Package ‘mgsub’

July 28, 2021

Type Package

Title Safe, Multiple, Simultaneous String Substitution

Version 1.7.3

BugReports https://github.com/bmewing/mgsub/issues

Description Designed to enable simultaneous substitution in strings in a safe fashion.
Safe means it does not rely on placeholders (which can cause errors in same length matches).

License MIT + file LICENSE

Encoding UTF-8

ByteCompile true

RoxygenNote 7.1.1

Suggests covr, testthat, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Mark Ewing [aut, cre]

Maintainer Mark Ewing <b.mark@ewingsonline.com>

Repository CRAN

Date/Publication 2021-07-28 19:50:01 UTC

R topics documented:
censor_worker .................................................. 2
fast_replace ....................................................... 2
filter_overlap .................................................... 3
get_matches ....................................................... 3
mgsub ............................................................... 4
mgsub_censor ..................................................... 4
worker ............................................................. 5

Index 6

1
censor_worker

### Description

The hard worker doing everything for mgsub_censor

### Usage

```r
censor_worker(string, pattern, censor, split = any(nchar(censor) > 1),
  seed = NULL, ...)
```

### Arguments

- **string**: a character vector where replacements are sought
- **pattern**: Character string to be matched in the given character vector
- **censor**: character to use in censoring - see details
- **split**: if a multicharacter censor pattern is provided, should it be split to preserve original string length
- **seed**: optional parameter to fix sampling of multicharacter censors
- **...**: arguments to pass to regexpr family

---

fast_replace

### Description

Fast escape function for limited case where only one pattern provided actually matches anything

### Usage

```r
fast_replace(string, pattern, replacement, ...)
```

### Arguments

- **string**: a character vector where replacements are sought
- **pattern**: Character string to be matched in the given character vector
- **replacement**: Character string equal in length to pattern or of length one which are a replacement for matched pattern.
- **...**: arguments to pass to gsub
**filter_overlap**

*Filter overlaps from matches*

**Description**

Helper function used to identify which results from gregexpr overlap other matches and filter out shorter, overlapped results

**Usage**

```r
filter_overlap(x)
```

**Arguments**

- `x` Matrix of gregexpr results, 4 columns, index of column matched, start of match, length of match, end of match. Produced exclusively from a worker function in mgsub

---

**get_matches**

*Get all matches*

**Description**

Helper function to be used in a loop to check each pattern provided for matches

**Usage**

```r
get_matches(string, pattern, i, ...)
```

**Arguments**

- `string` a character vector where replacements are sought
- `pattern` Character string to be matched in the given character vector
- `i` an iterator provided by a looping function
- `...` arguments to pass to gregexpr
**mgsub**  
*Safe, multiple gsub*

**Description**  
mgsub - A safe, simultaneous, multiple global string replacement wrapper that allows access to multiple methods of specifying matches and replacements.

**Usage**  
mgsub(string, pattern, replacement, recycle = FALSE, ...)

**Arguments**
- **string**: a character vector where replacements are sought  
- **pattern**: Character string to be matched in the given character vector  
- **replacement**: Character string equal in length to pattern or of length one which are a replacement for matched pattern.  
- **recycle**: logical. should replacement be recycled if lengths differ?  
- **...**: arguments to pass to `regexp / sub`

**Value**
Converted string.

**Examples**
- mgsub("hey, ho", pattern = c("hey", "ho"), replacement = c("ho", "hey"))  
- mgsub("developer", pattern = c("e", "p"), replacement = c("p", "e"))  
- mgsub("The chemical Dopaziamine is fake",
  pattern = c("dopa(\.*?\)\", "fake"),
  replacement = c("mega\\\1\", "real"),
  ignore.case = TRUE)

**mgsub_censor**  
*Safe, multiple censoring of text strings*

**Description**  
mgsub_censor - A safe, simultaneous, multiple global string censoring (replace matches with a censoring character like '*')

**Usage**  
mgsub_censor(string, pattern, censor = "*", split = any(nchar(censor) > 1), seed = NULL, ...)

---

**Description**  
mgsub - A safe, simultaneous, multiple global string replacement wrapper that allows access to multiple methods of specifying matches and replacements.

**Usage**  
mgsub(string, pattern, replacement, recycle = FALSE, ...)

**Arguments**
- **string**: a character vector where replacements are sought  
- **pattern**: Character string to be matched in the given character vector  
- **replacement**: Character string equal in length to pattern or of length one which are a replacement for matched pattern.  
- **recycle**: logical. should replacement be recycled if lengths differ?  
- **...**: arguments to pass to `regexp / sub`

**Value**
Converted string.

**Examples**
- mgsub("hey, ho", pattern = c("hey", "ho"), replacement = c("ho", "hey"))  
- mgsub("developer", pattern = c("e", "p"), replacement = c("p", "e"))  
- mgsub("The chemical Dopaziamine is fake",
  pattern = c("dopa(.*)\", "fake"),
  replacement = c("mega\1\", "real"),
  ignore.case = TRUE)
**Arguments**

- **string**: a character vector to censor
- **pattern**: regular expressions used to identify where to censor
- **censor**: character to use in censoring - see details
- **split**: if a multicharacter censor pattern is provided, should it be split to preserve original string length
- **seed**: optional parameter to fix sampling of multicharacter censors
- **...**: arguments to pass to `regexpr`/`sub`

**Details**

When censor is provided as a >1 length vector or as a multicharacter string with split = TRUE, it will be sampled to return random censoring patterns. This can be helpful if you want to create cartoonish swear censoring. If needed, the randomization can be controlled with the seed argument.

**Value**

Censored string.

**Examples**

```r
mgsub_censor("Flowers for a friend", pattern=c("low"), censor="*")
```

**Description**

The hard worker doing everything for mgsub

**Usage**

```r
worker(string, pattern, replacement, ...)
```

**Arguments**

- **string**: a character vector where replacements are sought
- **pattern**: Character string to be matched in the given character vector
- **replacement**: Character string equal in length to pattern or of length one which are a replacement for matched pattern.
- **...**: arguments to pass to `regexpr` family
Index

censor_worker, 2
fast_replace, 2
filter_overlap, 3
get_matches, 3
mgsub, 4
mgsub_censor, 4
regexpr, 4, 5
sub, 4, 5
worker, 5