

Package ‘svSweave’

October 14, 2022

Type Package

Version 1.0.0

Date 2022-05-10

Title 'SciViews' - 'Sweave', 'Knitr' and R Markdown Companion
Functions

Description Functions to enumerate and reference figures, tables and equations in R Markdown documents that do not support these features (thus not 'bookdown' or 'quarto'). Supporting functions for using 'Sweave' and 'Knitr' with 'LyX'.

Maintainer Philippe Grosjean <phgrosjean@sciviews.org>

Depends R (>= 2.6.0)

Imports utils, knitr, rmarkdown

Suggests covr, testthat, spelling

License GPL-2

URL <https://github.com/SciViews/svSweave>,
<https://www.sciviews.org/svSweave/>

BugReports <https://github.com/SciViews/svSweave/issues>

RoxygenNote 7.1.1

VignetteBuilder knitr

Encoding UTF-8

Language en-US

NeedsCompilation no

Author Philippe Grosjean [aut, cre] (<<https://orcid.org/0000-0002-2694-9471>>)

Repository CRAN

Date/Publication 2022-05-10 09:30:02 UTC

R topics documented:

clean_lyx	2
fig_id	5
new_labelling	5
subsettable	7

Index	9
--------------	----------

clean_lyx	<i>Clean up, weave or tangle Sweave files produced by LyX with the SciViews Sweave module</i>
-----------	---

Description

These functions process `.Rnw` files produced by LyX and the SciViews Sweave module (not the standard Sweave commands to contain R chunks (embedded R code processed by Sweave). Unfortunately, LyX use to write two lines feeds for each line of code, introducing extra lines in the R chunks. Moreover, tabulations are interpreted as 8 spaces, while R code use to consider a tabulation as equivalent to 4 spaces. `clean_lyx()` corrects these little problems, and it should not affect R noweb files produced by a different software.

Usage

```
clean_lyx(RnwCon, RnwCon2 = RnwCon, encoding = "UTF-8")
```

```
cleanLyxRnw(RnwCon, RnwCon2 = RnwCon, encoding = "UTF-8")
```

```
tangle_lyx(
  file,
  driver = Rtangle(),
  syntax = getOption("SweaveSyntax"),
  encoding = "UTF-8",
  width = 80,
  useFancyQuotes = TRUE,
  annotate = TRUE,
  logFile = file.path(tempdir(), ".lyxSweave.log"),
  ...
)
```

```
tangleLyxRnw(
  file,
  driver = Rtangle(),
  syntax = getOption("SweaveSyntax"),
  encoding = "UTF-8",
  width = 80,
  useFancyQuotes = TRUE,
  annotate = TRUE,
```

```
    logFile = file.path(tempdir(), ".lyxSweave.log"),
    ...
)

purl_lyx(
  file,
  encoding = "UTF-8",
  width = 80,
  useFancyQuotes = TRUE,
  logFile = file.path(tempdir(), ".lyxSweave.log"),
  ...
)

purlLyxRnw(
  file,
  encoding = "UTF-8",
  width = 80,
  useFancyQuotes = TRUE,
  logFile = file.path(tempdir(), ".lyxSweave.log"),
  ...
)

weave_lyx(
  file,
  driver = RweaveLatex(),
  syntax = getOption("SweaveSyntax"),
  encoding = "UTF-8",
  width = 80,
  useFancyQuotes = TRUE,
  logFile = file.path(tempdir(), ".lyxSweave.log"),
  ...
)

weaveLyxRnw(
  file,
  driver = RweaveLatex(),
  syntax = getOption("SweaveSyntax"),
  encoding = "UTF-8",
  width = 80,
  useFancyQuotes = TRUE,
  logFile = file.path(tempdir(), ".lyxSweave.log"),
  ...
)

knit_lyx(
  file,
  encoding = "UTF-8",
  width = 80,
```

```

    useFancyQuotes = TRUE,
    logFile = file.path(tempdir(), ".lyxSweave.log"),
    ...
)

knitLyxRnw(
  file,
  encoding = "UTF-8",
  width = 80,
  useFancyQuotes = TRUE,
  logFile = file.path(tempdir(), ".lyxSweave.log"),
  ...
)

