Package ‘chronicle’

June 25, 2021

Type Package
Title Grammar for Creating R Markdown Reports
Version 0.3
Description A system for creating R Markdown reports with a sequential syntax.
Depends R (>= 3.5.0), magrittr
License GPL (>= 3)
Encoding UTF-8
Imports data.table, DT, dplyr, dygraphs, ggplot2, glue, knitr, plotly,
prettydoc, purrr, readr, rlang, rmarkdown, rmdformats, scales,
stat, stats, viridis, zoo
Suggests bookdown, devtools, flexdashboard, pagedown, rolldown,
rticles, tufte, xaringan
VignetteBuilder knitr
RoxygenNote 7.1.1
NeedsCompilation no
Author Philippe Heymans Smith [aut, cre]
Maintainer Philippe Heymans Smith <pheymanss@gmail.com>
Repository CRAN
Date/Publication 2021-06-25 05:00:02 UTC

R topics documented:

- add_barplot ........................................... 2
- add_boxplot .......................................... 4
- add_chunk ............................................ 5
- add_code ............................................... 7
- add_density .......................................... 8
- add_dygraph ......................................... 9
- add_histogram ....................................... 11
- add_image ........................................... 13
- add_lineplot ....................................... 14
add_barplot

Description

Add a bar plot to a chronicle report

Usage

add_barplot(
  report = "",
  dt,
  bars,
  value = NULL,
  break_bars_by = NULL,
  up_to_n_bars = 20,
  horizontal = FALSE,
  sort_by_value = FALSE,
  sort_decreasing = TRUE,
add_barplot

ggtheme = "minimal",
x_axis_label = NULL,
y_axis_label = NULL,
plot_palette = NULL,
plot_palette_generator = NULL,
barplot_title = NULL,
title_level = 2,
echo = FALSE,
message = FALSE,
warning = FALSE,
fig_width = NULL,
fig_height = NULL
)

Arguments

**report**  Character string containing all the R Markdown chunks previously added. Default is ", an empty report.

**dt**  Table with the data for the plot.

**bars**  Name of the columns containing the different groups.

**value**  Name of the columns to use as values on the y axis of the plot. If NULL (default), counts will be used.

**break_bars_by**  Name of the categorical variable used to break each bar

**up_to_n_bars**  Plot up to this number of bars. If there are more distinct values in 'bars', the function will summarise them into an 'Others' category. Default is 20

**horizontal**  Plot the bars horizontally. Default is FALSE.

**sort_by_value**  Sort the bars by value. Default is FALSE.

**sort_decreasing**  Sort the values decreasingly. Default is TRUE, but sort_by_value must also be TRUE.

**ggtheme**  ggplot2 theme function to apply. Default is ggplot2::theme_minimal.

**x_axis_label**  Label for the x axis.

**y_axis_label**  Label for the y axis.

**plot_palette**  Character vector of hex codes specifying the colors to use on the plot.

**plot_palette_generator**  Palette from the viridis package used in case plot_palette is unspecified or insufficient for the number of colors required.

**barplot_title**  Title of the bar plot section on the report. If NULL, chronicle will try to parse a generic title using make_title()

**title_level**  Level of the section title of this plot (ie, number of # on Rmarkdown syntax.)

**echo**  Whether to display the source code in the output document. Default is FALSE.

**message**  Whether to preserve messages on rendering. Default is FALSE.

**warning**  Whether to preserve warnings on rendering. Default is FALSE.

**fig_width**  Width of the plot (in inches).

**fig_height**  Height of the plot (in inches).
Value

An rmarkdown file as a character string, now containing a chunk for adding the specified bar plot.

Examples

```r
html_report <- add_boxplot(report = '',
                           dt = iris,
                           bars = 'Species',
                           value = 'Sepal.Length')

cat(html_report)
```

**add_boxplot**  
*Add a box plot to a chronicle report*

Description

Add a box plot to a chronicle report

Usage

```r
add_boxplot(
  report = '',
  dt,
  value,
  groups = NULL,
  split_groups_by = NULL,
  jitter = TRUE,
  ggtheme = NULL,
  x_axis_label = NULL,
  y_axis_label = NULL,
  plot_palette = NULL,
  plot_palette_generator = NULL,
  boxplot_title = NULL,
  title_level = 2,
  echo = FALSE,
  message = FALSE,
  warning = FALSE,
  fig_width = NULL,
  fig_height = NULL
)
```

Arguments

- **report**: Character string containing all the R Markdown chunks previously added. Default is '', an empty report.
- **dt**: Table with the data for the plot.
- **value**: Name of the column to use as values on the y axis of the plot.
**add_chunk**

**groups** Name of the column containing the different groups.

**split_groups_by** Column to split each group.

**jitter** Whether to add the actual values of each observation over the box plots. Only done when dt has 1000 rows or less.

**ggtheme** ggplot2 theme function to apply. Default is ggplot2::theme_minimal.

**x_axis_label** Label for the x axis.

**y_axis_label** Label for the y axis.

**plot_palette** Character vector of hex codes specifying the colors to use on the plot.

**plot_palette_generator** Palette from the viridis package used in case plot_palette is unspecified or insufficient for the number of colors required.

**boxplot_title** Title of the box plot section on the report. If NULL, chronicle will try to parse a generic title using make_title()

**title_level** Level of the section title of this plot (ie, number of # on Rmarkdown syntax.)

**echo** Whether to display the source code in the output document. Default is FALSE.

**message** Whether to preserve messages on rendering. Default is FALSE.

**warning** Whether to preserve warnings on rendering. Default is FALSE.

**fig_width** Width of the plot (in inches).

**fig_height** Height of the plot (in inches).

**Value**

An rmarkdown file as a character string, now containing a chunk for adding the specified box plot.

**Examples**

```r
html_report <- add_boxplot(report = '',
    dt = iris,
    value = 'Sepal.Length',
    groups = 'Species', jitter = TRUE)

cat(html_report)
```

**Description**

Transforms a function call into an Rmarkdown chunk
add_chunk

Usage

```r
add_chunk(
  report = "", 
  fun, 
  params, 
  chunk_title = NULL, 
  title_level = 2, 
  echo = FALSE, 
  message = FALSE, 
  warning = FALSE, 
  fig_width = NULL, 
  fig_height = NULL, 
  guess_title = TRUE
)
```

Arguments

- **report**: Character string containing all the R Markdown chunks previously added. Default is ", an empty report.
- **fun**: Function to call.
- **params**: List of parameters to be passed to fun.
- **chunk_title**: Title of the Rmarkdown chunk. If NULL, chronicle will try to parse a generic title based on the function and parameters passed using make_title().
- **title_level**: Level of the section title of this plot (ie, number of # on Rmarkdown syntax.)
- **echo**: Whether to display the source code in the output document. Default is FALSE.
- **message**: Whether to preserve messages on rendering. Default is FALSE.
- **warning**: Whether to preserve warnings on rendering. Default is FALSE.
- **fig_width**: Width of the plot (in inches).
- **fig_height**: Height of the plot (in inches).
- **guess_title**: If TRUE, tries to generate a generic title for chronicle::make_* family of functions (eg 'Sepal.Length vs Sepal.Width by Species' for make_scatter)

Value

An rmarkdown chunk as a character string.

Examples

```r
library(chronicle)
html_chunk <- add_chunk(fun = chronicle::make_barplot, 
params = list(dt = 'iris', 
  value = 'Sepal.Width', 
  bars = 'Species'))
cat(html_chunk)
```
add_code

Add formatted code chunks to a chronicle R Markdown report

Description

Beware that code indentation of the chronicle call will affect the indentation of the chunk, so make sure not to leave unintended indentation in the 'code' parameter on this function call.

