

Package ‘r2rtf’

December 4, 2020

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

Version 0.2.0

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

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports grDevices

Suggests stringi, knitr, rmarkdown, testthat, emmeans, dplyr, tidyr, devtools, covr, ggplot2, haven

URL <https://merck.github.io/r2rtf/>

Config/testthat/edition 3

NeedsCompilation no

Author Yilong Zhang [aut, cre],
Siruo Wang [aut],
Simiao Ye [aut],
Madhusudhan Ginnaram [aut],
Merck Sharp & Dohme Corp [cph]

Maintainer Yilong Zhang <yilong.zhang@merck.com>

Repository CRAN

Date/Publication 2020-12-04 06:20:02 UTC

R topics documented:

adae	3
adsl	4
as_rtf_colheader	4

as_rtf_color	5
as_rtf_end	5
as_rtf_font	6
as_rtf_footnote	6
as_rtf_init	7
as_rtf_margin	7
as_rtf_new_page	8
as_rtf_page	8
as_rtf_pageby	9
as_rtf_paragraph	9
as_rtf_source	10
as_rtf_subline	10
as_rtf_table	11
as_rtf_title	11
border_type	12
cell_size	12
check_args	13
color_table	14
convert	14
font_format	15
font_type	15
footnote_source_space	16
HAMD17	16
inch_to_twip	17
justification	17
match_arg	18
obj_rtf_border	19
obj_rtf_text	21
rtf_body	22
rtf_colheader	27
rtf_encode	30
rtf_encode_figure	32
rtf_encode_list	33
rtf_encode_table	34
rtf_figure	35
rtf_footnote	36
rtf_group_by_enhance	39
rtf_page	39
rtf_pageby	41
rtf_page_footer	42
rtf_page_header	43
rtf_paragraph	45
rtf_read_png	46
rtf_source	47
rtf_subline	50
rtf_table_content	52
rtf_text	54
rtf_title	55

<i>adae</i>	3
set_margin	57
spacing	57
tbl_1	58
tbl_2	58
tbl_3	58
unicode_latex	59
utf8Tortf	59
write_rtf	60
write_rtf_para	60
Index	61

<i>adae</i>	<i>An Adverse Event Dataset</i>
-------------	---------------------------------

Description

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

Usage

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>

Source

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

adsl	<i>A Subject Level Demographic Dataset</i>
------	--

Description

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

Usage

```
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>

Source

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

as_rtf_colheader	<i>Create Column Header RTF Encode</i>
------------------	--

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	<i>Create RTF Color Encode</i>
--------------	--------------------------------

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	<i>End RTF Encode</i>
------------	-----------------------

Description

End RTF Encode

Usage

```
as_rtf_end()
```

Specification

- Add symbol right curly bracket at the end of code.

as_rtf_font	<i>Create RTF Font Encode</i>
-------------	-------------------------------

Description

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.

as_rtf_footnote	<i>Create Footnote RTF Encode</i>
-----------------	-----------------------------------

Description

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	<i>Create RTF Header Encode</i>
-------------	---------------------------------

Description

Create RTF Header Encode

Usage

```
as_rtf_init()
```

Specification

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

as_rtf_margin	<i>Create RTF Page Margin Encode</i>
---------------	--------------------------------------

Description

Create RTF Page Margin Encode

Usage

```
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

```
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

```
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	<i>RTF Table Page By Encoding</i>
---------------	-----------------------------------

Description

RTF Table Page By Encoding

Usage

```
as_rtf_pageby(tbl)
```

Arguments

tbl	A data frame.
-----	---------------

Specification

- Collect all attributes from tbl object.
- Define table attributes using `rtf_table_content()`.

as_rtf_paragraph	<i>Create Paragraph RTF Encode</i>
------------------	------------------------------------

Description

Create Paragraph RTF Encode

Usage

```
as_rtf_paragraph(text)
```

Arguments

text	A character string.
------	---------------------

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_source	<i>Create Data Source RTF Encode</i>
---------------	--------------------------------------

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	<i>Create Table Subline RTF Encode</i>
----------------	--

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	<i>Combine RTF Table Encoding</i>
--------------	-----------------------------------

Description

Combine RTF Table Encoding

Usage

```
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	<i>Create Table Title RTF Encode</i>
--------------	--------------------------------------

Description

Create Table Title RTF Encode

Usage

```
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	<i>RTF Border Type Dictionary</i>
-------------	-----------------------------------

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	<i>Calculate Cell Size in Twips</i>
-----------	-------------------------------------

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

<code>arg</code>	An argument to be checked.
<code>type</code>	A character vector of candidate argument type.
<code>length</code>	A numeric value of argument length or NULL
<code>dim</code>	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

```
## 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)
```

color_table	<i>RTF Text Color Dictionary</i>
-------------	----------------------------------

