Package 'bannerCommenter'

May 7, 2026

Type Package

Title Make Banner Comments with a Consistent Format

Version 1.0.0

Author Bill Venables <Bill.Venables@gmail.com>

Maintainer Bill Venables <Bill.Venables@gmail.com>

Description A convenience package for use while drafting code. It facilitates making stand-out comment lines decorated with bands of characters. The input text strings are converted into R comment lines, suitably formatted. These are then displayed in a console window and, if possible, automatically transferred to a clipboard ready for pasting into an R script. Designed to save time when drafting R scripts that will need to be navigated and maintained by other programmers.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests knitr, rmarkdown, dplyr, datasets

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2021-03-23 08:30:09 UTC

Contents

banner *Make a decorated comment in an R script*

Description

Make a decorated multi-line comment from input strings and, if possible, transfer it to the clipboard ready for pasting into an R script (via the `print` method).

Usage

```
banner(
  x,
  ...,
  emph = FALSE,
  snug = FALSE,
  upper = emph,
  centre = !fold,
  leftSideHashes = 2 + emph,
  rightSideHashes = leftSideHashes,
  minHashes = (!snug) * (65 + 10 * emph),
  numLines = 1 + emph,
  bandChar = "#",
  center = centre,
  fold = FALSE,
  maxChar = 75
)

section(..., emph = TRUE, centre = TRUE, fold = TRUE)

boxup(..., rightSideHashes = 1, bandChar = "-")

open_box(
  ...,
  minHashes = 0,
  rightSideHashes = 0,
  centre = FALSE,
  bandChar = "-",
  center
)

block(
  ...,
  leftSideHashes = 3,
  rightSideHashes = 0,
  centre = FALSE,
  minHashes = 0,
  numLines = 0,
```

```
  center
)
```

Arguments

x	A string, first line of the comment. If "", the zero-length string, only the top lines of the banner are made. If missing, in an interactive session the user will be prompted for the input strings, one per line, in the console.
...	Zero or more additional strings as extra lines. Strings may contain newline characters resulting in further line breaks.
emph	A logical value: Do you want this to be an emphasised comment?
snug	A logical value: Do you want the decoration to hug the strings closely?
upper	A logical value: Do you want the strings converted to upper case?
centre	A logical value: Do you want the text strings centred? (alternative: left justified)
leftSideHashes	A positive integer: How many hashes go on the left side?
rightSideHashes	A non-negative integer: How many hashes go on the right side?
minHashes	A non-negative integer: What is the minimum number of hashes in the boundary lines?
numLines	A non-negative integer: How many lines of hashes above and below do you want?
bandChar	A single character. Used instead of # for all characters in the bands around the text, apart from the first character of every line.
center	Alternative spelling of centre.
fold	Logical: should the text be folded to ensure lines are not too long?
maxChar	Ingeter: maximum length allowed in any line if fold is TRUE.

Value

A character string vector returned invisibly, but automatically displayed in the console

Functions

- section: Make a prominent banner such as might be useful at the beginning of a major code section
- boxup: Make a minimally boxed banner comment
- open_box: Make a boxed banner coment open at the right
- block: Make a simple block of comment lines

Examples

```
banner("This should appear clearly and stand out.",
       "The lines are left justified by default.")
section("This is the first line in a section heading",
        "and this is the second\nand this the third.")
boxup("This is a less obtrusive comment",
      "centred on multiple lines", center = TRUE, bandChar = ".")
banner("This is an important side issue.", "Take note!",
      snug = TRUE, bandChar = "=")
open_box("This is a succinctly presented comment",
         "left justified and open at the right",
         "on multiple lines")

block("This is a chatty comment.  Entering it this way just",
      "saves a tiny bit of typing but it can be useful if",
      "you need multiple initial hash marks, as you may when",
      "using Emacs/ESS, for example.",
      "Or if you want the lines centred for some odd reason.",
      center = TRUE)
## some authors like to use lines of a uniform length to separate code sections:
boxup("")
```

---

copy_to_clipboard *Transfer text strings to the clipboard ready for paste*

Description

This is only guaranteed for Windows; in the case of Linux you will need to have the `xclip` command installed and visible on the `PATH` and for Mac OS you will need to have `pbcopy` similarly available. In any case the transfer to the clipboard is only activated while in an interactive session.

Usage

```
copy_to_clipboard(x, ..., file = con)
```

Arguments

`x`	a characeter string vector
`...`	additional arguments as for `cat`
`file`	a file or connection (usually left at the default)

Details

It behaves like `base::cat` but differs in three respects.

First, if `file` is left missing, in an interactive session, the default file is a clipboard device, if possible.

Second, the return value is `invisible(x)` rather than `invisible(NULL)` as it is for `base::cat`.

Third, it only has a copying side-effect if used in an interactive session. In a non-interactive session it merely returns the x argument, invisibly.

Note the on Windows the function utils::writeClipboard offers a much more extensive range of possibilities for communicating with the clipboard device, but this facility is only available on Windows.

Value

x, invisibly (as for a print method)

---

print.banner *Print method for banner objects*

Description

As well as printing the comment string in the console window the same text strings are transferred to a clipboard, if possible, ready for pasting into the R script currently being drafted.

Usage

```
## S3 method for class 'banner'
print(x, ...)
```

Arguments

x	A character string vector as procuced by banner()
...	Not used

Value

x itself, invisibly. A side effect is that x is transferred to a clipboard device, if possible

---

yaml_header *Functions for YAML comments*

Description

A facility for generating a comment block for inserting at the top of an R script to insert a yaml header and for setting chunk options within scripts to use the RStudio facility for automatically rendering scripts into documents.

Usage

```
yaml_header(title, author, ...)

chunk_options(...)
```

Arguments

title	A character string; if omitted may be supplied via the console
author	A character string; if omitted may be supplied via the console
...	Extra arguments, for yaml_header currently ignored; for chunk_options comma separated chunk options

Value

An incomplete yaml header, or a line of chunk options, in the clipboard by default

Examples

```
yaml_header(title = "My script",
            author = "Bill Venables")
chunk_options(comment="",
              fig.height=7,
              fig.width=8,
              out.width="90%")
```

Index