Usage

add_code(
    report = "",
    code,
    code_title = NULL,
    title_level = 2,
    eval = TRUE,
    echo = TRUE,
    message = FALSE,
    warning = FALSE,
    fig_width = NULL,
    fig_height = NULL
)

Arguments

report Character string containing all the R Markdown chunks previously added. Default is "", an empty report.

code The code that will be added to the report. Mind the indentation on the call, since spaces between quotations will be preserved.

code_title The title of the text section. Default is NULL.

title_level Level of the section title of this text (ie, number of # on Rmarkdown syntax.)

eval Run the code instead of just display it. Default is TRUE.

echo Whether to display the source code in the output document. Default is FALSE.

message Whether to preserve messages on rendering. Default is FALSE.

warning Whether to preserve warnings on rendering. Default is FALSE.

fig_width Width of the figures printed from this code.

fig_height Height of the figures printed from this code.

Value

The text of the Rmarkdown report plus an additional section with the code chunk.
Examples

```r
html_report <- add_code(report = '',
    code_title = 'Code comes after this title',
    code = '
f <- function(x, y){paste(x,y)},

f("a", "b")',
    eval = FALSE,
    echo = TRUE,
    fig_width = 12,
    fig_height = 8)
cat(html_report)
```

---

**add_density**

Add a density plot to a chronicle report

**Description**

Add a density plot to a chronicle report

**Usage**

```r
add_density(
    report = '',
    dt,
    value,
    groups = NULL,
    faceted = TRUE,
    scales = "fixed",
    ggtheme = NULL,
    x_axis_label = NULL,
    plot_palette = NULL,
    plot_palette_generator = NULL,
    density_title = NULL,
    title_level = 2,
    echo = FALSE,
    message = FALSE,
    warning = FALSE,
    fig_width = NULL,
    fig_height = NULL
)
```

**Arguments**

- **report**: Character string containing all the R Markdown chunks previously added. Default is '', an empty report.
- **dt**: data.frame containing the data to plot.
- **value**: Name of the column to use as values on the y axis of the plot.
- **groups**: Name of the column containing the different groups.
add_dygraph

If TRUE (default), each group will be plotted separately.

From ggplot2::facet_wrap: Should scales be 'fixed', 'free', or free in one dimension ('free_x', 'free_y'). Default is 'fixed'.

ggplot2 theme function to apply. Default is ggplot2::theme_minimal.

Label for the x axis.

Character vector of hex codes specifying the colors to use on the plot.

Palette from the viridis package used in case plot_palette is unspecified or insufficient for the number of colors required.

Title of the density plot section on the report. If NULL, chronicle will try to parse a generic title using make_title()

Level of the section title of this plot (ie, number of # on Rmarkdown syntax.)

Whether to display the source code in the output document. Default is FALSE.

Whether to preserve messages on rendering. Default is FALSE.

Whether to preserve warnings on rendering. Default is FALSE.

Width of the plot (in inches).

Height of the plot (in inches).

An rmarkdown file as a character string, now containing a chunk for adding the specified density plot.

html_report <- add_density(report = "",
                          dt = iris,
                          value = 'Sepal.Length',
                          groups = 'Species')
cat(html_report)

---

Add a dygraph to a chronic report

Add a dygraph to a chronic report
Usage

```r
add_dygraph(
    report = "",
    dt,  
    value, 
    date,
    groups = NULL,
    y_axis_label = NULL,
    plot_palette = NULL,
    plot_palette_generator = NULL,
    dygraph_title = NULL,
    title_level = 2,
    echo = FALSE,
    message = FALSE,
    warning = FALSE,
    fig_width = NULL,
    fig_height = NULL
)
```

Arguments

- **report**: Character string containing all the R Markdown chunks previously added. Default is ", an empty report.
- **dt**: Data to plot
- **value**: Name of the column of the data frame containing the numerical variables of the time series.
- **date**: Name of the column containing the date variable. It must be already a date or time object.
- **groups**: Name of the columns containing the different groups.
- **y_axis_label**: Label for the y axis. x axis is the date (or time) so it is not needed
- **plot_palette**: Character vector of hex codes specifying the colors to use on the plot.
- **plot_palette_generator**: Palette from the viridis package used in case plot_palette is unspecified or insufficient for the number of colors required.
- **dygraph_title**: Title for the Rmarkdown section containing the dygraph
- **title_level**: Level of the section title of this plot (ie, number of # on Rmarkdown syntax.)
- **echo**: Whether to display the source code in the output document. Default is FALSE.
- **message**: Whether to preserve messages on rendering. Default is FALSE.
- **warning**: Whether to preserve warnings on rendering. Default is FALSE.
- **fig_width**: Width of the plot (in inches).
- **fig_height**: Height of the plot (in inches).

Value

An R Markdown file as a character string, now containing a chunk for the specified dygraph.
add_histogram

Examples

dat <- data.frame(x = c(rnorm(100, 2, 4),
rnorm(100, 6, 1),
rnorm(100, 8, 2)),
group = c(rep('A', 100),
rep('B', 100),
rep('C', 100)),
date = rep(seq(as.Date("2020-01-01"),
as.Date("2020-04-09"),
'days'),
3))
html_report <- add_dygraph(report = '',
dt = dat,
value = 'x',
date = 'date')
cat(html_report)

add_histogram Add a histogram plot to a chronicle report

Description

Add a histogram plot to a chronicle report

Usage

add_histogram(
    report = "",
dt,
value,
groups = NULL,
binwidth = NULL,
bins = NULL,
scales = "fixed",
ggtheme = NULL,
x_axis_label = NULL,
plot_palette = NULL,
plot_palette_generator = NULL,
histogram_title = NULL,
title_level = 2,
echo = FALSE,
message = FALSE,
warning = FALSE,
fig_width = NULL,
fig_height = NULL)
**Arguments**

- **report**: Character string containing all the R Markdown chunks previously added. Default is "", an empty report.
- **dt**: data.frame containing the data to plot.
- **value**: Name of the column to use as values on the y axis of the plot.
- **groups**: Name of the column containing the different groups.
- **binwidth**: Width of the histogram bins.
- **bins**: Number of bins. Overridden by binwidth. Defaults to 30.
- **scales**: From ggplot2::facet_wrap: Should scales be 'fixed', 'free', or free in one dimension ('free_x', 'free_y'). Default is 'fixed'.
- **ggtheme**: ggplot2 theme function to apply. Default is ggplot2::theme_minimal.
- **x_axis_label**: Label for the x axis.
- **plot_palette**: Character vector of hex codes specifying the colors to use on the plot.
- **plot_palette_generator**: Palette from the viridis package used in case plot_palette is unspecified or insufficient for the number of colors required.
- **histogram_title**: Title of the histogram plot section on the report. If NULL, chronicle will try to parse a generic title using make_title() 
- **title_level**: Level of the section title of this plot (ie, number of # on Rmarkdown syntax.)
- **echo**: Whether to display the source code in the output document. Default is FALSE.
- **message**: Whether to preserve messages on rendering. Default is FALSE.
- **warning**: Whether to preserve warnings on rendering. Default is FALSE.
- **fig_width**: Width of the plot (in inches).
- **fig_height**: Height of the plot (in inches).

**Value**

An rmarkdown chunk as a character string, now containing a chunk for adding the histogram plot.

