

Package ‘calcUnique’

May 8, 2026

Type Package

Title Simple Wrapper for Computationally Expensive Functions

Version 0.1.2

Author Stephen Froehlich

Maintainer Stephen Froehlich <stephen.froehlich@gmail.com>

Description This is a one-function package that will pass only unique values to a computationally-expensive function that returns an output of the same length as the input. In importing and working with tidy data, it is common to have index columns, often including time stamps that are far from unique. Some functions to work with these such as text conversion to other variable types (e.g. as.POSIXct()), various grep()-based functions, and often the cut() function are relatively slow when working with tens of millions of rows or more.

License MIT + file LICENSE

URL <https://github.com/stephenbfroehlich/calcUnique>

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

Suggests testthat

NeedsCompilation no

Repository CRAN

Date/Publication 2020-05-04 21:50:07 UTC

Contents

| | |
|----------------------|----------|
| calcUnique | 2 |
| Index | 4 |

| | |
|------------|--|
| calcUnique | <i>Pass only unique values to a computationally expensive function that returns an output of the same length as the input.</i> |
|------------|--|

Description

In importing and working with tidy data, it is common to have index columns, often including time stamps that are far from unique. Some functions to work with these such as text conversion, various `grep()`-based functions, and often the `/codecut()` function are relatively slow when working with tens of millions of rows or more.

Usage

```
calcUnique(x, .f, ...)
```

Arguments

| | |
|------------------|--|
| <code>x</code> | A list or vector to be passed to <code>.f</code> |
| <code>.f</code> | The function to be called. It take as in input the incoming <code>x</code> vector or list and it must return a vector or list of the same length as the input. |
| <code>...</code> | Any other arguments to be passed to <code>.f</code> . |

Details

This wrapper function pares down the items to process to only unique values using the `unique()` function. For a function that takes in a vector or list and returns a vector or list the same length, the inputs and outputs are the same as they would be otherwise ... it just happens faster.

Value

The normal output of `.f` as long as it is of the same length as `x`

Examples

```
#Create a sample of some date text with repeats
ts_sample <-
  sample(
    as.character(
      seq(from = as.POSIXct('2020-03-01'), to = as.POSIXct('2020-03-15'), by = 'day')
    ),
    size = 30, replace = TRUE
  )

#Now convert the time text back to POSIXct timestamps:
as.POSIXct(ts_sample)

#Do the same with the calcUnique function:
calcUnique(ts_sample, as.POSIXct)
```

```
#Just to show that the output is the same with and without calcUnique:  
all.equal(as.POSIXct(ts_sample),calcUnique(ts_sample, as.POSIXct))  
  
#An example for when the function doesn't take the vector as the first argument:  
gsub("00","$$", ts_sample)  
calcUnique(ts_sample, function(i) gsub("00","$$", i))
```

Index

`calcUnique`, 2