Description

RTF Text Color Dictionary

Usage

```
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	<i>Convert Symbol to ANSI and Unicode Encoding</i>
---------	--

Description

Convert Symbol to ANSI and Unicode Encoding

Usage

```
convert(
  text,
  load_stringi = suppressMessages(suppressWarnings(require("stringi")))
)
```

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

RTF Text Format Dictionary

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

RTF Text Font Dictionary

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.

footnote_source_space *Derive Space Adjustment*

Description

Derive Space Adjustment

Usage

```
footnote_source_space(tbl)
```

Arguments

tbl A data frame.

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.

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

```
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/>

Source

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

inch_to_twip	<i>Convert Inches to Twips</i>
--------------	--------------------------------

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	<i>RTF Text Justification Dictionary</i>
---------------	--

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

<code>arg</code>	a character vector (of length one unless <code>several.ok</code> is TRUE) or NULL.
<code>choices</code>	a character vector of candidate values
<code>several.ok</code>	logical specifying if <code>arg</code> 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

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

```
## 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)
```

obj_rtf_border	<i>Create an RTF Table Border Object</i>
----------------	--

Description

Create an RTF Table Border Object

Usage

```
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_nrow = NULL
)
```

Arguments

tbl	A data frame.
border_left	Left border type. Default is "single". 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 <code>r2rtf:::border_type()\$name</code> .
border_right	Right border type. Default is "single". 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 <code>r2rtf:::border_type()\$name</code> .
border_top	Top border type. Default is NULL. 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 <code>r2rtf:::border_type()\$name</code> .
border_bottom	Bottom border type. Default is "double" indicating double line bottom border. 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 <code>r2rtf:::border_type()\$name</code> .

<code>border_first</code>	First top border type of the whole table. Default is "double" indicating double line bottom border. All possible input can be found in <code>r2rtf:::border_type()\$name</code> .
<code>border_last</code>	Last bottom border type of the whole table. Default is "double" indicating double line bottom border. All possible input can be found in <code>r2rtf:::border_type()\$name</code> .
<code>border_color_left</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>border_color_right</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>border_color_top</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>border_color_bottom</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>border_color_first</code>	First top border color type of the whole table. Default is NULL for black. All possible input can be found in <code>grDevices::colors()</code> .
<code>border_color_last</code>	Last bottom border color type of the whole table. Default is NULL for black. All possible input can be found in <code>grDevices::colors()</code> .
<code>border_width</code>	Border width in twips. Default is 15 for 0.0104 inch.
<code>cell_height</code>	Cell height in inches. Default is 0.15 for 0.15 inch.
<code>cell_justification</code>	Justification type for cell. Default is "c" for center justification. All possible input can be found in <code>r2rtf:::justification()\$type</code> .
<code>cell_nrow</code>	Number of rows required in each cell. Default is NULL.

Value

the same `tbl` with additional border attributes

Specification

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

obj_rtf_text

Create an RTF Text Object

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 <code>r2rtf::font_type()\$type</code> .
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 <code>r2rtf::font_format()\$type</code> .
text_font_size	Text font size. Default is 9. 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 <code>grDevices::colors()</code> .

<code>text_background_color</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>text_justification</code>	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. <code>c("c","l","r")</code> . All possible input can be found in <code>r2rtf::justification()</code> \$type.
<code>text_indent_first</code>	A value of first indent.
<code>text_indent_left</code>	A value of left indent.
<code>text_indent_right</code>	A value of right indent.
<code>text_space</code>	A value of text space.
<code>text_space_before</code>	Line space before text in twips. Default is 15 for 0.0104 inch.
<code>text_space_after</code>	Line space after text in twips. Default is 15 for 0.0104 inch.
<code>text_new_page</code>	A logical value to control whether display text in new page. Default is FALSE.
<code>text_hyphenation</code>	A logical value to control whether display text linked with hyphenation. Default is TRUE.
<code>text_convert</code>	A logical value to convert special characters. Default is TRUE.

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.

