Package ‘commentr’

October 12, 2022

Type Package

Title Print Nicely Formatted Comments for Use in Script Files

Version 1.0.4

License GPL-2

Date 2016-03-19

Author Erik Bulow

Maintainer Erik Bulow <erik.bulow@rccvast.se>

Description Functions to
produce nicely formatted comments to use in R-scripts (or
Latex/HTML/markdown etc). A comment with formatting is printed to the
console and can then be copied to a script.

Imports stringr

Suggests testthat

BugReports https://bitbucket.org/cancercentrum/commentr/issues?
status=new&status=open

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-03-19 11:05:39

R topics documented:

block_comment 2
comment_width 4

Index **6**

block_comment *Create nicely formated comments*

Description

Functions to automate creation of nicely formated comments. Comments are printed to the console and automaticaly pasted to clipboard (and can thereafter be copied to script files.

Usage

```
block_comment(description = "", empty_lines_first = 1,
  empty_lines_last = empty_lines_first, allign = "center", token = "#",
  html = FALSE, clipboard = TRUE, verbose = TRUE, ...)

header_comment(title, description = "", author = getOption("name"),
  contact = getOption("contact"), client = author,
  date_created = format(Sys.time(), "%Y-%m-%d"),
  date_updated = date_created, source = getwd(), tab = 17, token = "#",
  html = FALSE, clipboard = TRUE, verbose = TRUE, ...)

line_comment(title = "", ..., clipboard = TRUE, verbose = TRUE,
  token = "#", html = FALSE)
```

Arguments

`description` additional information about the script. Could be a single line or a full paragraph. Empty character string as default.

`empty_lines_first` number of empty lines to be added before the text in a block comment. 1 by default.

`empty_lines_last` number of empty lines to be added after the text in a block comment. Same as `empty_lines_first` by default.

`allign` specify text allignment. "center" as default with other options "left" and "right". If the text is too long to be fit on one line, it is split to a left aligned paragraph (`allign` being ignored).

`token` a character string specifying the comment character to be used. Default is `"#"`, which should be used in R scripts.

`html` Should the comment be used in a HTML- or R Markdown-document (FALSE by default). If so, the comment starts with "<!–" and ends with "–>". `token = "%"` could be used to create comments for LaTeX and `token = "*"` for Stata etcetera. The length of the character string is not restricted but should normally be just one character. If a longer character string is used, the width of the comment would be multiplied by the number of characters. This might result in a quite messy output.

`clipboard` Should the output be copied to clipboard? (`TRUE` by default). Corrently only supported on Mac.

`verbose` Should the comment be printed to the console? (`TRUE` by default). This could be used if `clipboard = TRUE` and run on a Mac.

`...` arguments passed to comment_width to specify the width of the comment.

`title` a short and descriptive title.

`author` name of the script author(s). From `getOption("name")` as default.

`contact` How to contact the author of the script. From `getOption("contact")` as default.

`client` Name (and possible contact information) to the client for who the script was written. Sets author as default (assuming you wrote the script for yourself).

`date_created` A character string defining the date when the script was created. Todays date as default. The date is specified as a string and does not need to be exact. For example "Many years ago" would be accepted if you want to add a heading to an old script.

`date_updated` A character string defining the date when the script was last updated. Same as date_created by default.

`source` is the path to where the script was originaly stored . This information might be good if the document is printed, forgotten and found. Current directory as default.

`tab` "tabulation length" to add space betwen left column titles and right column text.

Details

The `header_comment` function uses global options for name, contact and textwidth as default. Width is a standard option in R but name and contact are not. It might be a good idea to define those as global options in a .Rprofile-script (use Google if you do not know the concept!). Dates for creation and last update are both set to the same date by default. The update date has to be manually updated in the script

Value

There is no object returned from the function call. There is just a printed message to the console that could be copied to the beginning of a script).

Functions

There are 3 functions to help with script comments

- header_comment: a header introduction to be included at the top of the script
- block_comment: a block comment to make a title for a new section in the script
- line_comment: a one line comment

Examples

```
## Not run:

# If global options specifies "author" and "contact", these do not need to be specified every time:
header_comment("Test", "This is a little test")

header_comment("Test", "This is just a test!", width = "script_width")

header_comment("Smaller block", "This is a smaller test block!", width = 55, tab = 17)

header_comment("Smaller block", "This is a small test block but with a longer extra description
   that has to be split from a single line into a full paragraph.", width = 55, tab = 17)

## End(Not run)

header_comment("Nice script", "This is a very nice script that is well documented",
     author       = "Jane Doe",
     contact      = "jane@doe.se",
     client       = "John Doe",
     date_created = "2014-07-03",
     width        = "a4landscape")

block_comment("A title for a new section in the script")
block_comment("A shorter box", width = 50)
block_comment("A compact title", empty_lines_first = 0, allign = "left")
## Not run:
block_comment("A longer descriptive text that has to be
   separated into several lines in order to fit.
   Then it is no longer alligned to 'center' even if so specified!", allign = "center")

## End(Not run)
line_comment("A comment in the middle of a line")
line_comment("A comment in the middle of a shorter line", 50)
```

comment_width *Function to specify the width of the comment*

Description

Helper function to the commet family of functions. Can take both numerical values as the width in characters or a prespecified standard through a character string.

Usage

```
comment_width(width = "option", a4portrait_width = 80)
```

Arguments

`width` is the prefered width. Could be numeric (number of characters), a paper size (currently `"a4portrait"`, `"a4landscape"`, `"a3portrait"` or `"a3landscape"`), `"script_width"` (= `getOption("width") - 5`) or `"option"` (to get data from global option `"comment_width"`). Default is `"option"` but if a global option does not exist, "a4portrait".

`a4portrait_width` specifies the number of characters that can be printed on a single line on a a4 paper in portrait orientation. This value is usually 80 (or in the range from 60 to 75). Here a lower value is set by default due to experimentation on the authors own computer. Please contact the author if this value seems strange. Note however that it is uslly more sufficient to use a global option for the `width` parameter than to manually change this value (whih is however possible for increased flexibility). This value depends on margins and font size when printing.

Details

The width is specified in order to fit the comment on a printed page or in a script file showed in RStudio (hence the -5 in `getOption("width") - 5` to adjust for the line numbering in the script file compared to the console. It is also possible to use a global option `comment_width`. This is not a standard option and has to be manually specified (for example in a .Rprofile-file).

Value

An integer specifiing the text width (in number of characters) to be used in a comment.

Examples

```
comment_width()
comment_width(42)
comment_width("a4portrait")

## Not run:
# We can set a global option for the comment_width
options(comment_width = 80)
header_comment("Test", "A small test")

## End(Not run)
```

Index

block_comment, 2

comment *(block_comment)*, 2
comment_width, *3*, 4
commentr *(block_comment)*, 2

header_comment *(block_comment)*, 2

line_comment *(block_comment)*, 2