Package 'kableExtra'

May 23, 2018

Type Package

Title Construct Complex Table with 'kable' and Pipe Syntax

Version 0.9.0

Description Build complex HTML or 'LaTeX' tables using 'kable()' from 'knitr' and the piping syntax from 'magrittr'. Function 'kable()' is a light weight table generator coming from 'knitr'. This package simplifies the way to manipulate the HTML or 'LaTeX' codes generated by 'kable()' and allows users to construct complex tables and customize styles using a readable syntax.

License MIT + file LICENSE

LazyData TRUE

URL http://haozhu233.github.io/kableExtra/, https://github.com/haozhu233/kableExtra

BugReports https://github.com/haozhu233/kableExtra/issues

Depends R (>= 3.1.0)

Imports knitr (>= 1.16), magrittr, stringr (>= 1.0), xml2 (>= 1.1.1), rvest, rmarkdown (>= 1.6.0), readr, scales, viridisLite, stats, grDevices, htmltools, rstudioapi

Suggests testthat, magick, formattable, dplyr

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 6.0.1

NeedsCompilation no

Author Hao Zhu [aut, cre] (<https://orcid.org/0000-0002-3386-6076>), Thomas Travison [ctb], Timothy Tsai [ctb], Will Beasley [ctb], Yihui Xie [ctb], GuangChuang Yu [ctb], Stéphane Laurent [ctb],

Rob Shepherd [ctb],
Yoni Sidi [ctb],
Brian Salzer [ctb],
George Gui [ctb],
Yeliang Fan [ctb]

Maintainer Hao Zhu <haozhu233@gmail.com>

Repository CRAN

Date/Publication 2018-05-21 07:22:58

R topics documented:

kableExtra-package *kableExtra*

Description

When we are talking about table generators in R, knitr's `kable()` function wins lots of flavor by its ultimate simplicity. Unlike those powerful table rendering engines such as xtable, the philosophy behind knitr::kable() is to make it easy for programmers to use. Just as it claimed in its function description, "this is a very simple table generator. It is simple by design. It is not intended to replace any other R packages for making tables. - Yihui".

However, the ultimate simplicity of `kable()` also brought troubles to some of us, especially for new R users, who may not have a lot of experience on generating tables in R. It is not rare to see people including experienced users asking questions like how to center/left-align a table on Stack Overflow. Also, for me personally, I found myself repeatedly parsing CSS into `kable()` for some very simple features like striped lines. For LaTeX, it's even worse since I'm almost Stack Overflow dependent for LaTeX... That's why this package `kableExtra` was created.

I hope with `kableExtra`, you can

- Use default base `kable()` (Or a good alternative for markdown tables is `pander::pander()`) for all simple tables
- Use `kable()` with `kableExtra` to generate 90 tables in either HTML or LaTeX
- Only have to mess with raw HTML/LaTeX in the last 10 `kableExtra` cannot solve the problem

For a full package documentation, please visit the package documentation site for more information

Features

Pipable syntax: `kableExtra` is NOT a table generating package. It is a package that can "add features" to a `kable` output using a syntax that every useR loves - the pipe. We see similar approaches to deal with plots in packages like `ggvis` and `plotly`. There is no reason why we cannot use it with tables.

Unified functions for both HTML and PDF: Most functionalities in `kableExtra` can work in both HTML and PDF. In fact, as long as you specifies format in `kable` (which can be set globally through option `knitr.table.format`), functions in this package will pick the right way to manipulate the table be themselves. As a result, if users want to left align the table, `kable_styling(kable(...), position = "left")` will work in both HTML and PDF.

Note

If you found a feature on the documentation site that is not available in the version of `kableExtra` you are using, try to install the pre-release version from github. You can do so by running `devtools::install_github("hao`

Also, note that This package can load required LaTeX package automatically in vanilla rmarkdown. For customized rmarkdown templates, it is recommended to load related LaTeX packages manually.

add_footnote *Add footnote*

Description

Add footnote to your favorite kable output.