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 = "",
  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_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,
  group_by = NULL,
  page_by = NULL,
  new_page = FALSE,
  pageby_header = TRUE,
  last_row = TRUE
)

```

Arguments

<code>tbl</code>	A data frame.
<code>col_rel_width</code>	Column relative width in a vector e.g. <code>c(2,1,1)</code> refers to 2:1:1. Default is <code>NULL</code> for equal column width.
<code>as_colheader</code>	A boolean value to indicate whether to add default column header to the table. Default is <code>TRUE</code> to use data frame column names as column header.

<code>border_left</code>	Left border type. Default is "single". To vary left border by column, use character vector with length of vector equal to number of columns displayed e.g. <code>c("single","single","single")</code> . All possible input can be found in <code>r2rtf:::border_type()\$name</code> .
<code>border_right</code>	Right border type. Default is "single". To vary right border by column, use character vector with length of vector equal to number of columns displayed e.g. <code>c("single","single","single")</code> . All possible input can be found in <code>r2rtf:::border_type()\$name</code> .
<code>border_top</code>	Top border type. Default is NULL. To vary top border by column, use character vector with length of vector equal to number of columns displayed e.g. <code>c("single","single","single")</code> . 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 <code>r2rtf:::border_type()\$name</code> .
<code>border_bottom</code>	Bottom border type. Default is "double" indicating double line bottom border. To vary bottom border by column, use character vector with length of vector equal to number of columns displayed e.g. <code>c("single","single","single")</code> . All possible input can be found in <code>r2rtf:::border_type()\$name</code> .
<code>border_first</code>	First top border type of the whole table. Default is "double" indicating double line bottom border. All possible input can be found in <code>r2rtf:::border_type()\$name</code> .
<code>border_last</code>	Last bottom border type of the whole table. Default is "double" indicating double line bottom border. All possible input can be found in <code>r2rtf:::border_type()\$name</code> .
<code>border_color_left</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices:::colors()</code> .
<code>border_color_right</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices:::colors()</code> .
<code>border_color_top</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices:::colors()</code> .
<code>border_color_bottom</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices:::colors()</code> .
<code>border_color_first</code>	First top border color type of the whole table. Default is NULL for black. All possible input can be found in <code>grDevices:::colors()</code> .
<code>border_color_last</code>	Last bottom border color type of the whole table. Default is NULL for black. All possible input can be found in <code>grDevices:::colors()</code> .
<code>border_width</code>	Border width in twips. Default is 15 for 0.0104 inch.
<code>cell_height</code>	Cell height in inches. Default is 0.15 for 0.15 inch.

<code>cell_justification</code>	Justification type for cell. Default is "c" for center justification. All possible input can be found in <code>r2rtf::justification()\$type</code> .
<code>cell_nrow</code>	Number of rows required in each cell. Default is NULL.
<code>text_font</code>	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. <code>c(1,2,3)</code> . All possible input can be found in <code>r2rtf::font_type()\$type</code> .
<code>text_format</code>	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. <code>c("i","u","ib")</code> . All possible input can be found in <code>r2rtf::font_format()\$type</code> .
<code>text_font_size</code>	Text font size. Default is 9. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. <code>c(9,20,40)</code> .
<code>text_color</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>text_background_color</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>text_justification</code>	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. <code>c("c","l","r")</code> . All possible input can be found in <code>r2rtf::justification()\$type</code> .
<code>text_indent_first</code>	A value of first indent.
<code>text_indent_left</code>	A value of left indent.
<code>text_indent_right</code>	A value of right indent.
<code>text_space</code>	A value of text space.
<code>text_space_before</code>	Line space before text in twips. Default is 15 for 0.0104 inch.
<code>text_space_after</code>	Line space after text in twips. Default is 15 for 0.0104 inch.
<code>text_convert</code>	A logical value to convert special characters. Default is TRUE.
<code>group_by</code>	A character vector of variable names in tbl.
<code>page_by</code>	Column names in a character vector to group by table in sections. Default is NULL.
<code>new_page</code>	A boolean value to indicate whether to separate grouped table into pages by sections. Default is FALSE.
<code>pageby_header</code>	A boolean value to display pageby header at the beginning of each page.
<code>last_row</code>	A boolean value to indicate whether the table contains the last row of the final table. Default is TRUE.

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(tbl_1)
tbl_1 %>%
  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()
```

<code>rtf_colheader</code>	<i>Add Column Header Attributes to Table</i>
----------------------------	--

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_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

<code>tbl</code>	A data frame.
<code>colheader</code>	A character string that uses " " to separate column names. Default is NULL for a blank column header.
<code>col_rel_width</code>	A Column relative width in a vector e.g. <code>c(2,1,1)</code> refers to 2:1:1. Default is NULL for equal column width.
<code>border_left</code>	Left border type. Default is "single". To vary left border by column, use character vector with length of vector equal to number of columns displayed e.g. <code>c("single","single","single")</code> . All possible input can be found in <code>r2rtf:::border_type()\$name</code> .
<code>border_right</code>	Right border type. Default is "single". To vary right border by column, use character vector with length of vector equal to number of columns displayed e.g. <code>c("single","single","single")</code> . All possible input can be found in <code>r2rtf:::border_type()\$name</code> .
<code>border_top</code>	Top border type. Default is NULL. To vary top border by column, use character vector with length of vector equal to number of columns displayed e.g. <code>c("single","single","single")</code> . 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 <code>r2rtf:::border_type()\$name</code> .
<code>border_bottom</code>	Bottom border type. Default is "double" indicating double line bottom border. To vary bottom border by column, use character vector with length of vector equal to number of columns displayed e.g. <code>c("single","single","single")</code> . All possible input can be found in <code>r2rtf:::border_type()\$name</code> .
<code>border_color_left</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>border_color_right</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>border_color_top</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .

<code>border_color_bottom</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>border_width</code>	Border width in twips. Default is 15 for 0.0104 inch.
<code>cell_height</code>	Cell height in inches. Default is 0.15 for 0.15 inch.
<code>cell_justification</code>	Justification type for cell. Default is "c" for center justification. All possible input can be found in <code>r2rtf:::justification()\$type</code> .
<code>cell_nrow</code>	Number of rows required in each cell. Default is NULL.
<code>text_font</code>	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. <code>c(1,2,3)</code> . All possible input can be found in <code>r2rtf:::font_type()\$type</code> .
<code>text_format</code>	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. <code>c("i","u","ib")</code> . All possible input can be found in <code>r2rtf:::font_format()\$type</code> .
<code>text_font_size</code>	Text font size. Default is 9. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. <code>c(9,20,40)</code> .
<code>text_color</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>text_background_color</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>text_justification</code>	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. <code>c("c","l","r")</code> . All possible input can be found in <code>r2rtf:::justification()\$type</code> .
<code>text_indent_first</code>	A value of first indent.
<code>text_indent_left</code>	A value of left indent.
<code>text_indent_right</code>	A value of right indent.
<code>text_space</code>	A value of text space.
<code>text_space_before</code>	Line space before text in twips. Default is 15 for 0.0104 inch.
<code>text_space_after</code>	Line space after text in twips. Default is 15 for 0.0104 inch.
<code>text_convert</code>	A logical value to convert special characters. Default is TRUE.

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(tbl_1)
tbl_1 %>%
  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"
)
```

Arguments

<code>tbl</code>	A data frame for table or a list of binary string for figure.
<code>doc_type</code>	The <code>doc_type</code> of input, default is table.
<code>page_title</code>	A character of title displaying location. Default is "all" for all pages. Possible values are "first", "last" and "all".

page_footnote	A character of title displaying location. Default is "last" for all pages. Possible values are "first", "last" and "all".
page_source	A character of title displaying location. Default is "last" for all pages. Possible values are "first", "last" and "all".

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_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(tbl_1)
data(tbl_2)
```

```

data(tbl_3)
## convert tbl_2 to the table body. Add a table column header to table body.
t2 <- tbl_2 %>%
  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_figure *Render Figure to RTF Encoding*

Description

Render Figure to RTF Encoding

Usage

```

rtf_encode_figure(
  tbl,
  page_title = "all",
  page_footnote = "last",
  page_source = "last"
)

```

Arguments

tbl	A data frame for table or a list of binary string for figure.
page_title	A character of title displaying location. Default is "all" for all pages. Possible values are "first", "last" and "all".
page_footnote	A character of title displaying location. Default is "last" for all pages. Possible values are "first", "last" and "all".
page_source	A character of title displaying location. Default is "last" for all pages. Possible values are "first", "last" and "all".

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.

rtf_encode_list

Render List to RTF Encoding

Description

Render List to RTF Encoding

Usage

```
rtf_encode_list(
    tbl,
    page_title = "all",
    page_footnote = "last",
    page_source = "last"
)
```

Arguments

tbl	A data frame for table or a list of binary string for figure.
page_title	A character of title displaying location. Default is "all" for all pages. Possible values are "first", "last" and "all".
page_footnote	A character of title displaying location. Default is "last" for all pages. Possible values are "first", "last" and "all".
page_source	A character of title displaying location. Default is "last" for all pages. Possible values are "first", "last" and "all".

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.

<code>rtf_encode_table</code>	<i>Render Table to RTF Encoding</i>
-------------------------------	-------------------------------------

Description

Render Table to RTF Encoding

Usage

```
rtf_encode_table(
  tbl,
  page_title = "all",
  page_footnote = "last",
  page_source = "last"
)
```

Arguments

<code>tbl</code>	A data frame for table or a list of binary string for figure.
<code>page_title</code>	A character of title displaying location. Default is "all" for all pages. Possible values are "first", "last" and "all".
<code>page_footnote</code>	A character of title displaying location. Default is "last" for all pages. Possible values are "first", "last" and "all".
<code>page_source</code>	A character of title displaying location. Default is "last" for all pages. Possible values are "first", "last" and "all".

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

