# Package ‘grobblR’

October 13, 2022

**Title** Creating Flexible, Reproducible 'PDF' Reports

**Version** 0.2.1

**Description** A tool which allows users the ability to intuitively create
flexible, reproducible portable document format reports comprised of
aesthetically pleasing tables, images, plots and/or text.

**Depends** R (>= 3.3)

**Imports** dplyr, ggplot2, glue, graphics, grDevices, grid, gridExtra,
methods, magrittr, png, purrr, RCurl, stringr, tibble, tools

**License** MIT + file LICENSE

**Language** en-US

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**Suggests** knitr, rmarkdown, R6, scales, testthat

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Calvin Floyd [aut, cre, cph]

**Maintainer** Calvin Floyd <calvin.michael.floyd@gmail.com>

**Repository** CRAN

**Date/Publication** 2021-09-14 17:30:02 UTC

## R topics documented:

---

`add_aesthetic` *Add an Aesthetic*

---

## Description

Add an aesthetic to a grob matrix object.

## Usage

```
add_aesthetic(
  grob_object,
  aesthetic,
  value = NULL,
  group = c("cells", "column_names", "column_headings")
)
```

## Arguments

| | |
|---|---|
| `grob_object` | The R6 object outputted by either `grob_matrix` or `grob_text`. |
| `aesthetic` | The matrix aesthetic the user wishes to add. |
| `value` | A single value or a matrix of values the user wants to apply to the group of matrix / text elements for the given aesthetic.<br>If a matrix of values is supplied, then the matrix must be of the same dimensions as the chosen subset of the matrix / text. |
| `group` | The group of the grob matrix object the user wants to add the aesthetic to.<br>For objects initialized by `grob_matrix`, the user can add an aesthetic to the ’cells’, the ’column_names’ or the ’column_headings’. If the user is passing through an object initialized by `grob_text`, then only ’cells’ will be accepted. |

## Details

Accepted aesthetics:

**Matrix / Text**
- `background_alpha`
- `background_color`
- `border_color`
- `border_sides`
- `border_width`
- `font_face`
- `group_elements`
- `replace_na`
- `round_rect_radius`
- `text_align`
- `text_cex`
- `text_font`
- `text_color`
- `text_just`
- `text_v_align`
- `text_v_just`
- `text_rot`

To see descriptions of the aesthetics above, see the documentation of `ga_list`.

## Value

The R6 object of the grob matrix class with its aesthetics properties altered.

## Examples

```
df = data.frame(var1 = c(5, 14, 6, 10), var2 = c(3, 30, 17, 7))
df %>%
  grob_matrix() %>%
  add_aesthetic(aesthetic = 'text_color', value = 'red', group = 'cells') %>%
  view_grob()
```

---

## add_column_headings *Add column headings to grob matrix*

---

## Description

Add column headings onto an object initialized by `grob_matrix`.

### Usage

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

### Arguments

| | |
|---|---|
| `mat` | The grob matrix object 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. Can either be numeric indices, or column names of the initial data frame / matrix passed through `grob_matrix`. Default is an empty list. If unaltered, the function will assume the user wants to apply `headings` to all columns of the `grob_matrix` - in which case only one `headings` is allowed. |

### Details

The user must add column headings **before** adding or altering any aesthetics.

### Value

The initial grob matrix object with column headings inserted into the appropriate areas.

### Examples

```
data.frame(var1 = c(5, 14, 6, 10), var2 = c(3, 30, 17, 7)) %>%
  grob_matrix() %>%
  add_column_headings(c('HEADING')) %>%
  view_grob()
```

---

## `add_structure` *Add a Structure*

---

### Description

Add a structure to a grob matrix / image / text object.

### Usage

```
add_structure(grob_object, structure, value)
```

## Arguments

| | |
|---|---|
| `grob_object` | The R6 object initialized by one of: • `grob_matrix` • `grob_image` • `grob_text` |
| `structure` | The structure the user wishes to add. |
| `value` | If `grob_object` is outputted by `grob_matrix`, then a single value, or a vector of values corresponding to each column of the initial object passed through `grob_matrix`, the user wants to apply to the grob matrix object. Otherwise, a single value to apply to the `structure`. |

## Details

Accepted structures:

**Matrix / Text**
- `column_widths_p`
- `n_lines`
- `padding_p`

**Image**
- `aspect_ratio_multiplier`
- `maintain_aspect_ratio`

To see descriptions of the structures above, see the documentation of `ga_list`.