**Examples**

```r
html_report <- add_histogram(report = "",
   dt = iris,
   value = 'Sepal.Length',
   groups = 'Species')

cat(html_report)
```
add_image

Description

Add an image to a chronicle Rmarkdown report

Usage

add_image(
  report = "", 
  image_path, 
  image_caption = NULL, 
  image_title = NULL, 
  title_level = 2, 
  fig_width = NULL, 
  fig_height = NULL
)

Arguments

- **report**: Character string containing all the R Markdown chunks previously added. Default is "", an empty report.
- **image_path**: The path to the image that will be added to the report.
- **image_caption**: A caption to be printed for the image.
- **image_title**: The title of the text section. Default is NULL.
- **title_level**: Level of the section title of this text (i.e., number of # on Rmarkdown syntax.)
- **fig_width**: Width of the figures printed from this code.
- **fig_height**: Height of the figures printed from this code.

Value

The text of the Rmarkdown report plus an additional section with the text.

Examples

library(chronicle)
report <- add_image(image_path = 'readme1.png',
  image_caption = 'This is the caption of the image',
  image_title = 'This is the image that I want to include')
add_lineplot  

Add a line plot to a chronicle report

Description

Add a line plot to a chronicle report

Usage

add_lineplot(
  report = "",
  dt, 
  x, 
  y, 
  groups = NULL, 
  faceted = NULL, 
  scales = NULL, 
  show_trend = NULL, 
  trend_method = NULL, 
  ggtheme = NULL, 
  x_axis_label = NULL, 
  y_axis_label = NULL, 
  plot_palette = NULL, 
  plot_palette_generator = NULL, 
  lineplot_title = NULL, 
  title_level = 2, 
  echo = FALSE, 
  message = FALSE, 
  warning = FALSE, 
  fig_width = NULL, 
  fig_height = NULL
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report</td>
<td>Character string containing all the R Markdown chunks previously added. Default is &quot;, an empty report.</td>
</tr>
<tr>
<td>dt</td>
<td>data.frame containing the data to plot.</td>
</tr>
<tr>
<td>x</td>
<td>Value on the x axis.</td>
</tr>
<tr>
<td>y</td>
<td>Value on the y axis.</td>
</tr>
<tr>
<td>groups</td>
<td>Name of the column containing the different groups.</td>
</tr>
<tr>
<td>faceted</td>
<td>If TRUE (default), each group will be plotted separately.</td>
</tr>
<tr>
<td>scales</td>
<td>From ggplot2::facet_wrap: Should scales be 'fixed', 'free', or free in one dimension ('free_x', 'free_y'). Default is 'fixed'.</td>
</tr>
<tr>
<td>show_trend</td>
<td>If TRUE, adds a ggplot2::geom_smooth() line to the plot.</td>
</tr>
</tbody>
</table>
add_quotes

```r
trend_method  # The method ggplot2::geom_smooth will use. Default is 'loess', which is a local polynomial regression fit
ggtheme       # ggplot2 theme function to apply. Default is ggplot2::theme_minimal.
x_axis_label  # Label for the x axis.
y_axis_label  # Label for the y axis.
plot_palette  # Character vector of hex codes specifying the colors to use on the plot.
plot_palette_generator
               # Palette from the viridis package, used in case plot_palette is unspecified or insufficient for the number of colors required.
lineplot_title  # Title of the line plot section on the report. If NULL, chronicle will try to parse a generic title using make_title()
title_level    # Level of the section title of this plot (ie, number of # on Rmarkdown syntax.)
echo           # Whether to display the source code in the output document. Default is FALSE.
message        # Whether to preserve messages on rendering. Default is FALSE.
warning        # Whether to preserve warnings on rendering. Default is FALSE.
fig_width      # Width of the plot (in inches).
fig_height     # Height of the plot (in inches).
```

**Value**

An R Markdown file as a character string, now containing a chunk for the specified line plot.

**Examples**

```r
html_report <- add_lineplot(report = "",
dt = ggplot2::mpg,
x = 'hwy',
y = 'cty',
groups = 'manufacturer',
faceted = FALSE)
cat(html_report)
```

---

**add_quotes**

*Adds additional quotations to character values*

**Description**

This is useful when assembling functions calls, where you specify parameter names and character values at the same time.

**Usage**

```r
add_quotes(x, except = NULL, single_quote = TRUE, collapse = NULL)
```
Arguments

- **x**: List or named vector
- **except**: Vector specifying the names of the elements that should not be enquoted.
- **single_quote**: Use single quotes ('') instead of double quotes (""). Default is TRUE.
- **collapse**: If not NULL, collapse the values into a single vector using this value as the separator. Default is NULL.

Value

The list or named vector, with additional quotes around the appropriate values

Examples

```r
params = list(a = TRUE, b = FALSE, c = 'ABC', d = 15)
add_quotes(params)
add_quotes(params, except = 'c')
```

Description

Add a raincloud plot to a chronicle report

Usage

```r
add_raincloud(
  report = "",
  dt,
  value,
  groups = NULL,
  adjust = 0.5,
  include_boxplot = TRUE,
  include_mean = FALSE,
  include_median = TRUE,
  force_all_jitter_obs = FALSE,
  ggtheme = "minimal",
  x_axis_label = NULL,
  plot_palette = NULL,
  plot_palette_generator = NULL,
  static = NULL,
  raincloud_title = NULL,
  title_level = 2,
  echo = FALSE,
  message = FALSE,
  warning = FALSE,
)```
```r
fig_width = NULL,
fig_height = NULL
)
```

**Arguments**

- **report**: Character string containing all the R Markdown chunks previously added. Default is ", an empty report.
- **dt**: data.frame containing the data to plot.
- **value**: Name of the column to use as values on the y axis of the plot.
- **groups**: Name of the column containing the different groups.
- **adjust**: Width of the kernel bins. The smaller the value, the higher the resolution of the density. For full details, see ?ggplot2::stat_density.
- **include_boxplot**: Include a boxplot over the raincloud. Default is TRUE.
- **include_mean**: Mark the median of each distribution. Default is TRUE.
- **include_median**: Mark the mean of each distribution. Default is FALSE.
- **force_all_jitter_obs**: When the data has more than 1000 observations, the function will sample 1000 observations in order to keep the object reasonably small. If you need to override it, set this value to TRUE.
- **ggtheme**: ggplot2 theme function to apply. Default is ggplot2::theme_minimal.
- **x_axis_label**: Label for the x axis.
- **plot_palette**: Character vector of hex codes specifying the colors to use on the plot.
- **plot_palette_generator**: Palette from the viridis package used in case plot_palette is unspecified or insufficient for the number of colors required.
- **static**: If TRUE, the output will be static ggplot chart instead of an interactive ggplotly chart. Default is FALSE.
- **raincloud_title**: Title of the raincloud plot section on the report. If NULL, chronicle will try to parse a generic title using make_title()
- **title_level**: Level of the section title of this plot (ie, number of # on Rmarkdown syntax.)
- **echo**: Whether to display the source code in the output document. Default is FALSE.
- **message**: Whether to preserve messages on rendering. Default is FALSE.
- **warning**: Whether to preserve warnings on rendering. Default is FALSE.
- **fig_width**: Width of the plot (in inches).
- **fig_height**: Height of the plot (in inches).

**Value**

An rmarkdown file as a character string, now containing a chunk for adding the specified raincloud plot.
add_scatterplot

Add a scatter plot to a chronicle report

Add a scatter plot to a chronicle report

**add_scatterplot**

