Package ‘r2rtf’

September 9, 2021

Title  Easily Create Production-Ready Rich Text Format (RTF) Table and Figure

Version  0.3.1

Description  Create production-ready Rich Text Format (RTF) table and figure with flexible format.

Depends  R (>= 3.5.0)

License  GPL-3

Encoding  UTF-8

VignetteBuilder  knitr

LazyData  true

RoxygenNote  7.1.1

Imports  grDevices

Suggests  stringi, knitr, markdown, emmeans, tidyrr, devtools, covr, dplyr, testthat, ggplot2


BugReports  https://github.com/Merck/r2rtf/issues

Config/testthat/edition  3

NeedsCompilation  no

Author  Yilong Zhang [aut, cre],
        Siruo Wang [aut],
        Simiao Ye [aut],
        Fansen Kong [aut],
        Nan Xiao [ctb],
        Madhusudhan Ginnaram [ctb],
        Ruchitbhai Patel [ctb],
        Huei-Ling Chen [ctb],
        Peikun Wu [ctb],
        Uday Preetam Palukuru [ctb],
        Daniel Woodie [ctb],
        Sarad Nepal [ctb],
        Jane Liao [ctb],
        Benjamin Wang [ctb].
R topics documented:

- `as_rtf_colheader` ........................................ 3
- `as_rtf_color` ............................................. 4
- `as_rtf_end` .................................................. 4
- `as_rtf_font` .................................................. 5
- `as_rtf_footnote` ............................................. 5
- `as_rtf_init` ................................................... 6
- `as_rtf_margin` ................................................. 6
- `as_rtf_new_page` ............................................. 7
- `as_rtf_page` ................................................... 7
- `as_rtf_pageby` ................................................ 8
- `as_rtf_paragraph` ............................................. 8
- `as_rtf_source` ................................................ 9
- `as_rtf_subline` ............................................... 9
- `as_rtf_table` ............................................... 10
- `as_rtf_title` ................................................ 10
- `border_type` .................................................. 11
- `cell_size` .................................................... 11
- `check_args` ................................................... 12
- `color_table` .................................................. 13
- `convert` ..................................................... 13
- `font_format` .................................................. 14
- `font_type` ..................................................... 14
- `footnote_source_space` ..................................... 15
- `inch_to_twip` ............................................... 15
- `justification` ............................................... 16
- `match_arg` .................................................... 16
- `nrow_paragraph` ............................................. 17
- `nrow_table` ................................................... 18
- `obj_rtf_border` ............................................. 19
- `obj_rtf_text` ............................................... 21
- `r2rtf_adae` .................................................. 23
- `r2rtf_adsl` .................................................. 23
- `r2rtf_HAMD17` ............................................... 24
- `r2rtf_tbl1` ................................................... 24
- `r2rtf_tbl2` ................................................... 25
- `r2rtf_tbl3` ................................................... 25
- `rtf_body` ..................................................... 25
as_rtf_colheader

Create Column Header RTF Encode

Description

Create Column Header RTF Encode

Usage

as_rtf_colheader(tbl)
Arguments

tbl  A data frame.

Specification

• Obtain column header attributes from tbl object.
• Extract column header total width from page col_width attribute.
• Define column header in RTF syntax using rtf_table_content().

as_rtf_color  Create RTF Color Encode

Description

Create RTF Color Encode

Usage

as_rtf_color(tbl)

Arguments

tbl  A data frame.

Specification

• Initiate RTF color using color_table() if use_color is TRUE in page attribute.
• Combine all components into a single RTF code string.

as_rtf_end  End RTF Encode

Description

End RTF Encode

Usage

as_rtf_end()

Specification

• Add symbol right curly bracket at the end of code.
Create RTF Font Encode

Usage

as_rtf_font()

Specification

- Initiate RTF font type using `font_type()`.
- Combine all components into a single RTF code string.

Create Footnote RTF Encode

Usage

as_rtf_footnote(tbl)

Arguments

 tbl  A data frame.

Specification

- Obtain footnote attributes from `tbl`.
- Define footnote in RTF syntax using `rtf_table_content()` if `as_table` attribute is TRUE.
- Define footnote in RTF syntax using `rtf_paragraph()` if `as_table` attribute is FALSE.
### as_rtf_init

**Create RTF Header Encode**

**Description**

Create RTF Header Encode

**Usage**

```r
as_rtf_init()
```

**Specification**

- Initiate RTF table by defining language #1033 (U.S. English).
- Define the initiation in RTF syntax.

### as_rtf_margin

**Create RTF Page Margin Encode**

**Description**

Create RTF Page Margin Encode

**Usage**

```r
as_rtf_margin(tbl)
```

**Arguments**

- `tbl` A data frame.

**Specification**

- Collect margin attributes from `tbl` object.
- Convert margin from inch to twip using `inch_to_twip()`.
- Define margin in RTF syntax.
**as_rtf_new_page**

*Create RTF New Page Encode*

**Description**

Create RTF New Page Encode

**Usage**

```r
as_rtf_new_page()
```

**Specification**

- Define new page in RTF syntax.

---

**as_rtf_page**

*Create RTF Page Size Encode*

**Description**

Create RTF Page Size Encode

**Usage**

```r
as_rtf_page(tbl)
```

**Arguments**

- `tbl` A data frame.

**Specification**

- Collect page attributes from `tbl` object.
- Convert page size from inch to twip using `inch_to_twip()`.
- Define page size in width, height and orientation (landscape or portrait) in RTF syntax.
as_rtf_pageby \hspace{1cm} RTF Table Page By Encoding

Description
RTF Table Page By Encoding

Usage
\texttt{as\_rtf\_pageby(tbl)}

Arguments
\begin{itemize}
  \item \texttt{tbl} \hspace{1cm} A data frame.
\end{itemize}

Specification
\begin{itemize}
  \item Collect all attributes from \texttt{tbl} object.
  \item Define table attributes using \texttt{rtf\_table\_content()}.  
\end{itemize}

as_rtf_paragraph \hspace{1cm} Create Paragraph RTF Encode

Description
Create Paragraph RTF Encode

Usage
\texttt{as\_rtf\_paragraph(text, combine = TRUE)}

Arguments
\begin{itemize}
  \item \texttt{text} \hspace{1cm} A character string.
  \item \texttt{combine} \hspace{1cm} A boolean to combine string or not.
\end{itemize}

Specification
\begin{itemize}
  \item Obtain title and subtitle text from \texttt{tbl} using \texttt{rtf\_text()}.  
  \item Define title and subtitle text font, size, format and color attributes.
  \item Return title/subtitle to header using \texttt{rtf\_paragraph()} if not NULL, otherwise return NULL to header.
\end{itemize}
as_rtf_source

Create Data Source RTF Encode

Description
Create Data Source RTF Encode

Usage
as_rtf_source(tbl)

Arguments
tbl A data frame.

Specification
- Obtain source attributes from tbl.
- Define source in RTF syntax using rtf_table_content() if as_table attribute is TRUE.
- Define source in RTF syntax using rtf_paragraph() if as_table attribute is FALSE.

as_rtf_subline

Create Table Subline RTF Encode

Description
Create Table Subline RTF Encode

Usage
as_rtf_subline(tbl)

Arguments
tbl A data frame.

Specification
- Obtain title and subtitle text from tbl using rtf_text().
- Define title and subtitle text font, size, format and color attributes.
- Return title/subtitle to header using rtf_paragraph() if not NULL, otherwise return NULL to header.
**as_rtf_table**  
*Combine RTF Table Encoding*

**Description**  
Combine RTF Table Encoding

**Usage**  
```r
as_rtf_table(tbl)
```

**Arguments**  
- `tbl`  
  A data frame.

**Specification**

- Calculate number of rows for table content, title, header, footnote and source for each page from ‘tbl’ object.
- Calculate number of pages using total number of rows divided by number of rows in each page.
- Extract first and last row for each page, assign border type and color attributes based on input from ‘rtf_body()’.
- Convert to RTF encoding using ‘rtf_table_content()’.
- Combine all components into a single code string.
- Add info attributes into ‘tbl’.

**as_rtf_title**  
*Create Table Title RTF Encode*

**Description**  
Create Table Title RTF Encode

**Usage**  
```r
as_rtf_title(tbl)
```

**Arguments**  
- `tbl`  
  A data frame.

**Specification**

- Obtain title attributes from tbl object.
- Define title in RTF syntax using `as_rtf_paragraph()` if it is not NULL, otherwise return NULL.
**border_type**

RTF Border Type Dictionary

**Description**

RTF Border Type Dictionary

**Usage**

border_type()

**Specification**

- Collect most commonly used border types for a table.
- Define the border types in RTF syntax.
- Create a mapping between border types and their RTF code.
- Return to ‘border_type()’ data frame to see all available border types.

---

**cell_size**

Calculate Cell Size in Twips

**Description**

Calculate Cell Size in Twips

**Usage**

cell_size(col_rel_width, col_total_width)

**Arguments**

- **col_rel_width** A vector of numbers separated by comma to indicate column relative width ratio.
- **col_total_width** A numeric number to indicate total column width.

**Specification**

- Convert inch to twip for cell size using .inch_to_twip().
check_args

Check Argument Types, Length or Dimension

Description

Check Argument Types, Length or Dimension

Usage

check_args(arg, type, length = NULL, dim = NULL)

Arguments

arg An argument to be checked.
type A character vector of candidate argument type.
length A numeric value of argument length or NULL
dim A numeric vector of argument dimension or NULL.

Details

if type, length or dim is NULL, the corresponding check will not be executed.

Value

Check failure detailed error message

Specification

• Check if arg is NULL.
• Extract the type, length and dim information from arg.
• Compare with target values and report error message if it does not match.

