

Package ‘clinify’

August 1, 2026

Type Package

Title Clinical Table Styling Tools and Utilities

Version 0.4.0

Depends R (>= 4.2)

Imports flextable, officer (>= 0.7.2), magrittr, dplyr, knitr,
htmltools, tidyselect, zoo

Suggests rmarkdown, testthat (>= 3.0.0), Tplyr, rvest, withr, xml2

Description

The primary motivation of this package is to take the things that are great about the R packages 'flextable' <<https://davidgohel.github.io/flextable/>> and 'officer' <<https://davidgohel.github.io/officer/>>, take the standard and complex pieces of formatting clinical tables for regulatory use, and simplify the tedious pieces.

License Apache License (>= 2)

Encoding UTF-8

RoxygenNote 7.3.2

Config/testthat/edition 3

VignetteBuilder knitr

URL <https://atorus-research.github.io/clinify/>

NeedsCompilation no

Author Mike Stackhouse [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-6030-723X>>),
Ross Didenko [aut],
Yevhenii Boiko [aut],
Marat Zakirov [ctb],
Roman Rogoza [ctb],
Atorus Research, Inc. [cph],
Incyte Corporation [cph]

Maintainer Mike Stackhouse <mike.stackhouse@atorusresearch.com>

Repository CRAN

Date/Publication 2026-08-01 20:20:02 UTC

Contents

as_clintable	2
clindoc	3
clinify_titles_default	4
clintable	5
clin_add_titles	7
clin_alt_pages	9
clin_auto_page	10
clin_column_headers	11
clin_col_widths	12
clin_default_table_width	13
clin_group_by	14
clin_group_pad	14
clin_header_pad	15
clin_page_by	17
clin_replace_pagenums	18
clin_row_height	19
clin_spanner_rule	20
clin_table_align	22
make_grouped_pagenums	23
new_title_footnote	23
print.clintable	24
write_clindoc	25

Index	27
-------	----

as_clintable	<i>Convert a flextable into a clintable object</i>
--------------	--

Description

Convert a flextable into a clintable object

Usage

as_clintable(x, page_by = NULL, group_by = NULL)

Arguments

x	A flextable object
page_by	A variable in the input dataframe to use for pagination
group_by	A variable which will be used for grouping and attached as a label above the table headers

Details

There is no `coerce_character` argument here, unlike `clintable()`. A flextable arrives with its cell text already rendered, so the numeric formatting this argument exists to avoid has already happened and coercing the source data is no longer an option. The nearest equivalent is `flextable::set_formatter(x, values = as.character)` before calling `as_clintable()`, which rewrites every body cell from the stored data. That is not the same operation: it replaces cell *content*, so any chunk level work already done on the body - `flextable::compose()`, `flextable::colformat_*()`, images, hyperlinks, equations - is discarded, and columns keep the right alignment flextable gave them for being numeric. Because that trade cannot be made safely on the user's behalf, it is left to the caller. To get the coercion without the trade, build with `clintable(x, coerce_character = TRUE)` instead.

Value

A clintable object

Examples

```
ft <- flextable::flextable(mtcars)
as_clintable(ft)
```

clindoc

Create a clindoc object

Description

These functions handle the conversion of a clintable object into a clindoc object.

Usage

```
clindoc(...)
```

```
as_clindoc(x)
```

Arguments

<code>...</code>	clintable objects to be converted. Or separately, a list of clintable objects
<code>x</code>	A clintable object to be converted.

Details

- `as_clindoc()` is intended for a single clintable object and extracts all necessary title and footnote information to apply to the document

- `clindoc()` can accept `clintable` objects as separate parameters or as a list of `clintable` objects. If a single `clintable` is passed, this function simply calls `as_clindoc()` internally.

When a multiple `clintable` objects are passed to `clindoc()`, titles and footnotes should be applied directly to the `clindoc` object using `clin_add_title()`, `clin_add_footnote()`, or `clin_add_footnote_page()`. Title and footnote information on the individual `clintable` objects will be ignored.

Value

a `clindoc` object, inherited from an `officer::rdocx` object

Examples

```
ct <- clintable(mtcars)

clindoc(ct)
```

clinify_titles_default

Apply Default Clinical Styling to Clintables

Description

These functions apply default styling to `clintable` objects used for clinical tables, including titles and footnotes. The styling includes removing borders, setting font properties, and adjusting table width, line spacing, and padding.

