Package 'wrswoR.benchmark'

July 26, 2020

Type Package

Title Benchmark and Correctness Data for Weighted Random Sampling Without Replacement

Version 0.2.1

Date 2020-07-25

Description Includes performance measurements and results of repeated experiment runs (for correctness checks) for code in the 'wrswoR' package.

License GPL-3

URL https://github.com/krlmlr/wrswoR.benchmark, https://github.com/krlmlr/wrswoR.benchmark

BugReports https://github.com/krlmlr/wrswoR.benchmark/issues

Depends R (>= 3.0.2)

Imports curl, lazyeval

Suggests dplyr, ggplot2, knitr, microbenchmark, rmarkdown, tibble, tidyr

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1.9000

NeedsCompilation no

Author Kirill Müller [aut, cre]

Maintainer Kirill Müller <krlmlr+r@mailbox.org>

Repository CRAN

Date/Publication 2020-07-26 08:40:02 UTC

R topics documented:

`p_values_7` *P-values for n = 7*

Description

Created by `data_raw/p_values_7.R`.

Examples

```
head(p_values_7)
```

`p_values_agg` *Aggregated p-values*

Description

Created by `data_raw/p_values_agg.R`.

Examples

```
head(p_values_agg)
head(p_values_agg_agg)
```

`timings` *Run time data*

Description

Run times measured on an Intel(R) Xeon(R) CPU X5680 clocked at 3.33 GHz with 12 MB cache, running RedHat Enterprise Linux, R 3.2.3 and gcc 4.8.5, using version 0.4 of the `wrswoR` package. The data are created by the corresponding scripts in the `data_raw` directory.

Usage

```
timings_sort
```

Format

An object of class `data.frame` with 25200 rows and 5 columns.

A data frame with 5 columns:

`prob` A description of the probability distribution used. See `data_raw/benchmark.R` for details.

`expr` Function name without the `sample_int_` prefix.

`time` Run time in nanoseconds, as measured by `microbenchmark::microbenchmark()`

`r` Ratio between the `size` and `n` arguments.

`n` The `n` argument.

Details

`timings` contains run times for a larger range of values for the `n` argument.

`timings_sort` contains run times for sorting probabilities with the given distributions.

`break_even` contains detailed run times for the analysis of break-even points between the various implementations.

Examples

```
head(timings)
head(break_even)
```

Index