## Value

The initial R6 object of the grob object class with its structure properties altered.

## Examples

```
df = data.frame(x = c(5, 14, 6, 10), y = c(3, 30, 17, 7))
df %>%
  grob_matrix() %>%
  add_structure(structure = 'column_widths_p', value = c(1, 4)) %>%
  view_grob()

gg = ggplot2::ggplot(data = df, mapping = ggplot2::aes(x = x, y = y)) +
  ggplot2::geom_line(color = 'red')

gg %>%
  grob_image() %>%
  view_grob()
```

## `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`.

## `alter_at` *Alter aesthetics / structures at certain areas of a grob matrix*

### Description

Flexibly alter the aesthetic / structure of a grob matrix object at specific points of the data.frame/matrix.

### Usage

```
alter_at(
  grob_object,
  .f = NULL,
  ...,
  columns = NULL,
  rows = NULL,
  data = NULL,
  structure = NULL,
  aesthetic = NULL,
  group = NULL
)
```

## Arguments

| | |
|---|---|
| `grob_object` | The R6 grob object class initialized by `grob_matrix`. |
| `.f` | A quosure style lambda ~ `fun(.)`, which the user wants to apply to the specific subset of cells. |
| `...` | Logical predicates defined in terms of the variables in the initial data frame / matrix, or if the user provides a new data.frame to evaluate via `data`. Multiple conditions are combined with `&`. Only rows where the condition evaluates to TRUE are evaluated.<br>If no logical predicates provided, then the entire columns will be altered. |
| `columns` | A character vector of column names, or numeric column indices, of the initial data.frame/matrix, or `data` if it is provided, the user wishes to alter. |
| `rows` | A numeric vector of row indices, of the initial data.frame/matrix, or `data` if it is provided, the user wishes to alter.<br>Ignored if the user is altering a structure and not an aesthetic. |
| `data` | A separate data.frame/matrix of the same dimensions as the initial data.frame/matrix which the `.f` function and any filters will be applied to.<br>Must match the dimensions of the subset of the initial data.frame/matrix the user is attempting to alter.<br>Ignored if the user is altering a structure and not an aesthetic. |
| `structure` | Which structure the user wants to make alterations to. If left `NULL` and `aesthetic` is left `NULL`, the function will look for the most previous altered structure, either via `add_structure`, or a previous application of `alter_at`.<br>View the documentation of `add_structure` for a list of accepted structures |
| `aesthetic` | Which aesthetic the user wants to make alterations to. If left `NULL` and `structure` is left `NULL`, the function will look for the most previous altered aesthetic, either via `add_aesthetic`, or a previous application of `alter_at`.<br>View the documentation of `add_aesthetic` for a list of accepted aesthetics. |
| `group` | Which group of elements the user wants to make alterations to. If left `NULL`, the function will look for the most previous altered group, either via `add_aesthetic`, or a previous application of `alter_at`. |

## Value

The R6 grob matrix object class with its aesthetic / structure properties altered.

## Examples

```
df = data.frame(var1 = c(5, 14, 6, 10), var2 = c(3, 30, 17, 7))
df %>%
  grob_matrix() %>%
  add_aesthetic(aesthetic = 'text_color', group = 'cells', value = 'red') %>%
  alter_at(
    .f = ~ 'blue',
    abs(var2 - var1) > 1
    ) %>%
  view_grob()

test_function = function(x) ifelse(x > 15, 2, 1)

df %>%
  grob_matrix() %>%
  alter_at(
    .f = ~ test_function(.),
    aesthetic = 'font_face',
    group = 'cells'
    ) %>%
  view_grob()

df %>%
  grob_matrix() %>%
  add_structure("column_widths_p", 1) %>%
  alter_at(
    .f = ~ 2,
    columns = 1
    ) %>%
  view_grob()
```

---

## alter_column_names *Alter column names of a grob matrix*

---

### Description

Alter column names of an object initialized by `grob_matrix`.

