing simulation results from estimating Cronbach’s alpha and its variance.

**Usage**

`alpha_res`

**Format**

A tibble with 1,000 rows and 3 variables:

- **A** estimate of alpha.
- **Var_A** estimate of the variance of alpha.
- **true_param** true alpha used to generate the data.

---

**calc_absolute**

Calculate absolute performance criteria and MCSE

**Description**

Calculates absolute bias, variance, mean squared error (mse) and root mean squared error (rmse). The function also calculates the associated Monte Carlo standard errors.
Usage

calc_absolute(
  res_dat,
  estimates,
  true_param,
  perfm_criteria = c("bias", "variance", "mse", "rmse")
)

Arguments

res_dat data frame or tibble containing the simulation results.
estimates name of the column containing the estimates.
true_param name of the column containing the true parameters.
perfm_criteria character or character vector indicating the performance criteria to be calculated.

Value

A tibble containing the number of simulation iterations, performance criteria estimate(s) and the associated MCSE.

Examples

calc_absolute(res_dat = t_res, estimates = est, true_param = true_param)


calc_coverage

Calculate confidence interval coverage, width and MCSE

Description

Calculates confidence interval coverage and width. The function also calculates the associated Monte Carlo standard errors. The confidence interval percentage is based on how you calculated the lower and upper bounds.

Usage

calc_coverage(
  res_dat,
  lower_bound,
  upper_bound,
  true_param,
  perfm_criteria = c("coverage", "width")
)
### Arguments

- **res_dat**: data frame or tibble containing the simulation results.
- **lower_bound**: name of the column containing the lower bound estimates of the confidence intervals.
- **upper_bound**: name of the column containing the upper bound estimates of the confidence intervals.
- **true_param**: name of the column containing the true parameters.
- **perfm_criteria**: character or character vector indicating the performance criteria to be calculated.

### Value

A tibble containing the number of simulation iterations, performance criteria estimate(s) and the associated MCSE.

### Examples

```r
calc_coverage(res_dat = t_res, lower_bound = lower_bound,
              upper_bound = upper_bound, true_param = true_param)
```

---

### Description

Calculates rejection rate. The function also calculates the associated Monte Carlo standard error.

### Usage

```r
calc_rejection(res_dat, p_values, alpha = 0.05)
```

### Arguments

- **res_dat**: data frame or tibble containing the simulation results.
- **p_values**: name of the column containing the p-values.
- **alpha**: number indicating the nominal alpha level. Default value is set to the conventional .05.

### Value

A tibble containing the number of simulation iterations, performance criteria estimate and the associated MCSE.
Examples

    calc_rejection(res_dat = t_res, p_values = p_val)

---

calc_relative  

*Calculate relative performance criteria and MCSE*

Description

Calculates relative bias, mean squared error (relative mse), and root mean squared error (relative rmse). The function also calculates the associated Monte Carlo standard errors.

Usage

    calc_relative(
        res_dat, estimates, true_param,
        perfm_criteria = c("relative bias", "relative mse", "relative rmse")
    )

Arguments

- `res_dat`: data frame or tibble containing the simulation results.
- `estimates`: name of the column containing the estimates.
- `true_param`: name of the column containing the true parameters.
- `perfm_criteria`: character or character vector indicating the performance criteria to be calculated.

Value

A tibble containing the number of simulation iterations, performance criteria estimate(s) and the associated MCSE.

Examples

    calc_relative(res_dat = t_res, estimates = est, true_param = true_param)
calc_relative_var  

Calculate jack-knife Monte Carlo SE for variance estimators

Description
Calculates relative bias, mean squared error (relative mse), and root mean squared error (relative rmse) of variance estimators. The function also calculates the associated jack-knife Monte Carlo standard errors.

Usage
```r
calc_relative_var(
  res_dat, 
  estimates, 
  var_estimates, 
  perfm_criteria = c("relative bias", "relative mse", "relative rmse")
)
```

Arguments
- `res_dat`: data frame or tibble containing the simulation results.
- `estimates`: name of the column containing the estimates.
- `var_estimates`: name of the column containing the variance estimates.
- `perfm_criteria`: character or character vector indicating the performance criteria to be calculated.

Value
A tibble containing the number of simulation iterations, performance criteria estimate(s) and the associated MCSE.

Examples
```r
calc_relative_var(res_dat = alpha_res, estimates = A, var_estimates = Var_A)
```

create_skeleton  

Open a simulation skeleton

Description
Creates and opens a .R file containing a skeleton for writing a Monte Carlo simulation study.

