Package 'grobblR'

April 1, 2020

Title The package intent on creating reports within the R environment as easy and intuitive as possible.

Version 1.0

Description Making use of graphical grid objects (grobs), grobblR intuitively combines objects of various types into a single, cohesive grob which can be converted into reports.

Depends R (>= 3.3.1)

Imports ggplot2,
graphics,
grDevices,
grid,
gridExtra,
png,
R6,
scales,
stringr

License 1.0.153

Language en-US

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

`add_column_headings` *Add column headings onto a matrix. Intended to be used with the group_elements aesthetic within* `grob_col`.

Description

Add column headings onto a matrix. Intended to be used with the group_elements aesthetic within `grob_col`.

Usage

```
add_column_headings(mat, headings = list(), heading_cols = list())
```

Arguments

`mat`	The matrix the column headings will be added onto.
`headings`	The headings to be added onto the initial matrix, in a list with each heading a separate element. The list must have the same amount of elements as the `heading_cols` parameter.
`heading_cols`	Which column positions of the initial matrix the `headings` will be placed above, in a list with each heading's column positions a separate element. The list must have the same amount of elements as the `headings` parameter.

Value

The initial matrix with column headings inserted into the first row.

`add_row_headings` *Add row headings onto a matrix. Intended to be used with the* `group_elements` *aesthetic within* `grob_col` *and* `ga_list`.

Description

Add row headings onto a matrix. Intended to be used with the `group_elements` aesthetic within `grob_col` and `ga_list`.

Usage

```
add_row_headings(mat, headings = list(), heading_rows = list())
```

Arguments

`mat`	The matrix the column headings will be added onto.
`headings`	The headings to be added onto the initial matrix, in a list with each heading a separate element. The list must have the same amount of elements as the `heading_rows` parameter.
`heading_rows`	Which row positions of the initial matrix the `headings` will be placed to the left of, in a list with each heading's row positions a separate element. The list must have the same amount of elements as the `headings` parameter.

Value

The initial matrix with row headings inserted into the first column.

`aes_matrix` *Create a matrix based off the dimensions of a data.frame/matrix and a single value to make up its cells. Designed to be used as an aesthetic matrix within* `ga_list`.

Description

Create a matrix based off the dimensions of a data.frame/matrix and a single value to make up its cells. Designed to be used as an aesthetic matrix within `ga_list`.

Usage

```
aes_matrix(df, value, column_names = FALSE)
```

Arguments

`df`	A data.frame/matrix the resulting matrix will get its dimensions from.
`value`	The single value that will make up the cells of the resulting matrix.
`column_names`	A TRUE/FALSE value indicating if the resulting aesthetic matrix is intended to be used for the column names.

Value

A matrix based on the dimensions of `df` and `value`.

Examples

```
df = data.frame(x = c(1, 2, 3), y = c(4, 5, 6))
aes_matrix(df, 'white')
```

`alter_cells` *Alter the values at specific row-column combinations of a matrix.*

Description

Alter the values at specific row-column combinations of a matrix.

Usage

```
alter_cells(mat, value, rows = NULL, columns = NULL)
```

Arguments

`mat`	The matrix the user wishes to alter cells of.
`value`	The single value that will replace specific cells of the matrix.
`rows`	The rows the user wishes to alter. See `alter_rows` for information on special inputs.
`columns`	The columns the user wishes to alter. See `alter_columns` for information on special inputs.

Value

A matrix with the desired cells altered.

Examples

```
df = data.frame(x = c(1, 2, 3), y = c(4, 5, 6))
mat = aes_matrix(df, 'white')
alter_cells(mat = mat, value = 'red', rows = c(1,2), columns = 1)
```

`alter_columns` *Alter the values at specific columns of a matrix.*

Description

Alter the values at specific columns of a matrix.

Usage

```
alter_columns(mat, value, columns = NULL)
```

Arguments

`mat`	The matrix the user wishes to alter cells of.
`value`	The single value that will replace specific cells of the matrix.
`columns`	The columns the user wishes to alter. Can be numeric column positions, or the user can input: • 'odd' - Only alter odd numbered columns. • 'even' - Only alter even numbered columns. • 'first' - Only alter the first column. • 'last' - Only alter the last column. Also, the user can provide the column name of the column they wish to alter.

Value

A matrix with the desired columns altered.