Usage

```
clinify_titles_default(x, ...)

clinify_footnotes_default(x, ...)

clinify_table_default(x, ...)

clinify_caption_default(x, ...)

clinify_grouplabel_default(x, ...)

clinify_docx_default()
```

Arguments

<code>x</code>	A <code>clintable</code> object representing the table (title or footnote).
<code>...</code>	Additional arguments (currently unused).

Value

A clintable object with the applied styling.

Examples

```
op <- options()

sect <- clinify_docx_default()

# Save out options to grab defaults
options(
  clinify_docx_default = sect,
  clinify_titles_default = clinify_titles_default,
  clinify_footnotes_default = clinify_footnotes_default,
  clinify_table_default = clinify_table_default,
  clinify_caption_default = clinify_caption_default,
  clinify_grouplabel_default = clinify_grouplabel_default
)

options(op)
```

clintable

Create a new clintable object

Description

A clintable object directly inherits from a flextable object. This function will pass all necessary parameters `flextable::flextable()` and convert the object to a clintable

Usage

```
clintable(
  x,
  page_by = NULL,
  group_by = NULL,
  use_labels = TRUE,
  coerce_character = FALSE,
  ...
)
```

Arguments

x	A data frame
page_by	A variable in the input dataframe to use for pagination
group_by	A character vector of variable names which will be used for grouping and attached as a label above the table headers

`use_labels` Use variable labels as column headers. Nested levels can be achieved using the string "||" as a delimiter. Horizontally adjacent cells using identical words will be merged, which can be adjusted afterwards using the `merge` argument of `clin_column_headers()`.

`coerce_character` Coerce every column of `x` to character before the flextable is built, so pre-formatted values render exactly as supplied. Defaults to `FALSE`, which leaves flextable's numeric formatting in place.

... Parameters to pass to `flextable::flextable()`

Value

A clintable object

Rendering values verbatim

flextable bakes cell text in when the table is built, and it formats a double column as a whole with `format(x, trim = TRUE, scientific = FALSE, big.mark = ",")`. Because that decision is column wide, a clinical summary column holding a count in one row and a statistic in another - necessarily a double - is reformatted against its neighbours: `c(86, 75.2)` renders the count as "86.0", `c(1234, 12.5)` renders it as "1,234.0", and `c(1234567.891, 2)` is rounded to seven significant digits as "1,234,568". Values that were already formatted upstream are therefore silently changed, and nothing errors to say so.

`coerce_character = TRUE` runs `as.character()` over every column first, so each value carries into the table as its own string and no column wide decision is made. It replaces the `lapply(x, as.character)` line that otherwise has to be written ahead of every table. Column label attributes survive the coercion, so `use_labels` still finds them. Factors coerce to their levels rather than their integer codes.

Two side effects are worth knowing about. Numeric columns lose the right alignment flextable's default theme gives them, since alignment follows column type; use `clin_table_align()` or `flextable::align()` to put it back. And flextable's formula selectors compare against the coerced values, so `bold(i = ~ n > 5)` becomes a string comparison and quietly selects different rows.

NA is left as NA

`as.character(NA)` is `NA_character_`, and flextable's default `na_str` is `""`, so an NA still renders as a blank cell. NA is deliberately not replaced with `""`, which is safe in a body column but changes the meaning of a pagination variable.

`clin_page_by()` splits where the page variable changes, as does `clin_group_by()` by default, and that comparison is `x != lag(x)`. It is NA wherever either side is NA, and those rows are dropped rather than treated as splits. So a `page_by`, `group_by`, or `caption_by` column that is padded - carrying its value only on the first row of each block, NA below - collapses to a single page with no group label. A variable used that way needs `clin_group_by(when = "notempty")`, which tests against `""` and handles NA just as well, and `clin_page_by()` offers no such option so its page variable has to carry a value on every row.

Padding and a change comparison do not go together whichever the pad is, but they fail differently, and the NA failure is the quieter one: `""` padding makes each padded row look like a change and

splits on every one of them, which is hard to miss, where NA padding drops the splits and leaves a plausible looking single page.

Examples

```
clintable(mtcars)

# A summary column holding a count and a mean is a double, so flextable
# would render the count 86 as "86.0". Coercion keeps it as written.
summary_dat <- data.frame(
  row_label = c("n", "Mean"),
  trt_a = c(86, 75.2)
)
clintable(summary_dat, coerce_character = TRUE)
```

clin_add_titles	<i>Add titles, footnotes, or a footnote page to a clintable or clindoc</i>
-----------------	--

Description

This function allows you to attach specified titles, footnotes, or a footnote page into clintable or clindoc object. The input can be provided either as a list of character vectors, or pre-built flextable.