```r
add_scatterplot(
  report = "",
  dt,
  x,
  y,
  groups = NULL,
  faceted = NULL,
  scales = NULL,
  show_trend = NULL,
  trend_method = NULL,
  ggtheme = NULL,
  x_axis_label = NULL,
  y_axis_label = NULL,
  plot_palette = NULL,
  plot_palette_generator = NULL,
  scatterplot_title = NULL,
  title_level = 2,
  echo = FALSE,
  message = FALSE,
  warning = FALSE,
  fig_width = NULL,
  fig_height = NULL
)
```

**Arguments**

- `report`  Character string containing all the R Markdown chunks previously added. Default is ", an empty report.
- `dt`  data.frame containing the data to plot.
- `x`  Value on the x axis.
add_scatterplot

- `y` Value on the y axis.
- `groups` Name of the column containing the different groups.
- `faceted` If TRUE (default), each group will be plotted separately.
- `scales` From ggplot2::facet_wrap: Should scales be 'fixed', 'free', or free in one dimension ('free_x', 'free_y'). Default is 'fixed'.
- `show_trend` If TRUE, adds a ggplot2::geom_smooth() line to the plot. Default is FALSE.
- `trend_method` The method ggplot2::geom_smooth will use. Default is 'loess', which is a local polynomial regression fit
- `ggtheme` ggplot2 theme function to apply. Default is ggplot2::theme_minimal.
- `x_axis_label` Label for the x axis.
- `y_axis_label` Label for the y axis.
- `plot_palette` Character vector of hex codes specifying the colors to use on the plot.
- `plot_palette_generator` Palette from the viridis package, used in case plot_palette is unspecified or insufficient for the number of colors required.
- `scatterplot_title` Title of the scatter plot section on the report. If NULL, chronicle will try to parse a generic title using make_title()
- `title_level` Level of the section title of this plot (ie, number of # on Rmarkdown syntax.)
- `echo` Whether to display the source code in the output document. Default is FALSE.
- `message` Whether to preserve messages on rendering. Default is FALSE.
- `warning` Whether to preserve warnings on rendering. Default is FALSE.
- `fig_width` Width of the plot (in inches).
- `fig_height` Height of the plot (in inches).

**Value**

An R Markdown file as a character string, now containing a chunk for the specified scatter plot.

**Examples**

```r
html_report <- add_scatterplot(report = "",
                            dt = ggplot2::mpg,
                            x = 'hwy',
                            y = 'cty',
                            groups = 'manufacturer',
                            faceted = FALSE)
cat(html_report)
```
add_table  

Add a table to a chronicle report

Description

Add a table to a chronicle report

Usage

```r
add_table(
    report = "",
    table, 
    table_title = NULL, 
    title_level = 2, 
    html_table_type = c("DT", "kable"), 
    table_params = NULL, 
    fig_width = NULL, 
    fig_height = NULL
)
```

Arguments

- **report**: Character string containing all the R Markdown chunks previously added. Default is "", an empty report.
- **table**: data.frame to print on the report.
- **table_title**: title of the table. Default is no title.
- **title_level**: Level of the section title of this plot (ie, number of # on Rmarkdown syntax.)
- **html_table_type**: Either print a knitr::kable table or a DT htmlwidget.
- **table_params**: A named list of additional parameters to be passed to either knitr::kable() or DT::datatable(), depending on html_table_type
- **fig_width**: Width of the figures printed from this code.
- **fig_height**: Height of the figures printed from this code.

Value

An R Markdown file as a character string, now containing a chunk for the specified table.

Examples

```r
html_report <- add_table(table = iris,
                           table_title = 'Iris measures',
                           html_table_type = 'kable')
cat(html_report)
```
add_text

Add text to a chronicle Rmarkdown report

Description

Add text to a chronicle Rmarkdown report

Usage

add_text(report = "", text, text_title = NULL, title_level = 2)

Arguments

report Character string containing all the R Markdown chunks previously added. Default is ", an empty report.
text The text that will be added to the report.
text_title The title of the text section. Default is NULL.
title_level Level of the section title of this text (ie, number of # on Rmarkdown syntax.) Default is 1.

Value

The text of the Rmarkdown report plus an additional section with the text.

Examples

html_report <- add_text(text = 'This is the text that will be seen outside of any chunk',
                         text_title = 'Text title')
cat(html_report)

add_title

Add a titled section to a chronicle Rmarkdown report

Description

Add a titled section to a chronicle Rmarkdown report

Usage

add_title(report = "", title, title_level = 1)

Arguments

report Character string containing all the R Markdown chunks previously added. Default is ", an empty report.
title The title to be added as a section.
title_level Level of the section title (ie, number of # on Rmarkdown syntax.)
Value

The text of the Rmarkdown report plus an additional section by the given title.

Examples

```r
html_report <- add_title(report = '',
                          title = 'Just the title here')
cat(html_report)
```

---

**add_violin**

*Add a violin plot to a chronicle report*

---

Description

Add a violin plot to a chronicle report

Usage

```r
add_violin(
  report = '',
  dt,
  value,
  groups = NULL,
  jitter = NULL,
  ggtheme = NULL,
  x_axis_label = NULL,
  y_axis_label = NULL,
  plot_palette = NULL,
  plot_palette_generator = NULL,
  violin_title = NULL,
  title_level = 2,
  echo = FALSE,
  message = FALSE,
  warning = FALSE,
  fig_width = NULL,
  fig_height = NULL
)
```

Arguments

- **report** Character string containing all the R Markdown chunks previously added. Default is ", an empty report.
- **dt** Table with the data for the plot.
- **value** Name of the column to use as values on the y axis of the plot.
- **groups** Name of the column containing the different groups.
assemble_call

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>jitter</td>
<td>Whether to add the actual values of each observation over the violin plots. Only done when dt has 1000 rows or less.</td>
</tr>
<tr>
<td>ggtheme</td>
<td>ggplot2 theme function to apply. Default is ggplot2::theme_minimal.</td>
</tr>
<tr>
<td>x_axis_label</td>
<td>Label for the x axis.</td>
</tr>
<tr>
<td>y_axis_label</td>
<td>Label for the y axis.</td>
</tr>
<tr>
<td>plot_palette</td>
<td>Character vector of hex codes specifying the colors to use on the plot.</td>
</tr>
<tr>
<td>plot_palette_generator</td>
<td>Palette from the viridis package used in case plot_palette is unspecified or insufficient for the number of colors required.</td>
</tr>
<tr>
<td>violin_title</td>
<td>Title of the violin plot section on the report. If NULL, chronicle will try to parse a generic title using make_title().</td>
</tr>
<tr>
<td>title_level</td>
<td>Level of the section title of this plot (ie, number of # on Rmarkdown syntax.)</td>
</tr>
<tr>
<td>echo</td>
<td>Whether to display the source code in the output document. Default is FALSE.</td>
</tr>
<tr>
<td>message</td>
<td>Whether to preserve messages on rendering. Default is FALSE.</td>
</tr>
<tr>
<td>warning</td>
<td>Whether to preserve warnings on rendering. Default is FALSE.</td>
</tr>
<tr>
<td>fig_width</td>
<td>Width of the plot (in inches).</td>
</tr>
<tr>
<td>fig_height</td>
<td>Height of the plot (in inches).</td>
</tr>
</tbody>
</table>

Value

An rmarkdown chunk as a character string, now containing a chunk for adding the violin plot.

Examples

```r
html_report <- add_violin(report = "",
                          dt = iris,
                          value = 'Sepal.Length',
                          groups = 'Species', jitter = TRUE)
cat(html_report)
```

assemble_call

Assembles a formatted function call from a function and a list of parameters

Description

Assembles a formatted function call from a function and a list of parameters

Usage

