Package 'r2rtf'

June 10, 2026

Title Easily Create Production-Ready Rich Text Format (RTF) Tables and Figures

Version 1.3.1

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

License GPL-3

URL https://merck.github.io/r2rtf/, https://github.com/Merck/r2rtf

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

Encoding UTF-8

VignetteBuilder knitr

LazyData true

Depends R (>= 3.5.0)

Imports grDevices, tools

Suggests covr, dplyr, emmeans, ggplot2, knitr, magrittr, officer, rmarkdown, stringi, testthat, tidyr, xml2

Config/testthat/edition 3

RoxygenNote 7.3.3

NeedsCompilation no

Author Yilong Zhang [aut],
Siruo Wang [aut],
Simiao Ye [aut],
Fansen Kong [aut],
Brian Lang [aut],
Benjamin Wang [aut],
Nan Xiao [ctb],
Madhusudhan Ginnaram [ctb],
Ruchitbhai Patel [aut, cre],
Huei-Ling Chen [ctb],
Peikun Wu [ctb],
Uday Preetham Palukuru [ctb],
Daniel Woodie [ctb],

Sarad Nepal [ctb],
Jane Liao [ctb],
Jeff Cheng [ctb],
Yirong Cao [ctb],
Amin Shirazi [ctb],
Yihui Xie [ctb],
Günter Milde [ctb] (Original author of the unimathsymbols.txt file),
Merck Sharp & Dohme Corp [cph]

Maintainer Ruchitbhai Patel <ruchitbhai.patel@merck.com>

Repository CRAN

Date/Publication 2026-06-10 05:20:12 UTC

Contents

assemble_docx *Assemble Multiple RTF Table Listing and Figure Into One Word Document*

Description

The function assemble multiple RTF table, listing, and figures into one document as Microsoft Word (i.e., docx).

Usage

```
assemble_docx(input, output, landscape = FALSE)
```

Arguments

input	Character vector of file path.
output	Character string to the output file path.
landscape	Logical vector to determine page direction.

Specification

- Transfer files to toggle fields format in Word
- Insert into Word file using officer

Examples

```
library(officer)
library(magrittr)

file <- replicate(2, tempfile(fileext = ".rtf"))
file1 <- head(iris) %>%
  rtf_body() %>%
  rtf_encode() %>%
  write_rtf(file[1])
file2 <- head(cars) %>%
  rtf_page(orientation = "landscape") %>%
  rtf_body() %>%
  rtf_encode() %>%
  write_rtf(file[2])
output <- tempfile(fileext = ".docx")

assemble_docx(
  input = file,
  output = output
)
```

assemble_rtf *Assemble Multiple RTF Table Listing and Figure Into One RTF Document*

Description

The function assemble multiple RTF table, listing, and figures into one document as RTF file.

Usage

```
assemble_rtf(input, output, landscape = FALSE)
```

Arguments

input	Character vector of file path.
output	Character string to the output file path.
landscape	Logical value to determine page direction.

Specification

- Read individual RTF files.
- Insert into one RTF file.

Examples

```
library(magrittr)

file <- replicate(2, tempfile(fileext = ".rtf"))
file1 <- head(iris) %>%
  rtf_body() %>%
  rtf_encode() %>%
  write_rtf(file[1])
file2 <- head(cars) %>%
  rtf_page(orientation = "landscape") %>%
  rtf_body() %>%
  rtf_encode() %>%
  write_rtf(file[2])
output <- tempfile(fileext = ".rtf")

assemble_rtf(
  input = file,
  output = output
)
```

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://github.com/phuse-org/phuse-scripts/tree/master/data/adam/cdisc

Source

https://github.com/phuse-org/phuse-scripts/tree/master/data/adam/cdisc

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://github.com/phuse-org/phuse-scripts/tree/master/data/adam/cdisc

Source

https://github.com/phuse-org/phuse-scripts/tree/master/data/adam/cdisc

---

`r2rtf_HAMD17` *An Efficacy Clinical Trial Data to Evaluate a Drug to Reduce Lower Back Pain*

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

This dataset was prepared by the Drug Information Association (DIA) scientific working group to investigate methods for handling missing data. It contains data from an efficacy clinical trial evaluating a drug to reduce lower back pain.