Usage

```
clin_add_titles(x, ls = NULL, ft = NULL, align = NULL, tokens = NULL)

clin_add_footnotes(x, ls = NULL, ft = NULL, align = NULL, tokens = NULL)

clin_add_footnote_page(x, ls = NULL, ft = NULL, align = NULL, tokens = NULL)
```

Arguments

x	a clintable object
ls	a list of character vectors, no more than 2 elements to a vector, or a data frame spec as described above
ft	A flextable object to use as the header
align	Where to place each line, as a character vector holding one value per element of ls (or a single value for all of them). Values are "left", "center", "right", "split", or NA to keep the default for that line. Cannot be used together with ft, or with a spec that already has an align column.
tokens	Replacements for {NAME} placeholders in the text, as a named list or character vector - tokens = list(FILE = "programs/t14-1-01.R") turns {FILE} into that path. Cannot be used together with ft.

Details

When using the `ls` parameter, each element of the list can contain no more than two elements within each character vector. In a title, a single element will align center. In a footnote, a single element will align to the left. For both titles and footnotes, two elements will align split down the middle, with the left side element aligning left and the right side element aligning right.

Use `align` to place a line somewhere other than its default. A line holding a single element can go "left", "center", or "right"; a line holding two elements is split down the middle by construction, which `align` spells "split". NA leaves a line where it would have landed anyway.

Instead of a list, `ls` can be a data frame holding every line for a table at once, so one object feeds the titles, the footnotes and a footnote page together. Each of the three functions takes the rows that belong to it and ignores the rest, and a surface with no rows is left alone. Rows are used in the order they are given.

<code>column</code>	holds
<code>type</code>	"title", "footnote", or "footnote_page" (plurals accepted)
<code>text1</code>	the line, or its left hand side
<code>text2</code>	the right hand side of a split line, blank or NA if there is none
<code>align</code>	as the <code>align</code> argument below, blank or NA for the default

Only `type` and `text1` are required. Reading the spec in is left to you - it is an ordinary data frame, so it can come from a spreadsheet, a CSV, a database, or be written out by hand.

`tokens` fills in {NAME} placeholders, which is how a program path or a run date gets into text that was written somewhere else. {PAGE} and {NUMPAGES} are left alone - those become real Word page number fields when the table renders, so do not pass them as tokens.

Value

A clintable object

Examples

```
clintable(mtcars) |>
  clin_add_titles(
    list(
      c("Left", "Right"),
      c("Just the middle")
    )
  ) |>
  clin_add_titles(
    list(
      c("Protocol: ABC", "Page {PAGE} of {NUMPAGES}"),
      "Table 14-2.01",
      "Summary of Demographic and Baseline Characteristics"
    ),
    # the title line stays centered, the one below it goes left
    align = c(NA, NA, "left")
  ) |>
  clin_add_footnotes(
```

```

      list(
        c(
          "Here's a footnote.",
          format(Sys.time(), "%H:%M %A, %B %d, %Y")
        )
      )
    ) |>
  clin_add_footnote_page(
    list(
      c(
        "Use when you have a lot of footnotes",
        "And you don't want to put them on every page"
      )
    )
  )

# Or keep every line for the table in one place and let each function take
# the rows that belong to it
spec <- data.frame(
  type = c("title", "title", "footnote"),
  text1 = c("Protocol: ABC", "Table 14-2.01", "Source: {FILE}"),
  text2 = c("Page {PAGE} of {NUMPAGES}", NA, NA),
  align = c("split", "center", "left")
)

clintable(mtcars) |>
  clin_add_titles(spec, tokens = list(FILE = "programs/t14-2-01.R")) |>
  clin_add_footnotes(spec, tokens = list(FILE = "programs/t14-2-01.R"))

```

clin_alt_pages

*Configure alternating pages during pagination of a clintable***Description**

This function configures alternating pages on a clintable object.

Usage

```
clin_alt_pages(x, key_cols, col_groups)
```

Arguments

x	A clintable object
key_cols	A character vector of variable names
col_groups	A list of character vectors of variable names

Value

A clintable object

Examples

```
ct <- clintable(mtcars)

clin_alt_pages(
  ct,
  key_cols = c("mpg", "cyl", "hp"),
  col_groups = list(
    c("disp", "drat", "wt"),
    c("qsec", "vs", "am"),
    c("gear", "carb")
  )
)
```

clin_auto_page	Enable Word Auto-Pagination Using Group Variable
----------------	--

Description

This function uses the applies the functionality `flectable::keep_with_next()` by automatically building the row indices using some grouping variable. Each group identified by the variable (i.e. when the value of the variable changes) will be set as a "keep_with_next" group in Word. Using this functionality, Word will attempt not to break that group across pages, enabling smoother pagination without having to do specific calculations of page breaks.

