Package: geovizr (via r-universe)

October 24, 2024

Type Package

Title Support for Knitr (Quarto/Rmd)

Version 1.1.0

Description Provide support functions for Quarto and Rmd documents.

License MIT + file LICENSE

URL https://geovizr.nenuial.org

BugReports http://github.com/nenuial/geovizr/issues/

Imports bookdown, chunkhooks, dplyr, fs, geodata, geotools, ggeo, ggplot2, glue, here, janitor, jsonlite, knitr, lifecycle, magrittr, pdftools, purrr, quarto, readr, rlang, rmarkdown, rnotion, stringr, tibble, tidyr, xfun, yaml

Suggests downlit, gt, kableExtra, magick, ragg, tinytex, tufte

Remotes Nenuial/geodata, Nenuial/geotools, Nenuial/ggeo, Nenuial/rnotion

Config/Needs/website nenuial/geopkg

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Repository https://nenuial.r-universe.dev

RemoteUrl https://github.com/Nenuial/geovizr

RemoteRef HEAD

RemoteSha 8ee76e9ac7b58fe035d9b423ce703f25a6fbfd86

Contents

`book_skeleton` *Install book files*

Description

Install book files

Usage

```
book_skeleton(path)
```

Arguments

path	The path to intall the ressources to

Value

TRUE

Examples

```
book_skeleton("./")
```

---

eng_center_text *Center text with width option*

Description

[Deprecated]

Usage

```
eng_center_text(options)
```

Arguments

options	Chunk options

Examples

```
# Not run: knitr engine for Rmd/Quarto documents
eng_center_text(list(code = "Some text"))
```

---

eng_classic_box *Classic box environment*

Description

[Deprecated]

Usage

```
eng_classic_box(options)
```

Arguments

options	Chunk options

Examples

```
# Not run: knitr engine for Rmd/Quarto documents
eng_classic_box(list(code = "Some text"))
```

`eng_document_ref` *Provide a knit engine for AlXxx LaTeX commands*

Description

[Deprecated]

Usage

```
eng_document_ref(options)
```

Arguments

`options`	Chunk options

Value

A string

Examples

```
# Not run: knitr engine for Rmd/Quarto documents
eng_document_ref(list(type = "article", author = "Some author", code = "Some text"))
```

`eng_exam_questions` *Question in exam document*

Description

[Deprecated] Use quarto extension with solution filter.

Usage

```
eng_exam_questions(options)
```

Arguments

`options`	Chunk options

Examples

```
# Not run: knitr engine for Rmd/Quarto documents
eng_exam_questions(code = "Some text")
```

`eng_image_legend` *Insert figure environment*

Description

[Deprecated]

Usage

```
eng_image_legend(options)
```

Arguments

`options`	Chunk options

Examples

```
# Not run: knitr engine for Rmd/Quarto documents
eng_image_legend(list(
  img.cap = "Image caption",
  img.width = 80,
  img.author = "Someone"
))
```

`eng_latex_raw` *Raw LaTeX*

Description

Raw LaTeX

Usage

```
eng_latex_raw(options)
```

Arguments

`options`	Chunk options Only the code part is used

Examples

```
# Not run: knitr engine for Rmd/Quarto documents
eng_latex_raw(list(code = "Some text with \\emph{emphasis}"))
```

`eng_legal_list` *Create paralist*

Description

Create paralist

Usage

```
eng_legal_list(options)
```

Arguments

`options`	Chunk options

Examples

```
# Not run: knitr engine for Rmd/Quarto documents
eng_legal_list(code = "Some text")
```

`eng_wrap_figure` *Wrap image*

Description

[Deprecated]

Usage

```
eng_wrap_figure(options)
```

Arguments

`options`	Chunk options

Examples

```
# Not run: knitr engine for Rmd/Quarto documents
eng_wrap_figure(list(
  fig.cap = "Image caption",
  out.width = "80%",
  wrap.width = "20%"
))
```

`ggeo_knit_theme` *Plot theme for knitted documents*

Description

Plot theme for knitted documents

Usage

```
ggeo_knit_theme(..., theme = "doc", mode = "light")
```

Arguments

`...` Arguments passed on to `ggplot2::theme`

`line` all line elements (`element_line()`)

`rect` all rectangular elements (`element_rect()`)

`text` all text elements (`element_text()`)

`title` all title elements: plot, axes, legends (`element_text()`; inherits from `text`)

`aspect.ratio` aspect ratio of the panel

`axis.title,axis.title.x,axis.title.y,axis.title.x.top,axis.title.x.bottom,axis.title` labels of axes (`element_text()`). Specify all axes' labels (`axis.title`), labels by plane (using `axis.title.x` or `axis.title.y`), or individually for each axis (using `axis.title.x.bottom`, `axis.title.x.top`, `axis.title.y.left`, `axis.title.y.right`). `axis.title.*.*` inherits from `axis.title.*` which inherits from `axis.title`, which in turn inherits from `text`

`axis.text,axis.text.x,axis.text.y,axis.text.x.top,axis.text.x.bottom,axis.text.y.lef` tick labels along axes (`element_text()`). Specify all axis tick labels (`axis.text`), tick labels by plane (using `axis.text.x` or `axis.text.y`), or individually for each axis (using `axis.text.x.bottom`, `axis.text.x.top`, `axis.text.y.left`, `axis.text.y.right`). `axis.text.*.*` inherits from `axis.text.*` which inherits from `axis.text`, which in turn inherits from `text`