```
rtf_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

```
## 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() %>%
attributes()

## End(Not run)
```

rtf_footnote	<i>Add Footnote Attributes to Table</i>
--------------	---

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_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_space = 1,
  text_space_before = 15,
  text_space_after = 15,
  text_convert = TRUE,
  as_table = TRUE
)
```

Arguments

<code>tbl</code>	A data frame.
<code>footnote</code>	A vector of character for footnote text.
<code>border_left</code>	Left border type. Default is "single". To vary left border by column, use character vector with length of vector equal to number of columns displayed e.g. <code>c("single","single","single")</code> . All possible input can be found in <code>r2rtf:::border_type()\$name</code> .
<code>border_right</code>	Right border type. Default is "single". To vary right border by column, use character vector with length of vector equal to number of columns displayed e.g. <code>c("single","single","single")</code> . All possible input can be found in <code>r2rtf:::border_type()\$name</code> .
<code>border_top</code>	Top border type. Default is NULL. To vary top border by column, use character vector with length of vector equal to number of columns displayed e.g. <code>c("single","single","single")</code> . 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 <code>r2rtf:::border_type()\$name</code> .
<code>border_bottom</code>	Bottom border type. Default is "double" indicating double line bottom border. To vary bottom border by column, use character vector with length of vector equal to number of columns displayed e.g. <code>c("single","single","single")</code> . All possible input can be found in <code>r2rtf:::border_type()\$name</code> .
<code>border_color_left</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>border_color_right</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>border_color_top</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>border_color_bottom</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>border_width</code>	Border width in twips. Default is 15 for 0.0104 inch.
<code>cell_height</code>	Cell height in inches. Default is 0.15 for 0.15 inch.
<code>cell_justification</code>	Justification type for cell. Default is "c" for center justification. All possible input can be found in <code>r2rtf:::justification()\$type</code> .
<code>cell_nrow</code>	Number of rows required in each cell. Default is NULL.
<code>text_font</code>	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. <code>c(1,2,3)</code> . All possible input can be found in <code>r2rtf:::font_type()\$type</code> .

<code>text_format</code>	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. <code>c("i","u","ib")</code> . All possible input can be found in <code>r2rtf:::font_format()\$type</code> .
<code>text_font_size</code>	Text font size. Default is 9. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. <code>c(9,20,40)</code> .
<code>text_color</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>text_background_color</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>text_justification</code>	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. <code>c("c","l","r")</code> . All possible input can be found in <code>r2rtf:::justification()\$type</code> .
<code>text_indent_first</code>	A value of first indent.
<code>text_indent_left</code>	A value of left indent.
<code>text_indent_right</code>	A value of right indent.
<code>text_space</code>	A value of text space.
<code>text_space_before</code>	Line space before text in twips. Default is 15 for 0.0104 inch.
<code>text_space_after</code>	Line space after text in twips. Default is 15 for 0.0104 inch.
<code>text_convert</code>	A logical value to convert special characters. Default is TRUE.
<code>as_table</code>	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(tbl_1)
```

```
tbl_1 %>%
  rtf_footnote("\\dagger Based on an ANCOVA model.") %>%
  attr("rtf_footnote")
```

rtf_group_by_enhance *Remove Duplicate Records*

Description

Remove Duplicate Records

Usage

```
rtf_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_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, 28),
  border_first = "double",
  border_last = "double",
  border_color_first = NULL,
  border_color_last = NULL,
  col_width = width - ifelse(orientation == "portrait", 2.25, 2.5)
)
```

Arguments

<code>tbl</code>	A data frame.
<code>orientation</code>	Orientation in 'portrait' or 'landscape'.
<code>width</code>	A numeric value of page width in inches. Default is 8.5 inch in portrait orientation or 11.0 inch in landscape orientation.
<code>height</code>	A numeric value of page width in inches. Default is 11.0 inch in portrait orientation or 8.5 inch in landscape orientation.
<code>margin</code>	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 <code>r2rtf:::set_margin("wma", orientation)</code>
<code>nrow</code>	Number of rows in each page. Default is 42 rows if portrait orientation and 26 rows in landscape orientation.
<code>border_first</code>	First top border type of the whole table. Default is "double" indicating double line bottom border. All possible input can be found in <code>r2rtf:::border_type()\$name</code> .
<code>border_last</code>	Last bottom border type of the whole table. Default is "double" indicating double line bottom border. All possible input can be found in <code>r2rtf:::border_type()\$name</code> .
<code>border_color_first</code>	First top border color type of the whole table. Default is NULL for black. All possible input can be found in <code>grDevices::colors()</code> .
<code>border_color_last</code>	Last bottom border color type of the whole table. Default is NULL for black. All possible input can be found in <code>grDevices::colors()</code> .
<code>col_width</code>	A numeric value of total column width in inch. Default is <code>width - ifelse(orientation == "portrait", 2, 2.5)</code>