Usage

```
clin_auto_page(x, group_var, when = c("change", "notempty"), drop = FALSE)
```

Arguments

x	A clintable object
group_var	A string containing a variable name of the input dataset used to calculate groups
when	Character string indicating when to apply padding: "notempty": Find allowable break points when the value in group_var is not empty. "change": Find allowable break points when the value in group_var changes from the previous row.
drop	Keep or drop the 'group_var' variable

Value

A clintable object

Examples

```
clintable(mtcars) |>
  clin_auto_page("gear")
```

clin_column_headers	<i>Set the column headers of the output clintable</i>
---------------------	---

Description

This function allows you to apply column headers named arguments and character vectors. Separate elements of the character vector are converted to separate levels of the output table header. The in which the headers are applied goes from top to bottom, so if you provide 3 elements for a column header, the first element is applied to the top and the second to the bottom. If one variable has three levels and other variable only have one or two, the columns with less levels to the header will bind to the bottom. So a column with two levels will apply to the second and third row, and a column with one level with apply the bottom row. Spanners are determined using cells of the same text value, where horizontally adjacent cells holding the same text are merged. Use the merge argument when a header row legitimately repeats a label across adjacent columns and those cells should be left alone - merged, they render as one label centred over the whole run, so the repeats are not there to read any more. That is most often wanted for the bottom row, which holds each column's own label: six columns each labelled "Baseline" come out as a single Baseline spanning all six unless merge = "spanners" keeps that row out of it. merge works a row at a time, so if a single row needs some of its repeated cells merged but not others, leave that row out of merge and span the intended cells with `flextable::merge_at()`.

Usage

```
clin_column_headers(x, ..., merge = TRUE)
```

Arguments

x	A clintable object
...	Named arguments providing the column header text. Separate levels of the header are determined using separate elements of a character vector.
merge	Controls the automatic merging of identical, adjacent header cells, which is what forms spanners. TRUE (the default) or "all" merges every header row, FALSE or "none" merges none of them, and "spanners" merges every row except the bottom one - the row holding the individual column labels. Merging can also be limited to specific header rows, numbered from the top down, using ordinary R subscripts: merge = 1:2 merges the top two rows only, merge = -3 merges every row except the third, and a logical vector as long as the header is deep toggles each row individually. Only the header is ever merged - the table body is left alone. One thing to know: a custom <code>clinify_table_default()</code> that calls <code>flextable::merge_h()</code> on the header will merge it again when the table renders, overriding whatever is set here.

Details

The same result can be achieved using column labels on the input dataframe to the clintable. If labels are present, header levels will be separated using the delimiter "||" within the label string.

Headers built that way can have their merging adjusted by calling `clin_column_headers()` with no header text and only the merge argument, which leaves the header text as it is. Called that way, any merging already on the header is cleared first - including merges applied by hand with `flextable::merge_at()` or `flextable::merge_v()` - so the rows named in merge end up being the only merged rows.

Value

A clintable object

Examples

```
clintable(iris) |>
  clin_column_headers(
    Sepal.Length = c("Flowers", "Sepal", "Length"),
    Sepal.Width = c("Flowers", "Sepal", "Width"),
    Petal.Length = c("Petal", "Length"),
    Petal.Width = c("Petal", "Width")
  )

# Keep the repeated bottom row cells separate, but still span
# "Flowers" and "Petal" across the columns above them
clintable(iris) |>
  clin_column_headers(
    Sepal.Length = c("Flowers", "Sepal", "Value"),
    Sepal.Width = c("Flowers", "Sepal", "Value"),
    Petal.Length = c("Petal", "Value"),
    Petal.Width = c("Petal", "Value"),
    merge = "spanners"
  )

# Headers coming from column labels can have their merging adjusted
# without restating the header text
iris2 <- iris
attr(iris2$Sepal.Length, "label") <- "Flowers||Value"
attr(iris2$Sepal.Width, "label") <- "Flowers||Value"

clintable(iris2) |>
  clin_column_headers(merge = 1)
```

clin_col_widths

Set column widths using percent

Description

Extraction of flextable print method with special handling of clintable pages and