Source

Drug Information Association (DIA) scientific working group on missing data

---

`r2rtf_tbl1` *Within Group Results from an ANCOVA Model*

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

```
rtf_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 `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`.

`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. The unit is twip.

`text_indent_left`
A value of text left indent. The unit is twip.

`text_indent_right`
A value of text right indent. The unit is twip.

`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 `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. Default is TRUE. If the value is FALSE, the pageby header is displayed in the first page of the pageby group. The special pageby value "-----" is to avoid displaying a pageby header for this group.
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_colheader` *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_hyphenation = TRUE,
  text_convert = TRUE
)
```

Arguments

`tbl`	A data frame.
`colheader`	A character string that uses " | " to separate column names. Default is NULL for a blank column header.
`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`.

`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. The unit is twip.

`text_indent_left`
: A value of text left indent. The unit is twip.

`text_indent_right`
: A value of text right indent. The unit is twip.

`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_hyphenation`
: A logical value to control whether display text linked with hyphenation.

`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_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

```
rtf_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 \code{rtf_encode}, a vector of RTF code.
For \code{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_figure(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_figure` *Add Figure Attributes*

Description

Add Figure Attributes

Usage

```
rtf_figure(tbl, fig_width = 5, fig_height = 5, fig_format = NULL)
```

Arguments

tbl	A data frame.
fig_width	the width of figures in inch
fig_height	the height of figures in inch
fig_format	the figure format defined in r2rtf:::fig_format()

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

```
## 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_figure(file) %>%
  rtf_figure() %>%
  attributes()

## End(Not run)
```

`rtf_footnote` *Add Footnote Attributes to Table*

Description

Add Footnote Attributes to Table

Usage

```
rtf_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 = "l",
  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 `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`.

`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. The unit is twip.

`text_indent_left`
A value of text left indent. The unit is twip.

`text_indent_right`
A value of text right indent. The unit is twip.

`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_page` *Add RTF File Page Information*

Description

Add RTF File Page Information

Usage

```
rtf_page(
  tbl,
  orientation = "portrait",
  width = ifelse(orientation == "portrait", 8.5, 11),
  height = ifelse(orientation == "portrait", 11, 8.5),
  margin = set_margin("wma", orientation),
  nrow = ifelse(orientation == "portrait", 40, 24),
  border_first = "double",
  border_last = "double",
  border_color_first = NULL,
  border_color_last = NULL,
  col_width = width - ifelse(orientation == "portrait", 2.25, 2.5),
  use_color = FALSE,
  use_i18n = FALSE
)
```

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, 2.5)
use_color	A logical value to use color in the output.
use_i18n	A logical value to enable internationalization fonts (e.g., SimSun for Chinese). Default is FALSE.

Value

the same data frame tbl with additional attributes for page features

Specification

- Check if all argument types and values are valid inputs.
- Add attributes to ‘tbl‘ based on the inputs.
- Register the use of color in page attributes.
- Return to ‘tbl‘ with page attributes.

Examples

```
library(dplyr) # required to run examples
data(r2rtf_tbl1)
r2rtf_tbl1 %>%
  rtf_page() %>%
  attr("page")
```

---

rtf_page_footer *Add RTF Page Footer Information*

Description

Add RTF Page Footer Information

Usage

```
rtf_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 `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. The unit is twip.
text_indent_left	A value of text left indent. The unit is twip.
text_indent_right	A value of text right indent. The unit is twip.
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` *Add RTF Page Header Information*

Description

Add RTF Page Header Information

Usage

```
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. The unit is twip.

`text_indent_left`
: A value of text left indent. The unit is twip.

`text_indent_right`
: A value of text right indent. The unit is twip.

`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_read_figure` *Read Figures into Binary Files*

Description

Supported format is listed in `r2rtf:::fig_format()`.

Usage

```
rtf_read_figure(file)
```

Arguments

`file`	A character vector of figure file paths.

Value

a list of binary data vector returned by `readBin`

Specification

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

Examples

```
## Not run:

# Read in PNG file in binary format
file <- tempfile("figure", fileext = ".png")
png(file)
plot(1:10)
dev.off()


rtf_read_figure(file)

# Read in EMF file in binary format
library(devEMF)
file <- tempfile("figure", fileext = ".emf")
emf(file)
plot(1:10)
dev.off()

rtf_read_figure(file)

## End(Not run)
```

`rtf_read_png` *Read PNG Figures into Binary Files*

Description

Read PNG Figures into Binary Files

Usage

```
rtf_read_png(file)
```

Arguments

`file`	A character vector of PNG file paths.