Examples

```
df = data.frame(x = c(1, 2, 3), y = c(4, 5, 6))
mat = aes_matrix(df, 'white')
alter_columns(mat = mat, value = 'red', columns = 1)
```

`alter_rows` *Alter the values at specific rows of a matrix.*

Description

Alter the values at specific rows of a matrix.

Usage

```
alter_rows(mat, value, rows = NULL)
```

Arguments

mat	The matrix the user wishes to alter cells of.
value	The single value that will replace specific cells of the matrix.
rows	The rows the user wishes to alter. Can be numeric row positions, or the user can also input: • 'odd' - Only alter odd numbered rows. • 'even' - Only alter even numbered rows. • 'first' - Only alter the first row. • 'last' - Only alter the last row. Also, the user can provide the row name of the column they wish to alter.

Value

A matrix with the desired rows altered.

Examples

```
df = data.frame(x = c(1, 2, 3), y = c(4, 5, 6))
mat = aes_matrix(df, 'white')
alter_rows(mat = mat, value = 'red', rows = c(1, 2))
```

---

column_names_to_row *Take a data.frame/matrix and insert its column names as the first row of the resulting matrix.*

Description

Take a data.frame/matrix and insert its column names as the first row of the resulting matrix.

Usage

```
column_names_to_row(df)
```

Arguments

df	The data.frame/matrix.

Value

A matrix of the initial data.frame/matrix with its column names as the first row.

`convert_to_grob` *Takes in an object, and converts it to a grob based on inputted aesthetics arguments.*

Description

Takes in an object, and converts it to a grob based on inputted aesthetics arguments.

Usage

```
convert_to_grob(x, height, width, units = c("mm", "cm", "inches"),
  aes_list = ga_list())
```

Arguments

`x`	The object which needs to be converted to a grob. Must be either: A data.frame/matrix, the file name of a .png image, a character string, a vector, a ggplot object, NA (for an empty grob), or already a grob.
`height`	The numeric height in mm of the desired grob.
`width`	The numeric width in mm of the desired grob.
`aes_list`	The list outputted by `ga_list` which contains elements to adjust aesthetics to the grob of `x`. Different type of grobs have different types of elements of this list which will affect its aesthetics.\ For character strings or matrices of dimensions n x p, the aesthetic elements can either be a single value which will be applied to the entire matrix, or a matrix of dimension n x p, which specifies how each element of the matrix will be adjusted. Note that column names and actual matrix elements are treated differently.\ Possible elements for character strings, matrices and images can be found in `ga_list`.

Value

A grob of x with aesthetics based on the aes_list parameter.

`ga_list` *Grob aesthetic list used to control aesthetics within 'grobblR'.*

Description

Grob aesthetic list used to control aesthetics within 'grobblR'.

Usage

```
ga_list(aspect_ratio_multiplier = NULL, background_color = NULL,
  background_alpha = NULL, border_color = NULL, border_sides = NULL,
  border_width = NULL, font_face = NULL, group_elements = NULL,
  text_color = NULL, text_align = NULL, text_v_align = NULL,
  text_just = NULL, text_v_just = NULL, text_cex = NULL,
  text_font = NULL, text_rot = NULL, round_rect_radius = NULL,
  column_widths = NULL, column_widths_p = NULL, padding_p = NULL,
  cell_font_face = NULL, cell_group_elements = NULL,
  cell_background_color = NULL, cell_background_alpha = NULL,
  cell_border_color = NULL, cell_border_sides = NULL,
  cell_border_width = NULL, cell_text_color = NULL,
  cell_text_align = NULL, cell_text_v_align = NULL,
  cell_text_just = NULL, cell_text_v_just = NULL,
  cell_text_cex = NULL, cell_text_font = NULL, cell_text_rot = NULL,
  cell_round_rect_radius = NULL, cell_column_widths = NULL,
  cell_column_widths_p = NULL, cell_padding_p = NULL,
  cell_color_gradient_columns = NULL,
  cell_color_gradient_binary = NULL, cell_color_binary_cut_off = NULL,
  cell_color_binary_high = NULL, cell_color_binary_low = NULL,
  cell_color_binary_equal = NULL, cell_color_gradient_max = NULL,
  cell_color_gradient_mid = NULL, cell_color_gradient_min = NULL,
  colname_font_face = NULL, colname_group_elements = NULL,
  colname_background_color = NULL, colname_background_alpha = NULL,
  colname_border_color = NULL, colname_border_sides = NULL,
  colname_border_width = NULL, colname_text_color = NULL,
  colname_text_align = NULL, colname_text_v_align = NULL,
  colname_text_just = NULL, colname_text_v_just = NULL,
  colname_text_cex = NULL, colname_text_font = NULL,
  colname_text_rot = NULL, colname_round_rect_radius = NULL,
  colname_column_widths = NULL, colname_column_widths_p = NULL,
  colname_padding_p = NULL, maintain_aspect_ratio = NULL,
  n_lines = NULL, str_sep = NULL)
```