Usage

```
add_footnote(input, label = NULL, notation = "alphabet",
  threeparttable = FALSE, escape = TRUE)
```

Arguments

input	The direct output of your kable function or your last kableExtra function.
label	A vector of footnotes you want to add. You don't need to add notations in your notes.
notation	You can select the format of your footnote notation from number, alphabet, symbol and none.
threeparttable	Boolean value indicating if a threeparttable scheme should be used.
escape	Logical value controlling if the label needs to be escaped. Default is TRUE.

Examples

```
x <- knitr::kable(head(mtcars), "html")
add_footnote(x, c("footnote 1", "footnote 2"), notation = "symbol")
```

add_header_above *Add a header row on top of current header*

Description

Tables with multiple rows of header rows are extremely useful to demonstrate grouped data. This function takes the output of a kable() function and adds an header row on top of it.

Usage

```
add_header_above(kable_input, header = NULL, bold = FALSE, italic = FALSE,
  monospace = FALSE, align = "c", escape = TRUE, line = TRUE)
```

Arguments

kable_input	Output of knitr::kable() with format specified
header	A (named) character vector with colspan as values. For example, c(" " = 1, "title" = 2) can be used to create a new header row for a 3-column table with "title" spanning across column 2 and 3. For convenience, when colspan equals to 1, users can drop the = 1 part. As a result, c(" ", "title" = 2) is the same as c(" " = 1, "title" = 2).
bold	A T/F value to control whether the text should be bolded.
italic	A T/F value to control whether the text should to be emphasized.
monospace	A T/F value to control whether the text of the selected column need to be monospaced (verbatim)
align	A character string for cell alignment. For HTML, possible values could be l, c, r plus left, center, right, justify, initial and inherit while for LaTeX, you can only choose from l, c & r.
escape	A T/F value showing whether special characters should be escaped.
line	A T/F value to control whether a line will appear underneath the header

Examples

```
x <- knitr::kable(head(mtcars), "html")
# Add a row of header with 3 columns on the top of the table. The column
# span for the 2nd and 3rd one are 5 & 6.
add_header_above(x, c(" ", "Group 1" = 5, "Group 2" = 6))
```

---

add_indent *Add indentations to row headers*

Description

Add indentations to row headers

Usage

```
add_indent(kable_input, positions)
```

Arguments

kable_input	Output of knitr::kable() with format specified
positions	A vector of numeric row numbers for the rows that need to be indented.

Examples

```
x <- knitr::kable(head(mtcars), "html")
# Add indentations to the 2nd & 4th row
add_indent(x, c(2, 4))
```

auto_index *Automatically figuring out the group_row index*

Description

This helper function allows users to build the `group_row` index more quickly and use `group_rows` in a way that is similar with `collapse_rows`.

Usage

```
auto_index(x)
```

Arguments

x	The index column. A vector. For example ‘c("a", "a", "b", "b", "b")“

cell_spec *Specify Cell/Text format*

Description

Specify Cell format before it gets into kable

Usage

```
cell_spec(x, format, bold = FALSE, italic = FALSE, monospace = FALSE,
  underline = FALSE, strikeout = FALSE, color = NULL, background = NULL,
  align = NULL, font_size = NULL, angle = NULL, tooltip = NULL,
  popover = NULL, link = NULL, extra_css = NULL, escape = TRUE,
  background_as_tile = TRUE, latex_background_in_cell = TRUE)

text_spec(x, format, bold = FALSE, italic = FALSE, monospace = FALSE,
  underline = FALSE, strikeout = FALSE, color = NULL, background = NULL,
  align = NULL, font_size = NULL, angle = NULL, tooltip = NULL,
  popover = NULL, link = NULL, extra_css = NULL, escape = TRUE,
  background_as_tile = TRUE, latex_background_in_cell = FALSE)
```

Arguments

x	Things to be formated. It could be a vector of numbers or strings.
format	Either "html" or "latex". It can also be set through `option(knitr.table.format)`, same as `knitr::kable()`.
bold	T/F for font bold.
italic	T/F for font italic.