```r
assemble_call(fun_name, params, non_char = NULL)
```
Arguments

fun_name       Name of the function to be called (must be a character or coercible to a character).
params         Named list or vector containing the parameters for the fun call.
non_char       Names of the parameters whose values should not be interpreted as character values

Value

A character string with the formatted function call.

Examples

chronicle::assemble_call(fun_name = 'base::sapply',
                          params = list(X = 'iris',
                                        FUN = 'class'))
chronicle::assemble_call(fun_name = 'base::sapply',
                          params = list(X = 'iris',
                                        FUN = 'class'),
                          non_char = c('X', 'FUN'))

check_cols  Warns if any of the passed column names is missing from the data provided.

Description

Warns if any of the passed column names is missing from the data provided.

Usage

check_cols(dt, cols)

Arguments

dt               A data.frame.
cols             A vector of column names.

Value

The vector of all columns present in dt.

Examples

chronicle::check_cols(mtcars, c('cyl', 'made_up_column'))
**file_extension**

*Parse the file extension for each R Markdown output format*

**Description**

Currently supports:

**Usage**

```r
file_extension(file_type)
```

**Arguments**

- `file_type`: R Markdown output formats.

**Details**

- `rmdformats` * prettydoc * bookdown * ioslides * tuftes_html * xaringan * rolldown * flexdashboard
- `slidy_presentation` * `html_document` * `html_notebook` * `pagedown`

**Value**

The file extension corresponding to the provided formats (".html", "pdf", ".md", ".docx", ".pptx")

**Examples**

```r
file_extension(c("prettydoc", "word_document", "tuftes_handout"))
```

---

**make_barplot**

*Create a bar plot from a data frame through ggplotly*

**Description**

Create a bar plot from a data frame through ggplotly

**Usage**

```r
make_barplot(
  dt,
  bars,
  value = NULL,
  break_bars_by = NULL,
  up_to_n_bars = 20,
  horizontal = FALSE,
  sort_by_value = horizontal,
  sort_decreasing = TRUE,
```
Arguments

- **dt**: data.frame containing the data to plot.
- **bars**: Name of the column containing the different groups.
- **value**: Name of the columns to use as value on the y axis of the plot. If NULL (default), counts will be used.
- **break_bars_by**: Name of the categorical variable used to break each bar.
- **up_to_n_bars**: Plot up to this number of bars. If there are more distinct values in 'bars', the function will summarise them into an 'Others' category. Default is 20.
- **horizontal**: Plot the bars horizontally. Default is FALSE.
- **sort_by_value**: Sort the bars by value. Default is FALSE unless horizontal is TRUE.
- **sort_decreasing**: Sort the values decreasingly. Default is TRUE, but sort_by_value must also be TRUE.
- **ggtheme**: ggplot2 theme function to apply. Default is ggplot2::theme_minimal.
- **x_axis_label**: Label for the x axis.
- **y_axis_label**: Label for the y axis.
- **plot_palette**: Character vector of hex codes specifying the colors to use on the plot.
- **plot_palette_generator**: Palette from the viridis package used in case plot_palette is unspecified or insufficient for the number of colors required.
- **static**: If TRUE (or if the dataset is over 10,000 rows), the output will be static ggplot chart instead of an interactive ggplotly chart. Default is FALSE.

Value

A plotly-ized version of a ggplot bar plot.

Examples

```
make_barplot(dt = iris, bars = 'Species', value = 'Sepal.Length')
make_barplot(dt = ggplot2::mpg,
            bars = 'manufacturer',
            break_bars_by = 'model',
            value = 'cty',
            horizontal = TRUE,
            sort_by_value = TRUE)
```
make_boxplot

Create a box plot from a data frame through ggplotly

Description

Create a box plot from a data frame through ggplotly

Usage

make_boxplot(
  dt,
  value,
  groups = NULL,
  split_groups_by = NULL,
  jitter = FALSE,
  ggtheme = "minimal",
  x_axis_label = NULL,
  y_axis_label = NULL,
  plot_palette = NULL,
  plot_palette_generator = "plasma",
  static = FALSE
)

Arguments

dt data.frame containing the data to plot.
value Name of the column to use as values on the y axis of the plot.
groups Name of the column containing the different groups.
split_groups_by Second column to split each group by (eg, create individual boxplots within the 'groups'.)
jitter Whether to add the actual values of each observation over the box plots. Only done when dt has 10,000 rows or less.
ggtheme ggplot2 theme function to apply. Default is ggplot2::theme_minimal.
x_axis_label Label for the x axis.
y_axis_label Label for the y axis.
plot_palette Character vector of hex codes specifying the colors to use on the plot.
plot_palette_generator Palette from the viridis package used in case plot_palette is unspecified or insufficient for the number of colors required.
static If TRUE (or if the dataset is over 10,000 rows), the output will be static ggplot chart instead of an interactive ggplotly chart. Default is FALSE.
**make_density**

**Value**

A plotly-ized version of a ggplot box plot.

**Examples**

```
make_boxplot(dt = ggplot2::mpg, value = 'hwy', groups = 'drv', jitter = TRUE)
```

---

**Description**

Create a density plot from a data frame through ggplotly

**Usage**

```
make_density(
  dt,
  value,
  groups = NULL,
  faceted = TRUE,
  scales = "fixed",
  ggtheme = "minimal",
  x_axis_label = NULL,
  plot_palette = NULL,
  plot_palette_generator = "plasma",
  static = FALSE
)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dt</td>
<td>data.frame containing the data to plot.</td>
</tr>
<tr>
<td>value</td>
<td>Name of the column to use as values on the y axis of the plot.</td>
</tr>
<tr>
<td>groups</td>
<td>Name of the column containing the different groups.</td>
</tr>
<tr>
<td>faceted</td>
<td>If TRUE (default), each group will be plotted separately.</td>
</tr>
<tr>
<td>scales</td>
<td>From ggplot2::facet_wrap: Should scales be 'fixed', 'free', or free in one dimension ('free_x', 'free_y'). Default is 'fixed'.</td>
</tr>
<tr>
<td>ggtheme</td>
<td>ggplot2 theme function to apply. Default is ggplot2::theme_minimal.</td>
</tr>
<tr>
<td>x_axis_label</td>
<td>Label for the x axis.</td>
</tr>
<tr>
<td>plot_palette</td>
<td>Character vector of hex codes specifying the colors to use on the plot.</td>
</tr>
<tr>
<td>plot_palette_generator</td>
<td>Palette from the viridis package used in case plot_palette is unspecified or insufficient for the number of colors required.</td>
</tr>
<tr>
<td>static</td>
<td>If TRUE (or if the dataset is over 10,000 rows), the output will be static ggplot chart instead of an interactive ggplotly chart. Default is FALSE.</td>
</tr>
</tbody>
</table>
**Value**

A plotly-ized version of a ggplot density plot.

**Examples**

```r
make_density(dt = iris, 
              value = 'Sepal.Length', 
              groups = 'Species')
make_density(dt = iris, 
              value = 'Sepal.Length', 
              groups = 'Species', 
              faceted = FALSE)
```

**make_dygraph**

_plot a time series from a data frame through dygraph's interactive html plot interface_

**Description**

Plot a time series from a data frame through dygraph’s interactive html plot interface

**Usage**

```r
make_dygraph(
  dt, 
  value, 
  date, 
  groups = NULL, 
  y_axis_label = NULL, 
  plot_palette = NULL, 
  plot_palette_generator = "plasma", 
  static = FALSE 
)
```

**Arguments**

- **dt**
  - data.frame containing the data to plot. It must have a numerical variable, a date variable, and optionally a grouping variable to split the data and plot them as individual time series inside the same plot.

- **value**
  - Name of the column of the data frame containing the numerical variables of the time series.

- **date**
  - Name of the column containing the date variable. It must be already a date or time object.

- **groups**
  - Name of the columns containing the different groups.