Arguments

`aspect_ratio_multiplier`
: A numeric value which controls how much to increase/decrease the aspect ratio of images or ggplots.

`background_color`
: Controls the background color of the elements of the matrix. If the matrix has no rownames or colnames, the default is white. If the matrix has column names, the default is white-gray90 on every odd-even row. Used with matrices.

`background_alpha`
: Controls the background alpha/opacity of the elements of the matrix. Values are used in `grid::gpar()`. Default is 1.0. Used with matrices.

`border_color`
: Controls the color of the selected borders. Default is gray40. Used with matrices.

`border_sides` Controls the borders of the elements of the matrix. The input is a string with the possible words "top", "bottom", "left", "right" separated by commas. For example, "top, left, right" will put borders on the top, left and right side of the grid cell, but not the bottom. Default is "", or no borders. Used with matrices.

`border_width` Controls the line width density/thickness of the selected borders. Values are used in `grid::gpar()`. Default is 4. Used with matrices.

`font_face` Controls the font face of the elements of the matrix (i.e. bold, italic, etc.). Values are used in `grid::gpar()`. Default for table elements is normal, or 1. Default for column name elements is "bold", or 2. Used with matrices or character strings.

`group_elements` A TRUE/FALSE argument on whether like, adjacent matrix elements should be grouped together into a single element. Default is FALSE.

`text_color` Controls the text color of the elements of the matrix. Default for table elements and row names is black, and a gray-blue color for column names. Used with matrices or character strings.

`text_align` Controls where the text in each grid cell will be centered around, horizontally. A numeric value between 0 and 1, with 0 being all the way to the left of the grid cell, and 1 being all the way to the right of the grid cell. Default is 0.5. Can also input 'left', 'right' or 'center', which will also make edits to `text_just` to make the text completely left-justified, right-justified or centered, respectively. Used with matrices or character strings.

`text_v_align` Controls where the text in each grid cell will be centered around, vertically. A numeric value between 0 and 1, with 0 being all the way to the bottom of the grid cell, and 1 being all the way to the top of the grid cell. Default is 0.5. Can also input 'top', 'bottom' or 'center', which will also make edits to `text_v_just` to make the text completely top-justified, bottom-justified or centered, respectively. Used with matrices or character strings.

`text_just` Controls the horizontal justification of the text in the matrix. A numeric value between 0 and 1, with 0 being left justification and 1 being right justification. Default is 0.5, or center justification. Can also input 'left', 'right' or 'center', which will also make edits to `text_align` to make the text completely left-justified, right-justified or centered, respectively. Used with matrices or character strings.

`text_v_just` Controls the vertical justification of the text in the matrix. A numeric value between 0 and 1, with 0 being bottom justification and 1 being top justification. Default is 0.5, or center justification. Can also input 'top', 'bottom' or 'center', which will also make edits to `text_v_align` to make the text completely top-justified, bottom-justified or centered, respectively. Used with matrices or character strings.

`text_cex` Controls the size of the text within the matrix. Default is automatic text sizing based on the length of the elements within the matrix, the row heights and the column widths. Used with matrices or character strings.

`text_font` Controls the font family of the text within the matrix. Default is sans. Used with matrices or character strings.

`text_rot` Controls the rotation in degrees of the text within the matrix. Default is 0 degrees. Used with matrices or character strings. **Please be aware that the automatic text sizing will not react properly if the text is angled at anything other than 0 degrees.**

`round_rect_radius` Controls the radius of the corners of the rectangles matrix text is laid on top of. Used with matrices.

`column_widths` If automatic column widths are not desired, the user can provide a vector of widths for each column in the matrix in whatever units are specified in the grob-layout. Used with matrices.