monospace	T/F for font monospaced (verbatim)
underline	A T/F value to control whether the text of the selected row need to be underlined
strikeout	A T/F value to control whether the text of the selected row need to be stricked out.
color	A character string for text color. Here please pay attention to the differences in color codes between HTML and LaTeX.
background	A character string for background color. Here please pay attention to the differences in color codes between HTML and LaTeX. Also note that in HTML, background defined in cell_spec won't cover the whole cell.
align	A character string for cell alignment. For HTML, possible values could be `l`, `c`, `r` plus `left`, `center`, `right`, `justify`, `initial` and `inherit` while for LaTeX, you can only choose from `l`, `c` & `r`.
font_size	A numeric input for font size. For HTML, you can also use options including `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, `xx-large`, `smaller`, `larger`, `initial` and `inherit`.
angle	0-360, degree that the text will rotate. Can be a vector.
tooltip	A vector of strings to be displayed as tooltip. Obviously, this feature is only available in HTML. Read the package vignette to see how to use bootstrap tooltip css to improve the loading speed and look.
popover	Similar with tooltip but can hold more contents. The best way to build a popover is through `spec_popover()`. If you only provide a text string, it will be used as content. Note that You have to enable this bootstrap module manually. Read the package vignette to see how.
link	A vector of strings for url links. Can be used together with tooltip and popover.
extra_css	Extra css text to be passed into the cell
escape	T/F value showing whether special characters should be escaped.
background_as_tile	T/F value indicating if you want to have round cornered tile as background in HTML.
latex_background_in_cell	T/F value. It only takes effect in LaTeX when `background` provided, Default value is `TRUE`. If it's `TRUE`, the background only works in a table cell. If it's `FALSE`, it works outside of a table environment.

`collapse_rows` *Collapse repeated rows to multirow cell*

Description

Collapse same values in columns into multirow cells. This feature does similar things with `group_rows`. However, unlike `group_rows`, it analyzes existing columns, finds out rows that can be grouped together, and make them multirow cells. Note that if you want to use `column_spec` to specify column styles, you should use `column_spec` before `collapse_rows`.

Usage

```
collapse_rows(kable_input, columns = NULL, valign = c("middle", "top",
  "bottom"), latex_hline = c("full", "major", "none", "custom"),
  row_group_label_position = c("identity", "stack"),
  custom_latex_hline = NULL, row_group_label_fonts = NULL,
  headers_to_remove = NULL)
```

Arguments

kable_input	Output of `knitr::kable()` with `format` specified
columns	Numeric column positions where rows need to be collapsed.
valign	Select from "top", "middle"(default), "bottom". The reason why "top" is not default is that the multirow package on CRAN win-builder is not up to date.
latex_hline	Option controlling the behavior of adding hlines to table. Choose from `full`, `major`, `none`, `custom`.
row_group_label_position	Option controlling positions of row group labels. Choose from `identity`, `stack`.
custom_latex_hline	Numeric column positions whose collapsed rows will be separated by hlines.
row_group_label_fonts	A list of arguments that can be supplied to group_rows function to format the row group label when `row_group_label_position` is `stack`
headers_to_remove	Numeric column positions where headers should be removed when they are stacked.

Examples

```
dt <- data.frame(a = c(1, 1, 2, 2), b = c("a", "a", "a", "b"))
x <- knitr::kable(dt, "html")
collapse_rows(x)
```

---

column_spec *Specify the look of the selected column*

Description

This function allows users to select a column and then specify its look.

