Package 'ftExtra'

August 30, 2020

Title Extensions for 'Flextable'

Version 0.0.3

Maintainer Atsushi Yasumoto <atusy.rpkg@gmail.com>

Description Build display tables easily by extending the functionality of the 'flextable' package. Features include spanning header, grouping rows, parsing markdown and so on.

License MIT + file LICENSE

URL https://ftextra.atusy.net, https://github.com/atusy/ftExtra

BugReports https://github.com/atusy/ftExtra/issues

Imports dplyr, jsonlite, flextable, tidyr, purrr, magrittr, rmarkdown, rlang, stringr, tibble, tidyselect (>= 1.1.0)

Suggests testthat (>= 2.1.0), knitr

VignetteBuilder knitr

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.1.1

SystemRequirements pandoc (>= 1.12.3) - http://pandoc.org

NeedsCompilation no

Author Atsushi Yasumoto [aut, cph, cre] (<https://orcid.org/0000-0002-8335-495X>), Romain François [ctb] (<https://orcid.org/0000-0002-2444-4226>)

Repository CRAN

Date/Publication 2020-08-30 01:00:22 UTC

R topics documented:

`as_flextable_methods` *method to convert object to flextable*

Description

This is a convenient function to let users create flextable bindings from any objects.

Usage

```
## S3 method for class 'grouped_df'
as_flextable(
  x,
  groups_to = c("titles", "merged", "asis"),
  groups_pos = c("left", "asis"),
  ...
)

## S3 method for class 'data.frame'
as_flextable(x, col_keys = names(x), ...)
```

Arguments

`x`	object to be transformed as flextable
`groups_to`	One of `titles`, `merged`, or `asis`. See examples for the result.
`groups_pos`	When `groups_to = "merged"`, grouping columns are reordered according to `group_pos`. Choices are `left` (default) or `asis`.
`...`	arguments for custom methods
`col_keys`	columns names/keys to display. If some column names are not in the dataset, they will be added as blank columns by default.

See Also

Other as_flextable methods: `as_flextable.glm()`, `as_flextable.grouped_data()`, `as_flextable.htest()`, `as_flextable.lm()`, `as_flextable.xtable()`

Examples

```
# For grouped_df
grouped_df <- iris %>%
  dplyr::group_by(Species) %>%
  dplyr::slice(1, 2)

as_flextable(grouped_df, groups_to = "titles")
as_flextable(grouped_df, groups_to = "titles", hide_grouplabel = TRUE)
as_flextable(grouped_df, groups_to = "merged")
as_flextable(grouped_df, groups_to = "asis")
# For data.frame
iris %>%
  head() %>%
  as_flextable()
```

---

as_paragraph_md *Convert a character vector into markdown paragraph(s)*

Description

Parse markdown cells and returns the "paragraph" object.

Usage

```
as_paragraph_md(
  x,
  auto_color_link = "blue",
  .from = "markdown+autolink_bare_uris"
)
```

Arguments

x	A character vector.
auto_color_link	A color of the link texts.
.from	Pandoc's `--from` argument (default: `'markdown+autolink_bare_uris'`).

Examples

```
if (rmarkdown::pandoc_available()) {
  library(flextable)
  ft <- flextable(
    data.frame(
      x = c("**foo** bar", "***baz***", "*qux*"),
      stringsAsFactors = FALSE
    )
  )
```

```
  ft <- compose(ft, j = "x", i = 1:2, value = as_paragraph_md(x))
  autofit(ft)
}
```

`colformat_md` *Format character columns as markdown text*

Description

Format character columns as markdown text

Usage

```
colformat_md(
  x,
  j = where(is.character),
  auto_color_link = "blue",
  .from = "markdown+autolink_bare_uris"
)
```

Arguments

`x`	A `flextable` object
`j`	Columns to be treated as markdown texts. Selection can be done by the semantics of `dplyr::select()`.
`auto_color_link`	A color of the link texts.
`.from`	Pandoc's `--from` argument (default: `'markdown+autolink_bare_uris'`).

Examples

```
if (rmarkdown::pandoc_available()) {
  d <- data.frame(
    x = c("**bold**", "*italic*"),
    y = c("^superscript^", "~subscript~"),
    z = c("***^ft^~Extra~** is*", "*Cool*")
  )
  colformat_md(flextable::flextable(d))
}
```

separate_header *Separate the header based on delimiters*

Description

Separate the header based on delimiters

Usage

```
separate_header(
  x,
  sep = "[_\\.]",
  theme_fun = flextable::theme_booktabs,
  ...
)
```

Arguments

x	A flextable object‘
sep	Separator between columns. If character, sep is interpreted as a regular expression. The default value is a regular expression that matches any sequence of non-alphanumeric values. If numeric, sep is interpreted as character positions to split at. Positive values start at 1 at the far-left of the string; negative value start at -1 at the far-right of the string. The length of sep should be one less than into.
theme_fun	a function theme to apply before returning the flextable. set to NULL for none.
...	Passed to theme_fun

Examples

```
iris %>%
  as_flextable() %>%
  separate_header()
```

span_header *Span the header based on delimiters*

Description

Span the header based on delimiters

Usage

```
span_header(x, sep = "[_\\.]", theme_fun = flextable::theme_booktabs, ...)
```

Arguments

x	A flextable object‘
sep	Separator between columns. If character, sep is interpreted as a regular expression. The default value is a regular expression that matches any sequence of non-alphanumeric values. If numeric, sep is interpreted as character positions to split at. Positive values start at 1 at the far-left of the string; negative value start at -1 at the far-right of the string. The length of sep should be one less than into.
theme_fun	a function theme to apply before returning the flextable. set to NULL for none.
...	Passed to theme_fun

Examples

```
iris %>%
  as_flextable() %>%
  span_header()
```

---

with_blanks *Specify blank columns easily via* col_keys

Description

Specify blank columns easily via col_keys

Usage

```
with_blanks(after = NULL, before = NULL)
```

Arguments

after, before	Blank columns are added after/before the selected columns. Selections can be done by the semantics of dplyr::select.

Examples

```
iris %>%
  as_flextable(col_keys = with_blanks(dplyr::ends_with("Width")))
```

Index