

Package: chronicle (via r-universe)

September 11, 2024

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

URL <https://github.com/pheymanss/chronicle>

BugReports <https://github.com/pheymanss/chronicle/issues>

Imports data.table, DT, dplyr, dygraphs, ggplot2, glue, knitr, plotly, prettydoc, purrr, readr, rlang, rmarkdown, rmdformats, scales, skimr, stats, viridis, zoo

Suggests bookdown, devtools, flexdashboard, pagedown, rolldown, rtables, tuftex, xaringan

LazyData true

VignetteBuilder knitr

RoxygenNote 7.1.1

Repository <https://pheymanss.r-universe.dev>

RemoteUrl <https://github.com/pheymanss/chronicle>

RemoteRef HEAD

RemoteSha e30b87e0f847f2cdb0303271a2a453d57b4dde1a

Contents

add_barplot	2
add_boxplot	4
add_chunk	6
add_code	7

add_density	8
add_dygraph	10
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

Index 46

add_barplot	<i>Add a bar plot to a chronicle report</i>
-------------	---

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,
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

```
html_report <- add_barplot(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

```
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.
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

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

 add_chunk

Transforms a function call into an Rmarkdown chunk

Description

Transforms a function call into an Rmarkdown chunk

Usage

```
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

```
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

```
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

```
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
)
```

 add_dygraph

Add a dygraph to a chronicle report

Description

Add a dygraph to a chronicle report

Usage

```
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.

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	<i>Add a histogram plot to a chronicle report</i>
---------------	---

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 <code>ggplot2::facet_wrap</code> : 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 <code>ggplot2::theme_minimal</code> .
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 <code>make_title()</code>
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

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

cat(html_report)
```

add_image

Add an image to a chronicle Rmarkdown report

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 (ie, 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	<i>Add a line plot to a chronicle report</i>
--------------	--

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

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.
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 <code>ggplot2::geom_smooth</code> will use. Default is 'loess', which is a local polynomial regression fit
ggtheme	ggplot2 theme function to apply. Default is <code>ggplot2::theme_minimal</code> .
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 <code>viridis</code> package, used in case <code>plot_palette</code> is unspecified or insufficient for the number of colors required.
lineplot_title	Title of the line plot section on the report. If NULL, <code>chronicle</code> will try to parse a generic title using <code>make_title()</code>
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

```
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

```
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

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

add_raincloud	<i>Add a raincloud plot to a chronicle report</i>
---------------	---

Description

Add a raincloud plot to a chronicle report

Usage

```
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,
```



```

    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 <code>?ggplot2::stat_density</code> .
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 <code>ggplot2::theme_minimal</code> .
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 <code>make_title()</code>
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.

Examples

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

cat(html_report)
```

add_scatterplot	<i>Add a scatter plot to a chronicle report</i>
-----------------	---

Description

Add a scatter plot to a chronicle report

Usage

```
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_table	<i>Add a table to a chronicle report</i>
-----------	--

Description

Add a table to a chronicle report

Usage

```
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

```
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

```
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

```
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.

jitter	Whether to add the actual values of each observation over the violin 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.
violin_title	Title of the violin 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 violin plot.

Examples

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

assemble_call	<i>Assembles a formatted function call from a function and a list of parameters</i>
---------------	---

Description

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

Usage

```
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	<i>Warns if any of the passed column names is missing from the data provided.</i>
------------	---

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	<i>Parse the file extension for each R Markdown output format</i>
----------------	---

Description

Currently supports:

Usage

```
file_extension(file_type)
```

Arguments

file_type R Markdown output formats.

Details

* rmdformats * prettydoc * bookdown * ioslides * tufte_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

```
file_extension(c('prettydoc', 'word_document', 'tufte_handout'))
```

make_barplot	<i>Create a bar plot from a data frame through ggplotly</i>
--------------	---

Description

Create a bar plot from a data frame through ggplotly

Usage

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