`column_widths_p` If automatic column widths are not desired, the user can provide a vector of width proportions (ideally adding to 1) for each column in the matrix in whatever units are specified in the grob-layout. Used with matrices. Overridden by `column_widths` argument.

`padding_p` Controls the amount of proportional padding around each matrix cell. Used with matrices.

`cell_font_face` Controls the `font_face` of matrix cells. Overridden by the `font_face` parameter.

`cell_group_elements` Controls the `group_elements` of matrix cells. Overridden by the `group_elements` parameter.

`cell_background_color` Controls the `background_color` of matrix cells. Overridden by the `background_color` parameter.

`cell_background_alpha` Controls the `background_alpha` of matrix cells. Overridden by the `background_alpha` parameter.

`cell_border_color` Controls the `border_color` of matrix cells. Overridden by the `border_color` parameter.

`cell_border_sides` Controls the `border_sides` of matrix cells. Overridden by the `border_sides` parameter.

`cell_border_width` Controls the `border_width` of matrix cells. Overridden by the `border_width` parameter.

`cell_text_color` Controls the `text_color` of matrix cells. Overridden by the `text_color` parameter.

`cell_text_align` Controls the `text_align` of matrix cells. Overridden by the `text_align` parameter.

`cell_text_v_align` Controls the `text_v_align` of matrix cells. Overridden by the `text_v_align` parameter.

`cell_text_just` Controls the `text_just` of matrix cells. Overridden by the `text_just` parameter.

`cell_text_v_just`
Controls the `text_v_just` of matrix cells. Overridden by the `text_v_just` parameter.

`cell_text_cex` Controls the `text_cex` of matrix cells. Overridden by the `text_cex` parameter.

`cell_text_font` Controls the `text_font` of matrix cells. Overridden by the `text_font` parameter.

`cell_text_rot` Controls the `text_rot` of matrix cells. Overridden by the `text_rot` parameter.

`cell_round_rect_radius`
Controls the `round_rect_radius` of matrix cells. Overridden by the `round_rect_radius` parameter.

`cell_column_widths`
Controls the `column_widths` of matrix cells. Overridden by the `column_widths` parameter.

`cell_column_widths_p`
Controls the `column_widths_p` of matrix cells. Overridden by the `column_widths_p` parameter.

`cell_padding_p` Controls the `padding_p` of matrix cells. Overridden by the `padding_p` parameter.

`cell_color_gradient_columns`
Controls the columns which a color gradient scale will be applied to. Integer values denoting the column numbers. Can only be applied to columns with all numeric values. Used with matrices.

`cell_color_gradient_binary`
A TRUE/FALSE value which signifies if a binary color gradient should be applied to the `color_gradient_columns`. Used with matrices.

`cell_color_binary_cut_off`
A cut-off value which the binary color gradient will be applied to. Default is 0. Used with matrices.

`cell_color_binary_high`
The color of the binary color gradient if the numeric element is greater than the `color_binary_cut_off`. Default is green. Used with matrices.

`cell_color_binary_low`
The color of the binary color gradient if the numeric element is less than the `color_binary_cut_off`. Default is red. Used with matrices.

`cell_color_binary_equal`
The color of the binary color gradient if the numeric element is equal to the `color_binary_cut_off`. Default is gray. Used with matrices.

`cell_color_gradient_max`
The high color for the gradual color gradient. Default is green. Used with matrices.

`cell_color_gradient_mid`
The middle color for the gradual color gradient. Default is yellow. Used with matrices.

cell_color_gradient_min
: The low color for the gradual color gradient. Default is red. Used with matrices.

colname_font_face
: Controls the font face of column names. Overridden by the font_face parameter.

colname_group_elements
: Controls the group_elements of column names. Overridden by the group_elements parameter.

colname_background_color
: Controls the background_color of column names. Overridden by the background_color parameter.

colname_background_alpha
: Controls the background_alpha of column names. Overridden by the background_alpha parameter.

colname_border_color
: Controls the border_color of column names. Overridden by the border_color parameter.

colname_border_sides
: Controls the border_sides of column names. Overridden by the border_sides parameter.

colname_border_width
: Controls the border_width of column names. Overridden by the border_width parameter.

colname_text_color
: Controls the text_color of column names. Overridden by the text_color parameter.