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(tbl_1)
tbl_1 %>%
  rtf_page() %>%
  attr("page")
```

rtf_pageby

Add Table Body Pageby Attributes to Table

Description

Add Table Body Pageby Attributes to Table

Usage

```
rtf_pageby(tbl, page_by, new_page, pageby_header)
```

Arguments

tbl	A data frame.
page_by	Column names in a character vector to group by table in sections. Default is NULL.
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.

rtf_page_footer	<i>Add RTF Page Footer Information</i>
-----------------	--

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 <code>r2rtf::font_type()\$type</code> .
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 <code>r2rtf::font_format()\$type</code> .
text_font_size	Text font size. Default is 9. 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 <code>grDevices::colors()</code> .

`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 first indent.

`text_indent_left` A value of left indent.

`text_indent_right` A value of right indent.

`text_space` A value of text space.

`text_space_before` Line space before text in twips. Default is 15 for 0.0104 inch.

`text_space_after` Line space after text in twips. Default is 15 for 0.0104 inch.

`text_convert` A logical value to convert special characters. Default is TRUE.

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

<code>tbl</code>	A data frame.
<code>text</code>	A character string.
<code>text_font</code>	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. <code>c(1,2,3)</code> . All possible input can be found in <code>r2rtf::font_type()\$type</code> .
<code>text_format</code>	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. <code>c("i","u","ib")</code> . All possible input can be found in <code>r2rtf::font_format()\$type</code> .
<code>text_font_size</code>	Text font size. Default is 9. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. <code>c(9,20,40)</code> .
<code>text_color</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>text_background_color</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>text_justification</code>	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. <code>c("c","l","r")</code> . All possible input can be found in <code>r2rtf::justification()\$type</code> .
<code>text_indent_first</code>	A value of first indent.
<code>text_indent_left</code>	A value of left indent.
<code>text_indent_right</code>	A value of right indent.
<code>text_space</code>	A value of text space.
<code>text_space_before</code>	Line space before text in twips. Default is 15 for 0.0104 inch.
<code>text_space_after</code>	Line space after text in twips. Default is 15 for 0.0104 inch.
<code>text_convert</code>	A logical value to convert special characters. Default is TRUE.

rtf_paragraph	<i>Paragraph to RTF Encode</i>
---------------	--------------------------------

Description

Paragraph to RTF Encode

Usage

```
rtf_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  
)
```

Arguments

text	rtf text obtained using <code>rtf_text()</code> 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.

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

```
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

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

```
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_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,
as_table = FALSE
)

```

Arguments

<code>tbl</code>	A data frame.
<code>source</code>	A character string.
<code>border_left</code>	Left border type. Default is "single". To vary left border by column, use character vector with length of vector equal to number of columns displayed e.g. <code>c("single","single","single")</code> . All possible input can be found in <code>r2rtf:::border_type()\$name</code> .
<code>border_right</code>	Right border type. Default is "single". To vary right border by column, use character vector with length of vector equal to number of columns displayed e.g. <code>c("single","single","single")</code> . All possible input can be found in <code>r2rtf:::border_type()\$name</code> .
<code>border_top</code>	Top border type. Default is NULL. To vary top border by column, use character vector with length of vector equal to number of columns displayed e.g. <code>c("single","single","single")</code> . 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 <code>r2rtf:::border_type()\$name</code> .
<code>border_bottom</code>	Bottom border type. Default is "double" indicating double line bottom border. To vary bottom border by column, use character vector with length of vector equal to number of columns displayed e.g. <code>c("single","single","single")</code> . All possible input can be found in <code>r2rtf:::border_type()\$name</code> .
<code>border_color_left</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>border_color_right</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>border_color_top</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>border_color_bottom</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>border_width</code>	Border width in twips. Default is 15 for 0.0104 inch.
<code>cell_height</code>	Cell height in inches. Default is 0.15 for 0.15 inch.

<code>cell_justification</code>	Justification type for cell. Default is "c" for center justification. All possible input can be found in <code>r2rtf:::justification()\$type</code> .
<code>cell_nrow</code>	Number of rows required in each cell. Default is NULL.
<code>text_font</code>	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. <code>c(1,2,3)</code> . All possible input can be found in <code>r2rtf:::font_type()\$type</code> .
<code>text_format</code>	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. <code>c("i","u","ib")</code> . All possible input can be found in <code>r2rtf:::font_format()\$type</code> .
<code>text_font_size</code>	Text font size. Default is 9. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. <code>c(9,20,40)</code> .
<code>text_color</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>text_background_color</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>text_justification</code>	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. <code>c("c","l","r")</code> . All possible input can be found in <code>r2rtf:::justification()\$type</code> .
<code>text_indent_first</code>	A value of first indent.
<code>text_indent_left</code>	A value of left indent.
<code>text_indent_right</code>	A value of right indent.
<code>text_space</code>	A value of text space.
<code>text_space_before</code>	Line space before text in twips. Default is 15 for 0.0104 inch.
<code>text_space_after</code>	Line space after text in twips. Default is 15 for 0.0104 inch.
<code>text_convert</code>	A logical value to convert special characters. Default is TRUE.
<code>as_table</code>	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(tbl_1)
tbl_1 %>%
  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_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_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 <code>r2rtf:::font_type()\$type</code> .