Usage

```
column_spec(kable_input, column, width = NULL, bold = FALSE,
  italic = FALSE, monospace = FALSE, underline = FALSE,
  strikeout = FALSE, color = NULL, background = NULL,
  border_left = FALSE, border_right = FALSE, width_min = NULL,
  width_max = NULL, extra_css = NULL, include_thead = FALSE)
```

Arguments

kable_input	Output of knitr::kable() with format specified
column	A numeric value or vector indicating which column(s) to be selected.
width	A character string telling HTML & LaTeX how wide the column needs to be, e.g. "10cm", "3in" or "30em".
bold	A T/F value to control whether the text of the selected column need to be bolded.
italic	A T/F value to control whether the text of the selected column need to be emphasized.
monospace	A T/F value to control whether the text of the selected column need to be monospaced (verbatim)
underline	A T/F value to control whether the text of the selected row need to be underlined
strikeout	A T/F value to control whether the text of the selected row need to be stricked out.
color	A character string for column text color. Here please pay attention to the differences in color codes between HTML and LaTeX.
background	A character string for column background color. Here please pay attention to the differences in color codes between HTML and LaTeX.
border_left	A logical variable indicating whether there should be a border line on the left of the selected column. In HTML, you can also pass in a character string for the CSS of the border line
border_right	A logical variable indicating whether there should be a border line on the right of the selected column. In HTML, you can also pass in a character string for the CSS of the border line
width_min	Only for HTML table. Normal column width will automatically collapse when the window cannot hold enough contents. With this width_min, you can set up a column with a width that won't collapse even when the window is not wide enough.
width_max	Only for HTML table. width_max defines the maximum width of table columns.
extra_css	Extra css text to be passed into the cells of the row. Note that it's not for the whole column but to each individual cells
include_thead	T/F. A HTML only feature to contoll whether the header row will be manipulated. Default is FALSE.

Examples

```
x <- knitr::kable(head(mtcars), "html")
column_spec(x, 1:2, width = "20em", bold = TRUE, italic = TRUE)
```

`footnote` *Add footnote (new)*

Description

`footnote` provides a more flexible way to add footnote. You can add mutiple sets of footnote using differeny notation system. It is also possible to specify footnote section header one by one and print footnotes as a chunk of texts.

Usage

```
footnote(kable_input, general = NULL, number = NULL, alphabet = NULL,
  symbol = NULL, footnote_order = c("general", "number", "alphabet",
  "symbol"), footnote_as_chunk = FALSE, escape = TRUE,
  threeparttable = FALSE, general_title = "Note: ", number_title = "",
  alphabet_title = "", symbol_title = "", title_format = "italic")
```

Arguments

`kable_input`	HTML or LaTeX table generated by `knitr::kable`
`general`	Text for general footnote comments. Footnotes in this section won't be labeled with any notations
`number`	A vector of footnote texts. Footnotes here will be numbered. There is no upper cap for the number of footnotes here
`alphabet`	A vector of footnote texts, Footnotes here will be labeled with abc. The vector here should not have more than 26 elements.
`symbol`	A vector of footnote texts, Footnotes here will be labeled with special symbols. The vector here should not have more than 20 elements.
`footnote_order`	The order of how to arrange `general`, `number`, `alphabet` and `symbol`.
`footnote_as_chunk`	T/F value. Default is FALSE. It controls whether the footnotes should be printed in a chunk (without line break).
`escape`	T/F value. It controls whether the contents and titles should be escaped against HTML or LaTeX. Default is TRUE.
`threeparttable`	T/F value for whether to use LaTeX package threeparttable. Threeparttable will force the width of caption and footnotes be the width of the original table. It's useful when you have long paragraph of footnotes.
`general_title`	Section header for general footnotes. Default is "Note: ".
`number_title`	Section header for number footnotes. Default is "".
`alphabet_title`	Section header for alphabet footnotes. Default is "".
`symbol_title`	Section header for symbol footnotes. Default is "".
`title_format`	Choose from "italic"(default), "bold" and "underline". Multiple options are possible.

Examples

```
dt <- mtcars[1:5, 1:5]
footnote(knitr::kable(dt, "html"), alphabet = c("Note a", "Note b"))
```

`footnote_marker_number` *Footnote marker*

Description

Put footnote mark in superscription in table. Unless you are using it in the `caption` of `kable`, you will need to put `escape = F` in `kable` (similar with `cell_spec`). Again, similar with `cell_spec`, the `format` option here can read default value from global option `knitr.table.format`.