Usage

```
clin_col_widths(x, ...)
```

Arguments

`x` A clintable object

`...` Named parameters where the names are columns in the flextable and the values are decimals representing the percent of total width of the table

Value

A clintable object

Examples

```
ct <- clintable(mtcars)

ct <- clin_alt_pages(
  ct,
  key_cols = c("mpg", "cyl", "hp"),
  col_groups = list(
    c("disp", "drat", "wt"),
    c("qsec", "vs", "am"),
    c("gear", "carb")
  )
) |>
  clin_col_widths(mpg = .2, cyl = .2, disp = .15, vs = .15)

print(ct)
```

clin_default_table_width

Get the Default Table Width for Clinical Documents

Description

This function calculates the default table width based on the page width and margins specified in the `clinify_docx_default` option.

Usage

```
clin_default_table_width()
```

Value

An rdocx object from the officer package

Examples

```
clin_default_table_width()
```

clin_group_by	<i>Configure a clintable to table by a grouping variable, which will be used as a label</i>
---------------	---

Description

Configure a clintable to table by a grouping variable, which will be used as a label

Usage

```
clin_group_by(x, group_by, caption_by = NULL, when = c("change", "notempty"))
```

Arguments

x	A clintable object
group_by	A character vector of variable names which will be used for grouping and attached as a label above the table headers
caption_by	A single element string of a variable name which will be used as a caption attached below the table body and above in the footer. Defaults to NULL.
when	Character string indicating how to identify groups and captions: • "change": Add padding when the value in group_by or caption_by changes from the previous row. • "notempty": Add padding when the value in group_by or caption_by is not empty.

Value

A clintable object

Examples

```
clintable(iris) |>
  clin_group_by("Species")
```

clin_group_pad	<i>Add Padding Between Groups in a Clinical Flextable</i>
----------------	---

Description

Adds top padding to rows in a ‘clintable’ based on changes in a grouping variable or non-empty values. Useful for visually separating groups in a table

Usage

```
clin_group_pad(
  x,
  pad_by,
  size = 9,
  when = c("change", "notempty"),
  drop = FALSE
)
```

Arguments

x	A clintable
pad_by	A string indicating the column name used to detect group changes.
size	Numeric value for the base padding size (default is 9).
when	Character string indicating when to apply padding: • "notempty": Add padding when the value in pad_by is not empty. • "change": Add padding when the value in pad_by changes from the previous row.
drop	Keep or drop the padding variable used to identify padding locations

Value

A clintable object with modified padding.

Examples

```
ct <- clintable(mtcars) |>
  clin_group_pad('gear')

ct <- clintable(mtcars) |>
  clin_group_pad('gear', size = 15)
```

clin_header_pad	<i>Set the spacing around a table's column headers</i>
-----------------	--

Description

Three pieces of vertical space shape the header block, and they are named here for where they sit rather than for the padding that produces them, because the mapping between the two is not obvious:

Usage

```
clin_header_pad(
    x,
    above = NULL,
    below = NULL,
    rule_to_body = NULL,
    rows = NULL
)
```

Arguments

x	A clintable object
above	Space above each header row, in points
below	Space below each header row, in points. The bottom row's is what sets how far the rule sits from the column labels
rule_to_body	Space between that rule and the first body row, in points. A single value - there is only one first body row per page
rows	Which header rows to space, as row numbers counting from the top. The default spaces every row; rows left out keep whatever spacing they already have

Details

- above is the space over each header row. On a single row header that is the buffer above the column labels; on a spanned header it also opens the space between the levels, which is what a blank row above the header normally looks like.
- below is the space under each header row. The one that matters most is the bottom row's, because a cell's bottom border sits at the bottom edge of the cell, *below* its padding - so this is what decides how far the rule is drawn from the column labels. It does not open space beneath the rule.
- rule_to_body is the space between that rule and the first row of the table body, which is the one that has to come from the body side.

above and below apply to every row of the header by default, which is the usual convention and matches `flextable::padding(part = "header")`. A header row that needs a different gap can be given one either by passing a value per row - `above = c(18, 34)` - or by aiming the call at particular rows with `rows`, which leaves the others alone. That matters because the spacing is applied as the table renders, after anything the caller did, so a call covering every row would otherwise overwrite a per-row `flextable::padding()` set beforehand.

rule_to_body is applied to the first row of every page, so a table split over pages keeps the same gap under the rule throughout. If a group label is added above the header it keeps its own spacing, since it is put there as the table renders.