- **y_axis_label**
  - Label for the y axis. x axis is the date (or time) so it is not needed
make_histogram

Description

Create a histogram plot from a data frame through ggplotly

Usage

make_histogram(
  dt,  
  value,  
  groups = NULL,  
  binwidth = NULL,  
  bins = 30,  
  plot_palette,  
  plot_palette_generator,  
  static)

plot_palette  Character vector of hex codes specifying the colors to use on the plot. Default is RColorBrewer’s Paired and Spectral colors concatenated.

plot_palette_generator

Palette from the viridis package used in case plot_palette is unspecified or insufficient for the number of colors required.

static  If TRUE (or if the dataset is over 10,000 rows), the output will be static ggplot chart instead of a dygraph. Default is FALSE.

Value

A dygraph of the numerical variable specified, optionally split by the values of 'groups'. If static is set to TRUE, it will return a ggplot line plot

Examples

dat <- data.frame(x = c(rnorm(100, 2, 4), rnorm(100, 6, 1), rnorm(100, 8, 2)),
group = c(rep('A', 100), rep('B', 100), rep('C', 100)),
date = rep(seq(as.Date("2020-01-01"), as.Date("2020-04-09"), 'days'), 3))

make_dygraph(dt = dat,
  value = 'x',
  date = 'date')

make_dygraph(dt = dat,
  value = 'x',
  groups = 'group',
  date = 'date')
make_lineplot

```r
make_lineplot(dt = iris,
value = 'Sepal.Length',
ggtheme = "minimal",
plot_palette = "plasma",
static = FALSE)
```

## Arguments

- **dt**: data.frame containing the data to plot.
- **value**: Name of the column to use as values on the y axis of the plot.
- **groups**: Name of the column containing the different groups.
- **binwidth**: Width of the histogram bins.
- **bins**: Number of bins. Overridden by binwidth. Defaults to 30.
- **scales**: From ggplot2::facet_wrap: Should scales be 'fixed', 'free', or free in one dimension ('free_x', 'free_y'). Default is 'fixed'.
- **ggtheme**: ggplot2 theme function to apply. Default is ggplot2::theme_minimal.
- **x_axis_label**: Label for the x axis.
- **plot_palette**: Character vector of hex codes specifying the colors to use on the plot.
- **plot_palette_generator**: Palette from the viridis package used in case plot_palette is unspecified or insufficient for the number of colors required.
- **static**: If TRUE (or if the dataset is over 10,000 rows), the output will be static ggplot chart instead of an interactive ggplotly chart. Default is FALSE.

## Value

A plotly-ized version of a grouped ggplot histogram plot.

## Examples

```r
make_histogram(dt = iris,
value = 'Sepal.Length',
groups = 'Species')
```

## Description

Create a line plot from a data frame through ggplotly
Usage

make_lineplot(
    dt,
    x,
    y,
    groups = NULL,
    faceted = FALSE,
    scales = "fixed",
    show_trend = FALSE,
    trend_method = "loess",
    ggtheme = "minimal",
    x_axis_label = NULL,
    y_axis_label = NULL,
    plot_palette = NULL,
    plot_palette_generator = "plasma",
    static = FALSE
)

Arguments

dt data.frame containing the data to plot.
    x Value on the x axis.
    y Value on the y axis.
    groups Name of the column containing the different groups.
    faceted If TRUE (default), each group will be plotted separately.
    scales From ggplot2::facet_wrap: Should scales be 'fixed', 'free', or free in one
        dimension ('free_x', 'free_y'). Default is 'fixed'.
    show_trend If TRUE, adds a ggplot2::geom_smooth() line to the plot.
    trend_method The method ggplot2::geom_smooth will use. Default is 'loess', which is a local
        polynomial regression fit
    ggtheme ggplot2 theme function to apply. Default is ggplot2::theme_minimal.
    x_axis_label Label for the x axis.
    y_axis_label Label for the y axis.
    plot_palette Character vector of hex codes specifying the colors to use on the plot.
    plot_palette_generator Palette from the viridis package, used in case plot_palette is unspecified or in-
        sufficient for the number of colors required.
    static If TRUE (or if the dataset is over 10,000 rows), the output will be static ggplot
        chart instead of an interactive ggplotly chart. Default is FALSE.

Value

A plotly-ized version of a grouped ggplot line plot.
Examples

```r
make_lineplot(dt = ggplot2::mpg,
          x = 'hwy',
          y = 'cty',
          groups = 'manufacturer',
          faceted = FALSE)

make_lineplot(dt = ggplot2::mpg,
          x = 'hwy',
          y = 'cty',
          groups = 'manufacturer',
          faceted = TRUE,
          scales = 'free')
```

Description

Create a raincloud plot from a data frame through ggplotly

Usage

```r
make_raincloud(
  dt,
  value,
  groups = NULL,
  adjust = 0.5,
  include_boxplot = TRUE,
  include_mean = FALSE,
  include_median = TRUE,
  force_all_jitter_obs = FALSE,
  ggtheme = "minimal",
  x_axis_label = NULL,
  plot_palette = NULL,
  plot_palette_generator = "plasma",
  static = FALSE
)
```

Arguments

- `dt` data.frame containing the data to plot.
- `value` Name of the column to use as values on the y axis of the plot.
- `groups` Name of the column containing the different groups.
- `adjust` Width of the kernel bins. The smaller the value, the higher the resolution of the density. For full details, see ?ggplot2::stat_density.
### make_scatterplot

Create a scatter plot from a data frame through ggplotly

**Description**

Create a scatter plot from a data frame through ggplotly

**Usage**

```r
make_scatterplot(
  dt,
  x,
  y,
  groups = NULL,
  faceted = FALSE,
  scales = "fixed",
  show_trend = FALSE,
  trend_method = "loess",
)```

**Examples**

```r
make_scatterplot(iris, Sepal.Length, Sepal.Width)
```
```r
make_scatterplot

    ggtheme = "minimal",
    x_axis_label = NULL,
    y_axis_label = NULL,
    plot_palette = NULL,
    plot_palette_generator = "plasma",
    static = FALSE)

Arguments

dt data.frame containing the data to plot.
x Value on the x axis.
y Value on the y axis.
groups Name of the column containing the different groups.
faceted If TRUE (default), each group will be plotted separately.
scales From ggplot2::facet_wrap: Should scales be 'fixed', 'free', or free in one di-
mension ('free_x', 'free_y'). Default is 'fixed'.
show_trend If TRUE, adds a ggplot2::geom_smooth() line to the plot.
trend_method The method ggplot2::geom_smooth will use. Default is 'loess', which is a local
                polynomial regression fit
ggtheme ggplot2 theme function to apply. Default is ggplot2::theme_minimal.
x_axis_label Label for the x axis.
y_axis_label Label for the y axis.
plot_palette Character vector of hex codes specifying the colors to use on the plot.
plot_palette_generator Palette from the viridis package, used in case plot_palette is unspecified or in-
sufficient for the number of colors required.
static If TRUE (or if the dataset is over 10,000 rows), the output will be static ggplot
        chart instead of an interactive ggplotly chart. Default is FALSE.

Value

A plotly-ized version of a grouped ggplot scatter plot.