Usage

```
footnote_marker_number(x, format, double_escape = FALSE)

footnote_marker_alphabet(x, format, double_escape = FALSE)

footnote_marker_symbol(x, format, double_escape = FALSE)
```

Arguments

`x`	a number. For example, for `footnote_marker_alphabet(2)` will return "b" in HTML.
`format`	Either `html` or `latex`. All functions here can read default value from global option `knitr.table.format`.
`double_escape`	T/F if output is in LaTeX, whether it should be double escaped. If you are using footnote_marker in `group_rows`` labeling `row` oradd_header_above, you need to set this to b

Examples

```
dt <- mtcars[1:5, 1:5]
colnames(dt)[1] <- paste0("mpg", footnote_marker_alphabet(2, "html"))
rownames(dt)[2] <- paste0(rownames(dt)[2], footnote_marker_alphabet(1, "html"))
footnote(knitr::kable(dt, "html"), alphabet = c("Note a", "Note b"))
```

group_rows *Put a few rows of a table into one category*

Description

Group a few rows in a table together under a label.

Usage

```
group_rows(kable_input, group_label = NULL, start_row = NULL,
  end_row = NULL, index = NULL,
  label_row_css = "border-bottom: 1px solid;", latex_gap_space = "0.3em",
  escape = TRUE, latex_align = "l", colnum = NULL, bold = T,
  italic = F, hline_before = F, hline_after = F,
  extra_latex_after = NULL)
```

Arguments

kable_input	Output of knitr::kable() with format specified
group_label	A character string for the name of the group
start_row	A numeric value that tells the function in which row the group starts. Note that the counting excludes header rows and other group labeling rows
end_row	A numeric value that tells the function in which row the group ends.
index	A named vector providing the index for robust row-grouping tasks. Basically, you can use it in the same way as add_header_above().
label_row_css	A character string for any customized css used for the labeling row. By default, the labeling row will have a solid black line underneath. Only useful for HTML documents.
latex_gap_space	A character value telling LaTeX how large the gap between the previous row and the group labeling row. Only useful for LaTeX documents.
escape	A T/F value showing whether special characters should be escaped.
latex_align	Adjust justification of group_label in latex only. Value should be "c" for centered on row, "r" for right justification, or "l" for left justification. Default Value is "l" If using html, the alignment can be set by using the label_row_css parameter.
colnum	A numeric that determines how many columns the text should span. The default setting will have the text span the entire length.
bold	A T/F value to control whether the text should be bolded.
italic	A T/F value to control whether the text should to be emphasized.
hline_before	A T/F value that addes a horizontal line before the group_row label. Default value is False.
hline_after	A replicate of hline.after in xtable. It addes a hline after the row
extra_latex_after	Extra LaTeX text to be added after the row.

Examples

```
x <- knitr::kable(head(mtcars), "html")
# Put Row 2 to Row 5 into a Group and label it as "Group A"
group_rows(x, "Group A", 2, 5)
```

`html_dependency_bsTable` *HTML dependency for Twitter bootstrap (table only)*

Description

HTML dependency for Twitter bootstrap (table only)

Usage

```
html_dependency_bsTable()
```

`html_dependency_kePrint` *HTML dependency for js script to enable bootstrap tooltip and popup message*

Description

HTML dependency for js script to enable bootstrap tooltip and popup message

Usage

```
html_dependency_kePrint()
```

`kableExtra_latex_packages` *LaTeX Packages*

Description

This function shows all LaTeX packages that is supposed to be loaded for this package in a rmarkdown yaml format.

Usage

```
kableExtra_latex_packages()
```

kable_as_image *Convert a LaTeX table to an image and place it in a rmarkdown document*

Description

This is a LaTeX-only function. This function will render the raw LaTeX code (could be codes generated by other table packages like `xtable`) to generate a table, convert it to an image and put it back to a rmarkdown environment. It is a "better than nothing" solution to print high quality tables in rmarkdown Word document. By using this, you need to take the responsibility of explaining to your collaborators why they can't make edits to the tables in Word.