Usage
```r
create_skeleton()
```
evaluate_by_row

Examples

## Not run:
create_skeleton()

## End(Not run)

evaluate_by_row  Evaluate a simulation function on each row of a data frame or tibble

Description

Evaluates a simulation function on each row of a data frame or tibble containing parameter values. Returns a single tibble with parameters and simulation results. The function uses `furrr::future_pmap`, which allows for easy parallelization.

Usage

```r
evaluate_by_row(
  params,  # data frame or tibble containing simulation parameter values. Each row should represent a separate set of parameter values.
  sim_function,  # function to be evaluated, with argument names matching the variable names in params. The function must return a data.frame, tibble, or vector.
  ...,  # additional arguments passed to sim_function.
  .progress = FALSE,  # A single logical. Should a progress bar be displayed? Only works with multisession, multicore, and multiprocess futures. Note that if a multicore/multisession future falls back to sequential, then a progress bar will not be displayed.
  .options = furrr::future_options(),  # The future specific options to use with the workers. This must be the result from a call to `furrr_options()`.
  system_time = TRUE  # logical indicating whether to print computation time. TRUE by default.
)
```

Arguments

- **params**: data frame or tibble containing simulation parameter values. Each row should represent a separate set of parameter values.
- **sim_function**: function to be evaluated, with argument names matching the variable names in params. The function must return a data.frame, tibble, or vector.
- **...**: additional arguments passed to sim_function.
- **.progress**: A single logical. Should a progress bar be displayed? Only works with multisession, multicore, and multiprocess futures. Note that if a multicore/multisession future falls back to sequential, then a progress bar will not be displayed.
- **.options**: The future specific options to use with the workers. This must be the result from a call to `furrr_options()`.
- **system_time**: logical indicating whether to print computation time. TRUE by default.

Value

A tibble containing parameter values and simulation results.
Examples

```r
df <- data.frame(
  n = 3:5,
  lambda = seq(8, 16, 4)
)

evaluate_by_row(df, rpois)
```

<table>
<thead>
<tr>
<th>Tipton_Pusto</th>
<th>Results for Figure 2 of Tipton &amp; Pustejovsky (2015)</th>
</tr>
</thead>
</table>

Description

A dataset containing simulation results comparing small sample correction methods for cluster robust variance estimation in meta-analysis.

Usage

Tipton_Pusto

Format

A tibble with 15,300 rows and 8 variables:

- `num_studies` the number of studies included in the meta-analysis.
- `r` correlation between outcomes.
- `Isq` measure of heterogeneity of true effects.
- `contrast` type of contrast that was tested.
- `test` small sample method used.
- `q` the number of parameters in the hypothesis test.
- `rej_rate` the Type 1 error rate.
- `mcse` the Monte Carlo standard error for the estimate of the Type 1 error rate.

Source

t_res  

**t-test simulation results**

**Description**
A dataset containing simulation results from a study that just runs a t-test.

**Usage**
t_res

**Format**
A tibble with 1,000 rows and 5 variables:
- **est** estimate of the mean difference.
- **p_val** p-value from the t-test.
- **lower_bound** lower bound of the confidence interval.
- **upper_bound** upper bound of the confidence interval.
- **true_param** true mean difference used to generate the data.

welch_res  

**Welch t-test simulation results**

**Description**
A dataset containing simulation results from a study comparing Welch t-test to the conventional t-test.

**Usage**
welch_res

**Format**
A tibble with 16,000 rows and 11 variables:
- **n1** sample size for Group 1.
- **n2** sample size for Group 2.
- **mean_diff** true difference in means of two groups used to generate the data.
- **iterations** number of iterations.
- **seed** seed used to generate data.
- **method** indicates whether Welch or conventional t-test was used.
est  estimate of the mean difference.
var  variance of the estimate.
p_val p-value from the t-test.
lower_bound  lower bound of the confidence interval.
upper_bound  upper bound of the confidence interval.
Index

* datasets
  alpha_res, 2
  t_res, 9
  Tipton_Pusto, 8
  welch_res, 9

alpha_res, 2

calc_absolute, 2
calc_coverage, 3
calc_rejection, 4
calc_relative, 5
calc_relative_var, 6
create_skeleton, 6

evaluate_by_row, 7

furrr_options(), 7

t_res, 9
  Tipton_Pusto, 8

welch_res, 9