### Usage

```
alter_column_names(
  mat,
  column_names = list(),
  column_name_cols = list(),
  group_elements = TRUE
)
```

### Arguments

| | |
|---|---|
| `mat` | The grob matrix object the column names will be edited in. |
| `column_names` | The replacement column names, in a list with each column name a separate element. The list must have the same amount of elements as the `column_name_cols` parameter. |

| | |
|---|---|
| `column_name_cols` | Which column positions of the initial data frame / matrix the `column_names` will replace, in a list with each column name's column positions a separate element. The list must have the same amount of elements as the `column_names` parameter. Can either be numeric indices, or column names of the initial data frame / matrix passed through `grob_matrix`. Default is an empty list. If unaltered, the function will assume the user wants to apply `column_names` to all columns of the `grob_matrix` - in which case only one `column_names` is allowed. |
| `group_elements` | A boolean argument on whether like, adjacent column names should be grouped together. |

## Details

The user can only use this function if the initial data frame / matrix passed through `grob_matrix` had column names to begin with.

The underlying column names will be unaffected. So, if the user wants to use `alter_at` afterwards, he/she should select the original column names and not the replacements from this function.

## Value

The initial grob matrix object with column names edited in the appropriate areas.

## Examples

```
data.frame(var1 = c(5, 14, 6, 10), var2 = c(3, 30, 17, 7)) %>%
  grob_matrix() %>%
  alter_column_names(c('COLUMN NAME')) %>%
  view_grob()
```

---

`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, 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, or `NA` (for an empty 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. 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.

---

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

---

### Description

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

### Usage

```
convert_to_image_grob(.image, aes_list, height = numeric(), width = numeric())
```

### Arguments

| | |
|---|---|
| `.image` | 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. |

### Value

A grob of the raw .png file.

---

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

---

### Description

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

### Usage

```
convert_to_matrix_grob(
  .df,
  aes_list = list(),
  height = numeric(),
  width = numeric()
)
```

### 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. |
| `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. |

### Value

A grob of `.df`, with the corresponding aesthetics.

## ga_list *Grob Aesthetic / Structure List*

### Description

Grob aesthetic list used to control aesthetics and structures within grob_col, grob_row and grob_layout.

### 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,
  replace_na = NULL,
  round_rect_radius = NULL,
  column_widths_p = NULL,
  padding_p = NULL,
  maintain_aspect_ratio = NULL,
  n_lines = 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_replace_na = NULL,
  cell_round_rect_radius = NULL,
  cell_column_widths_p = NULL,
  cell_padding_p = 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_replace_na = NULL,
  colname_round_rect_radius = NULL,
  colname_column_widths_p = NULL,
  colname_padding_p = NULL
)
```

## Arguments

| | |
|---|---|
| `aspect_ratio_multiplier` | A numeric value which controls how much to increase/decrease the aspect ratio of images. |
| `background_color` | Controls the background color of the elements of the text / matrix. |
| `background_alpha` | Controls the background alpha/opacity of the elements of the text / matrix. |
| `border_color` | Controls the color of the selected borders. |
| `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. |
| `border_width` | Controls the line width density/thickness of the selected borders. |
| `font_face` | Controls the font face of the elements of the matrix. Currently only numeric font face's are accepted. See `gpar` for more information. |
| `group_elements` | A boolean argument on whether like, adjacent matrix elements should be grouped together into a single element. |
| `text_color` | Controls the text color of the elements of the text / matrix. |
| `text_align` | Controls where the text in each cell will be centered around, horizontally. A numeric value between 0 and 1, with 0 being all the way to the left of the cell, |

| | |
|---|---|
| | and 1 being all the way to the right of the 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. |
| `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.<br>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. |
| `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. |
| `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. |
| `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. |
| `text_font` | Controls the font family within the text / matrix. Default is 'sans'. |
| `text_rot` | Controls the rotation in degrees of the text within the matrix. Default is 0 degrees<br>**Please be aware that the automatic text sizing will not react properly if the text is angled at anything other than 0 degrees.** |
| `replace_na` | Controls what `NA` values in matrix will be replaced with.<br>Default is an empty string. |
| `round_rect_radius` | Controls the radius of the corners of the rectangles matrix text is laid on top of. |
| `column_widths_p` | If automatic column widths are not desired, the user can provide a vector of width proportions corresponding to each column of the matrix. |
| `padding_p` | Controls the amount of proportional padding around each matrix cell. |
| `maintain_aspect_ratio` | A boolean argument 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. |
| `n_lines` | The maximum number of lines is desired for the character string to be broken up into. |
| `cell_background_alpha, cell_background_color, cell_border_color, cell_border_sides, cell_border_width` | These arguments correspond to that aesthetic / structure for cells of a matrix.<br>All are overridden by the corresponding arguments without `cell_` in front of them. |

`colname_background_alpha, colname_background_color, colname_border_color, colname_border_sides, colna`

These arguments correspond to that aesthetic / structure for column names of a matrix.

All are overridden by the corresponding arguments without `colname_` in front of them.

## Details

Most of the matrix aesthetics / structures are inputted into `gpar`. More in-depth details on input possibilities can be found in its documentation.2

To see which of the arguments are aesthetics and what types of grobs they apply to, view the documentation of `add_aesthetic`.

To see which of the arguments are structures and what types of grobs they apply to, view the documentation of `add_structure`.

For the color aesthetics (most notably `background_color`), inputting "none" will remove the color entirely, so the color will appear transparent.

## Value

A list with all possible aesthetic / structure elements.

---

`grob_col` *Grob Column*

---

## Description

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

## Usage