Examples

```r
## Not run:
tbl <- as.data.frame(matrix(1:9, nrow = 3))
check_args(arg = tbl, type = c("data.frame"))

vec <- c("a", "b", "c")
check_args(arg = vec, type = c("character"), length = 3)

## End(Not run)
```
**Description**

RTF Text Color Dictionary

**Usage**

```r
color_table()
```

**Specification**

- Collect all possible colors from R graphics devices.
- Define the colors to RGB conversion in RTF syntax.
- Combine all RGB components into a single RTF code string.
- Create a mapping between colors and their RTF code.
- Return to `color_table()` data frame to see the complete mapping.

---

**convert**  
*Convert Symbol to ANSI and Unicode Encoding*

**Description**

Convert Symbol to ANSI and Unicode Encoding

**Usage**

```r
convert(
  text,
  load_stringi = class(try(stringi::stri_replace_all_fixed, silent = TRUE)) != "try-error"
)
```

**Arguments**

- `text` A string to be converted.
- `load_stringi` A logical value to load stringi or not

**Specification**

- Define commonly used symbols in RTF syntax (superscript, subscript, greater than or equal to, less than or equal to, line break).
- Define Pattern for latex code.
- Declare fixed string in the pattern (no regular expression).
font_format

Description

RTF Text Format Dictionary

Usage

font_format()

Specification

• Collect most commonly used font formats (normal, bold, italics, underline, strike, superscript, and subscript).
• Define font format types in "", "b", "i", "u", "s", "^", "_".
• Create a mapping between font formats and their RTF code.

font_type

Description

RTF Text Font Dictionary

Usage

font_type()

Specification

• Collect most commonly used fonts (Times New Roman, Times New Roman Greek, and Arial Greek, etc.).
• Define font types from 1 to 10.
• Define font styles.
• Create a mapping between font types and their RTF code.
Derive Space Adjustment

Description

Derive Space Adjustment

Usage

footnote_source_space(tbl, text_indent_reference = "table")

Arguments

tbl A data frame.
text_indent_reference

The reference start point of text indent. Accept table or page_margin

Value

a value indicating the amount of space adjustment

Specification

• Collect page width, page margins and table width attributes from ‘tbl’ object.
• Convert the attributes from inch to twip using ‘inch_to_twip’.
• Derive the adjusted space by discounting page margins and table width from page width, then divided by 2.
• Set the adjusted space to 0 if previous derivation returns to negative value.

Convert Inches to Twips

Description

Convert Inches to Twips

Usage

inch_to_twip(inch)

Arguments

inch Value in inches.

Specification

• Convert inch to twips using conversion factor 1:1440.
justification

RTF Text Justification Dictionary

Description

RTF Text Justification Dictionary

Usage

justification()

Specification

- Collect most commonly used alignments for texts or rows (left, center, right, decimal, and justified).
- Define alignments/justifications in "l", "c", "r", "d", "j".
- Define the alignments/justifications for texts in RTF syntax.
- Define the alignments/justifications for rows in RTF syntax.
- Create a mapping between justifications and their RTF code.

match_arg

Argument Verification Using Partial Matching

Description

Similar to match.arg(), match_arg matches arg against a table of candidate values as specified by choices.

Usage

match_arg(arg, choices, several.ok = FALSE)

Arguments

- arg: a character vector (of length one unless several.ok is TRUE) or NULL.
- choices: a character vector of candidate values
- several.ok: logical specifying if arg should be allowed to have more than one element.

Details

This function resolves errors from match.arg() with "" as arg input.

Value

The matched elements of arg or in case of match failure a detailed error message.
nrow_paragraph

Specification

- Convert arg and choices inputs from numeric to characters.
- Input choices imputation if it is missing.
- Input arg imputation if it is NULL.
- Input several.ok check for arg length.
- Compare arg with choices values and report error message if it does not match.

Examples

```r
## Not run:
match_arg(arg = c(2, 1), choices = c(4, 3, 1, 2), several.ok = TRUE)
match_arg(arg = c("c", "b"), choices = c("a", "b", "c", "d"), several.ok = TRUE)
## End(Not run)
```

---

nrow_paragraph

Calculate Number of Rows for a Paragraph

Description

Calculate number of rows for a paragraph like title, subline, footnote, source

Usage

`nrow_paragraph(tbl, size, padding = 0.2)`

Arguments

<table>
<thead>
<tr>
<th>tbl</th>
<th>A data frame’s rtf_title, rtf_subline, rtf_footnote, or rtf_source attribute containing strwidth attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>size</td>
<td>Size of a line in inches</td>
</tr>
<tr>
<td>padding</td>
<td>Cell padding in inches</td>
</tr>
</tbody>
</table>

Value

an integer (number of rows) for title, subline, footnote, or source

Specification

- tbl is a data frame’s `rtf_title`, `rtf_subline`, `rtf_footnote`, or `rtf_source` attribute containing `strwidth` attribute.
- Return an integer (number of rows) for title, subline, footnote, or source
Examples

```
library(dplyr) # required for running example
tbl <- head(iris) %>%
  rtf_title(title = "Iris example") %>%
  rtf_footnote(footnote = c("footnote 1", "footnote 2")) %>%
  rtf_body()

r2rtf:::nrow_paragraph(attr(tbl, "rtf_title"), 6.25)
r2rtf:::nrow_paragraph(attr(tbl, "rtf_footnote"), 6.25)
```

---

**nrow_table**  
*Calculate Number of Lines Broken to for Each Table Row*

**Description**

Calculate number of lines broken to for each row of a table

**Usage**

```
nrow_table(tbl, size, page_size = size, padding = 0.2)
```

**Arguments**

- `tbl`  
  A data frame with attributes or a data frame’s `rtf_footnote` or `rtf_source` attributes
- `size`  
  Table size in inches
- `page_size`  
  Page size in inches
- `padding`  
  Cell padding in inches

**Value**

A numeric vector of number of maximum lines broken to for each row

**Specification**

- `tbl` is a data frame.
- Size is table’s width in inches.
- `Page_size` is page’s width in inches.
- Return to a numeric vector of number of maximum lines broken to for each row.

**Examples**

```
library(dplyr) # required for running example
tbl <- iris[c(1:4, 50:54), ] %>%
  rtf_title(title = "Iris example") %>%
  rtf_body()
r2rtf:::nrow_table(tbl, size = 2.55)
```
Create an RTF Table Border Object

**Description**

Create an RTF Table Border Object

**Usage**

```r
obj_rtf_border(
  tbl,
  border_left = "single",
  border_right = "single",
  border_top = "",
  border_bottom = "",
  border_first = "single",
  border_last = "single",
  border_color_left = NULL,
  border_color_right = NULL,
  border_color_top = NULL,
  border_color_bottom = NULL,
  border_color_first = NULL,
  border_color_last = NULL,
  border_width = 15,
  cell_height = 0.15,
  cell_justification = "c",
  cell_vertical_justification = "top",
  cell_nrow = NULL
)
```

**Arguments**

- `tbl`: A data frame.
- `border_left`: Left border type. To vary left border by column, use character vector with length of vector equal to number of columns displayed e.g. `c("single","single","single")`. All possible input can be found in `r2rtf::border_type()$name`.
- `border_right`: Right border type. To vary right border by column, use character vector with length of vector equal to number of columns displayed e.g. `c("single","single","single")`. All possible input can be found in `r2rtf::border_type()$name`.
- `border_top`: Top border type. To vary top border by column, use character vector with length of vector equal to number of columns displayed e.g. `c("single","single","single")`. If it is the first row in a table for this page, the top border is set to "double" otherwise the border is set to "single". All possible input can be found in `r2rtf::border_type()$name`. 
**border_bottom**  
Bottom border type. To vary bottom border by column, use character vector with length of vector equal to number of columns displayed e.g. `c("single","single","single")`. All possible input can be found in `r2rtf::border_type()$name`.

**border_first**  
First top border type of the whole table. All possible input can be found in `r2rtf::border_type()$name`.

**border_last**  
Last bottom border type of the whole table. All possible input can be found in `r2rtf::border_type()$name`.

**border_color_left**  
Left border color type. Default is NULL for black. To vary left border color by column, use character vector with length of vector equal to number of columns displayed e.g. `c("white","red","blue")`. All possible input can be found in `grDevices::colors()`.

**border_color_right**  
Right border color type. Default is NULL for black. To vary right border color by column, use character vector with length of vector equal to number of columns displayed e.g. `c("white","red","blue")`. All possible input can be found in `grDevices::colors()`.

**border_color_top**  
Top border color type. Default is NULL for black. To vary top border color by column, use character vector with length of vector equal to number of columns displayed e.g. `c("white","red","blue")`. All possible input can be found in `grDevices::colors()`.

**border_color_bottom**  
Bottom border color type. Default is NULL for black. To vary bottom border color by column, use character vector with length of vector equal to number of columns displayed e.g. `c("white","red","blue")`. All possible input can be found in `grDevices::colors()`.

**border_color_first**  
First top border color type of the whole table. Default is NULL for black. All possible input can be found in `grDevices::colors()`.

**border_color_last**  
Last bottom border color type of the whole table. Default is NULL for black. All possible input can be found in `grDevices::colors()`.

**border_width**  
Border width in twips. Default is 15 for 0.0104 inch.

**cell_height**  
Cell height in inches. Default is 0.15 for 0.15 inch.

**cell_justification**  
Justification type for cell. All possible input can be found in `r2rtf:::justification()$type`.

**cell_vertical_justification**  
Vertical justification type for cell. All possible input can be found in `r2rtf:::vertical_justification()$type`.

**cell_nrow**  
Number of rows required in each cell.

**Value**

the same tbl with additional border attributes
obj_rtf_text

Specifying

• Input checks using check_args(), match_arg() and stopifnot().
• Define border attributes based on the input.
• Register use_color attribute.
• Return tbl with attributes.