```

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

```

Arguments

<code>dt</code>	data.frame containing the data to plot.
<code>bars</code>	Name of the column containing the different groups.
<code>value</code>	Name of the columns to use as value on the y axis of the plot. If NULL (default), counts will be used.
<code>break_bars_by</code>	Name of the categorical variable used to break each bar
<code>up_to_n_bars</code>	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.
<code>horizontal</code>	Plot the bars horizontally. Default is FALSE.
<code>sort_by_value</code>	Sort the bars by value. Default is FALSE unless horizontal is TRUE.
<code>sort_decreasing</code>	Sort the values decreasingly. Default is TRUE, but <code>sort_by_value</code> must also be TRUE.
<code>ggtheme</code>	ggplot2 theme function to apply. Default is <code>ggplot2::theme_minimal</code> .
<code>x_axis_label</code>	Label for the x axis.
<code>y_axis_label</code>	Label for the y axis.
<code>plot_palette</code>	Character vector of hex codes specifying the colors to use on the plot.
<code>plot_palette_generator</code>	Palette from the viridis package used in case <code>plot_palette</code> is unspecified or insufficient for the number of colors required
<code>static</code>	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	<i>Create a box plot from a data frame through ggplotly</i>
--------------	---

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.

Value

A plotly-ized version of a ggplot box plot.

Examples

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

 make_density

Create a density plot from a data frame through ggplotly

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

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.
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'.
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 ggplot density plot.

Examples

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

make_dygraph	<i>Plot a time series from a data frame through dygraph's interactive html plot interface</i>
--------------	---

Description

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

Usage

```
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

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_histogram

Create a histogram plot from a data frame through ggplotly

Description

Create a histogram plot from a data frame through ggplotly

Usage

```
make_histogram(
  dt,
  value,
  groups = NULL,
  binwidth = NULL,
  bins = 30,
```

```

    scales = "fixed",
    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.
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

```

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

```

make_lineplot

Create a line plot from a data frame through ggplotly

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 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 line plot.

Examples

```
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')
```

make_raincloud	<i>Create a raincloud plot from a data frame through ggplotly</i>
----------------	---

Description

Create a raincloud plot from a data frame through ggplotly

Usage

```
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 <code>?ggplot2::stat_density</code> .

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 (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 raincloud plot.

Examples

```
make_raincloud(dt = iris, value = 'Sepal.Width')
make_raincloud(dt = iris, value = 'Sepal.Width', adjust = 1)
make_raincloud(dt = iris, value = 'Petal.Length', groups = 'Species', static = TRUE, adjust = 1)
make_raincloud(dt = iris, value = 'Sepal.Length', groups = 'Species', adjust = 1)
```

make_scatterplot	<i>Create a scatter plot from a data frame through ggplotly</i>
------------------	---

Description

Create a scatter plot from a data frame through ggplotly

Usage

```
make_scatterplot(
  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

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

```

```
faceted = TRUE,
scales = 'free')
```

make_title	<i>Guess a title out of function parameters</i>
------------	---

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	<i>Create a violin plot from a data frame through ggplotly</i>
-------------	--

Description

Create a violin plot from a data frame through ggplotly

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

<code>dt</code>	data.frame containing the data to plot.
<code>value</code>	Name of the column to use as values on the y axis of the plot.
<code>groups</code>	Name of the column containing the different groups.
<code>jitter</code>	Whether to add the actual values of each observation over the violin plots. Only done when <code>dt</code> has 10,000 rows or less.
<code>ggtheme</code>	ggplot2 theme function to apply. Default is <code>ggplot2::theme_minimal</code> .
<code>x_axis_label</code>	Label for the x axis.
<code>y_axis_label</code>	Label for the y axis.
<code>plot_palette</code>	Character vector of hex codes specifying the colors to use on the plot.
<code>plot_palette_generator</code>	Palette from the <code>viridis</code> package used in case <code>plot_palette</code> is unspecified or insufficient for the number of colors required
<code>static</code>	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

```
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_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](https://github.com/juba/rmdformats) outputs. Default is "downcute", and possible values are "downcute", "robobook", "material", "readthedown", "html_clean", "html_dooco".
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".
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](https://www.datadreaming.org/post/r-markdown-theme-gallery/) outputs. Default is "simplex".
rticles_template	The theme to be used for [rticles](https://github.com/rstudio/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

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

plot_columns	<i>Plot all columns of a table</i>
--------------	------------------------------------

Description

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

Usage

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

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	<i>Render the report using all objects from the global environment</i>
---------------	--

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,
  rticles_template = "arxiv_article",
  html_theme = "simplex",
  custom_output = NULL
)
```