Also, if a filename is provided, user has the option to "save" the table to an image file like `ggplot2::ggsave()`.

Note that, if you are using this function on a Windows computer, you need to install Ghostscript before you can use this feature. It is essential for magick to read PDFs on Windows. Website for Ghostscript: https://ghostscript.com/

The idea of this function was coming from this StackOverflow question. The approach was learned and adopted from the texpreview package, which allows you to preview the results of TeX code in the Viewer panel.

Usage

```
kable_as_image(kable_input, filename = NULL, file_format = "png",
  latex_header_includes = NULL, keep_pdf = FALSE, density = 300,
  keep_tex = FALSE)
```

Arguments

`kable_input`	Raw LaTeX code to generate a table. It doesn't have to came from `kable` or `kableExtra`.
`filename`	Character String. If specified, the image will be saved under the specified (path &) name. You don't need to put file format like ".png" here.
`file_format`	Character String to specify image format, such as `png`, `jpeg`, `gif`, `tiff`, etc. Default is `png`.
`latex_header_includes`	A character vector of extra LaTeX header stuff. Each element is a row. You can have things like `c("\\usepackage{threeparttable}", "\\usepackage{icons}")`
`keep_pdf`	A T/F option to control if the mid-way standalone pdf should be kept. Default is `FALSE`.
`density`	Resolution to read the PDF file. Default value is 300, which should be sufficient in most cases.
`keep_tex`	A T/F option to control if the latex file that is initially created should be kept. Default is `FALSE`.

kable_styling *HTML table attributes*

Description

This function provides a cleaner approach to modify the style of HTML tables other than using the `table.attr` option in `knitr::kable()`. Note that those bootstrap options requires Twitter bootstrap theme, which is not avaiable in some customized template being loaded.

Usage

```
kable_styling(kable_input, bootstrap_options = "basic",
  latex_options = "basic", full_width = NULL, position = "center",
  font_size = NULL, row_label_position = "l", ...)
```

Arguments

`kable_input`	Output of `knitr::kable()` with `format` specified
`bootstrap_options`	A character vector for bootstrap table options. Please see package vignette or visit the w3schools' Bootstrap Page for more information. Possible options include `basic`, `striped`, `bordered`, `hover`, `condensed` and `responsive`.
`latex_options`	A character vector for LaTeX table options. Please see package vignette for more information. Possible options include `basic`, `striped`, `hold_position`, `HOLD_position`, `scale_down` & `repeat_header`. `striped` will add alternative row colors to the table. It will imports `LaTeX` package `xcolor` if enabled. `hold_position` will "hold" the floating table to the exact position. It is useful when the `LaTeX` table is contained in a `table` environment after you specified captions in `kable()`. It will force the table to stay in the position where it was created in the document. A stronger version: `HOLD_position` requires the `float` package and specifies `[H]`. `scale_down` is useful for super wide table. It will automatically adjust the table to page width. `repeat_header` in only meaningful in a longtable environment. It will let the header row repeat on every page in that long table.
`full_width`	A `TRUE` or `FALSE` variable controlling whether the HTML table should have 100% width. Since HTML and pdf have different flavors on the preferable format for `full_width`. If not specified, a HTML table will have full width by default but this option will be set to `FALSE` for a LaTeX table
`position`	A character string determining how to position the table on a page. Possible values include `left`, `center`, `right`, `float_left` and `float_right`. Please see the package doc site for demonstrations. For a `LaTeX` table, if `float_*` is selected, `LaTeX` package `wrapfig` will be imported.
`font_size`	A numeric input for table font size

row_label_position
: A character string determining the justification of the row labels in a table. Possible values inclued `l` for left, `c` for center, and `r` for right. The default value is `l` for left justifcation.

`...`
: extra options for HTML or LaTeX. See `details`.