Description

Create an RTF Text Object

Usage

obj_rtf_text(
  text,
  text_font = 1,
  text_format = NULL,
  text_font_size = 9,
  text_color = NULL,
  text_background_color = NULL,
  text_justification = "l",
  text_indent_first = 0,
  text_indent_left = 0,
  text_indent_right = 0,
  text_space = 1,
  text_space_before = 15,
  text_space_after = 15,
  text_new_page = FALSE,
  text_hyphenation = TRUE,
  text_convert = TRUE
)

Arguments

text A character string.
text_font Text font type. Default is 1 for Times New Roman. To vary text font type by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(1,2,3). All possible input can be found in r2rtf:::font_type()$type.
text_format Text format type. Default is NULL for normal. Combination of format type are permitted as input for e.g. "ub" for bold and underlined text. To vary text format by column, use character vector with length of vector equal to number of columns displayed e.g. c("i","u","ib"). All possible input can be found in r2rtf:::font_format()$type.
text_font_size  Text font size. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(9,20,40).

text_color  Text color type. Default is NULL for black. To vary text color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in grDevices::colors().

text_background_color  Text background color type. Default is NULL for white. To vary text color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in grDevices::colors().

text_justification  Justification type for text. Default is "c" for center justification. To vary text justification by column, use character vector with length of vector equal to number of columns displayed e.g. c("c","l","r"). All possible input can be found in r2rtf:::justification()$type.

text_indent_first  A value of text indent in first line.

text_indent_left  A value of text left indent.

text_indent_right  A value of text right indent.

text_space  Line space between paragraph in twips. Default is 0.

text_space_before  Line space before a paragraph in twips.

text_space_after  Line space after a paragraph in twips.

text_new_page  A logical value to control whether display text in new page.

text_hyphenation  A logical value to control whether display text linked with hyphenation.

text_convert  A logical value to convert special characters.

Value

the same text (data frame or text) with additional attributes

Specification

- Input checks using check_args(), match_arg() and stopifnot().
- Define text attributes based on the input.
- Return text with attributes.
**r2rtf_adae**  
*An Adverse Event Dataset*

**Description**
A dataset containing the adverse event information of a clinical trial following CDISC ADaM standard.

**Usage**
r2rtf_adae

**Format**
A data frame with 1191 rows and 55 variables.

**Details**
Definition of each variable can be found in [https://bitbucket.cdisc.org/projects/CED/repos/sdtm-adam-pilot-project/browse](https://bitbucket.cdisc.org/projects/CED/repos/sdtm-adam-pilot-project/browse)

**Source**

---

**r2rtf_adsl**  
*A Subject Level Demographic Dataset*

**Description**
A dataset containing the demographic information of a clinical trial following CDISC ADaM standard.

**Usage**
r2rtf_adsl

**Format**
A data frame with 254 rows and 51 variables.

**Details**
Definition of each variable can be found in [https://bitbucket.cdisc.org/projects/CED/repos/sdtm-adam-pilot-project/browse](https://bitbucket.cdisc.org/projects/CED/repos/sdtm-adam-pilot-project/browse)
**Source**

https://bitbucket.cdisc.org/projects/CED/repos/sdtm-adam-pilot-project/browse

---

<table>
<thead>
<tr>
<th>r2rtf_HAMD17</th>
<th>An Efficacy Clinical Trial Data to Evaluate a Drug to Reduce Lower Back Pain</th>
</tr>
</thead>
</table>

**Description**

A dataset prepared by the Drug Information Association scientific working group to investigate a drug to reduce lower back pain.

**Usage**

r2rtf_HAMD17

**Format**

A data frame with 831 rows and 6 variables.

**Details**

Definition of each variable can be found in [https://missingdata.lshtm.ac.uk/dia-working-group/](https://missingdata.lshtm.ac.uk/dia-working-group/)

**Source**

https://missingdata.lshtm.ac.uk/dia-working-group/

---

<table>
<thead>
<tr>
<th>r2rtf_tbl1</th>
<th>Within Group Results from an ANCOVA Model</th>
</tr>
</thead>
</table>

**Description**

A dataset containing within group results from an ANCOVA model.

**Usage**

r2rtf_tbl1

**Format**

A data frame with 2 rows and 8 variables.
### r2rtf_tbl2

**Between Group Results from an ANCOVA Model**

**Description**
A dataset containing between group results from an ANCOVA model.

**Usage**

```
r2rtf_tbl2
```

**Format**
A data frame with 1 row and 3 variables.

---

### r2rtf_tbl3

**Root Mean Square Error from an ANCOVA model**

**Description**
A dataset containing root mean square error from an ANCOVA model.

**Usage**

```
r2rtf_tbl3
```

**Format**
A data frame with 1 row and 1 variable.

---

### rtf_body

**Add Table Body Attributes to the Table**

**Description**
Add Table Body Attributes to the Table
Usage

```r
tft_body(
  tbl,
  col_rel_width = rep(1, ncol(tbl)),
  as_colheader = TRUE,
  border_left = "single",
  border_right = "single",
  border_top = NULL,
  border_bottom = NULL,
  border_first = "single",
  border_last = "single",
  border_color_left = NULL,
  border_color_right = NULL,
  border_color_top = NULL,
  border_color_bottom = NULL,
  border_color_first = NULL,
  border_color_last = NULL,
  border_width = 15,
  cell_height = 0.15,
  cell_justification = "c",
  cell_vertical_justification = "top",
  cell_nrow = NULL,
  text_font = 1,
  text_format = NULL,
  text_font_size = 9,
  text_color = NULL,
  text_background_color = NULL,
  text_justification = NULL,
  text_indent_first = 0,
  text_indent_left = 0,
  text_indent_right = 0,
  text_space = 1,
  text_space_before = 15,
  text_space_after = 15,
  text_convert = TRUE,
  group_by = NULL,
  page_by = NULL,
  new_page = FALSE,
  pageby_header = TRUE,
  pageby_row = "column",
  subline_by = NULL,
  last_row = TRUE
)
```

Arguments

- **tbl**: A data frame.
- **col_rel_width**: Column relative width in a vector e.g. `c(2,1,1)` refers to 2:1:1. Default is NULL
for equal column width.

**as_colheader**  
A boolean value to indicate whether to add default column header to the table. Default is TRUE to use data frame column names as column header.

**border_left**  
Left border type. To vary left border by column, use character vector with length of vector equal to number of columns displayed e.g. c("single","single","single"). All possible input can be found in `r2rtf:::border_type()`$name.

**border_right**  
Right border type. To vary right border by column, use character vector with length of vector equal to number of columns displayed e.g. c("single","single","single"). All possible input can be found in `r2rtf:::border_type()`$name.

**border_top**  
Top border type. To vary top border by column, use character vector with length of vector equal to number of columns displayed e.g. c("single","single","single"). If it is the first row in a table for this page, the top border is set to "double" otherwise the border is set to "single". All possible input can be found in `r2rtf:::border_type()`$name.

**border_bottom**  
Bottom border type. To vary bottom border by column, use character vector with length of vector equal to number of columns displayed e.g. c("single","single","single"). All possible input can be found in `r2rtf:::border_type()`$name.

**border_first**  
First top border type of the whole table. All possible input can be found in `r2rtf:::border_type()`$name.

**border_last**  
Last bottom border type of the whole table. All possible input can be found in `r2rtf:::border_type()`$name.

**border_color_left**  
Left border color type. Default is NULL for black. To vary left border color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in `grDevices::colors()`.

**border_color_right**  
Right border color type. Default is NULL for black. To vary right border color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in `grDevices::colors()`.

**border_color_top**  
Top border color type. Default is NULL for black. To vary top border color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in `grDevices::colors()`.

**border_color_bottom**  
Bottom border color type. Default is NULL for black. To vary bottom border color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in `grDevices::colors()`.

**border_color_first**  
First top border color type of the whole table. Default is NULL for black. All possible input can be found in `grDevices::colors()`.

**border_color_last**  
Last bottom border color type of the whole table. Default is NULL for black. All possible input can be found in `grDevices::colors()`.
border_width   Border width in twips. Default is 15 for 0.0104 inch.
cell_height   Cell height in inches. Default is 0.15 for 0.15 inch.
cell_justification   Justification type for cell. All possible input can be found in \texttt{r2rtf:::justification()}$\texttt{type}$.
cell_vertical_justification   Vertical justification type for cell. All possible input can be found in \texttt{r2rtf:::vertical_justification}$\texttt{type}$.
cell_nrow   Number of rows required in each cell.
text_font   Text font type. Default is 1 for Times New Roman. To vary text font type by column, use numeric vector with length of vector equal to number of columns displayed e.g. \texttt{c(1,2,3)}. All possible input can be found in \texttt{r2rtf:::font_type()}$\texttt{type}$.
text_format   Text format type. Default is NULL for normal. Combination of format type are permitted as input for e.g. "ub" for bold and underlined text. To vary text format by column, use character vector with length of vector equal to number of columns displayed e.g. \texttt{c("i","u","ib"). All possible input can be found in \texttt{r2rtf:::font_format()}$\texttt{type}$.
text_font_size   Text font size. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. \texttt{c(9,20,40)}.
text_color   Text color type. Default is NULL for black. To vary text color by column, use character vector with length of vector equal to number of columns displayed e.g. \texttt{c("white","red","blue"). All possible input can be found in \texttt{grDevices::colors()}.text_background_color   Text background color type. Default is NULL for white. To vary text color by column, use character vector with length of vector equal to number of columns displayed e.g. \texttt{c("white","red","blue"). All possible input can be found in \texttt{grDevices::colors()}.text_justification   Justification type for text. Default is "c" for center justification. To vary text justification by column, use character vector with length of vector equal to number of columns displayed e.g. \texttt{c("c","l","r"). All possible input can be found in \texttt{r2rtf:::justification()}$\texttt{type}$.
text_indent_first   A value of text indent in first line.
text_indent_left   A value of text left indent.
text_indent_right   A value of text right indent.
text_space   Line space between paragraph in twips. Default is 0.
text_space_before   Line space before a paragraph in twips.
text_space_after   Line space after a paragraph in twips.
text_convert   A logical value to convert special characters.
group_by   A character vector of variable names in \texttt{tbl}.
page_by   Column names in a character vector to group by table in sections.
new_page  A boolean value to indicate whether to separate grouped table into pages by sections. Default is FALSE.

pageby_header  A boolean value to display pageby header at the beginning of each page.

pageby_row  A character vector of location of page_by variable. Possible input are 'column' or 'first_row'.

subline_by  Column names in a character vector to subline by table in sections.

last_row  A boolean value to indicate whether the table contains the last row of the final table.