Called a second time, this refines what the first call set rather than replacing it: arguments this call does not name keep their earlier value.

Spacing is given in points, which is what flextable measures cell padding in. Whatever is set here replaces the header padding clinify starts with.

Value

A clintable object

Examples

```
# A blank row's worth of space around each header row, the rule close under
# the labels, and a little air before the body starts
clintable(mtcars) |>
  clin_header_pad(above = 18, below = 4, rule_to_body = 6)
```

 clin_page_by

Configure pagination using a page variable

Description

Configure pagination using a page variable

Usage

```
clin_page_by(x, page_by, max_rows = 10)
```

Arguments

x	A clintable object
page_by	A variable in the input dataframe to use for pagination
max_rows	If no page_by, the maximum rows allowed per page

Value

A clintable object

Examples

```
dat <- mtcars
dat["page"] <- c(
  rep(1, 10),
  rep(2, 10),
  rep(3, 10),
  c(4, 4)
)

clintable(dat) |>
  clin_page_by("page")

clintable(mtcars) |>
  clin_page_by(max_rows = 10)
```

clin_replace_pagenums *Replace Table Cells with Word Page Number Fields*

Description

This helper function will find placeholder text and replace the fields of the flextable object with the appropriate page number fields. The function will search for the text {PAGE} and replace with the word field for current pages, and {NUMPAGES} for total pages. This allows you to current and total page fields within Word documents. Note that this is intended to be used in the defaults for clinify_titles_default or clinify_footnotes_default.

Usage

```
clin_replace_pagenums(x)
```

Arguments

x A clintable object

Value

A clintable object

Examples

```
title <- new_title_footnote(
  list(
    # We'll add tools to automate paging
    c("Protocol: CDISCPIL0T01", "Page {PAGE} of {NUMPAGES}"),
    c("Table 14-2.01"),
    c("Summary of Demographic and Baseline Characteristics")
  ),
  "titles"
)

title <- clin_replace_pagenums(title)

footnote <- new_title_footnote(
  list(
    # We'll add tools to automate paging
    c("Page {PAGE}", "Total Pages: {NUMPAGES}")
  ),
  "footnotes"
)

footnote <- clin_replace_pagenums(footnote)
```

clin_row_height	<i>Set the row height of a clintable</i>
-----------------	--

Description

Regulatory outputs are usually specified to an exact row pitch, and the rendered height of a row is what decides how much fits on a page. flextable leaves rows at a nominal quarter inch with a rule of "auto", which lets the renderer size them however it likes. This records the pitch you want and applies it when the table renders.

Usage

```
clin_row_height(
  x,
  body = NULL,
  title = NULL,
  footnote = NULL,
  header = NULL,
  header_leading = NULL,
  rule = c("atleast", "exact", "auto"),
  unit = c("pt", "in", "cm", "mm")
)
```

Arguments

x	A clintable object
body	Row pitch for the table body
title	Row pitch for the title lines
footnote	Row pitch for the footnote lines, and for a footnote page
header	Row pitch for the column header rows. A floor under rule = "atleast", so a header cell of several lines still grows
header_leading	Leading of the lines within the header, as a multiple of single spacing - 0.75 draws them three quarters as far apart. This is the only measurement here that is not a length, because flextable and Word both express leading as a multiple, so unit does not apply to it
rule	How the renderer should treat the pitch. "atleast" (the default) makes it a floor, so a cell whose text wraps grows past it instead of being clipped. "exact" pins the row to the pitch and clips anything that does not fit, which is the only way to get a pitch at or below the font's line height. "auto" lets the renderer decide, ignoring the pitch.
unit	Unit the pitches are given in. Row pitch is normally specified in points, so that is the default.

Details

The surfaces are set separately because they are separate tables: the table body, the column header, and the title and footnote blocks that go into the Word header and footer. Group label and caption rows, which clinify inserts while it renders, take the body pitch.

The column header has two levers rather than one, and they do different things. `header` bounds the header rows the same way `body` bounds the body rows, so with the default `rule = "atleast"` it is a floor: a header cell holding three lines still grows past it. `header_leading` is what closes the gap *between* those lines, so it is the one to reach for when a wrapped arm label sits looser than a reference output. They can be used together.

Called a second time, this refines what the first call set rather than replacing it: arguments this call does not name keep their earlier value. So a house wide pitch can be set once and a single table can add an exception without restating the rest.