Value

a list of binary data vector returned by `readBin`

Specification

- Deprecated: rtf_read_png. Use rtf_read_figure instead

`rtf_rich_text` *Text to Formatted RTF Encode*

Description

Text to Formatted RTF Encode

Usage

```
rtf_rich_text(
  text,
  theme = list(.emph = list(format = "i"), .strong = list(format = "b"))
)
```

Arguments

`text`	Plain text.
`theme`	Named list defining themes for tags. See `rtf_text()` for details on possible formatting.

Specification

- Validate if theme list items correspond to `font_type()` arguments.
- Create regex expressions to match “ and ‘.tag‘ in text.
- Extract tagged text from input text.
- Extract tags from tagged text.
- Extract text from tagged text.
- Validate that lengths of extractions are all the same.
- Validate that tags are defined in the ‘theme‘ argument.
- Execute `rtf_text()` with extracted text and relevant formatting.
- Reinsert encoded formatted text to original input text.

Examples

```
rtf_rich_text(
  text = paste(
    "This is {.emph important}.",
    "This is {.strong relevant}.", "This is {.zebra ZEBRA}."
  ),
  theme = list(
    .emph = list(format = "i"),
    .strong = list(format = "b"),
    .zebra = list(color = "white", background_color = "black")
  )
)
```

`rtf_source` *Add Data Source Attributes to the Table*

Description

Add Data Source Attributes to the Table

Usage

```
rtf_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,
  text_indent_reference = "table",
  text_space = 1,
  text_space_before = 15,
  text_space_after = 15,
  text_convert = TRUE,
  as_table = FALSE
)
```

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`

`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. The unit is twip.

`text_indent_left`
A value of text left indent. The unit is twip.

`text_indent_right`
A value of text right indent. The unit is twip.

`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 data source of a table

Specification

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

Examples

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

`rtf_subline` *Add Subline Attributes to Table*

Description

Add subline attributes to the object

Usage

```
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_hyphenation = TRUE,
  text_convert = TRUE
)
```

Arguments

`tbl`	A data frame.
`text`	A character vector of subline
`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. The unit is twip.

`text_indent_left`
: A value of text left indent. The unit is twip.

`text_indent_right`
: A value of text right indent. The unit is twip.

`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_hyphenation`
: A logical value to control whether display text linked with hyphenation.

`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_title` *Add Title Attributes to Table*

Description

Add title, subtitle, and other attributes to the object

Usage

```
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_hyphenation = TRUE,
  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. The unit is twip.

`text_indent_left`
: A value of text left indent. The unit is twip.

`text_indent_right`
: A value of text right indent. The unit is twip.

`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_hyphenation`
: A logical value to control whether display text linked with hyphenation.

`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")
```

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 less than 128 (including all character on a keyboard), the character is as is. If the unicode of a character is larger or equal to 128, the character will be encoded.

Specification

- Define rules for character by setting 128 as cutoff.
- If the unicode of a character is 128 or under (including all character on a keyboard), the character is as is.
- If the unicode of a character is larger than 128, the character will be encoded.

References

Burke, S. M. (2003). RTF Pocket Guide. " O'Reilly Media, Inc.".

write_docx *Write an RTF Table or Figure to a DOCX File*

Description

The write_docx function writes an RTF encoding string to a .docx file by first writing to a temporary RTF file and then converting it to DOCX using LibreOffice.

Usage

```
write_docx(rtf, file)
```

Arguments

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

Details

This function requires LibreOffice to be installed on the system. The function uses the internal `rtf_convert_format()` function to perform the conversion from RTF to DOCX format.

Currently only Unix/Linux/macOS systems are supported.

Specification

- Write RTF encoding to a temporary RTF file.
- Convert RTF to DOCX using LibreOffice command-line tool via `rtf_convert_format()`.

`write_html` *Write an RTF Table or Figure to an HTML File*

Description

The write_html function writes an RTF encoding string to an .html file by first writing to a temporary RTF file and then converting it to HTML using LibreOffice.

Usage

```
write_html(rtf, file)
```

Arguments

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

Details

This function requires LibreOffice to be installed on the system. The function uses the internal `rtf_convert_format()` function to perform the conversion from RTF to HTML format.

Currently only Unix/Linux/macOS systems are supported.

Specification

- Write RTF encoding to a temporary RTF file.
- Convert RTF to HTML using LibreOffice command-line tool via `rtf_convert_format()`.

`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

```
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.

Index