Value

the same data frame tbl with additional attributes for table body

Specification

- Validate if input tbl argument is of type data.frame.
- Validate if input column relative width argument is of type integer or numeric.
- Validate if input column header argument is of type logical.
- Validate if input border and border color arguments are of type character.
- Validate if input border width and cell height arguments are of type integer or numeric.
- Validate if input cell justification argument is of type character.
- Validate if input text font, font size, space before and space after arguments are of type integer or numeric.
- Validate if input text format, color, background color and justification arguments are of type character.
- Validate if input group by and page by arguments are of type character.
- Validate if input new page, pageby header and last row arguments are of type integer or numeric.
- Validate if input border left, right, top, bottom, first and last arguments are valid using border_type()$name.
- Validate if input border color left, right, top, bottom, first and last arguments are valid using colors().
- Validate if input text color and background color arguments are valid using colors().
- Validate if input cell justification and text justification arguments are valid using justification()$type.
- Validate if input text font argument is valid using font_type()$type.
- Validate if input text format argument is valid using font_format()$type.
- Validate if input border width, cell height and text font size arguments are greater than 0.
- Validate if input text space before and text space after arguments are greater than or equal to 0.
- Add default page attributes if missing for input table data frame using rtf_page().
- Add page attribute use_color as TRUE if the input text, background or border color arguments are not black.
• Add column header attribute rtf_colheader if input column header argument is TRUE using rtf_colheader().
• Add black as default text color attribute if input text background color argument is not NULL and text color argument is NULL.
• Define matrices of same dimensions as input table data frame for non missing input arguments for border top, bottom, left, right, first and last.
• Define matrices of same dimensions as input table data frame for non missing input arguments for border color top, bottom, left, right, first and last.
• Define matrices of same dimensions as input table data frame for non missing input arguments for text font, format, color, background color, justification and font size.
• Add the defined matrices as attributes to input table data frame.
• Define pageby attributes using input page by, new page, pageby header arguments and rtf_pageby().
• Define table body attributes of tbl based on the input.
• Return tbl.

Examples

library(dplyr) # required to run examples
data(r2rtf_tbl1)
r2rtf_tbl1 %>%
  rtf_body(
    col_rel_width = c(3, 1, 3, 1, 3, 1, 3, 5),
    text_justification = c("l", rep("c", 7)),
    last_row = FALSE
  ) %>%
  attributes()

---

rtf_by_subline  Add Sublineby Attributes to Table

Description

Add Sublineby Attributes to Table

Usage

rtf_by_subline(tbl, subline_by)

Arguments

  tbl    A data frame.
  subline_by    Column names in a character vector to subline by table in sections.
Add Column Header Attributes to Table

Description
Add Column Header Attributes to Table

Usage

rtf_colheader(
  tbl,
  colheader = NULL,
  col_rel_width = NULL,
  border_left = "single",
  border_right = "single",
  border_top = "single",
  border_bottom = "",
  border_color_left = NULL,
  border_color_right = NULL,
  border_color_top = NULL,
  border_color_bottom = NULL,
  border_width = 15,
  cell_height = 0.15,
  cell_justification = "c",
  cell_vertical_justification = "bottom",
  cell_nrow = NULL,
  text_font = 1,
  text_format = NULL,
  text_font_size = 9,
  text_color = NULL,
  text_background_color = NULL,
  text_justification = "c",
  text_indent_first = 0,
  text_indent_left = 0,
  text_indent_right = 0,
  text_space = 1,
  text_space_before = 15,
  text_space_after = 15,
  text_convert = TRUE
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tbl</td>
<td>A data frame.</td>
</tr>
<tr>
<td>colheader</td>
<td>A character string that uses &quot;</td>
</tr>
</tbody>
</table>
col_rel_width  A Column relative width in a vector e.g. c(2,1,1) refers to 2:1:1. Default is NULL for equal column width.

border_left  Left border type. To vary left border by column, use character vector with length of vector equal to number of columns displayed e.g. c("single","single","single"). All possible input can be found in r2rtf:::border_type()$name.

border_right  Right border type. To vary right border by column, use character vector with length of vector equal to number of columns displayed e.g. c("single","single","single"). All possible input can be found in r2rtf:::border_type()$name.

border_top  Top border type. To vary top border by column, use character vector with length of vector equal to number of columns displayed e.g. c("single","single","single"). If it is the first row in a table for this page, the top border is set to "double" otherwise the border is set to "single". All possible input can be found in r2rtf:::border_type()$name.

border_bottom  Bottom border type. To vary bottom border by column, use character vector with length of vector equal to number of columns displayed e.g. c("single","single","single"). All possible input can be found in r2rtf:::border_type()$name.

border_color_left  Left border color type. Default is NULL for black. To vary left border color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in grDevices::colors().

border_color_right  Right border color type. Default is NULL for black. To vary right border color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in grDevices::colors().

border_color_top  Top border color type. Default is NULL for black. To vary top border color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in grDevices::colors().

border_color_bottom  Bottom border color type. Default is NULL for black. To vary bottom border color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in grDevices::colors().

border_width  Border width in twips. Default is 15 for 0.0104 inch.

cell_height  Cell height in inches. Default is 0.15 for 0.15 inch.

cell_justification  Justification type for cell. All possible input can be found in r2rtf:::justification()$type.

cell_vertical_justification  Vertical justification type for cell. All possible input can be found in r2rtf:::vertical_justification()

cell_nrow  Number of rows required in each cell.

text_font  Text font type. Default is 1 for Times New Roman. To vary text font type by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(1,2,3). All possible input can be found in r2rtf:::font_type()$type.
**rtf_colheader**

- **text_format**: Text format type. Default is NULL for normal. Combination of format type are permitted as input for e.g. "ub" for bold and underlined text. To vary text format by column, use character vector with length of vector equal to number of columns displayed e.g. c("i","u","ib"). All possible input can be found in `r2rtf:::font_format()`$type.
- **text_font_size**: Text font size. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(9,20,40).
- **text_color**: Text color type. Default is NULL for black. To vary text color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in `grDevices::colors()`.
- **text_background_color**: Text background color type. Default is NULL for white. To vary text color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in `grDevices::colors()`.
- **text_justification**: Justification type for text. Default is "c" for center justification. To vary text justification by column, use character vector with length of vector equal to number of columns displayed e.g. c("c","l","r"). All possible input can be found in `r2rtf:::justification()`$type.
- **text_indent_first**: A value of text indent in first line.
- **text_indent_left**: A value of text left indent.
- **text_indent_right**: A value of text right indent.
- **text_space**: Line space between paragraph in twips. Default is 0.
- **text_space_before**: Line space before a paragraph in twips.
- **text_space_after**: Line space after a paragraph in twips.
- **text_convert**: A logical value to convert special characters.

**Value**

The same data frame tbl with additional attributes for table column header.

**Specification**

- Input checks using `check_args()`, `match_arg()` and `stopifnot()`. The required argument is tbl, i.e. A data frame must define by tbl.
- Set default page attributes and register `use_color` attribute.
- Define column header attributes of tbl based on the input.
- Return tbl.
Examples

library(dplyr) # required to run examples
data(r2rtf_tbl1)
r2rtf_tbl1 %>%
  rtf_colheader(
    colheader = "Treatment | N | Mean (SD) | N | Mean (SD) | N |
                Mean (SD) | LS Mean (95% CI)\dagger",
    text_format = c("b", "u", "u", "u", "i")
  ) %>%
  attr("rtf_colheader")

---

**rtf_convert_format**  
**Convert RTF file to Other Format**

**Description**

This is a experimental function.

**Usage**

```r
rtf_convert_format(
  input,
  output_file = NULL,
  output_dir = ".",
  format = "pdf",
  overwrite = FALSE
)
```

**Arguments**

- `input`: A vector of file paths for the input file to be converted.
- `output_file`: A vector of filename for the output file. Default is the same filename for input.
- `output_dir`: The output directory for the converted output_dir.
- `format`: Converted file format extension. Currently support "pdf" or "docx"
- `overwrite`: logical; should existing destination files be overwritten?

**Details**

Convert RTF files to PDF or DOCX files. Require libreoffice7.1.

**Value**

A vector of file paths for the converted files.
**rtf_encode**  
*Render to RTF Encoding*

**Description**

This function extracts table/figure attributes and render to RTF encoding that is ready to save to an RTF file.

**Usage**

```r
ttf_encode(
  tbl,
  doc_type = "table",
  page_title = "all",
  page_footnote = "last",
  page_source = "last",
  verbose = FALSE
)
```

**Arguments**

- **tbl**: A data frame for table or a list of binary string for figure.
- **doc_type**: The doc_type of input, default is table.
- **page_title**: A character of title displaying location. Possible values are "first", "last" and "all".
- **page_footnote**: A character of title displaying location. Possible values are "first", "last" and "all".
- **page_source**: A character of title displaying location. Possible values are "first", "last" and "all".
- **verbose**: a boolean value to return more details of RTF encoding.