colname_text_align
: Controls the text_align of column names. Overridden by the text_align parameter.

colname_text_v_align
: Controls the text_v_align of column names. Overridden by the text_v_align parameter.

colname_text_just
: Controls the text_just of column names. Overridden by the text_just parameter.

colname_text_v_just
: Controls the text_v_just of column names. Overridden by the text_v_just parameter.

colname_text_cex
: Controls the text_cex of column names. Overridden by the text_cex parameter.

colname_text_font
: Controls the text_font of column names. Overridden by the text_font parameter.

colname_text_rot
: Controls the text_rot of column names. Overridden by the text_rot parameter.

colname_round_rect_radius
: Controls the round_rect_radius of column names. Overridden by the round_rect_radius parameter.

colname_column_widths
: Controls the column_widths of column names. Overridden by the column_widths parameter.

colname_column_widths_p
: Controls the column_widths_p of column names. Overridden by the column_widths_p parameter.

colname_padding_p
: Controls the padding_p of column names. Overridden by the padding_p parameter.

maintain_aspect_ratio
: A TRUE/FALSE value which indicates whether the aspect ratio of the image should be maintained. Default is FALSE - meaning the image will be stretched to fit the designated grid area. Used with images.

n_lines
: The maximum number of lines is desired for the character string to be broken up into. Used with character strings.

str_sep
: The separator within the character string which designates where a new line should start. Used with character strings.

Value

A list with all possible aesthetic elements that can be adjusted, with the class of "grob_aes_list".

---

grob_col *The grob-column function where an object is converted a grob. Works within* grob_row *and* grob_layout.

Description

The grob-column function where an object is converted a grob. Works within grob_row and grob_layout.

Usage

```
grob_col(..., p = 1, aes_list = ga_list(), border = F,
  border_aes_list = ga_list(), title = "",
  title_aes_list = ga_list(), title_p = 0.15, caption = "",
  caption_aes_list = ga_list(), caption_p = 0.15, padding_p = 0.05,
  width = NA_real_, padding = NA_real_, hjust = 0.5, vjust = 0.5)
```

Arguments

...	Either the object to be converted to a grob, or a combination of grob-rows which need to be converted to sub-grobs.
p	The numeric proportion of the width given to the outer grob-row which should be given to the grob-column outputted by this function. Defaults to 1.
aes_list	The list outputted by ga_list, which controls aesthetics object within the grob-column.
border	A TRUE/FALSE argument corresponding to whether or not a border around the outputted grob-column is desired. Defaults to FALSE.
border_aes_list	The list outputted by ga_list, which controls aesthetics of the borders. Only two aesthetics that can be tweaked for borders are border_color, border_width and border_sides. Ignored if border is set to FALSE.
title	A character string which will be displayed as the title of the grob-column.
title_aes_list	The list outputted by ga_list, which controls aesthetics of the title of the grob-column.
title_p	The numeric proportion of height within the grob-column which will be used by the title grob. A numeric value between 0 and 0.25.
caption	A character string which will be displayed as the caption of the grob-column.
caption_aes_list	The list outputted by ga_list, which controls aesthetics of the caption of the grob-column.
caption_p	The numeric proportion of height within the grob-column which will be used by the caption grob. A numeric value between 0 and 0.25.
padding_p	The proportion of the minimum of the height and width which will be used for the padding around the edge of the grob-column. Overridden by any numeric value provided in the padding parameter.
width	The numeric width of the grob-column in the units supplied by the grob-layout. Overrides the p parameter.
padding	The numeric amount of padding around the edge of the grob-column in the units supplied by the grob-layout. Overrides the padding_p parameter.
hjust	A numeric value which will determine the alignment of the grob horizontally within its designated area. A value of 0 means moving the grob all the way to the left, a value of 1 means moving the grob all the way to the right and a value of 0.5 means keeping the grob in the middle. Defaults to 0.5. The grob-column is moved around within its padding, so if there is no padding, then hjust will be rendered useless.
vjust	A numeric value which will determine the alignment of the grob vertically within its designated area. A value of 0 means moving the grob all the way to the bottom, a value of 1 means moving the grob all the way to the top and a value of 0.5 means keeping the grob in the middle. Defaults to 0.5. The grob-column is moved around within its padding, so if there is no padding, then vjust will be rendered useless.

