Package 'fastpos'

September 29, 2020

Type Package

Title Finds the Critical Sequential Point of Stability for a Pearson Correlation

Version 0.4.1

Date 2020-09-25

Description Finds the critical sample size (``critical point of stability'') for a correlation to stabilize in Schoenbrodt and Perugini's definition of sequential stability (see <doi:10.1016/j.jrp.2013.05.009>).

License GPL-3

Imports Rcpp (>= 1.0.1), plyr, MASS, future

LinkingTo Rcpp, RcppArmadillo, RcppProgress

URL https://github.com/johannes-titz/fastpos

BugReports https://github.com/johannes-titz/fastpos/issues

Encoding UTF-8

RoxygenNote 7.1.1

Suggests knitr, rmarkdown, testthat (>= 2.1.0), covr, microbenchmark

VignetteBuilder knitr

NeedsCompilation yes

Author Johannes Titz [aut, cre, cph]

Maintainer Johannes Titz <johannes.titz@gmail.com>

Repository CRAN

Date/Publication 2020-09-29 08:50:08 UTC

R topics documented:

create_pop *Creates a population with a specified correlation.*

Description

The correlation will be exactly the one specified. The used method is described here: https://stats.stackexchange.com/question
a-random-variable-with-a-defined-correlation-to-an-existing-variables/15040#15040

Usage

```
create_pop(rho, size)
```

Arguments

rho	Population correlation.
size	Population size.

Value

Two-dimensional population matrix with a specific correlation.

Examples

```
pop <- create_pop(0.5, 100000)
cor(pop)
```

find_critical_pos *Find the critical point of stability*

Description

Run simulations for one or several population correlations and return the critical points of stability (POS). The critical point of stability is the sample size at which a certain percentage of studies will fall into an a priori specified interval and stay in this interval if the sample size is increased further.

Usage

```
find_critical_pos(
  rhos,
  precision = 0.1,
  precision_rel = FALSE,
  sample_size_min = 20,
  sample_size_max = 1000,
  replace = TRUE,
  n_studies = 10000,
  confidence_levels = c(0.8, 0.9, 0.95),
```

```
  pop_size = 1e+06,
  n_cores = 1
)
```

Arguments

rhos	Vector of population correlations (can also be a single correlation).
precision	Precision around the correlation which is acceptable (defaults to 0.1). The precision will determine the corridor of stability which is just rho+-precision.
precision_rel	Whether the precision is absolute (rho+-precision or relative rho+-rho*precision), boolean (defaults to FALSE).
sample_size_min	Minimum sample size for each study (defaults to 20).
sample_size_max	Maximum sample size for each study (defaults to 1e3).
replace	Whether drawing samples is with replacement or not.
n_studies	Number of studies to run for each rho (defaults to 10e3).
confidence_levels	Confidence levels for point of stability. This corresponds to the quantile of the distribution of all found critical sample sizes (defaults to c(.8, .9, .95)).
pop_size	Population size (defaults to 1e6).
n_cores	Number of cores to use for simulation.

Value

A data frame containing all the above information, as well as the points of stability.

Examples

```
find_critical_pos(rhos = 0.5)
find_critical_pos(rhos = c(0.4, 0.5), n_studies = 1e3)
```

---

simulate_pos *Simulate several points of stability*

Description

Runs several simulations and returns the points of stability, which can then be further processed to calculate the critical point of stability.

Usage

```
simulate_pos(
  x_pop,
  y_pop,
  n_studies,
  sample_size_min,
  sample_size_max,
  replace,
  lower_limit,
  upper_limit
)
```

Arguments

x_pop	First vector of population.
y_pop	Second vector of population.
n_studies	How many studies to conduct.
sample_size_min	Minimum sample size to start in corridor of stability.
sample_size_max	How many participants to draw at maximum.
replace	Whether drawing samples is with replacement or not.
lower_limit	Lower limit of corridor of stability.
upper_limit	Upper limit of corridor of stability.

Value

Vector of sample sizes at which corridor of stability was reached.

Examples

```
pop <- fastpos::create_pop(0.5, 1000000)
simulate_pos(pop[,1], pop[,2], 100, 20, 1000, TRUE, 0.4, 0.6)
```

Index