Examples

make_scatterplot(dt = ggplot2::mpg,
    x = 'hwy',
    y = 'cty',
    groups = 'manufacturer',
    faceted = FALSE)

make_scatterplot(dt = ggplot2::mpg,
    x = 'hwy',
    y = 'cty',
    groups = 'manufacturer',
    static = TRUE)
```
make_violin

Create a violin plot from a data frame through ggplotly

Description

Create a violin plot from a data frame through ggplotly

make_violin

| make_violin | Create a violin plot from a data frame through ggplotly |

Description

Guessed a title out of function parameters

make_title

Guess a title out of function parameters

Description

Detects which make_* function is passed and builds a generic name based on its parameters.

Usage

make_title(fun, params)

Arguments

fun chronicle make_* function
params parameters for fun

Value

A generic title for the plot

Examples

make_title(fun = chronicle::make_barplot,
params = list(value = 'Amount',
bars = 'Country',
break_bars_by = 'Region'))

make_title(fun = chronicle::make_raincloud,
params = list(value = 'value',
groups = 'species'))

| make_title | Guess a title out of function parameters |

Description

Detects which make_* function is passed and builds a generic name based on its parameters.

Usage

make_title(fun, params)

Arguments

fun chronicle make_* function
params parameters for fun

Value

A generic title for the plot

Examples

make_title(fun = chronicle::make_barplot,
params = list(value = 'Amount',
bars = 'Country',
break_bars_by = 'Region'))

make_title(fun = chronicle::make_raincloud,
params = list(value = 'value',
groups = 'species'))

| make_title | Guess a title out of function parameters |

Description

Detects which make_* function is passed and builds a generic name based on its parameters.

Usage

make_title(fun, params)

Arguments

fun chronicle make_* function
params parameters for fun

Value

A generic title for the plot

Examples

make_title(fun = chronicle::make_barplot,
params = list(value = 'Amount',
bars = 'Country',
break_bars_by = 'Region'))

make_title(fun = chronicle::make_raincloud,
params = list(value = 'value',
groups = 'species'))
make_violin

Usage

make_violin(
  dt,
  value,
  groups = NULL,
  jitter = TRUE,
  ggtheme = "minimal",
  x_axis_label = NULL,
  y_axis_label = NULL,
  plot_palette = NULL,
  plot_palette_generator = "plasma",
  static = FALSE
)

Arguments

dt data.frame containing the data to plot.
value Name of the column to use as values on the y axis of the plot.
groups Name of the column containing the different groups.
jitter Whether to add the actual values of each observation over the violin plots. Only done when dt has 10,000 rows or less.

Arguments

ggtheme ggplot2 theme function to apply. Default is ggplot2::theme_minimal.
x_axis_label Label for the x axis.
y_axis_label Label for the y axis.
plot_palette Character vector of hex codes specifying the colors to use on the plot.
plot_palette_generator Palette from the viridis package used in case plot_palette is unspecified or insufficient for the number of colors required.
static If TRUE (or if the dataset is over 10,000 rows), the output will be static ggplot chart instead of an interactive ggplotly chart. Default is FALSE.

Value

A plotly-ized version of a ggplot violin plot.

Examples

make_violin(dt = ggplot2::mpg, value = 'hwy', groups = 'drv')
output_config  

Build the yaml output specification for an R Markdown

Description

Currently supported: prettydoc, ioslides, tufte, flexdashboard, slidy_presentation, html_document, html_notebook.

Usage

```r
output_config(
  output_format,
  title = NULL,
  author = NULL,
  include_date = TRUE,
  number_sections = FALSE,
  table_of_content = FALSE,
  table_of_content_depth = 1,
  fig_width = 8,
  fig_height = 5,
  plot_palette = NULL,
  plot_palette_generator = "plasma",
  rmdformats_theme = "downcute",
  prettydoc_theme = "leonids",
  docx_reference_file = NULL,
  pptx_reference_file = NULL,
  html_theme = "simplex",
  rticles_template = "arxiv_article",
  custom_output = NULL
)
```

Arguments

- `output_format`: The format of the R Markdown file.
- `title`: Title of the report. If NULL (default), no title will be added.
- `author`: Author of the report. If NULL (default), no author will be added.
- `include_date`: Whether or not to include the date as part of the header. Default is TRUE.
- `number_sections`: Whether or not to number the sections and subsections of the report.
- `table_of_content`: Whether or not to include a table of content at the beginning of the report.
- `table_of_content_depth`: The depth of sections and subsections to be displayed on the table of content.
- `fig_width`: Set the global figure width or the rmarkdown file.
- `fig_height`: Set the global figure height or the rmarkdown file.
plot_columns

- `plot_palette`: Character vector of hex codes to use on plots.
- `plot_palette_generator`: Palette from the viridis package used in case plot_palette is unspecified or insufficient for the number of colors required. Default value is 'plasma', and possible values are 'viridis', 'inferno', 'magma', 'plasma', 'cividis'.
- `rmdformats_theme`: The theme to be used for rmdformats outputs. Default is "downcute", and possible values are "downcute", "robobook", "material", "readthedown", "html_clean", "html_docco".
- `prettydoc_theme`: Name of the theme used on prettydoc. Default is "leonids", and possible values are "cayman", "tactile", "architect", "leonids", "hpstr".
- `docx_reference_file`: The path for a blank Microsoft Word document to use as template for the 'word_document' output.
- `pptx_reference_file`: The path for a blank Microsoft PowerPoint document to use as template for the 'powerpoint_presentation' output.
- `html_theme`: The theme to be used for html_document outputs. Default is "simplex".
- `rticles_template`: The theme to be used for rticles. Default is "arxiv_article".
- `custom_output`: [Experimental] This is to get output formats not currently supported. It should be a YAML element with the corresponding output.

### Value

The lines needed in the yaml header of an R Markdown file to render as the specified output type.

### Examples

```r
cat(output_config('prettydoc'))
cat(output_config('ioslides'))
```

### plot_columns

*Plot all columns of a table*

### Description

Make raincloud plots for each numerical variable on a table, and barplots for each categorical variable.

### Usage

```r
plot_columns(dt, by_column = NULL)
```
render_report

Arguments

  dt                Table to be plotted.
  by_column         Name of the column to use as groups for all the other plots

Value

  A list of plotly::ggplotly objects, one for each column of the table.

Examples

  chronicle::plot_columns(dt = iris, by_column = 'Species')

render_report          Render the report using all objects from the global environment

Description

  Render the report using all objects from the global environment

Usage

  render_report(
    report = "",
    output_format = "rmdformats",
    filename = paste("report", gsub(x = Sys.Date(), pattern = "-", replacement = ""), sep = "-_"),
    title = NULL,
    author = NULL,
    include_date = TRUE,
    directory = getwd(),
    keep_rmd = FALSE,
    render_reports = TRUE,
    number_sections = FALSE,
    table_of_content = FALSE,
    table_of_content_depth = 1,
    fig_width = 9,
    fig_height = 5,
    plot_palette = NULL,
    plot_palette_generator = "plasma",
    rmdformats_theme = "downcute",
    prettydoc_theme = "leonids",
    docx_reference_file = NULL,
    pptx_reference_file = NULL,
    rtticles_template = "arxiv_article",
    html_theme = "simplex",
    custom_output = NULL
  )
Arguments

report  Character string containing all the R Markdown chunks previously added (through chronicle::add_* functions.) Default is ", an empty report.

output_format  The format of the R Markdown file. Default is prettydoc. Currently supported: 'bookdown', 'github_document', 'html_document', 'html_notebook', 'ioslides', 'pagedown', 'powerpoint_presentation', 'pdf', 'prettydoc', 'rmdformats', 'rolldown', 'rticles', 'slidy_presentation', 'tufte_handout', 'tufte_html', 'word_document'. Also 'felxdashboard' and 'xaringan' technically compile, but the layout is stiff in flexdashboard and altogether incorrect in xaringan.

filename  The name of the .html file(s) created. If NULL (default), no author will be added.

title  Title of the report. If NULL (default), no title will be added.

author  Author of the report. If NULL (default), no author will be added.

include_date  Whether or not to include the date as part of the header. Default is TRUE.

directory  The directory in which to render the .html report

keep_rmd  Whether or not to keep the .Rmd file. Default is false.

render_reports  Whether or not to render the reports. Default is TRUE. Set render_reports = FALSE and keep_rmd = TRUE to only build the R Markdown files

number_sections  Whether or not to number the sections and subsections for the report.

table_of_content  Whether or not to include a table of content at the beginning of the report. Some formats does not allow overriding this.

table_of_content_depth  The depth of sections and subsections to be displayed on the table of content.

fig_width  Set the global figure width or the rmarkdown file.

fig_height  Set the global figure height or the rmarkdown file.

plot_palette  Character vector of hex codes to use on plots.

plot_palette_generator  Palette from the [viridis](https://cran.r-project.org/web/packages/viridis/vignettes/intro-to-viridis.html#the-color-scales) package used in case plot_palette is unspecified (or insufficient for the number of colors required.) Default value is 'plasma', and possible values are 'viridis', 'inferno', 'magma', 'plasma', 'cividis', 'mako', 'rocket', and 'turbo'.

rmdformats_theme  The theme to be used for [rmdformats](https://github.com/juba/rmdformats) outputs. Default is "downcute", and possible values are "downcute", "robobook", "material", "readthedown", "html_clean", "html_docco".

prettydoc_theme  Name of the theme used on [prettydoc](https://prettydoc.statr.me/themes.html). Default is "leonids", and possible values are "cayman", "tactile", "architect", "leonids", "hpstr".

The path for a blank Microsoft Word document to use as template for the 'word_document' output.

doctorx_reference_file

The path for a blank Microsoft PowerPoint document to use as template for the 'powerpoint_presentation' output.

rticles_template

The theme to be used for [rticles](https://github.com/rstudio/rticles). Default is "arxiv_article".

html_theme

The theme to be used for [html_document](https://www.datadreaming.org/post/r-markdown-theme-gallery/) outputs. Default is "simplex".

custom_output

[Experimental] A custom element for a yaml structure to specify as the output format of the R Markdown file. This is to get output formats not currently supported.#'

Value

Renders the report as an HTML file.

Examples