Details

For LaTeX, extra options includes:

- `repeat_header_method` can either be `append`(default) or `replace`
- `repeat_header_text` is just a text string you want to append on or replace the caption.
- `stripe_color` allows users to pick a different color for their strip lines.
- `latex_table_env` character string to define customized table environment such as tabu or tabularx.You shouldn't expect all features could be supported in self-defined environments.

Examples

```
x_html <- knitr::kable(head(mtcars), "html")
kable_styling(x_html, "striped", position = "left", font_size = 7)

x_latex <- knitr::kable(head(mtcars), "latex")
kable_styling(x_latex, latex_options = "striped", position = "float_left")
```

---

landscape *Print the table on an isolated landscape page in PDF*

Description

This function will put the table on an single landscape page. It's useful for wide tables that cann't be printed on a portrait page.

Usage

```
landscape(kable_input, margin = NULL)
```

Arguments

`kable_input`
: Output of `knitr::kable()` with `format` specified

`margin`
: Customizable page margin for special needs. Values can be "1cm", "1in" or similar.

Examples

```
landscape(knitr::kable(head(mtcars), "latex"))
```

linebreak *Make linebreak in LaTeX Table cells*

Description

This function generate LaTeX code of `makecell` so that users can have linebreaks in their table

Usage

```
linebreak(x, align = c("l", "c", "r"), double_escape = F)
```

Arguments

`x`	A character vector
`align`	Choose from "l", "c" or "r"
`double_escape`	Whether special character should be double escaped. Default is FALSE.

magic_mirror *Magic mirror that returns kable's attributes*

Description

Mirror mirror tell me, how does this kable look like?

Usage

```
magic_mirror(kable_input)
```

Arguments

`kable_input`	The output of kable

Examples

```
magic_mirror(knitr::kable(head(mtcars), "html"))
```

rmd_format *Rmarkdown Format*

Description

If the export format of the Rmarkdown document exist,

Usage

```
rmd_format()
```

row_spec *Specify the look of the selected row*

Description

This function allows users to select a row and then specify its look. It can also specify the format of the header row when row = 0.

Usage

```
row_spec(kable_input, row, bold = FALSE, italic = FALSE,
  monospace = FALSE, underline = FALSE, strikeout = FALSE, color = NULL,
  background = NULL, align = NULL, font_size = NULL, angle = NULL,
  extra_css = NULL, hline_after = FALSE, extra_latex_after = NULL)
```

Arguments

kable_input	Output of knitr::kable() with format specified
row	A numeric value or vector indicating which row(s) to be selected. You don't need to count in header rows or group labeling rows.
bold	A T/F value to control whether the text of the selected row need to be bolded.
italic	A T/F value to control whether the text of the selected row need to be emphasized.
monospace	A T/F value to control whether the text of the selected row need to be monospaced (verbatim)
underline	A T/F value to control whether the text of the selected row need to be underlined
strikeout	A T/F value to control whether the text of the selected row need to be stricked out.
color	A character string for row text color. For example, "red" or "#BBBBBB".
background	A character string for row background color. Here please pay attention to the differences in color codes between HTML and LaTeX.
align	A character string for cell alignment. For HTML, possible values could be l, c, r plus left, center, right, justify, initial and inherit while for LaTeX, you can only choose from l, c & r.
font_size	A numeric input for font size. For HTML, you can also use options including xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, initial and inherit.
angle	0-360, degree that the text will rotate.
extra_css	Extra css text to be passed into the cells of the row. Note that it's not for the whole row.
hline_after	T/F. A replicate of hline.after in xtable. It addes a hline after ther row
extra_latex_after	Extra LaTeX text to be added after the row. Similar with add.to.row in xtable

Examples

```
x <- knitr::kable(head(mtcars), "html")
row_spec(x, 1:2, bold = TRUE, italic = TRUE)
```

---

save_kable *Save kable to files*

Description

Save kable to files

Usage

```
save_kable(x, file, bs_theme = "simplex", self_contained = TRUE)
```

Arguments

x	A piece of HTML code for tables, usually generated by kable and kableExtra
file	save to files
bs_theme	Which Bootstrap theme to use
self_contained	Will the files be self-contained?