<code>text_format</code>	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. <code>c("i","u","ib")</code> . All possible input can be found in <code>r2rtf::font_format()\$type</code> .
<code>text_font_size</code>	Text font size. Default is 9. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. <code>c(9,20,40)</code> .
<code>text_color</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>text_background_color</code>	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. <code>c("white","red","blue")</code> . All possible input can be found in <code>grDevices::colors()</code> .
<code>text_justification</code>	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. <code>c("c","l","r")</code> . All possible input can be found in <code>r2rtf::justification()\$type</code> .
<code>text_indent_first</code>	A value of first indent.
<code>text_indent_left</code>	A value of left indent.
<code>text_indent_right</code>	A value of right indent.
<code>text_space</code>	A value of text space.
<code>text_space_before</code>	Line space before text in twips. Default is 15 for 0.0104 inch.
<code>text_space_after</code>	Line space after text in twips. Default is 15 for 0.0104 inch.
<code>text_convert</code>	A logical value to convert special characters. Default is TRUE.

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_table_content *Create RTF Table Body Encode*

Description

Create RTF Table Body Encode

Usage

```
rtf_table_content(
  tbl,
  col_total_width = attr(tbl, "page")$col_width,
  use_border_bottom = FALSE,
  border_left = attr(tbl, "border_left"),
  border_right = attr(tbl, "border_right"),
  border_top = attr(tbl, "border_top"),
  border_bottom = attr(tbl, "border_bottom"),
  border_color_left = attr(tbl, "border_color_left"),
  border_color_right = attr(tbl, "border_color_right"),
  border_color_top = attr(tbl, "border_color_top"),
  border_color_bottom = attr(tbl, "border_color_bottom"),
  border_width = attr(tbl, "border_width"),
  col_rel_width = attr(tbl, "col_rel_width"),
  cell_height = attr(tbl, "cell_height"),
  cell_justification = attr(tbl, "cell_justification"),
  text_font = attr(tbl, "text_font"),
  text_format = attr(tbl, "text_format"),
  text_color = attr(tbl, "text_color"),
  text_background_color = attr(tbl, "text_background_color"),
  text_justification = attr(tbl, "text_justification"),
  text_font_size = attr(tbl, "text_font_size"),
  text_space_before = attr(tbl, "text_space_before"),
  text_space_after = attr(tbl, "text_space_after"),
  text_convert = attr(tbl, "text_convert")
)
```

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.
border_left	Left border type. Default is the corresponding attribute from tbl.

<code>border_right</code>	Right border type. Default is the corresponding attribute from <code>tbl</code> .
<code>border_top</code>	Top border type. Default is the corresponding attribute from <code>tbl</code> .
<code>border_bottom</code>	Bottom border type. Default is the corresponding attribute from <code>tbl</code> .
<code>border_color_left</code>	Left border color. Default is the corresponding attribute from <code>tbl</code> .
<code>border_color_right</code>	Right border color. Default is the corresponding attribute from <code>tbl</code> .
<code>border_color_top</code>	Top border color. Default is the corresponding attribute from <code>tbl</code> .
<code>border_color_bottom</code>	Bottom border color. Default is the corresponding attribute from <code>tbl</code> .
<code>border_width</code>	Border width in twips. Default is the corresponding attribute from <code>tbl</code> .
<code>col_rel_width</code>	Column relative width in a vector eg. <code>c(2,1,1)</code> refers to 2:1:1
<code>cell_height</code>	Height for cell in twips. Default is the corresponding attribute from <code>tbl</code> .
<code>cell_justification</code>	Justification for cell. Default is the corresponding attribute from <code>tbl</code> .
<code>text_font</code>	Text font type. Default is the corresponding attribute from <code>tbl</code> .
<code>text_format</code>	Text format. Default is the corresponding attribute from <code>tbl</code> .
<code>text_color</code>	Text color. Default is the corresponding attribute from <code>tbl</code> .
<code>text_background_color</code>	Text background color. Default is the corresponding attribute from <code>tbl</code> .
<code>text_justification</code>	Justification for text. Default is the corresponding attribute from <code>tbl</code> .
<code>text_font_size</code>	Text font size. Default is the corresponding attribute from <code>tbl</code> .
<code>text_space_before</code>	Line space before text. Default is the corresponding attribute from <code>tbl</code> .
<code>text_space_after</code>	Line space after text. Default is the corresponding attribute from <code>tbl</code> .
<code>text_convert</code>	A logical value to convert special characters. Default is the corresponding attribute from <code>tbl</code> .