**Value**

- For `rtf_encode`, a vector of RTF code.
- For `write_rtf`, no return value.

**Specification**

- Input check for doc_type ("table" or "figure").
- Input check for title, footnote and source position ("all", "first" or "last").
- If doc_type is "table" and class is data.frame then run rtf_encode_table(tbl).
- If doc_type is "table" and class is list then run rtf_encode_list(tbl).
- If doc_type is "figure" then run rtf_encode_figure(tbl).
Examples

library(dplyr) # required to run examples

# Example 1
head(iris) %>%
  rtf_body() %>%
  rtf_encode() %>%
  write_rtf(file = file.path(tempdir(), "table1.rtf"))

# Example 2
## Not run:
library(dplyr) # required to run examples
file <- file.path(tempdir(), "figure1.png")
png(file)
plot(1:10)
dev.off()

# Read in PNG file in binary format
rtf_read_png(file) %>%
  rtf_figure() %>%
  rtf_encode(doc_type = "figure") %>%
  write_rtf(file = file.path(tempdir(), "figure1.rtf"))

## End(Not run)
# Example 3

## convert tbl_1 to the table body. Add title, subtitle, two table
## headers, and footnotes to the table body.
data(r2rtf_tbl2)
## convert r2rtf_tbl2 to the table body. Add a table column header to table body.
t2 <- r2rtf_tbl2 %>%
  rtf_colheader(
    colheader = "Pairwise Comparison | Difference in LS Mean(95% CI)\dagger | p-Value",
    text_justification = c("l", "c", "c")
  ) %>%
  rtf_body(
    col_rel_width = c(8, 7, 5),
    text_justification = c("l", "c", "c"),
    last_row = FALSE
  )

# concatenate a list of table and save to an RTF file
t2 %>%
  rtf_encode() %>%
  write_rtf(file.path(tempdir(), "table2.rtf"))
**rtf_encode_list**

**Description**

Render List to RTF Encoding

**Usage**

`rtf_encode_list(tbl)`

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>tbl</code></td>
<td>A data frame for table or a list of binary string for figure.</td>
</tr>
</tbody>
</table>

**Specification**

- Collect footnote attributes from `tbl` object.
- Define page, margin, header, footnote, source and new_page in RTF syntax.
- Define page height and width in RTF syntax.
- Initiate RTF using `as_rtf_init()` and `as_rtf_font()`.
- Get page title display location ("all", "first", "last") from arg input and display it in page accordingly.
- Get page footnote display location ("all", "first", "last") from arg input and display it in page accordingly.
- Get page source display location ("all", "first", "last") from arg input and display it in page accordingly.
- Translate all `tbl` attributes into RTF syntax.
- Combine all components into a single RTF code string.
**Specification**

- Collect color attributes from tbl object.
- Initiate RTF using `as_rtf_init()`, `as_rtf_font()` and color syntax obtained from previous step.
- Translate all tbl attributes into RTF syntax.
- Combine all components into a single RTF code string.

---

**Description**

Render Table to RTF Encoding

**Usage**

```r
trf_encode_table(tbl, verbose = FALSE)
```

**Arguments**

- `tbl` A data frame for table or a list of binary string for figure.
- `verbose` a boolean value to return more details of RTF encoding.

**Specification**

- Initiate RTF using `as_rtf_init()`, `as_rtf_font()` and `as_rtf_color()`.
- Define page, margin, header, footnote, source and new_page in RTF syntax.
- Define column header, first border and last border type in RTF syntax.
- Check whether footnote and source will be displayed as table if they exist.
- Define table content in RTF syntax.
- Get page title display location ("all", "first", "last") from arg input and display it in page accordingly.
- Get page footnote display location ("all", "first", "last") from arg input and display it in page accordingly.
- Get page source display location ("all", "first", "last") from arg input and display it in page accordingly.
- Translate all tbl attributes into RTF syntax.
- Combine all components into a single RTF code string.
**rtf_figure**  
*Add Figure Attributes*

**Description**

Add Figure Attributes

**Usage**

```r
tft_figure(tbl, fig_width = 5, fig_height = 5)
```

**Arguments**

- `tbl`: A data frame.
- `fig_width`: the width of figures in inch
- `fig_height`: the height of figures in inch

**Value**

the same data frame `tbl` with additional attributes for figure body

**Specification**

- If page attributes are NULL then assign default page attributes using `rtf_page()` function.
- Check if input width and height are greater than zero.
- Define figure width and height attributes based on the inputs.
- Return to `tbl` with figure width and height attributes.

**Examples**

```r
## Not run:
library(dplyr) # required to run examples
tbl <- file.path(tempdir(), "figure1.png")
png(file)
plot(1:10)
dev.off()

# Read in PNG file in binary format
rtf_read_png(file) %>%
  rtf_figure() %>%
  attributes()

## End(Not run)
```
rtf_footnote  Add Footnote Attributes to Table

Description
Add Footnote Attributes to Table

Usage

```r
tft_footnote(
  tbl,
  footnote = "",
  border_left = "single",
  border_right = "single",
  border_top = "",
  border_bottom = "single",
  border_color_left = NULL,
  border_color_right = NULL,
  border_color_top = NULL,
  border_color_bottom = NULL,
  border_width = 15,
  cell_height = 0.15,
  cell_justification = "c",
  cell_vertical_justification = "top",
  cell_nrow = NULL,
  text_font = 1,
  text_format = NULL,
  text_font_size = 9,
  text_color = NULL,
  text_background_color = NULL,
  text_justification = "1",
  text_indent_first = 0,
  text_indent_left = 0,
  text_indent_right = 0,
  text_indent_reference = "table",
  text_space = 1,
  text_space_before = 15,
  text_space_after = 15,
  text_convert = TRUE,
  as_table = TRUE
)
```

Arguments

- `tbl` A data frame.
- `footnote` A vector of character for footnote text.
border_left  Left border type. To vary left border by column, use character vector with length of vector equal to number of columns displayed e.g. c("single","single","single"). All possible input can be found in \texttt{r2rtf::border_type()$name}.

border_right Right border type. To vary right border by column, use character vector with length of vector equal to number of columns displayed e.g. c("single","single","single"). All possible input can be found in \texttt{r2rtf::border_type()$name}.

border_top Top border type. To vary top border by column, use character vector with length of vector equal to number of columns displayed e.g. c("single","single","single"). If it is the first row in a table for this page, the top border is set to "double" otherwise the border is set to "single". All possible input can be found in \texttt{r2rtf::border_type()$name}.

border_bottom Bottom border type. To vary bottom border by column, use character vector with length of vector equal to number of columns displayed e.g. c("single","single","single"). All possible input can be found in \texttt{r2rtf::border_type()$name}.

border_color_left Left border color type. Default is NULL for black. To vary left border color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in \texttt{grDevices::colors()}. 

border_color_right Right border color type. Default is NULL for black. To vary right border color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in \texttt{grDevices::colors()}. 

border_color_top Top border color type. Default is NULL for black. To vary top border color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in \texttt{grDevices::colors()}. 

border_color_bottom Bottom border color type. Default is NULL for black. To vary bottom border color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in \texttt{grDevices::colors()}. 

border_width Border width in twips. Default is 15 for 0.0104 inch.

cell_height Cell height in inches. Default is 0.15 for 0.15 inch.

cell_justification Justification type for cell. All possible input can be found in \texttt{r2rtf::justification()$type}. 

cell_vertical_justification Vertical justification type for cell. All possible input can be found in \texttt{r2rtf::vertical_justification()}. 

cell_nrow Number of rows required in each cell. 

text_font Text font type. Default is 1 for Times New Roman. To vary text font type by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(1,2,3). All possible input can be found in \texttt{r2rtf::font_type()$type}. 

text_format Text format type. Default is NULL for normal. Combination of format type are permitted as input for e.g. "ub" for bold and underlined text. To vary text format by column, use character vector with length of vector equal to number
of columns displayed e.g. c("i","u","ib"). All possible input can be found in r2rtf:::font_format()$type.

text_font_size  Text font size. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(9,20,40).

text_color     Text color type. Default is NULL for black. To vary text color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in grDevices::colors().

text_background_color  Text background color type. Default is NULL for white. To vary text background color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in grDevices::colors().

text_justification  Justification type for text. Default is "c" for center justification. To vary text justification by column, use character vector with length of vector equal to number of columns displayed e.g. c("c","l","r"). All possible input can be found in r2rtf:::justification()$type.

text_indent_first  A value of text indent in first line.

text_indent_left   A value of text left indent.

text_indent_right  A value of text right indent.

text_indent_reference  The reference start point of text indent. Accept table or page_margin.

text_space     Line space between paragraph in twips. Default is 0.

text_space_before  Line space before a paragraph in twips.

text_space_after   Line space after a paragraph in twips.

text_convert  A logical value to convert special characters.

as_table  A logical value to display it as a table.

Value

the same data frame tbl with additional attributes for table footnote

Specification

- Define footnote attributes of tbl based on the input.
- Return tbl.

Examples

library(dplyr) # required to run examples
data(r2rtf_tbl1)
r2rtf_tbl1 %>%
  rtf_footnote("\dagger Based on an ANCOVA model.") %>
  attr("rtf_footnote")
### rtf_group_by_enhance  
**Remove Duplicate Records**

**Description**
Remove Duplicate Records

**Usage**
```r
tft_group_by_enhance(tbl, group_by, page_index)
```

**Arguments**
- `tbl`: A data frame.
- `group_by`: A character vector of variable names in `tbl`.
- `page_index`: A numeric vector of page index.

**Value**
Return `tbl`.

**Specification**
- Define `id` variable to split data frame.
- Remove duplicate records within each splitted data frame.