Details

The individual grob-column is obtained with `grob_col()$grob`.

Value

An R6 class object which contains all the information needed to create the grob-column.

`grob_image` *Converts a raw .png file to a grob, with flexible aesthetics.*

Description

Converts a raw .png file to a grob, with flexible aesthetics.

Usage

```
grob_image(img_path, aes_list, height = numeric(), width = numeric(),
  units = c("mm", "cm", "inches"))
```

Arguments

`img_path`	The local path to the raw .png file.
`aes_list`	The list outputted by `ga_list` which gives the image grob its aesthetics.
`height`	A numeric value designating the total height of the matrix grob in mm.
`width`	A numeric value designating the total width of the matrix grob in mm.
`units`	The units of the given height and width for the grob. Options are 'mm', 'cm' or 'inches', with the default of 'mm'.

Value

A grob of the raw .png file.

`grob_layout` *The main* `grobblR` *function which contains and organizes* `grob_col`*'s and* `grob_row`*'s, giving the overall grob-layout its shape.*

Description

The main `grobblR` function which contains and organizes `grob_col`'s and `grob_row`'s, giving the overall grob-layout its shape.

Usage

```
grob_layout(..., height = 280, width = 216, units = c("mm", "cm",
  "inches"), title = "", title_aes_list = ga_list(), title_p = 0.1,
  caption = "", caption_aes_list = ga_list(), caption_p = 0.05,
  padding_p = 0.05, padding = NA_real_, page_number = "")
```

Arguments

`...`	The combination of grob-rows and grob-columns which will help give the main grob-layout outputted its shape and look.
`height`	The numeric height of the grob-layout in the `units` supplied. Default is 280 mm - which is the height of an upright 8.5 x 11 inches piece of copy paper.
`width`	The numeric width of the grob in the `units` supplied. Default is 216 mm - which is the width of an upright 8.5 x 11 inches piece of copy paper.
`units`	The units of the given height and width for the grob. Options are 'mm', 'cm' or 'inches'. Default is 'mm'.
`title`	A character string which will be displayed as the title of the grob-layout.
`title_aes_list`	The list outputted by `ga_list`, which controls aesthetics of the title of the grob-layout.
`title_p`	The numeric proportion of height within the grob-layout and its allotted space which will be used by the title grob. A numeric value between 0 and 0.25.
`caption`	A character string which will be displayed as the caption at the bottom of the grob-layout.
`caption_aes_list`	The list outputted by `ga_list`, which controls aesthetics of the caption of the grob-layout.
`caption_p`	The numeric proportion of height within the grob-layout and its allotted space which will be used by the caption grob. A numeric value between 0 and 0.25.
`padding_p`	The proportion of the minimum of the height and width which will be used for the padding around the edge of the grob-layout. Overridden by any numeric value provided in the `padding` parameter.
`padding`	The numeric amount of padding around the edge of the grob-layout in the `units` supplied.
`page_number`	A single value that can be converted to an integer for the page number in the bottom right of the grob-layout within the padding. If it cannot be converted to an integer, the page number will not appear.

Value

An R6 class object containing all information necessary to create the overall grob-layout. The grob itself is called with `grob_layout()$grob`.

Examples

```
gl = grob_layout(
  grob_row(grob_col(1), grob_col(2)),
  grob_row(grob_col(3))
  )
# to retrieve the grob-layout
gl$grob
```

`grob_matrix` *Converts a data.frame/matrix to a grob, with flexible aesthetics.*

Description

Converts a data.frame/matrix to a grob, with flexible aesthetics.

Usage

```
grob_matrix(df, aes_list = ga_list(), m_type = 1, height = numeric(),
  width = numeric(), padding = numeric(), units = c("mm", "cm",
  "inches"), text_cex_adj = 0.2)
```

Arguments

`df`	The data.frame/matrix to be converted to a grob.
`aes_list`	The list outputted by `ga_list` which gives the data.frame/matrix grob its aesthetics.
`m_type`	A integer value which indicates what the default aesthetics of the table will be. Default is 1. The possible options: 1. Plain theme. 2. Table theme. 3. Column name theme. 4. Caption theme. 5. Title theme.
`height`	A numeric value designating the total height of the matrix grob in mm.
`width`	A numeric value designating the total width of the matrix grob in mm.
`text_cex_adj`	A numeric value used to adjust the automatic text cex sizing.