The height is applied to whole parts, so it is a pitch for every row of the surface rather than a per-row height. Anything already set with `flextable::height()` or `flextable::height_all()` is replaced. Because the pitch is applied after the default styling functions run, it also holds when an organisation's `clinify_table_default()` sets a house pitch of its own - the table's own setting wins.

Value

A clintable object

Examples

```
clintable(mtcars) |>
  clin_row_height(body = 15.35, title = 11.4, footnote = 11.4)

# Or in inches
clintable(mtcars) |>
  clin_row_height(body = 0.213, unit = "in")

# A wrapped header label sitting too loose: bound the header rows and close
# up the lines inside them
clintable(mtcars) |>
  clin_column_headers(mpg = "Miles\nper\ngallon") |>
  clin_row_height(header = 13, header_leading = 0.75)
```

clin_spanner_rule

Draw a rule beneath the spanners in a clintable's column headers

Description

Regulatory arm spanner tables carry a thin horizontal rule directly under each spanner label, running across only the columns that spanner covers, so that the label reads as a heading over its own block of columns. The columns are worked out from the header that is on the table, so the rule follows the spanners as the layout changes instead of having to be given as column numbers that then have to be kept in step with it.

Usage

```
clin_spanner_rule(x, border = TRUE, rows = NULL)
```

Arguments

x	A clintable object
border	The pen to draw the rule with. TRUE, the default, draws the 1pt solid black rule these tables conventionally use. An officer::fp_border() draws in whatever width, style and colour it carries, which is how to get a dashed or a hairline rule. FALSE draws no rule, which is how to stop a house style from underlining the spanners.
rows	Header rows to rule, numbered from the top down. NULL, the default, rules every row above the bottom one, so a header of any depth has all of its spanners underlined.

Details

A spanner is a run of header cells that has been merged together, which is what clin_column_headers() makes of adjacent cells holding the same text. Two kinds of run are deliberately left alone:

- A run of blank cells. That is the empty space over a stub column, or over a trailing p-value column, rather than a spanner - clinify fills the header levels a column does not use, so those cells merge into a run of their own.
- Anything in the bottom row of the header. That row holds the individual column labels, so a merged run in it is a label sitting over two columns rather than a spanner, and the rule under the bottom row is the one the styling function draws across the whole table.

Called a second time, this refines what the first call set rather than replacing it: arguments this call does not name keep their earlier value.

The rule is drawn as the table renders, after the default styling function has run. That is what makes it survive a house style: the stock clinify_table_default() opens with flextable::border_remove(), which would wipe a border applied any earlier, and a house style is free to draw its header rules in a pen of its own.

Value

A clintable object

Examples

```
df <- data.frame(
  stub = c("Male", "Female"),
  a_lo = c("5 (10%)", "7 (14%)"),
  a_hi = c("2 (4%)", "3 (6%)"),
  b_lo = c("6 (12%)", "8 (16%)"),
  b_hi = c("1 (2%)", "4 (8%)")
)

ct <- clintable(df, use_labels = FALSE) |>
```

```

clin_column_headers(
  stub = "",
  a_lo = c("Drug A (N=50)", "Low"),
  a_hi = c("Drug A (N=50)", "High"),
  b_lo = c("Drug B (N=50)", "Low"),
  b_hi = c("Drug B (N=50)", "High")
)

# A rule under each arm spanner, over that arm's two columns only, with the
# stub left un-ruled
clin_spanner_rule(ct)

# The same rule in a dashed pen
clin_spanner_rule(ct, border = officer::fp_border(style = "dashed"))

```

clin_table_align	<i>Set how a table sits across the page</i>
------------------	---

Description

flextable centres a table on the page. Regulatory outputs are usually flush left, and a narrow table sometimes wants to be centred deliberately. The choice is recorded on the clintable and applied when the table renders, after the default styling function has run, so it holds even when an organisation's own `clinify_table_default()` rebuilds the table properties.

Usage

```
clin_table_align(x, align)
```

Arguments

x	A clintable object
align	One of "left", "center", or "right"

Details

This is only about where the table sits across the page. How wide it is, and how that width is divided between the columns, is `clin_col_widths()`.

Value

A clintable object

Examples

```

clintable(mtcars) |>
  clin_table_align("left")

```

make_grouped_pagenums *Assign Page Numbers to Presorted Grouped Data*

Description

Assigns sequential page numbers to elements of a vector, grouping by unique values and allocating a specified number of rows per page. The input vector must be presorted by group.

Usage

```
make_grouped_pagenums(var, rows)
```

Arguments

var	A vector of group labels, presorted so that identical values are contiguous.
rows	Integer. The maximum number of rows per page.

