Package ‘rmdwc’

July 23, 2025

Type Package

Title Count Words and Characters in R Markdown and Jupyter Notebooks

Version 0.3.1

Date 2025-05-20

Description Computes word, character, and non-whitespace character counts in R Markdown documents and Jupyter notebooks, with or without code chunks. Returns results as a data frame.

Imports jsonlite, knitr, rstudioapi

Suggests testthat

License GPL-3

URL https://github.com/sigbertklinke/rmdwc

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Sigbert Klinke [aut, cre]

Maintainer Sigbert Klinke <sigbert@hu-berlin.de>

Repository CRAN

Date/Publication 2025-05-20 12:00:02 UTC

Contents

ipynbcount — *Count text elements in Jupyter Notebook files*

Description

This function extracts text from specific cell types (e.g., markdown) in one or more `.ipynb` files and counts the number of characters, words, and lines. It optionally excludes certain patterns (e.g., code fences). The function uses a helper function `rmdcount()` to perform the counting on the extracted text.

Usage

```
ipynbcount(
  files,
  celltype = c("markdown"),
  space = "[[:space:]]",
  word = "[[:space:]]+",
  line = "\n",
  exclude = "```\\{.*?```"
)
```

Arguments

`files` character: vector of paths to `.ipynb` (Jupyter Notebook) files.

`celltype` character: vector indicating which cell types to include (default is `'markdown'`). Valid values include `'markdown'` and `'code'`.

`space` character: pattern to split a text at spaces (default: `'[[:space:]]'`)

`word` character: pattern to split a text at word boundaries (default: `'[[:space:]]+'`)

`line` character: pattern to split lines (default: `'\n'`)

`exclude` character: pattern to exclude text parts, e.g. code chunks (default: '```\\{.*?```')

Details

This function assumes that the notebook files are valid JSON and contain a list of cells under the `cells` field. It temporarily writes the extracted content to a file to reuse the `rmdcount()` logic.

Value

A data frame with counts of characters, words, and lines for each file. Additional columns include `file` (base name) and `path` (directory).

Examples

```
file <- system.file('ipynb/example_data_analysis.ipynb', package="rmdwc")
ipynbcount(file)                                 # without code
ipynbcount(file, celltype=c("markdown", "code"))   # with code
```

rmdcount *Word, character and non-whitespace characters count*

Description

rmdcount counts lines, words, bytes, characters and non-whitespace characters in R Markdown files excluding code chunks. txtcount counts lines, words, bytes, characters and non-whitespace characters in plain text files.
Note that the counts may differ a bit from unix wc and Libre Office because it depends on the definition of a line, a word and a character.

Usage

```
rmdcount(
  files = NULL,
  space = "[[:space:]]",
  word = "[[:space:]]+",
  line = "\n",
  exclude = "```\\{.*?```"
)

txtcount(
  files = NULL,
  space = "[[:space:]]",
  word = "[[:space:]]+",
  line = "\n"
)
```

Arguments

files	character: file name(s)
space	character: pattern to split a text at spaces (default: '[[:space:]]')
word	character: pattern to split a text at word boundaries (default: '[[:space:]]+')
line	character: pattern to split lines (default: '\n')
exclude	character: pattern to exclude text parts, e.g. code chunks (default: '```\\{.*?```')

Details

We define:

Line the number of lines. It differs from unix wc -l since wc counts the number of newlines.

Word it is considered to be a character or characters delimited by white space. However, a "word" is in general a fuzzy concept, for example is "3.141593" a word? Therefore different programs may count differently, for more details see the discussion to the Libreoffice bug Word count gives wrong results - Another Example Comment 5.

The following approach is used to detect lines, words, characters and non-whitespace characters.

lines `strsplit(rmd, line)[[1]]` with `line='\n'`

bytes `charToRaw(rmd)`

words `strsplit(rmd, word)[[1]]` with `word='[[:space:]]+'`

characters `strsplit(rmd, '')[[1]]`

non-whitespace characters `strsplit(gsub(space, '', rmd), '')[[1]]` with `space='[[:space:]]'`

If `txtcount` is used then code chunks are deleted with `` gsub('```\\{.*?```', '', rmd) `` before counting.

Value

a data frame with following elements

file basename of file

lines number of lines

words number of words

bytes number of bytes

chars number of characters

nonws number of non-whitespace characters

path path of file

Examples

```
# count excluding code chunks
files <- system.file('rmarkdown/rstudio_pdf.Rmd', package="rmdwc")
rmdcount(files)
# count including code chunks
txtcount(files) # or rmdcount(files, exclude='')
# count for a set of R Markdown docs
files <- list.files(path=system.file('rmarkdown', package="rmdwc"),
                    pattern="*.Rmd", full.names=TRUE)
rmdcount(files)
# use of rmdcount() in a R Markdown document
if (interactive()) {
  files <- system.file('rmarkdown/rstudio_pdf.Rmd', package="rmdwc")
  file.edit(files) # SAVE(!) the file and knit it
}
# count including code chunks
files <- system.file('rmarkdown/rstudio_pdf.Rmd', package="rmdwc")
txtcount(files)
```

rmdcountAddin *rmdcountAddin*

Description

Applies `rmdcount` to the current R Markdown document

Usage

```
rmdcountAddin()
```

Value

nothing

Examples

```
if (interactive()) rmdcountAddin()
```

rmdwcl *Word-, character and non-whitespace characters count for a text*

Description

Counts words, characters and non-whitespace characters in a string. Is used in `rmdcount`, see details there.

Usage

```
rmdwcl(rmd, space = "[[:space:]]", word = "[[:space:]]+", line = "\n")
```

Arguments

`rmd`	character: R Markdown document as string
`space`	character: pattern to split a text at spaces (default: `'[[:space:]]'`)
`word`	character: pattern to split a text at word boundaries (default: `'[[:space:]]+'`)
`line`	character: pattern to split lines (default: `'\n'`)

Value

a list

Examples

```
file  <- system.file('rmarkdown/rstudio_pdf.Rmd', package="rmdwc")
fcont <- readChar(file, file.info(file)$size)
rmdwcl(fcont)
```

Index