

Package: dchunkr (via r-universe)

August 14, 2024

Title Work with Chunks of Data

Version 0.0.0.9001

Description Helps to work with chunks of data in parallel and to cache the results of each chunk. It's basic approach to handling somewhat large datasets and long runtimes.

License MIT + file LICENSE

URL <https://github.com/2DegreesInvesting/dchunkr>,
<https://2degreesinvesting.github.io/dchunkr/>,
<https://github.com/maurolepore/dchunkr>,
<https://maurolepore.github.io/dchunkr/>

BugReports <https://github.com/maurolepore/dchunkr/issues>

Imports dplyr, fs, furrr, rappdirs, rlang, tibble, tidyr, tidyselect, utils

Suggests future, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Repository <https://maurolepore.r-universe.dev>

RemoteUrl <https://github.com/maurolepore/dchunkr>

RemoteRef HEAD

RemoteSha 97483508d464c9d41d4a557973c8eaca0f4314b0

Contents

add_file	2
cache_path	2
nest_chunk	3
order_rows	4
pick_undone	4

Index**6**

add_file	<i>Adds the file column to a data frame</i>
----------	---

Description

Adds the file column to a data frame

Usage

```
add_file(data, parent, ext = ".rds")
```

Arguments

data	A data frame.
parent	A directory.
ext	An extension.

Value

A data frame.

Examples

```
data <- tibble::tibble(chunk = 1)
add_file(data, parent = tempdir())
```

cache_path	<i>Create a path to the cache directory, ensuring it exists</i>
------------	---

Description

path() constructs a relative path, path_wd() constructs an absolute path from the current working directory.

Usage

```
cache_path(..., cache_dir = rappdirs::user_cache_dir("dchunkr"))
```

Arguments

...	character vectors, if any values are NA, the result will also be NA. The paths follow the recycling rules used in the tibble package, namely that only length 1 arguments are recycled.
cache_dir	Character. A directory for the cache.

Value

Character. A path.

See Also

[path_home\(\)](#), [path_package\(\)](#) for functions to construct paths relative to the home and package directories respectively.

Examples

```
dir <- withr::local_tempfile()
fs::dir_exists(dir)
cache_path("b", cache_dir = dir)
fs::dir_exists(dir)
```

nest_chunk

Nest a data frame by chunks containing all elements of a group

Description

Nest a data frame by chunks containing all elements of a group

Usage

```
nest_chunk(data, .by, chunks)
```

Arguments

data	A data frame.
.by	A column which values should be.
chunks	Integer. Number of chunks.

Value

A nested data frame.

Examples

```
data <- tibble::tibble(id = c(1, 1, 1, 2, 3))
out <- nest_chunk(data, .by = "id", chunks = 3)
out$data
```

order_rows	<i>Order the rows of a data frame</i>
------------	---------------------------------------

Description

Order the rows of a data frame

Usage

```
order_rows(data, .fun = c("identity", "rev", "sample"))
```

Arguments

data	Data frame.
.fun	Character. A function name.

Value

Data frame.

Examples

```
withr::local_seed(123)

data <- tibble::tibble(x = 1:5)
order_rows(data)
order_rows(data, "rev")
order_rows(data, "sample")
```

pick_undone	<i>Add and filter (un)done files in a data frame</i>
-------------	--

Description

Add and filter (un)done files in a data frame

Usage

```
pick_undone(data)
```

Arguments

data	A data frame with the column file.
------	------------------------------------

Value

A data frame.

Examples

```
data <- tibble::tibble(file = c(
  withr::local_tempdir(pattern = "exists_"),
  withr::local_tempfile(pattern = "doesnt_exist_")
))
data

pick_undone(data)
```

Index

`add_file`, 2

`cache_path`, 2

`nest_chunk`, 3

`order_rows`, 4

`path_home()`, 3

`path_package()`, 3

`pick_undone`, 4