Specification

- Define table begin and end in RTF syntax.
- Define cell justification using `'justification()'`, then convert 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.

`rtf_text`*Text to RTF Encode*

Description

Text to RTF Encode

Usage

```
rtf_text(  
    text,  
    font = 1,  
    font_size = 12,  
    format = NULL,  
    color = NULL,  
    background_color = NULL,  
    text_convert = TRUE  
)
```

Arguments

<code>text</code>	Plain text.
<code>font</code>	Text font type.
<code>font_size</code>	Text font size.
<code>format</code>	Text format.
<code>color</code>	Text color.
<code>background_color</code>	Text background color.
<code>text_convert</code>	A logical value to convert special characters. Default is TRUE.

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	<i>Add Title Attributes to Table</i>
-----------	--------------------------------------

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_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 <code>r2rtf:::font_type()\$type</code> .
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 <code>r2rtf:::font_format()\$type</code> .
text_font_size	Text font size. Default is 9. 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 <code>grDevices:::colors()</code> .

`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 first indent.

`text_indent_left`
A value of left indent.

`text_indent_right`
A value of right indent.

`text_space`
A value of text space.

`text_space_before`
Line space before text in twips. Default is 15 for 0.0104 inch.

`text_space_after`
Line space after text in twips. Default is 15 for 0.0104 inch.

`text_convert`
A logical value to convert special characters. Default is TRUE.

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(tbl_1)
tbl_1 %>%
  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.

tbl_1	<i>Within Group Results from an ANCOVA Model</i>
-------	--

Description

A dataset containing within group results from an ANCOVA model.

Usage

tbl_1

Format

A data frame with 2 rows and 8 variables.

tbl_2	<i>Between Group Results from an ANCOVA Model</i>
-------	---

Description

A dataset containing between group results from an ANCOVA model.

Usage

tbl_2

Format

A data frame with 1 row and 3 variables.

tbl_3	<i>Root Mean Square Error from an ANCOVA model</i>
-------	--

Description

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

Usage

tbl_3

Format

A data frame with 1 row and 1 variable.

unicode_latex	<i>Dictionary of Unicode and Latex Code</i>
---------------	---

Description

A dataset containing the mapping between unicode and latex code.

Usage

```
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

<http://milde.users.sourceforge.net/LUCR/Math/data/unimathsymbols.txt>

utf8Tortf	<i>Convert a UTF-8 Encoded Character String to a RTF Encoded String</i>
-----------	---

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

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

write_rtf	<i>Write an RTF Table or Figure to an RTF File</i>
-----------	--

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.

write_rtf_para	<i>Write a Paragraph to an RTF File</i>
----------------	---

Description

Write a Paragraph to an RTF File

Usage

```
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

- adae, 3
- adsl, 4
- HAMD17, 16
- tbl_1, 58
- tbl_2, 58
- tbl_3, 58
- unicode_latex, 59

adae, 3

adsl, 4

as_rtf_colheader, 4

as_rtf_color, 5

as_rtf_end, 5

as_rtf_font, 6

as_rtf_footnote, 6

as_rtf_init, 7

as_rtf_margin, 7

as_rtf_new_page, 8

as_rtf_page, 8

as_rtf_pageby, 9

as_rtf_paragraph, 9

as_rtf_source, 10

as_rtf_subline, 10

as_rtf_table, 11

as_rtf_title, 11

border_type, 12

cell_size, 12

check_args, 13

color_table, 14

convert, 14

font_format, 15

font_type, 15

footnote_source_space, 16

HAMD17, 16

inch_to_twip, 17

justification, 17

match_arg, 18

obj_rtf_border, 19

obj_rtf_text, 21

rtf_body, 22

rtf_colheader, 27

rtf_encode, 30

rtf_encode_figure, 32

rtf_encode_list, 33

rtf_encode_table, 34

rtf_figure, 35

rtf_footnote, 36

rtf_group_by_enhance, 39

rtf_page, 39

rtf_page_footer, 42

rtf_page_header, 43

rtf_pageby, 41

rtf_paragraph, 45

rtf_read_png, 46

rtf_source, 47

rtf_subline, 50

rtf_table_content, 52

rtf_text, 54

rtf_title, 55

set_margin, 57

spacing, 57

tbl_1, 58

tbl_2, 58

tbl_3, 58

unicode_latex, 59

utf8Tortf, 59

write_rtf, 60

write_rtf_para, 60