---

### rtf_nline_matrix  
**Calculate Number of Lines of a String Matrix**

**Description**
Calculate each string matrix (e.g., table body in matrix format) row’s maximum number of lines broken to given a specific cell size

**Usage**
```r
tft_nline_matrix(text, strwidth, size)
```
Arguments

text a matrix of string
strwidth a matrix of string width in inches
size a matrix of cell size in inches

# @section Specification:
• text is a matrix of string
• strwidth is a matrix of string width in inches
• size is a matrix of cell size in inches
• Return a vector of integer (number of lines)

Value

a vector of integer (number of lines)

Examples

text <- matrix("this is a sentence", nrow = 2, ncol = 2)
strwidth <- matrix(6:9, nrow = 2)
size <- matrix(1:4, nrow = 2)
r2rtf::rtf_nline_matrix(text = text, strwidth = strwidth, size = size)

rtf_nline_vector Calculate Number of Lines of a String Vector

Description

Calculate number of lines that a string vector (e.g., title, subline, footnote, source) broken to given
a specific cell size

Usage

rtf_nline_vector(text, strwidth, size)

Arguments

text a vector of string
strwidth a vector of string width in inches
size a vector of cell size in inches

Value

a vector of integer (number of lines)
**rtf_nrow**

**Specification**

- text is a vector of string
- strwidth is a vector of string width in inches
- size is a vector of cell size in inches
- Return a vector of integer (number of lines)

**Examples**

```r
c(1:4, 50:54), df> df
```

---

**Description**

Add number of row attributes for a table

**Usage**

```r
rtf_nrow(tbl)
```

**Arguments**

- `tbl` A data frame

**Value**

a data frame with number of row attributes

**Specification**

- tbl is a data frame.
- Return to a data frame with number of row attributes.

**Examples**

```r
library(dplyr) # required for running example
tbl <- iris[c(1:4, 50:54), df> df
```

```r
c(1:4, 50:54), df> df
```
Add RTF File Page Information

Arguments

- **tbl**: A data frame.
- **orientation**: Orientation in 'portrait' or 'landscape'.
- **width**: A numeric value of page width in inches.
- **height**: A numeric value of page width in inches.
- **margin**: A numeric vector of length 6 for page margin. The value set left, right, top, bottom, header and footer margin in order. Default value depends on the page orientation and set by `r2rtf:::set_margin("wma", orientation)`.
- **nrow**: Number of rows in each page.
- **border_first**: First top border type of the whole table. All possible input can be found in `r2rtf:::border_type()$name`.
- **border_last**: Last bottom border type of the whole table. All possible input can be found in `r2rtf:::border_type()$name`.
- **border_color_first**: First top border color type of the whole table. Default is NULL for black. All possible input can be found in `grDevices::colors()`.
- **border_color_last**: Last bottom border color type of the whole table. Default is NULL for black. All possible input can be found in `grDevices::colors()`.
- **col_width**: A numeric value of total column width in inch. Default is `width - ifelse(orientation == "portrait", 2.25, 2.5)`.
Add Pageby Attributes to Table

Description

Add Pageby Attributes to Table

Usage

```r
rtf_pageby(
  tbl,
  page_by = NULL,
  new_page = FALSE,
  pageby_header = TRUE,
  pageby_row = "column"
)
```

Arguments

- **tbl**: A data frame.
- **page_by**: Column names in a character vector to group by table in sections.
- **new_page**: A boolean value to indicate whether to separate grouped table into pages by sections. Default is FALSE.
- **pageby_header**: A boolean value to display pageby header at the beginning of each page.
- **pageby_row**: A character vector of location of page_by variable. Possible input are 'column' or 'first_row'.

Examples

```r
library(dplyr) # required to run examples
data(r2rtf_tbl1)
r2rtf_tbl1 %>%
  rtf_page() %>%
  attr("page")
```
Add RTF Page Footer Information

**Usage**

```r
tft_page_footer(
  tbl,
  text,
  text_font = 1,
  text_format = NULL,
  text_font_size = 12,
  text_color = NULL,
  text_background_color = NULL,
  text_justification = "c",
  text_indent_first = 0,
  text_indent_left = 0,
  text_indent_right = 0,
  text_space = 1,
  text_space_before = 15,
  text_space_after = 15,
  text_convert = TRUE
)
```

**Arguments**

- **tbl**
  A data frame.

- **text**
  A character string.

- **text_font**
  Text font type. Default is 1 for Times New Roman. To vary text font type by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(1,2,3). All possible input can be found in `r2rtf:::font_type()`$type.

- **text_format**
  Text format type. Default is NULL for normal. Combination of format type are permitted as input for e.g. "ub" for bold and underlined text. To vary text format by column, use character vector with length of vector equal to number of columns displayed e.g. c("i","u","ib"). All possible input can be found in `r2rtf:::font_format()`$type.

- **text_font_size**
  Text font size. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(9,20,40).

- **text_color**
  Text color type. Default is NULL for black. To vary text color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in `grDevices::colors()`.
text_background_color
Text background color type. Default is NULL for white. To vary text color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in \textit{grDevices::colors(}).

text_justification
Justification type for text. Default is "c" for center justification. To vary text justification by column, use character vector with length of vector equal to number of columns displayed e.g. c("c","l","r"). All possible input can be found in \textit{r2rtf:::justification()}$type.

text_indent_first
A value of text indent in first line.

text_indent_left
A value of text left indent.

text_indent_right
A value of text right indent.

text_space
Line space between paragraph in twips. Default is 0.

text_space_before
Line space before a paragraph in twips.

text_space_after
Line space after a paragraph in twips.

text_convert
A logical value to convert special characters.

---

**rtf_page_header**

\textit{Add RTF Page Header Information}

**Description**

Add RTF Page Header Information

**Usage**

```r
rtf_page_header(
  tbl,
  text = "Page \pagenumber of \pagefield",
  text_font = 1,
  text_format = NULL,
  text_font_size = 12,
  text_color = NULL,
  text_background_color = NULL,
  text_justification = "r",
  text_indent_first = 0,
  text_indent_left = 0,
  text_indent_right = 0,
  text_space = 1,
  text_space_before = 15,
  text_space_after = 15,
  text_convert = TRUE
)
```
Arguments

- **tbl**
  A data frame.

- **text**
  A character string.

- **text_font**
  Text font type. Default is 1 for Times New Roman. To vary text font type by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(1,2,3). All possible input can be found in `r2rtf:::font_type()`$type.

- **text_format**
  Text format type. Default is NULL for normal. Combination of format type are permitted as input for e.g. "ub" for bold and underlined text. To vary text format by column, use character vector with length of vector equal to number of columns displayed e.g. c("i","u","ib"). All possible input can be found in `r2rtf:::font_format()`$type.

- **text_font_size**
  Text font size. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(9,20,40).

- **text_color**
  Text color type. Default is NULL for black. To vary text color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in `grDevices::colors()`.

- **text_background_color**
  Text background color type. Default is NULL for white. To vary text color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in `grDevices::colors()`.

- **text_justification**
  Justification type for text. Default is "c" for center justification. To vary text justification by column, use character vector with length of vector equal to number of columns displayed e.g. c("c","l","r"). All possible input can be found in `r2rtf:::justification()`$type.

- **text_indent_first**
  A value of text indent in first line.

- **text_indent_left**
  A value of text left indent.

- **text_indent_right**
  A value of text right indent.

- **text_space**
  Line space between paragraph in twips. Default is 0.

- **text_space_before**
  Line space before a paragraph in twips.

- **text_space_after**
  Line space after a paragraph in twips.

- **text_convert**
  A logical value to convert special characters.
Paragraph to RTF Encode

Usage

```r
ttf_paragraph(
  text,
  justification = "c",
  indent_first = 0,
  indent_left = 0,
  indent_right = 0,
  space = 1,
  space_before = 180,
  space_after = 180,
  new_page = FALSE,
  hyphenation = TRUE,
  cell = FALSE
)
```

Arguments

- **text**: rtf text obtained using `rtf_text()` function.
- **justification**: Justification for text.
- **indent_first**: First indent.
- **indent_left**: Left indent.
- **indent_right**: Right indent.
- **space**: Paragraph space.
- **space_before**: Line space before text.
- **space_after**: Line space after text.
- **new_page**: A boolean value to indicate whether to start a new page.
- **hyphenation**: A boolean value to indicate whether to use hyphenation.
- **cell**: A boolean value to indicate if paragraph is in table cell.

Specification

- Validate if input paragraph justification is valid using `justification()`.
- Validate if input paragraph spacing is valid using `spacing()`.
- Validate if input indent and space arguments are numeric.
• Add left curly bracket followed by RTF syntax: two backward slashes followed by pard, to start of code.
• Add RTF syntax: two backward slashes followed by pagebb, if new_page argument is TRUE.
• Add RTF syntax: two backward slashes followed by sb, at start of line space_before argument.
• Add RTF syntax: two backward slashes followed by sa, at start of line space_after argument.
• Define paragraph space based on input argument for space and spacing().
• Add RTF syntax: two backward slashes followed by fi, at start of line indent_first argument.
• Add RTF syntax: two backward slashes followed by li, at start of line indent_left argument.
• Add RTF syntax: two backward slashes followed by ri, at start of line indent_right argument.
• Define paragraph justification based on input argument for justification and justification().
• Add RTF syntax: two backward slashes followed by hyphpar, if hyphenation argument is TRUE.
• Add RTF syntax: two backward slashes followed by hyphpar0, if hyphenation argument is FALSE.
• Add RTF syntax: two backward slashes followed by par, followed by right curly bracket to end of code.
• Combine all components into a single RTF code string.