```
grob_col(
  ...,
  p = 1,
  width = NA_real_,
  aes_list = ga_list(),
  border = FALSE,
  border_aes_list = ga_list(),
  title = "",
  title_aes_list = ga_list(),
  title_p = 0.15,
  title_height = NA_real_,
  caption = "",
  caption_aes_list = ga_list(),
  caption_p = 0.15,
  caption_height = NA_real_,
  padding_p = 0.05,
  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. |
| `width` | The numeric width of the grob-column in millimeters. Overrides the `p` parameter. |
| `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. |
| `title_height` | The numeric height in mm within the grob-column which will be used by the title grob. Will override `title_p` if provided. |
| `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. |
| `caption_height` | The numeric height in mm within the grob-column which will be used by the caption grob. Will override `caption_p` if provided. |
| `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. |
| `padding` | The numeric amount of padding around the edge of the grob-column in millimeters. 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. |

## Value

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

## Examples

```
grob_col(
  "grob-column",
  aes_list = ga_list(
    text_color = "red",
    background_color = "gray90"
    )
  ) %>%
 view_grob(100, 100)
```

---

`grob_image` *Grob Image*

---

## Description

Initialize a grob image object, to be used within `grob_col`.

## Usage

```
grob_image(x)
```

## Arguments

| | |
|---|---|
| `x` | Either a `ggplot` object, a file path to .png image or a URL to a .png image. |

## Value

An R6 object of the grob image class.

## Examples

```
gg = data.frame(x = c(5, 14, 6, 10), y = c(3, 30, 17, 7)) %>%
  ggplot2::ggplot(mapping = ggplot2::aes(x = x, y = y)) +
  ggplot2::geom_line(color = 'red')

gg %>%
  grob_image() %>%
  view_grob()
```

---

`grob_layout` *Grob Layout*

---

## 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,
  title = "",
  title_aes_list = ga_list(),
  title_p = 0.1,
  title_height = NA_real_,
  caption = "",
  caption_aes_list = ga_list(),
  caption_p = 0.05,
  caption_height = NA_real_,
  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 structure and aesthetics. |
| `height` | The numeric height of the grob-layout in millimeters.<br>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 millimeters.<br>Default is 216 mm - which is the width of an upright 8.5 x 11 inches piece of copy paper. |
| `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 the grob-layout's height will be used by the title grob. |
| `title_height` | The numeric height in mm within the grob-layout which will be used by the title grob. Will override `title_p` if provided. |
| `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. |
| `caption_height` | The numeric height in mm within the grob-layout which will be used by the caption grob. Will override `caption_p` if provided. |
| `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.<br>Overridden by any numeric value provided in the `padding` parameter. |
| `padding` | The numeric amount of padding around the edge of the grob-layout in millimeters. |
| `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 its padding. If it cannot be converted to an integer, the page number will not appear. |

## Details

Learn more in `vignette("grob_layout")`

## Value

An R6 class object containing all information necessary to create the overall grob-layout.

## Examples

```
grob_layout(
  grob_row(grob_col(1, border = TRUE), grob_col(2, border = TRUE)),
  grob_row(grob_col(3, border = TRUE))
  ) %>%
  view_grob(100, 100)
```

## grob_matrix *Grob Matrix*

### Description

Initialize a grob matrix object, to be used within grob_col.