```

Arguments

RnwCon	A connection object or a character string corresponding to the path to a R noweb file to be read.
RnwCon2	Idem, but where the cleaned up R noweb file should be written (by default, on the same file or connection).
encoding	The encoding of the .Rnw file. It is UTF-8 by default, but you can change it here.
file	The Sweave source file.
driver	The actual function to do the process, see Sweave() .
syntax	NULL or an object of class 'SweaveSyntax' or a character string with its name, see Sweave() .
width	The width used for outputs, 80 characters by default.
useFancyQuotes	Do we use fancy quotes in R outputs?
annotate	Is the R code extracted from the .Rnw file annotated?
logFile	The file to use to log results of weaving/tangling the document.
...	Further arguments passed to the driver's setup function of Sweave() or Stangle() .

Value

For [clean_lyx\(\)](#), a list for Sweave options found in the document; NULL for the other functions: these functions are invoked for their side effects. The function [weave_lyx\(\)](#) uses the standard Sweave driver (but it uses knitr for LyX documents that use the SciViews Knitr module), while [knit_lyx\(\)](#) does the same, but using the knitr driver. Similarly, [purl_lyx\(\)](#) is the knitr counterpart of [tangle_lyx\(\)](#) standard tangling function.

Author(s)

Philippe Grosjean

See Also

[knitr::knit\(\)](#), [utils::Sweave\(\)](#)

fig_id	<i>Create a figure id from a chunk label</i>
--------	--

Description

This function looks at the current chunk label and returns `id="fig:label"` that is usable in the `out.extra=` field of the R chunk. It allows to refer to a figure generated from a chunk with this label. Use `out.extra=chunk_id()` to set the id, or use `fig_id_auto()`.

Usage

```
fig_id(label)
```

```
fig_id_auto()
```

Arguments

`label` The label to use. If provided, it supersedes the chunk label.

Value

A string to set the id like `id="fig:label"`. For `fig_id_auto()`, the function installs a hook in 'knitr' to add an id automatically for each plot make by changing `out.extra=`.

Examples

```
fig_id("my_label")
```

new_labelling	<i>Reference figures, tables and equations in R Markdown documents</i>
---------------	--

Description

These functions return closures that allow for constructing a series of numbered items and to reference them. The number is create the first time a label is encountered, and provided again for further use of the same label.

Usage

```
new_labelling(
  type = c("arabic", "roman"),
  string_html = paste0("<style>.+----%%%::after{content:\"####\"}</style>",
    "<span class=\"figheader\">Figure\\ ####: </span>@@@"),
  string_latex = "@@@@",
  string_word = "[Figure\\ ####:]{#++++:%%}% @@@@",
  string_ref_html = "<a class=\".+----%%%\" href=\"#++++:%%%\"></a>",
```

```

string_ref_latex = "\\ref{++++:%%%",
string_ref_word = "[####](#++++:%%%",
name = "fig"
)

newLabelling(
  type = c("arabic", "roman"),
  string_html = paste0("<style>++++-%%%::after{content:\\\"####\\\"}</style>",
    "<span class=\\\"figheader\\\">Figure\\ \\####: </span>@@@@"),
  string_latex = "@@@@",
  string_word = "[Figure\\ \\####:]{#++++:%%%} @@@@",
  string_ref_html = "<a class=\\\"++++-%%%\\\" href=\\\"#++++:%%%\\\"></a>",
  string_ref_latex = "\\ref{++++:%%%",
  string_ref_word = "[####](#++++:%%%",
  name = "fig"
)

fig(
  caption = "",
  label = knitr::opts_current$get("label"),
  ref = NULL,
  reset = FALSE
)

tab(
  caption = "",
  label = knitr::opts_current$get("label"),
  ref = NULL,
  reset = FALSE
)

eq(label, ref, reset = FALSE)