---

scroll_box *Put a HTML table into a scrollable box*

Description

This function will put a HTML kable object in a fixed-height, fixed-width or both box and make it scrollable.

Usage

```
scroll_box(kable_input, height = NULL, width = NULL,
  box_css = "border: 1px solid #ddd; padding: 5px; ", extra_css = NULL)
```

Arguments

kable_input	A HTML kable object
height	A character string indicating the height of the box, e.g. "50px"
width	A character string indicating the width of the box, e.g. "100px"
box_css	CSS text for the box
extra_css	Extra CSS styles

Examples

```
## Not run:
# Specify table size by pixels
kable(cbind(mtcars, mtcars), "html") %>%
    kable_styling() %>%
    scroll_box(width = "500px", height = "200px")

# Specify by percent
kable(cbind(mtcars, mtcars), "html") %>%
    kable_styling() %>%
    scroll_box(width = "100%", height = "200px")

## End(Not run)
```

---

spec_angle *Generate rotation angle for continuous values*

Description

Generate rotation angle for continuous values

Usage

```
spec_angle(x, begin, end, scale_from = NULL)
```

Arguments

x	continuous vectors of values
begin	Smallest degree to rotate. Default is 0
end	Largest degree to rotate. Default is 359.
scale_from	input range (vector of length two). If not given, is calculated from the range of x

---

spec_color *Generate viridis Color code for continuous values*

Description

Generate viridis Color code for continuous values

Usage

```
spec_color(x, alpha = 1, begin = 0, end = 1, direction = 1,
  option = "D", na_color = "#BBBBBB", scale_from = NULL)
```

Arguments

x	continuous vectors of values
alpha	The alpha transparency, a number in [0,1], see argument alpha in hsv.
begin	The (corrected) hue in [0,1] at which the viridis colormap begins.
end	The (corrected) hue in [0,1] at which the viridis colormap ends.
direction	Sets the order of colors in the scale. If 1, the default, colors are ordered from darkest to lightest. If -1, the order of colors is reversed.
option	A character string indicating the colormap option to use. Four options are available: "magma" (or "A"), "inferno" (or "B"), "plasma" (or "C"), "viridis" (or "D", the default option) and "cividis" (or "E").
na_color	color code for NA values
scale_from	input range (vector of length two). If not given, is calculated from the range of x

---

spec_font_size *Generate common font size for continuous values*

Description

Generate common font size for continuous values

Usage

```
spec_font_size(x, begin = 8, end = 16, na_font_size = 12,
  scale_from = NULL)
```

Arguments

x	continuous vectors of values
begin	Smalles font size to be used. Default is 10.
end	Largest font size. Default is 20.
na_font_size	font size for NA values
scale_from	input range (vector of length two). If not given, is calculated from the range of x

spec_popover *Setup bootstrap popover*

Description

Setup bootstrap popover

Usage

```
spec_popover(content = NULL, title = NULL, trigger = "hover",
  position = "right")
```

Arguments

content	content for pop-over message
title	title for pop-over message.
trigger	Controls how the pop-over message should be triggered. Possible values include hover (default), click, focus and manual.
position	How the tooltip should be positioned. Possible values are right(default), top, bottom, left & auto.

spec_tooltip *Setup bootstrap tooltip*

Description

Setup bootstrap tooltip

Usage

```
spec_tooltip(title, position = "right")
```

Arguments

title	text for hovering message
position	How the tooltip should be positioned. Possible values are right(default), top, bottom, left & auto.

`usepackage_latex` *Load a LaTeX package*

Description

Load a LaTeX package using R code. Just like `\usepackage{}` in LaTeX

Usage

```
usepackage_latex(name, options = NULL)
```

Arguments

`name`	The LaTeX package name
`options`	The LaTeX options for the package

Examples

```
usepackage_latex("xcolor")
```

Index