```r
# report_demo <- add_title(title = 'This is how a chronicle report looks', title_level = 1) %>%
# add_density(dt = iris, groups = 'Species', value = 'Sepal.Length', faceted = F) %>%
# add_boxplot(dt = iris, groups = 'Species', value = 'Sepal.Length') %>%
# add_barplot(dt = iris, bars = 'Species', value = 'Sepal.Length') %>%
# add_table(table = iris,
#   table_title = 'This is the iris dataset. Smells good!',
#   html_table_type = 'kable') %>%
# add_table(table = mpg,
#   table_title = 'And this is mpg',
#   html_table_type = 'DT') %>%
# render_report(report = report_demo,
#   title = 'Demo Output',
#   author = 'This is the author',
#   filename = 'demo_output',
#   output_format = 'prettydoc',
#   keep_rmd = TRUE)
```

**Report Columns**

**HTML interactive report detailing each column on a table**

**Description**

Creates an Rmarkdown report plotting each column of a dataset. Categorical columns are plotted in bar plots, and numerical columns are plotted in box plots. If 'by_column' is provided, these plots will be grouped by the values of that column.
Usage

```r
report_columns(
  dt,
  by_column = NULL,
  filename = NULL,
  output_format = "rmdformats",
  title = NULL,
  author = NULL,
  plot_palette = NULL,
  plot_palette_generator = "plasma",
  horizontal_bars = TRUE,
  sort_bars_value = TRUE,
  sort_bars_decreasingly = TRUE,
  rmdformats_theme = "downcute",
  prettydoc_theme = "leonids",
  number_sections = TRUE,
  table_of_content = TRUE,
  table_of_content_depth = 1,
  fig_width = 9,
  fig_height = 4,
  directory = getwd(),
  keep_rmd = FALSE,
  render_reports = TRUE
)
```

Arguments

dt           Table to be studied.
by_column     Name of the column to use as groups for all the other plots. Default is NULL.
filename      Name of the output file. If not supplied, a generic name will be created.
output_format The format of the R Markdown output. Default is 'rmdformats'.
title         Title of the report. If NULL (default), no title will be added.
author        Author of the report. Default is NULL.
plot_palette  Character vector of hex codes to use on plots.
plot_palette_generator
               Palette from the viridis package used in case plot_palette is unspecified (or insufficient for the number of colors required.) Default value is 'plasma', and possible values are 'viridis', 'inferno', 'magma', 'plasma', 'cividis'.
horizontal_bars Plot bars for categorical variables horizontally. Default is FALSE
sort_bars_value Sort the bars by value. Default is FALSE.
sort_bars_decreasingly Sort the bars decreasingly. Default is TRUE.
rmdformats_theme
The theme to be used for rmdformats outputs. Default is "downcute", and possible values are "downcute", "robobook", "material", "readthedown", "html_clean", "html_docco".

prettydoc_theme
Name of the theme used on prettydoc. Default is leonids.

number_sections
Whether or not to number the sections and subsections fo the report.

table_of_content
Whether or not to include a table of content at the beginning of the report.

table_of_content_depth
The depth of sections and subsections to be displayed on the table of content.

fig_width
Set the global figure width or the rmarkdown file.

fig_height
Set the global figure height or the rmarkdown file.

directory
The directory in which to render the .html report.

keep_rmd
Whether or not to keep the .Rmd file. Default is false.

render_reports
Whether or not to render the reports. Default is TRUE. Set render_reports = FALSE and keep_rmd = TRUE to only build the R Markdown files

Value
Creates an HTML file with a plot for each column on the given table: a box plot for each numerical variable, and a bar plot for each categorical variable.

Examples

# chronicle::report_columns(dt = iris,
# by_column = 'Species',
# horizontal_bars = TRUE,
# keep_rmd = TRUE)

rmd_title_level
Returns the count of '#' corresponding to a given title level

description
Returns the count of '#' corresponding to a given title level

Usage

rmd_title_level(level)

Arguments

level R Markdown title level
Value

'##', '###' and so on, depending on the title level

Examples

rmd_title_level(1)
rmd_title_level(3)

---

**set_classes**

*Change column classes with a named vector*

Description

Change column classes with a named vector

Usage

```r
set_classes(
  dt,
  character = NULL,
  integer = NULL,
  double = NULL,
  logical = NULL,
  factor = NULL
)
```

Arguments

- `dt` Table whose column types will be changed
- `character` The columns that will be coerced to character.
- `integer` The columns that will be coerced to integer.
- `double` The columns that will be coerced to double.
- `logical` The columns that will be coerced to logical.
- `factor` The columns that will be coerced to factor.

Value

Changes by reference the types of the specified columns

Examples

```r
library(chronicle)
iris_changed <- chronicle::set_classes(dt = iris,
  character = 'Species',
  integer = c('Sepal.Length', 'Sepal.Width'))
purrr::map_chr(iris_changed, class)
```
Index

add_barplot, 2
add_boxplot, 4
add_chunk, 5
add_code, 7
add_density, 8
add_dygraph, 9
add_histogram, 11
add_image, 13
add_lineplot, 14
add_quotes, 15
add_raincloud, 16
add_scatterplot, 18
add_table, 20
add_text, 21
add_title, 21
add_violin, 22
assemble_call, 23
check_cols, 24
file_extension, 25
make_barplot, 25
make_boxplot, 27
make_density, 28
make_dygraph, 29
make_histogram, 30
make_lineplot, 31
make_raincloud, 33
make_scatterplot, 34
make_title, 36
make_violin, 36
output_config, 38
plot_columns, 39
render_report, 40
report_columns, 42
rmd_title_level, 44
set_classes, 45