Package 'quartabs'

May 9, 2026

Title Dynamically Generate Tabset Panels in 'Quarto' HTML Documents

Version 0.1.1

Description Dynamically generate tabset panels <https://quarto.org/docs/output-formats/html-basics.html#tabsets> in 'Quarto' HTML documents using a data frame as input.

License MIT + file LICENSE

URL https://sayuks.github.io/quartabs/, https://github.com/sayuks/quartabs

BugReports https://github.com/sayuks/quartabs/issues

Imports stats

Suggests altdoc, dplyr (>= 1.0.0), DT, flextable, gt (>= 0.9.0), htmltools, knitr, plotly, purrr, quarto, reactable, sessioninfo, spelling, testthat (>= 3.0.0), tibble, tidyr, tinytable, utils

Config/testthat/edition 3

Encoding UTF-8

Language en-US

RoxygenNote 7.3.2

NeedsCompilation no

Author Yusuke Sasaki [aut, cre]

Maintainer Yusuke Sasaki <sayuks.dev@gmail.com>

Repository CRAN

Date/Publication 2025-03-31 02:20:02 UTC

Contents

render_tabset *Dynamically Generate Tabset Panels in Quarto HTML Documents*

Description

`render_tabset()` takes a data frame as input and outputs the markdown that generates the tabset to stdout (console). **Only works with Quarto HTML documents.** See Get started for details.

Usage

```
render_tabset(
  data,
  tabset_vars,
  output_vars,
  layout = NULL,
  heading_levels = NULL,
  pills = FALSE,
  tabset_width = "default"
)
```

Arguments

`data`	A data frame.
`tabset_vars`	Columns to use as tabset labels. Internally passed to the `select` argument of `subset()`. Accepts raw column names, strings, numbers and logical values.
`output_vars`	Columns to display in each tabset panel. Internally passed to the `select` argument of `subset()`. Accepts raw column names, strings, numbers and logical values.
`layout`	`NULL` or a character vector of length 1 for specifying layout in tabset panel. If not `NULL`, `layout` must begin with at least three or more repetitions of ":" (e.g. ":::"). Closing div (e.g. ":::") is inserted automatically. See for details: `https://quarto.org/docs/authoring/figures.html#complex-layouts`.
`heading_levels`	`NULL` or a vector consisting of natural numbers and missing values. The length is equal to the number of columns specified in `tabset_vars`. This controls whether it is partially (or entirely) displayed as normal header instead of tabset.

- If `heading_levels` is a `NULL`, all output is tabset.
- If `heading_levels` is a vector of positive natural number, the elements of the vector correspond to the columns specified in `tabset_vars`.
  - If the element is integer, the tabset column is displayed as headers with their level, not tabset. (e.g. 2 means h2 header). Levels 1 to 6 are recommended. The reason is that quarto supports headers up to 6. 7 and above will also work, but they are displayed as normal text. In addition, considering the chapter format, it is preferable to gradually increase the level, as in 1, 2 and 3.
  - If the element is NA, tabset is displayed.

pills	Logical, use pills or not. See https://getbootstrap.com/docs/5.2/components/navs-tabs/#pills for details. If heading_levels is specified, this will be ignored.
tabset_width	Character, one of "default", "fill" and "justified". See https://getbootstrap.com/docs/5.2/components/navs-tabs/#fill-and-justify for details. If heading_levels is specified, this will be ignored.

Details

- Write #| results: asis at the beginning of the chunk or results='asis' in the chunk options.
- If multiple tabset_vars are given, create nested tabsets.
- For columns specified in output_vars, columns of type list are output with print() and normal columns are output with cat().
- The data is sorted internally by tabset_vars.
- If tabset_vars or output_vars have "factor", "Date" and "POSIXt" columns, they are converted internally to character. This is to prevent it being displayed as numeric when cat() is executed. Sorting by tabset_vars is performed before conversion to string.

Value

NULL invisibly. This function outputs the markdown that generates the tabset to stdout (console).

Limitations

- layout is intended for simplified use cases and complex layouts may not work.
- When outputting tables or figures that use JavaScript (such as {plotly}, {leaflet}, {DT}, {reactable}, etc.), it seems JavaScript dependencies need to be resolved. A simple solution is to wrap the output in htmltools::div() and create a dummy plot in another chunk. See the Get started for details.
- When tabset_vars and output_vars have the following columns, they may not display well:
  - A column of type list contains a named vector or list (This is for output_vars. tabset_vars must not contain list columns).
  - Classes with their own printing methods, such as "difftime", "ts", .etc.
- When specifying a list-type column that includes ggplot objects in output_vars, setting the chunk option echo: fenced may cause the plots to not display correctly.

References

As this function is focused on quickly and dynamically generating tabsets and chunks, it is difficult to customize it on a chunk-by-chunk basis. The regular way to dynamically create chunks is to use functions such as knitr::knit(), knitr::knit_child(), knitr::knit_expand(), etc. For more information on these, see the following links.

- Heiss, Andrew. 2024. "Guide to Generating and Rendering Computational Markdown Content Programmatically with Quarto." November 4, 2024. doi:10.59350/pa44jcc302.

- https://bookdown.org/yihui/rmarkdown-cookbook/child-document.html#child-document
- https://bookdown.org/yihui/rmarkdown-cookbook/knit-expand.html

Examples

```
# sample data
df <- data.frame(
  group1 = c(rep("A", 3), rep("B", 3)),
  group2 = rep(c("X", "Y", "Z"), 2),
  value1 = 1:6,
  value2 = letters[1:6]
)

# Here are examples of the output before it is converted to tabset.
# If you want it to actually work, in the .qmd file,
# set `results='asis'` in the chunk options or
# write `#| results: asis` at the beginning of the chunk.

# Basic usage
render_tabset(df, group1, value1)

# Nested tabset, two outputs side by side with a width of 1:1
render_tabset(
  df,
  c(group1, group2),
  c(value1, value2),
  layout = "::: {layout-ncol=2}"
)

# Use heading instead of tabset
render_tabset(
  df,
  c(group1, group2),
  value1,
  heading_levels = c(2, 3)
)
```

Index