

# Package ‘benchmarkmeData’

August 19, 2019

**Type** Package

**Title** Data Set for the 'benchmarkme' Package

**Version** 1.0.2

**Maintainer** Colin Gillespie <csgillespie@gmail.com>

**Description** Crowd sourced benchmarks from running the  
'benchmarkme' package.

**License** GPL-2 | GPL-3

**URL** <https://github.com/csgillespie/benchmarkme-data>

**BugReports** <https://github.com/csgillespie/benchmarkme-data/issues>

**Depends** R (>= 3.5.0)

**Imports** dplyr, graphics, tibble, utils

**Suggests** benchmarkme, covr, DT, testthat

**Encoding** UTF-8

**LazyData** TRUE

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**Author** Colin Gillespie [aut, cre]

**Repository** CRAN

**Date/Publication** 2019-08-19 08:40:02 UTC

## R topics documented:

benchmarkmeData-package . . . . .	2
get_datatable_past . . . . .	2
is_blas_optimize . . . . .	3
make_data_set . . . . .	3
past_results . . . . .	4
past_results_v2 . . . . .	4
plot_past . . . . .	4
summarise_results . . . . .	5

**Index****6**

benchmarkmeData-package

*The benchmarkmeData package***Description**

This package contains the results from users running the **benchmarkme** package. The key function is `plot_past()`.

**Author(s)**

<csgillespie@gmail.com>

**See Also**

<https://github.com/csgillespie/benchmarkme-data>

**Examples**

```
plot_past("prog")
```

get\_datatable\_past

*Interactive table of results***Description**

A summary of past results

**Usage**

```
get_datatable_past(test_group, blas_optimize = NULL, cores = 0)
```

**Arguments**

test_group	One of "prog", "matrix_fun", "matrix_cal", "read5", "read50", "read200", "write5", "write50" or "write200". Default value prog.
blas_optimize	Default NULL. The default behaviour is to plot all results. To plot only the BLAS optimized results, set to TRUE, otherwise FALSE.
cores	Default 0, i.e. no parallel.

**Examples**

```
## Need the DT package
## View all results for prog test
get_datatable_past("prog")
```

---

is_blas_optimize	<i>BLAS optimize</i>
------------------	----------------------

---

**Description**

Try to determine parallel BLAS, which implies non-standard R! Compare user with elapsed time. If user » elapsed, then parallel BLAS

**Usage**

```
is_blas_optimize(results)
```

**Arguments**

results	The output from a benchmark_* call.
---------	-------------------------------------

---

make_data_set	<i>Functions for manipulating uploaded results</i>
---------------	--

---

**Description**

Functions used for moving and creating the past\_results\_v2 data set from uploaded data. The move\_files function is used to moved files from the server to another location, whilst removing any empty data sets.

**Usage**

```
make_data_set(from)
```

```
move_files(from, to)
```

**Arguments**

from	A directory containing the uploaded results.
to	Destination directory

**Note**

One of the unit tests uploads an empty results file. Files where the results are NULL are moved to a sub-directory (called) empty in the to directory. If the empty directory doesn't exist, it is created.

Currently these functions are specific to my set-up.

---

past_results	<i>Benchmarking results</i>
--------------	-----------------------------

---

**Description**

A summary of past benchmarks.

**Format**

A data frame

---

past_results_v2	<i>Benchmarking results</i>
-----------------	-----------------------------

---

**Description**

A summary of past benchmarks.

**Format**

A data frame

---

plot_past	<i>Scatter plot of past benchmarks</i>
-----------	--

---

**Description**

Plot the previous benchmarks. This function creates two figures.

- Figure 1: Total benchmark time over all benchmarks (in seconds) on the y-axis.
- Figure 2: Relative time (compared to the smallest benchmark).

The data set used is `data(past_results_v2)`.

**Usage**

```
plot_past(test_group, blas_optimize = NULL, cores = 0, log = "y")
```

**Arguments**

test_group	One of "prog", "matrix_fun", "matrix_cal", "read5", "read50", "read200", "write5", "write50" or "write200". Default value prog.
blas_optimize	Default NULL. The default behaviour is to plot all results. To plot only the BLAS optimized results, set to TRUE, otherwise FALSE.
cores	Default 0, i.e. no parallel.
log	By default the y axis is plotted on the log scale. To change, set the the argument equal to the empty parameter string, "".

**Examples**

```
## Plot all past results for the `prog` benchmark
plot_past("prog", blas_optimize = NULL)
```

---

summarise\_results      *Selecting results*

---

**Description**

Selects and aggregates over the past\_results\_v2 data set or the results input data set..

**Usage**

```
summarise_results(res)

select_results(test_group, results = NULL, blas_optimize = NULL,
               cores = 0)
```

**Arguments**

res	A list containing benchmark results and system information.
test_group	One of "prog", "matrix_fun", "matrix_cal", "read5", "read50", "read200", "write5", "write50" or "write200". Default value prog.
results	Default NULL. If NULL the past_results_v2 data set is used. Otherwise, the input data set.
blas_optimize	Default NULL. The default behaviour is to plot all results. To plot only the BLAS optimized results, set to TRUE, otherwise FALSE.
cores	Default 0, i.e. no parallel.

**Value**

A data frame

**Examples**

```
select_results("prog", blas_optimize = NULL)
```

# Index

## \*Topic **package**

benchmarkmeData-package, 2

benchmarkmeData

(benchmarkmeData-package), 2

benchmarkmeData-package, 2

get\_datatable\_past, 2

is\_blas\_optimize, 3

make\_data\_set, 3

move\_files (make\_data\_set), 3

past\_results, 4

past\_results\_v2, 4

plot\_past, 4

select\_results (summarise\_results), 5

summarise\_results, 5