Value

A grob of df, with the corresponding aesthetics.

grob_row *The grob-row function which helps gives the grob from the* grob_layout *function its shape. Works hand-in-hand with the* grob_col *function.*

Description

The grob-row function which helps gives the grob from the grob_layout function its shape. Works hand-in-hand with the grob_col function.

Usage

```
grob_row(..., p = 1, border = F, border_aes_list = ga_list(),
  title = "", title_aes_list = ga_list(), title_p = 0.15,
  caption = "", caption_aes_list = ga_list(), caption_p = 0.15,
  padding_p = 0.05, padding = NA_real_, height = NA_real_)
```

Arguments

...	A series of grob_col's.
p	The numeric proportion of the given height which should be given to sub-grobs outputted in the grob-row. Defaults to 1. Overridden if a height is supplied.
border	A TRUE/FALSE argument corresponding to whether or not a border around the outputted grob-row is desired. Defaults to FALSE.
border_aes_list	The list outputted by ga_list, which controls aesthetics of the borders. Only two aesthetics that can be tweaked for borders are border_color, border_width and border_sides. Ignored if border is set to FALSE.
title	A character string which will be displayed as the title of the grob-row.
title_aes_list	The list outputted by ga_list, which controls aesthetics of the title of the grob-row.
title_p	The numeric proportion of height within the grob-row which will be used by the title grob. A numeric value between 0 and 0.25.
caption	A character string which will be displayed as the caption of the grob-row.
caption_aes_list	The list outputted by ga_list, which controls aesthetics of the caption of the grob-row.
caption_p	The numeric proportion of height within the grob-row which will be used by the caption grob. A numeric value between 0 and 0.25.
padding_p	The proportion of the minimum of the height and width which will be used for the padding around the edge of the grob-row. Overridden by any numeric value provided in the padding parameter.
padding	The numeric amount of padding around the edge of the grob-row in the units supplied by the grob-layout. Overrides the padding_p parameter.

height	The numeric height of the grob-row in the units supplied by the grob-layout. Overrides the p parameter.
border_sides	Controls the borders around the total grob-row. The input is a string with the possible words "top", "bottom", "left", "right" separated by ", ". For example, "top, left, right" will put borders on the top, left and right side of the grid cell, but not the bottom. Default is "top, bottom, left, right", or all borders.

Details

The individual grob-row is obtained with grob_row()$grob.

Value

An R6 class object which contains all the information needed to carry on to its grob-columns and create the grob-row.

---

grob_to_pdf *Converts a single grob-layout to a PDF, or combines multiple grob-layouts into a multiple page PDF document.*

Description

Converts a single grob-layout to a PDF, or combines multiple grob-layouts into a multiple page PDF document.

Usage

```
grob_to_pdf(..., file_name = character(), add_page_numbers = FALSE,
  meta_data_title = character())
```

Arguments

...	The single grob or series of grobs which will be converted to a PDF document.
file_name	The desired file name of the resulting PDF document in character format.
add_page_numbers	If TRUE, page numbers will be added to the bottom right corners of the pages of the document, based on the order of the grob-layouts listed.
meta_data_title	Title string to embed as the /Title field in the file. If not provided, it will default to the file_name provided.

Details

In the case of multiple page documents, the dimensions of the overall document will be determined by the dimensions of the first grob-layout listed.

Value

A PDF document of the grob-layout(s) which will be saved to the working directory.

`line_creator` *Breaks down character strings into one or several lines, and determines if it would fit into a specific height and width.*

Description

Breaks down character strings into one or several lines, and determines if it would fit into a specific height and width.

Usage

```
line_creator(cex_val, string, height = numeric(), width = numeric(),
  units = c("mm", "cm", "inches"), sep = "\n")
```

Arguments

`cex_val`	The text cex multiplier applied to the string.
`string`	The character string needed to be broken down into several lines.
`height`	A numeric value designating the total height of the matrix grob in mm.
`width`	A numeric value designating the total width of the matrix grob in mm.
`sep`	The separator within the character string which designates where a new line should start.

Value

A list containing a vector with each index equal to a line of the broken-down string, a TRUE/FALSE value indicating whether the lines will fit within equal sized rows and the widths in mm of each of the lines.

Index