---

**rtf_read_png**

**Read PNG Figures into Binary Files**

**Description**

Read PNG Figures into Binary Files

**Usage**

```r
rtf_read_png(file)
```

**Arguments**

- `file` A character vector of PNG file paths.

**Value**

A list of binary data vector returned by `readBin`

**Specification**

- Read PNG figures into binary file using `lapply` and `readBin`
rtf_source

Examples

## Not run:
file <- file.path(tempdir(), "figure1.png")
png(file)
plot(1:10)
dev.off()

# Read in PNG file in binary format
rtf_read_png(file)

## End(Not run)

---

rtf_source  Add Data Source Attributes to the Table

Description

Add Data Source Attributes to the Table

Usage

```r
tft_source(
  tbl,
  source = "",
  border_left = "single",
  border_right = "single",
  border_top = "",
  border_bottom = "single",
  border_color_left = NULL,
  border_color_right = NULL,
  border_color_top = NULL,
  border_color_bottom = NULL,
  border_width = 15,
  cell_height = 0.15,
  cell_justification = "c",
  cell_vertical_justification = "top",
  cell_nrow = NULL,
  text_font = 1,
  text_format = NULL,
  text_font_size = 9,
  text_color = NULL,
  text_background_color = NULL,
  text_justification = "c",
  text_indent_first = 0,
  text_indent_left = 0,
  text_indent_right = 0,
)```

```
Arguments

tbl A data frame.
source A character string.
border_left Left border type. To vary left border by column, use character vector with length of vector equal to number of columns displayed e.g. c("single","single","single"). All possible input can be found in r2rtf:::border_type()$name.
border_right Right border type. To vary right border by column, use character vector with length of vector equal to number of columns displayed e.g. c("single","single","single"). All possible input can be found in r2rtf:::border_type()$name.
border_top Top border type. To vary top border by column, use character vector with length of vector equal to number of columns displayed e.g. c("single","single","single"). If it is the first row in a table for this page, the top border is set to "double" otherwise the border is set to "single". All possible input can be found in r2rtf:::border_type()$name.
border_bottom Bottom border type. To vary bottom border by column, use character vector with length of vector equal to number of columns displayed e.g. c("single","single","single"). All possible input can be found in r2rtf:::border_type()$name.
border_color_left Left border color type. Default is NULL for black. To vary left border color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in grDevices::colors().
border_color_right Right border color type. Default is NULL for black. To vary right border color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in grDevices::colors().
border_color_top Top border color type. Default is NULL for black. To vary top border color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in grDevices::colors().
border_color_bottom Bottom border color type. Default is NULL for black. To vary bottom border color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in grDevices::colors().
border_width Border width in twips. Default is 15 for 0.0104 inch.
cell_height Cell height in inches. Default is 0.15 for 0.15 inch.
cell_justification
Justification type for cell. All possible input can be found in `r2rtf:::justification()`$type.

cell_vertical_justification
Vertical justification type for cell. All possible input can be found in `r2rtf:::vertical_justification()`$type.

cell_nrow
Number of rows required in each cell.

text_font
Text font type. Default is 1 for Times New Roman. To vary text font type by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(1,2,3). All possible input can be found in `r2rtf:::font_type()`$type.

text_format
Text format type. Default is NULL for normal. Combination of format type are permitted as input for e.g. "ub" for bold and underlined text. To vary text format by column, use character vector with length of vector equal to number of columns displayed e.g. c("i","u","ib"). All possible input can be found in `r2rtf:::font_format()`$type.

text_font_size
Text font size. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(9,20,40).

text_color
Text color type. Default is NULL for black. To vary text color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in `grDevices:::colors()`.

text_background_color
Text background color type. Default is NULL for white. To vary text color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in `grDevices:::colors()`.

text_justification
Justification type for text. Default is "c" for center justification. To vary text justification by column, use character vector with length of vector equal to number of columns displayed e.g. c("c","l","r"). All possible input can be found in `r2rtf:::justification()`$type.

text_indent_first
A value of text indent in first line.

text_indent_left
A value of text left indent.

text_indent_right
A value of text right indent.

text_indent_reference
The reference start point of text indent. Accept `table` or `page_margin`.

text_space
Line space between paragraph in twips. Default is 0.

text_space_before
Line space before a paragraph in twips.

text_space_after
Line space after a paragraph in twips.

text_convert
A logical value to convert special characters.

as_table
A logical value to display it as a table.

Value
the same data frame `tbl1` with additional attributes for data source of a table
Specification

- Define data source attributes of `tbl` based on the input.
- Return `tbl`.

Examples

```r
library(dplyr) # required to run examples
data(r2rtf_tbl1)
r2rtf_tbl1 %>%
  rtf_source("Source: [study999:adam-adef]") %>%
  attr("rtf_source")
```

---

**rtf_strwidth**  
*Calculate String Width in Inches*

**Description**

Calculate string width in inches based on font (Times New Roman, Arial, etc.), font size, font style (bold, italic, bold-italic etc.), and text indent.

**Usage**

```r
rtf_strwidth(tbl)
```

**Arguments**

- `tbl`  
  A data frame

**Value**

an object with string width corresponding to each cell in the data frame `tbl`

**Specification**

- `tbl` is a data frame.
- Return an object with string width corresponding to each cell in the data frame `tbl`.

**Examples**

```r
library(dplyr)
tbl <- data.frame(
  x = rep("This is a long sentence", 5),
  y = "short"
)
tbl %>%
  rtf_body(text_font = c(1, 3)) %>%
r2rtf:::rtf_strwidth()
```
**Description**

Add subline attributes to the object

**Usage**

```r
rtf_subline(
  tbl,
  text,
  text_font = 1,
  text_format = NULL,
  text_font_size = 12,
  text_color = NULL,
  text_background_color = NULL,
  text_justification = "l",
  text_indent_first = 0,
  text_indent_left = 0,
  text_indent_right = 0,
  text_indent_reference = "table",
  text_space = 1,
  text_space_before = 180,
  text_space_after = 180,
  text_convert = TRUE
)
```

**Arguments**

- `tbl` A data frame.
- `text` A character vector of subline text.
- `text_font` Text font type. Default is 1 for Times New Roman. To vary text font type by column, use numeric vector with length of vector equal to number of columns displayed e.g. `c(1,2,3). All possible input can be found in `r2rtf:::font_type()`$type.
- `text_format` Text format type. Default is NULL for normal. Combination of format type are permitted as input for e.g. "ub" for bold and underlined text. To vary text format by column, use character vector with length of vector equal to number of columns displayed e.g. c("i","u","ib"). All possible input can be found in `r2rtf:::font_format()`$type.
- `text_font_size` Text font size. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(9,20,40).
- `text_color` Text color type. Default is NULL for black. To vary text color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in `grDevices::colors()`.
text_background_color
   Text background color type. Default is NULL for white. To vary text color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in grDevices::colors().

text_justification
   Justification type for text. Default is "c" for center justification. To vary text justification by column, use character vector with length of vector equal to number of columns displayed e.g. c("c","l","r"). All possible input can be found in r2rtf:::justification()$type.

text_indent_first
   A value of text indent in first line.

text_indent_left
   A value of text left indent.

text_indent_right
   A value of text right indent.

text_indent_reference
   The reference start point of text indent. Accepts table or page_margin.

text_space
   Line space between paragraph in twips. Default is 0.

text_space_before
   Line space before a paragraph in twips.

text_space_after
   Line space after a paragraph in twips.

text_convert
   A logical value to convert special characters.

Value
   the same data frame tbl with additional attributes for table title

Specification
   • Define title attributes of tbl based on the input.
   • Return tbl.

---

rtf_subset  Pass Table Attributes to Subset Table

Description
   Pass original table attributes assigned through like rtf_page, rtf_title, rtf_body... to subsetted table because original attributes won’t be automatically carried over.

Usage
   rtf_subset(tbl, row = 1:nrow(tbl), col = 1:ncol(tbl))
**Arguments**

- **tbl**
  - A data frame with attributes.

- **row**
  - a numeric vector for the row index to keep in the subsetted data.

- **col**
  - a numeric vector for the column index to keep in the subsetted data.

**Value**

the subsetted data frame `tbl_sub` with original attributes from `tbl`

**Specification**

- `tbl` is a data frame with attributes to be subsetted.
- Return a data frame `tbl_sub` subsetted from `tbl` with original table attributes.

**Examples**

```r
library(dplyr)
data(r2rtf_tbl1)
sub_table <- r2rtf_tbl1 %>%
  rtf_body() %>%
r2rtf:::rtf_subset(row = 1:2, col = c(1, 4:5))

attributes(sub_table)
```

---

**Description**

Create RTF Table Body Encode

**Usage**

```r
rtf_table_content(
  tbl,
  col_total_width = attr(tbl, "page")$col_width,
  use_border_bottom = FALSE
)
```

**Arguments**

- **tbl**
  - A data frame.

- **col_total_width**
  - Column total width for the table. Default is the corresponding attribute from `tbl`.