### Usage

```
grob_matrix(x)
```

### Arguments

| | |
|---|---|
| x | Either a data.frame, a matrix or a vector. |

### Details

Learn more in `vignette("grob_matrix")`

### Value

An R6 object of the grob matrix class.

### Examples

```
data.frame(
  v1 = c(15, 4, 16, 11),
  v2 = c(10, 30, 3, 10)
  ) %>%
  grob_matrix() %>%
  view_grob()
```

## grob_row *Grob Row*

### Description

The grob-row function which helps gives the grob from the grob_layout function its shape. Encompasses grob_col within the overall grob-layout.

## Usage

```
grob_row(
  ...,
  p = 1,
  height = NA_real_,
  border = FALSE,
  border_aes_list = ga_list(),
  title = "",
  title_aes_list = ga_list(),
  title_p = 0.15,
  title_height = NA_real_,
  caption = "",
  caption_aes_list = ga_list(),
  caption_p = 0.15,
  caption_height = NA_real_,
  padding_p = 0.05,
  padding = 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.<br>Overridden if a `height` is supplied. |
| `height` | The numeric height of the grob-row in millimeters.<br>Overrides the `p` parameter. |
| `border` | A TRUE/FALSE argument corresponding to whether or not a border around the outputted grob-row is desired.<br>Defaults to FALSE. |
| `border_aes_list` | The list outputted by `ga_list`, which controls aesthetics of the borders.<br>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. |
| `title_height` | The numeric height in mm within the grob_column which will be used by the title grob. Will override `title_p` if provided. |
| `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. |
| `caption_height` | The numeric height in mm within the grob_column which will be used by the caption grob. Will override `caption_p` if provided. |
| `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.<br>Overridden by any numeric value provided in the `padding` parameter. |
| `padding` | The numeric amount of padding around the edge of the grob-row in millimeters.<br>Overrides the `padding_p` parameter. |

## Value

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

## Examples

```
grob_row(
  grob_col(1, border = TRUE),
  grob_col(2, border = TRUE)
  ) %>%
  view_grob(100, 100)
```

---

`grob_text` *Grob Text*

---

## Description

Initialize a grob text object, to be used within `grob_col`.

## Usage

```
grob_text(x)
```

## Arguments

| | |
|---|---|
| `x` | A single character string. |

## Value

An R6 object of the grob matrix class.

## Examples

```
"The quick brown fox jumps over the lazy dog" %>%
  grob_text() %>%
  view_grob()
```

---

`grob_to_pdf` *Grob Layout to PDF*

---

## 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_layout`, or series of `grob_layout`'s 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.

## Examples

```
grob_layout(
  grob_row(
    grob_col(1, border = TRUE),
    grob_col(2, border = TRUE),
    border = TRUE
    ),
  grob_row(
    grob_col(3, border = TRUE),
    grob_col(
      grob_row(grob_col(4, border = TRUE), border = TRUE),
      grob_row(grob_col(5, border = TRUE), border = TRUE),
      border = TRUE
      ),
    border = TRUE
    )
  ) %>%
  grob_to_pdf(
    file_name = file.path(tempdir(), "test.pdf"),
    meta_data_title = "Test PDF"
    )
```

---

`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"),
  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. |
| `units` | millimeters |
| `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.

---

## `view_grob` *View Grob*

---

### Description

View an grob outputted by one of the `grob_` functions with a given width and height.

### Usage

```
view_grob(grob, height = NA_real_, width = NA_real_)
```

### Arguments

`grob` An object outputted by one of the following functions:

- `grob_matrix`
- `grob_image`
- `grob_row`
- `grob_col`
- `grob_layout`

| | |
|---|---|
| `height` | The numeric height in millimeters the user wishes to view the grob in. |
| `width` | The numeric width in millimeters the user wishes to view the grob in. |

### Details

Plotted with `gridExtra::grid.arrange()`.

## Examples

```
df = data.frame(
  x = c(15, 4, 16, 11),
  y = c(10, 30, 3, 10)
  )

df %>%
  grob_matrix() %>%
  view_grob()

gg = ggplot2::ggplot(data = df, mapping = ggplot2::aes(x = x, y = y)) +
  ggplot2::geom_line(color = 'red')

gg %>%
  grob_image() %>%
  view_grob()
```

# Index