# Package: fastpos (via r-universe)

September 18, 2024

**Type** Package

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

**Version** 0.5.1.9000

**Date** 2022-08-15

**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, plyr, MASS, lifecycle, tibble, stats, pbmcapply

**LinkingTo** Rcpp, RcppArmadillo, RcppProgress

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

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

**Encoding** UTF-8

**RoxygenNote** 7.2.1

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

**Roxygen** list(markdown = TRUE)

**Repository** https://johannes-titz.r-universe.dev

**RemoteUrl** https://github.com/johannes-titz/fastpos

**RemoteRef** HEAD

**RemoteSha** f0fa8bc27a257e9d013828503fde6916170b66d0

## Contents

## 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(rho = 0.5, size = 1e6)
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(
  rho,
  precision_absolute = 0.1,
  confidence_levels = c(0.8, 0.9, 0.95),
  sample_size_min = 20,
  sample_size_max = 1000,
  n_studies = 10000,
  n_cores = 1,
  pop_size = 1e+06,
  replace = TRUE,
  precision_relative = NA,
  lower_limit = NA,
  upper_limit = NA,
  progress = show_progress(),
  precision = lifecycle::deprecated(),
  precision_rel = lifecycle::deprecated(),
  rhos = lifecycle::deprecated()
)
```

## Arguments

| | |
|---|---|
| `rho` | Vector of population correlations (can also be a single correlation). |
| `precision_absolute` | Precision around the correlation which is acceptable (defaults to 0.1). The precision will determine the corridor of stability which is just rho+-precision. Can be a single value or a vector (different values for different rhos). |
| `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)). A single value can also be used. Note that this value is fixed for all rhos! You cannot specify different levels for different rhos. |
| `sample_size_min` | Minimum sample size for each study (defaults to 20). A vector can be used (different values for different rhos). |
| `sample_size_max` | Maximum sample size for each study (defaults to 1e3). A vector can be used (different values for different rhos). If you get a warning that the corridor of stability was not reached, you should increase this value. But note that this will increase the time for the simulation. |
| `n_studies` | Number of studies to run for each rho (defaults to 1e4). A vector can be used (different values for different rhos). |
| `n_cores` | Number of cores to use for simulation. Defaults to 1. Under Windows only 1 core is supported because forking is used. |
| `pop_size` | Population size (defaults to 1e6). This is the size of the population from which value pairs for correlations are drawn. This value should usually not be decreased as it can lead to less accurate results. |
| `replace` | Whether drawing samples is with replacement or not. Default is TRUE, which usually should not be changed. This parameter is mainly of interest for researchers studying the method in more detail. A vector can be used (different values for different rhos). |
| `precision_relative` | Relative precision around the correlation (rho+-rho*precision), if set, it will overwrite precision_absolute. A vector can be used (different values for different rhos). |
| `lower_limit` | Lower limit of corridor, overrides precision parameters. A vector can be used (different values for different rhos). If used, upper_limit must also be set. |

| | |
|---|---|
| upper_limit | Upper limit of corridor, overrides precision parameters. A vector can be used (different values for different rhos). If used, lower_limit must also be set. |
| progress | Should progress bar be displayed? Logical, default is to show progress when run in interactive mode. |
| precision | **[Deprecated]**, use precision_absolute instead |
| precision_rel | **[Deprecated]**, use precision_relative instead |
| rhos | **[Deprecated]**, use rho instead |

## Value

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

The critical points of stability follow directly after the first column (rho) and are named pos.confidence-level, e.g. pos.80, pos.90, pos.95 for the default confidence levels.

## Examples

```
find_critical_pos(rho = 0.5, n_studies = 1e3)
find_critical_pos(rho = 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. This function should only be used if you need the specific points of stability. For instance, if you want to study the method in more detail and the higher level functions are not sufficient.

## Usage

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

## 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. |
| progress | Should progress bar be displayed? Boolean, default is FALSE. |

## Details

If you just want to calculate a quantile of the distribution, use the main function of the package `find_critical_pos()`).

## Value

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

## Examples

```
# set up a population
pop <- fastpos::create_pop(rho = 0.5, size = 1e6)
# create a distribution of points of stability
pos <- simulate_pos(x_pop = pop[,1], y_pop = pop[,2], n_studies = 100,
                    sample_size_min = 20, sample_size_max = 1e3,
                    replace = TRUE, lower_limit = 0.4, upper_limit = 0.6,
                    progress = TRUE)
# calculate quantiles or any other parameter of the distribution
quantile(pos, c(.8, .9, .95))
```

# Index