- **use_border_bottom**
  - A logical value of using the bottom border. Default is the corresponding attribute from `tbl`.
Specification

- Define table begin and end in RTF syntax.
- Define cell justification using 'justification()' and 'vertical_justification', then covert the cell from inch to twip using 'inch_to_twip()' in RTF syntax.
- Define cell border type using 'border_type()' and cell border width in RTF syntax.
- Define cell border color using 'color_table()' in RTF syntax.
- Define cell background color using input variable text_background_color in RTF syntax.
- Define cell size using 'cell_size()' in RTF syntax.
- Combine cell component attributes into a single code string.
- Define cell content in encoded RTF syntax.
- Check if cell content format is a valid value.
- Combine cell content and content component attributes into a single code string.

```r
text_to_rtfEncode(
    text = character(),
    font = integer(),
    font_size = integer(),
    format = character(),
    color = character(),
    background_color = character(),
    text_convert = logical()
)
```

Arguments

- `text` - Plain text.
- `font` - Text font type.
- `font_size` - Text font size.
- `format` - Text format.
- `color` - Text color.
- `background_color` - Text background color.
- `text_convert` - A logical value to convert special characters.
 Specification

- Set font color default value to black if background color value is not NULL and color value is NULL.
- Validate if input font type is valid using font_type().
- Validate if input font format is valid using font_format().
- Validate if input table color is valid using color_table().
- Convert latex character to Unicode using convert().
- Add left curly bracket to start of code and right curly bracket to the end of code.
- Combine all components into a single code string.

```
rtf_title(tbl,
          title = NULL,
          subtitle = NULL,
          text_font = 1,
          text_format = NULL,
          text_font_size = 12,
          text_color = NULL,
          text_background_color = NULL,
          text_justification = "c",
          text_indent_first = 0,
          text_indent_left = 0,
          text_indent_right = 0,
          text_indent_reference = "table",
          text_space = 1,
          text_space_before = 180,
          text_space_after = 180,
          text_convert = TRUE)
```

Arguments

- tbl: A data frame.
- title: Title in a character string.
- subtitle: Subtitle in a character string.
**text_font**
Text font type. Default is 1 for Times New Roman. To vary text font type by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(1,2,3). All possible input can be found in r2rtf:::font_type()$type.

**text_format**
Text format type. Default is NULL for normal. Combination of format type are permitted as input for e.g. "ub" for bold and underlined text. To vary text format by column, use character vector with length of vector equal to number of columns displayed e.g. c("i","u","ib"). All possible input can be found in r2rtf:::font_format()$type.

**text_font_size**
Text font size. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(9,20,40).

**text_color**
Text color type. Default is NULL for black. To vary text color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in grDevices::colors().

**text_background_color**
Text background color type. Default is NULL for white. To vary text color by column, use character vector with length of vector equal to number of columns displayed e.g. c("white","red","blue"). All possible input can be found in grDevices::colors().

**text_justification**
Justification type for text. Default is "c" for center justification. To vary text justification by column, use character vector with length of vector equal to number of columns displayed e.g. c("c","l","r"). All possible input can be found in r2rtf:::justification()$type.

**text_indent_first**
A value of text indent in first line.

**text_indent_left**
A value of text left indent.

**text_indent_right**
A value of text right indent.

**text_indent_reference**
The reference start point of text indent. Accept table or page_margin

**text_space**
Line space between paragraph in twips. Default is 0.

**text_space_before**
Line space before a paragraph in twips.

**text_space_after**
Line space after a paragraph in twips.

**text_convert**
A logical value to convert special characters.

**Value**
the same data frame tbl with additional attributes for table title

**Specification**
- Input checks using check_args(), match_arg() and stopifnot(). The required argument is tbl, i.e. A data frame must define by tbl.
- Set default page attributes and register use_color attribute.
• Define title attributes of tbl based on the input.
• Return tbl.

Examples

```
library(dplyr) # required to run examples
data(r2rtf_tbl1)
r2rtf_tbl1 %>%
  rtf_title(title = "ANCOVA of Change from Baseline at Week 8") %>%
  attr("rtf_title")
```

---

**set_margin**  
*Define Margin Type*

**Description**

Define Margin Type

**Usage**

```
set_margin(doc_type, orientation)
```

**Arguments**

- `doc_type`: doc_type in 'csr', 'wma', 'wmm' or 'narrow'
- `orientation`: Orientation in 'portrait' or 'landscape'.

**Specification**

- Define document margin by assigning margin values for left, right, top, bottom, header and footer.
- Define document orientation.

---

**spacing**  
*RTF Paragraph Spacing Dictionary*

**Description**

RTF Paragraph Spacing Dictionary

**Usage**

```
spacing()
```
Specification

- Collect most commonly used paragraph spacing (single-space, double-space, and 1.5-space).
- Define the paragraph spacing type in 1, 2, 1.5.
- Create a mapping between paragraph spacing and their RTF code.

---

<table>
<thead>
<tr>
<th>unicode_latex</th>
<th>Dictionary of Unicode and Latex Code</th>
</tr>
</thead>
</table>

Description

A dataset containing the mapping between unicode and latex code.

Usage

```r
unicode_latex
```

Format

A data frame with 681 rows and 3 variables.

- **unicode**: unicode, UTF-8 code
- **latex**: latex, latex code
- **int**: int, Converted integer of the UTF-8 code

Source


---

<table>
<thead>
<tr>
<th>update_border_first</th>
<th>Update First Border Line Based on Page Information</th>
</tr>
</thead>
</table>

Description

Update first border line type and color type based on page information.

Usage

```r
update_border_first(tbl)
```

Arguments

- **tbl**: A data frame
**update_border_last**

**Value**

a data frame tbl with updated top border type and top border color type attributes

**Specification**

- tbl is a data frame.
- Return a data frame tbl with updated top border type and top border color type attributes.

**Examples**

```r
library(dplyr)
tbl <- iris[c(1:3, 51:54), ] %>%
  rtf_body(page_by = "Species") %>%
  r2rtf:::update_border_first()
```

---

**update_border_last**

Update Last Border Line Based on Page Information

**Description**

Update last border line type and color type based on page information.

**Usage**

```r
update_border_last(tbl)
```

**Arguments**

- tbl  
  A data frame

**Value**

a data frame tbl with updated last border type and last border color type attributes

**Specification**

- tbl is a data frame.
- Return a data frame tbl with updated last border type and last border color type attributes.

**Examples**

```r
library(dplyr)
tbl <- iris[c(1:3, 51:54), ] %>%
  rtf_body(page_by = "Species") %>%
  r2rtf:::update_border_last()
```
utf8Tortf

Convert a UTF-8 Encoded Character String to a RTF Encoded String

Description

Convert a UTF-8 Encoded Character String to a RTF Encoded String

Usage

utf8Tortf(text)

Arguments

text

A string to be converted.

If the unicode of a character is 255 or under (including all character on a keyboard), the character is as is. If the unicode of a character is larger than 255, the character will be encoded.

Specification

• Define rules for character by setting 255 as cutoff.
• If the unicode of a character is 255 or under (including all character on a keyboard), the character is as is.
• If the unicode of a character is larger than 255, the character will be encoded.

References


---

vertical_justification

RTF Text Vertical Justification Dictionary

Description

RTF Text Vertical Justification Dictionary

Usage

vertical_justification()  

Specification

• Collect most commonly used vertical alignments for texts or rows (top and bottom).
• Create a mapping between justifications and their RTF code.
**write_rtf**  
Write an RTF Table or Figure to an RTF File

**Description**

The write_rtf function writes rtf encoding string to an .rtf file

**Usage**

```r
code
write_rtf(rtf, file)
```

**Arguments**

- `rtf`  
  A character rtf encoding string rendered by rtf_encode().
- `file`  
  A character string naming a file to save rtf file.

**Specification**

- Export a single RTF string into an file using `write` function.

---

**write_rtf_para**  
Write a Paragraph to an RTF File

**Description**

Write a Paragraph to an RTF File

**Usage**

```r
code
write_rtf_para(rtf, file)
```

**Arguments**

- `rtf`  
  rtf code for text paragraph, obtained using `rtf_paragraph(text,...)` function
- `file`  
  file name to save rtf text paragraph, eg. filename.rtf

**Specification**

- Define table color using `color_table()` and translate in RTF syntax.
- Initiate rtf using `as_rtf_init()` and `as_rtf_font()`.
- Combine the text with other components into a single RTF code string.
- Output the paragraph into a file.
Index

* datasets
  r2rtf_adae, 23
  r2rtf_ads1, 23
  r2rtf_HAMD17, 24
  r2rtf_tbl11, 24
  r2rtf_tbl12, 25
  r2rtf_tbl13, 25
  unicode_latex, 64

as_rtf_colheader, 3
as_rtf_color, 4
as_rtf_end, 4
as_rtf_font, 5
as_rtf_footnote, 5
as_rtf_init, 6
as_rtf_margin, 6
as_rtf_new_page, 7
as_rtf_page, 7
as_rtf_pageby, 8
as_rtf_paragraph, 8
as_rtf_source, 9
as_rtf_subline, 9
as_rtf_table, 10
as_rtf_title, 10

border_type, 11

cell_size, 11
check_args, 12
color_table, 13
convert, 13

font_format, 14
font_type, 14
footnote_source_space, 15

inch_to_twip, 15

justification, 16

match_arg, 16

nrow_paragraph, 17
nrow_table, 18

obj_rtf_border, 19
obj_rtf_text, 21

r2rtf_adae, 23
r2rtf_ads1, 23
r2rtf_HAMD17, 24
r2rtf_tbl11, 24
r2rtf_tbl12, 25
r2rtf_tbl13, 25
rtf_body, 25
rtf_by_subline, 30
rtf_colheader, 31
rtf_convert_format, 34
rtf_encode, 35
rtf_encode_figure, 36
rtf_encode_list, 37
rtf_encode_table, 38
rtf_figure, 39
rtf_footnote, 40
rtf_group_by_enhance, 43
rtf_nline_matrix, 43
rtf_nline_vector, 44
rtf_nrow, 45
rtf_page, 46
rtf_page_footer, 48
rtf_page_header, 49
rtf_pageby, 47
rtf_paragraph, 51
rtf_read_png, 52
rtf_source, 53
rtf_strwidth, 56
rtf_subline, 57
rtf_subset, 58
rtf_table_content, 59
rtf_text, 60
rtf_title, 61
INDEX

set_margin, 63
spacing, 63

unicode_latex, 64
update_border_first, 64
update_border_last, 65
utf8Tortf, 66

vertical_justification, 66

write_rtf, 67
write_rtf_para, 67