Arguments

report	Character string containing all the R Markdown chunks previously added (through <code>chronicle::add_*</code> functions.) Default is "", an empty report.
output_format	The format of the R Markdown file. Default is <code>prettydoc</code> . Currently supported: <code>'bookdown'</code> , <code>'github_document'</code> , <code>'html_document'</code> , <code>'html_notebook'</code> , <code>'ioslides'</code> , <code>'pagedown'</code> , <code>'powerpoint_presentation'</code> , <code>'pdf'</code> , <code>'prettydoc'</code> , <code>'rmdformats'</code> , <code>'roll-down'</code> , <code>'rticles'</code> , <code>'slidy_presentation'</code> , <code>'tufte_handout'</code> , <code>'tufte_html'</code> , <code>'word_document'</code> . Also <code>'felxdashboard'</code> and <code>'xaringan'</code> technically compile, but the layout is stiff in <code>flexdashborad</code> and altogether incorrect in <code>xaringan</code> .
filename	The name of the <code>.html</code> file(s) created. If <code>NULL</code> (default), no author will be added.
title	Title of the report. If <code>NULL</code> (default), no title will be added.
author	Author of the report. If <code>NULL</code> (default), no author will be added.
include_date	Whether or not to include the date as part of the header. Default is <code>TRUE</code> .
directory	The directory in which to render the <code>.html</code> report
keep_rmd	Whether or not to keep the <code>.Rmd</code> file. Default is <code>false</code> .
render_reports	Whether or not to render the reports. Default is <code>TRUE</code> . Set <code>render_reports = FALSE</code> and <code>keep_rmd = TRUE</code> to only build the R Markdown files
number_sections	Whether or not to number the sections and subsections fo the report.
table_of_content	Whether or not to include a table fo 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 <code>rmarkdown</code> file.
fig_height	Set the global figure height or the <code>rmarkdown</code> file.
plot_palette	Character vector of hex codes to use on plots.
plot_palette_generator	Palette from the <code>[viridis](https://cran.r-project.org/web/packages/viridis/vignettes/intro-to-viridis.html#the-color-scales)</code> package used in case <code>plot_palette</code> is unspecified (or insufficient for the number of colors required.) Default value is <code>'plasma'</code> , and possible values are <code>'viridis'</code> , <code>'inferno'</code> , <code>'magma'</code> , <code>'plasma'</code> , <code>'cividis'</code> , <code>'mako'</code> , <code>'rocket'</code> , and <code>'turbo'</code> .
rmdformats_theme	The theme to be used for <code>[rmdformats](https://github.com/juba/rmdformats)</code> outputs. Default is <code>"downcute"</code> , and possible values are <code>"downcute"</code> , <code>"robobook"</code> , <code>"material"</code> , <code>"readthedown"</code> , <code>"html_clean"</code> , <code>"html_docco"</code> .
prettydoc_theme	Name of the theme used on <code>[prettydoc](https://prettydoc.statr.me/themes.html)</code> . Default is <code>"leonids"</code> , and ossible values are <code>"cayman"</code> , <code>"tactile"</code> , <code>"architect"</code> , <code>"leonids"</code> , <code>"hpstr"</code> .

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

```
# 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

```
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](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. 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 fo 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	<i>Returns the count of '#' corresponding to a given title level</i>
-----------------	--

Description

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

Usage

```
rmd_title_level(level)
```

Arguments

level	R Markdonw 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

```
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

```
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](#), 6
[add_code](#), 7
[add_density](#), 8
[add_dygraph](#), 10
[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