```

Arguments

type	The type of numbering (arabic or roman).
string_html	The string prototyping the legend, with ++++ being the name (fig by default) @@ being a placeholder for the text, #### as a placeholder for the number, or %% as a placeholder for the label.
string_latex	Idem for LaTeX.
string_word	Idem for Word.
string_ref_html	Idem for reference in HTML format.
string_ref_latex	Idem for reference in LaTeX format.
string_ref_word	Idem for reference in Word format.

name	The name to use before the number, e.g., "Fig." to get "Fig. 1" as cross-reference text for the first figure. If you provide name = NULL, only the number is produced.
caption	The text of the caption.
label	A short string uniquely identifying the item within the collection. To set a label in an equation, give a name instead of a string.
ref	The reference to the label.
reset	If TRUE, the collection is reset. Useful if you want to restart numbering at the beginning of each chapter.

Details

A new labelling type is created using `new_labelling()` which is a function factory (a function that creates functions).

Value

The function returns a caption if `text =` is provided, or an anchor if it is missing. If `text=` contains a name, it returns a link. Same for the `label=` for `eq()`: if it is a text, a couple label + tag to place inside display equations is produced, and if it is a name, a link is provided. `new_labelling()` creates a new labelling function, which has the same arguments as `fig()`.

Author(s)

Philippe Grosjean

Examples

```
# These function are supposed to be used in an R Markdown document
# see the svSweave vignette
# Produce a caption that contains the required code to number and reference
# a figure in HTML documents
fig("A caption", label = "a_label")
# Produce a reference to that figure
fig$a_label
```

subsettable

Define a function as being 'subsettable' using \$ operator

Description

For labelling items like `fig()`, `tab()` or `eq()`, implements the `$` method to retrieve a reference and build a link to the element.

Usage

```
## S3 method for class 'subsettable_labelling_ref'
x$name
```

Arguments

x	A <code>subsettable_labelling_ref</code> function.
name	The value to use for the <code>ref=</code> argument.

Examples

```
eq(pythagoras) # Create a label / tag pair for R Markdown display equations  
eq$pythagoras # Create a link to the equation somewhere else in the document
```


Index

- * **Literate programming**
 - clean_lyx, 2
- * **automatic numbering of items in documents**
 - new_labelling, 5
- * **create 'subsettable' functions**
 - subsettable, 7
- * **utilities**
 - clean_lyx, 2
 - new_labelling, 5
 - subsettable, 7
- \$.subsettable_labelling_ref
 - (subsettable), 7

- clean_lyx, 2
- clean_lyx(), 2, 4
- cleanLyxRnw (clean_lyx), 2

- eq (new_labelling), 5
- eq(), 7

- fig (new_labelling), 5
- fig(), 7
- fig_id, 5
- fig_id_auto (fig_id), 5
- fig_id_auto(), 5

- knit_lyx (clean_lyx), 2
- knit_lyx(), 4
- knitLyxRnw (clean_lyx), 2
- knitr::knit(), 4

- new_labelling, 5
- new_labelling(), 7
- newLabelling (new_labelling), 5

- purl_lyx (clean_lyx), 2
- purl_lyx(), 4
- purlLyxRnw (clean_lyx), 2

- Stangle(), 4

- subsettable, 7
- Sweave(), 4

- tab (new_labelling), 5
- tab(), 7
- tangle_lyx (clean_lyx), 2
- tangle_lyx(), 4
- tangleLyxRnw (clean_lyx), 2

- utils::Sweave(), 4

- weave_lyx (clean_lyx), 2
- weave_lyx(), 4
- weaveLyxRnw (clean_lyx), 2