Details

The function splits the input vector into groups, then assigns page numbers within each group so that each page contains up to ‘rows’ items. Page numbers increment sequentially across groups. If the input is not presorted by group, the function will throw an error.

Value

An integer vector of the same length as ‘var’, indicating the assigned page number for each element.

Examples

```
library(dplyr)
iris |>
  mutate(
    page = make_grouped_pagenums(Species, 5)
  )
```

new_title_footnote *Create a new title or footnote flextable*

Description

Create a new title or footnote flextable

Usage

```
new_title_footnote(
  x,
  sect = c("titles", "footnotes", "footnote_page"),
  align = NULL
)
```

Arguments

x a list of character vectors, no more than 3 elements to a vector.

sect Either "titles" or "footnotes"

align Where to place each line - "left", "center", "right", "split", or NA for the default. One value per element of x, or a single value for all of them.

Value

A flextable object

Examples

```
title <- new_title_footnote(
  list(
    # We'll add tools to automate paging
    c("Protocol: CDISCPIL0T01", "Page {PAGE} of {NUMPAGES}"),
    c("Table 14-2.01"),
    c("Summary of Demographic and Baseline Characteristics")
  ),
  "titles"
)

footnote <- new_title_footnote(
  list(
    # We'll add tools to automate paging
    c("Page {PAGE}", "Total Pages: {NUMPAGES}")
  ),
  "footnotes"
)
```

print.clintable

Clintable print method

Description

Extraction of flextable print method with special handling of clintable pages and

Usage

```
## S3 method for class 'clintable'
print(x, n = 3, nrows = 15, apply_defaults = TRUE, ...)

## S3 method for class 'clintable'
knit_print(x, n = 3, nrows = 15, apply_defaults = TRUE, ...)
```

Arguments

x	A clintable object
n	Number of pages within the clintable to print. Only used when pagination is configured
nrows	Number of rows to print. Only used when rows aren't configured within the pagination method
apply_defaults	Apply default styles. These styles are stored in the options clinify_header_default, clinify_footer_default, and clinify_table_default respectively. Defaults to true.
...	Additional parameters passed to flextable print method

Value

Invisible

Examples

```
ct <- clintable(mtcars)

print(ct)

ct <- clin_alt_pages(
  ct,
  key_cols = c("mpg", "cyl", "hp"),
  col_groups = list(
    c("disp", "drat", "wt"),
    c("qsec", "vs", "am"),
    c("gear", "carb")
  )
)

print(ct)
```

write_clindoc

Clintable write method

Description

Write a clinify table out to a docx file

Usage

```
write_clindoc(x, file)
```

Arguments

x	a clintable object
file	The file path to which the file should be written

Value

Invisible

Examples

```
ct <- clintable(mtcars)

ct <- clin_alt_pages(
  ct,
  key_cols = c("mpg", "cyl", "hp"),
  col_groups = list(
    c("disp", "drat", "wt"),
    c("qsec", "vs", "am"),
    c("gear", "carb")
  )
)

# Get document object directly
doc <- clindoc(ct)

# Write out docx file
write_clindoc(ct, file.path(tempdir(), "demo.docx"))
```

Index

- * **Clinify Defaults**
 - clinify_titles_default, 4
- * **Print methods**
 - print.clintable, 24
- * **add_titles_footnotes**
 - clin_add_titles, 7

as_clindoc (clindoc), 3

as_clintable, 2

clin_add_footnote_page
(clin_add_titles), 7

clin_add_footnotes (clin_add_titles), 7

clin_add_titles, 7

clin_alt_pages, 9

clin_auto_page, 10

clin_col_widths, 12

clin_col_widths(), 22

clin_column_headers, 11

clin_default_table_width, 13

clin_group_by, 14

clin_group_pad, 14

clin_header_pad, 15

clin_page_by, 17

clin_replace_pagenums, 18

clin_row_height, 19

clin_spanner_rule, 20

clin_table_align, 22

clindoc, 3

clinify_caption_default
(clinify_titles_default), 4

clinify_docx_default
(clinify_titles_default), 4

clinify_footnotes_default
(clinify_titles_default), 4

clinify_grouplabel_default
(clinify_titles_default), 4

clinify_table_default
(clinify_titles_default), 4

clinify_titles_default, 4

clintable, 5

clintable(), 3

knit_print.clintable (print.clintable),
24

make_grouped_pagenums, 23

new_title_footnote, 23

print.clintable, 24

write_clindoc, 25