`axis.ticks,axis.ticks.x,axis.ticks.x.top,axis.ticks.x.bottom,axis.ticks.y,axis.ticks` tick marks along axes (`element_line()`). Specify all tick marks (`axis.ticks`), ticks by plane (using `axis.ticks.x` or `axis.ticks.y`), or individually for each axis (using `axis.ticks.x.bottom`, `axis.ticks.x.top`, `axis.ticks.y.left`, `axis.ticks.y.right`). `axis.ticks.*.*` inherits from `axis.ticks.*` which inherits from `axis.ticks`, which in turn inherits from `line`

axis.minor.ticks.x.top,axis.minor.ticks.x.bottom,axis.minor.ticks.y.left,axis.minor. minor tick marks along axes (element_line()). axis.minor.ticks.*.* inherit from the corresponding major ticks axis.ticks.*.*.

axis.ticks.length,axis.ticks.length.x,axis.ticks.length.x.top,axis.ticks.length.x.bo length of tick marks (unit)

axis.minor.ticks.length,axis.minor.ticks.length.x,axis.minor.ticks.length.x.top,axis length of minor tick marks (unit), or relative to axis.ticks.length when provided with rel().

axis.line,axis.line.x,axis.line.x.top,axis.line.x.bottom,axis.line.y,axis.line.y.lef lines along axes (element_line()). Specify lines along all axes (axis.line), lines for each plane (using axis.line.x or axis.line.y), or individually for each axis (using axis.line.x.bottom, axis.line.x.top, axis.line.y.left, axis.line.y.right). axis.line.*.* inherits from axis.line.* which inherits from axis.line, which in turn inherits from line

legend.background background of legend (element_rect(); inherits from rect)

legend.margin the margin around each legend (margin())

legend.spacing,legend.spacing.x,legend.spacing.y the spacing between legends (unit). legend.spacing.x & legend.spacing.y inherit from legend.spacing or can be specified separately

legend.key background underneath legend keys (element_rect(); inherits from rect)

legend.key.size,legend.key.height,legend.key.width size of legend keys (unit); key background height & width inherit from legend.key.size or can be specified separately

legend.key.spacing,legend.key.spacing.x,legend.key.spacing.y spacing between legend keys given as a unit. Spacing in the horizontal (x) and vertical (y) direction inherit from legend.key.spacing or can be specified separately.

legend.frame frame drawn around the bar (element_rect()).

legend.ticks tick marks shown along bars or axes (element_line())

legend.ticks.length length of tick marks in legend (unit)

legend.axis.line lines along axes in legends (element_line())

legend.text legend item labels (element_text(); inherits from text)

legend.text.position placement of legend text relative to legend keys or bars ("top", "right", "bottom" or "left"). The legend text placement might be incompatible with the legend's direction for some guides.

legend.title title of legend (element_text(); inherits from title)

legend.title.position placement of legend title relative to the main legend ("top", "right", "bottom" or "left").

legend.position the default position of legends ("none", "left", "right", "bottom", "top", "inside")

legend.position.inside A numeric vector of length two setting the placement of legends that have the "inside" position.

legend.direction layout of items in legends ("horizontal" or "vertical")

legend.byrow whether the legend-matrix is filled by columns (FALSE, the default) or by rows (TRUE).

legend.justification anchor point for positioning legend inside plot ("center" or two-element numeric vector) or the justification according to the plot area when positioned outside the plot

legend.justification.top,legend.justification.bottom,legend.justification.left,leger Same as legend.justification but specified per legend.position option.

legend.location Relative placement of legends outside the plot as a string. Can be "panel" (default) to align legends to the panels or "plot" to align legends to the plot as a whole.

legend.box arrangement of multiple legends ("horizontal" or "vertical")

legend.box.just justification of each legend within the overall bounding box, when there are multiple legends ("top", "bottom", "left", or "right")

legend.box.margin margins around the full legend area, as specified using margin()

legend.box.background background of legend area (element_rect(); inherits from rect)

legend.box.spacing The spacing between the plotting area and the legend box (unit)

panel.background background of plotting area, drawn underneath plot (element_rect(); inherits from rect)

panel.border border around plotting area, drawn on top of plot so that it covers tick marks and grid lines. This should be used with fill = NA (element_rect(); inherits from rect)

panel.spacing,panel.spacing.x,panel.spacing.y spacing between facet panels (unit). panel.spacing.x & panel.spacing.y inherit from panel.spacing or can be specified separately.

panel.grid,panel.grid.major,panel.grid.minor,panel.grid.major.x,panel.grid.major.y,p grid lines (element_line()). Specify major grid lines, or minor grid lines separately (using panel.grid.major or panel.grid.minor) or individually for each axis (using panel.grid.major.x, panel.grid.minor.x, panel.grid.major.y, panel.grid.minor.y). Y axis grid lines are horizontal and x axis grid lines are vertical. panel.grid.*.* inherits from panel.grid.* which inherits from panel.grid, which in turn inherits from line

panel.ontop option to place the panel (background, gridlines) over the data layers (logical). Usually used with a transparent or blank panel.background.

plot.background background of the entire plot (element_rect(); inherits from rect)

plot.title plot title (text appearance) (element_text(); inherits from title) left-aligned by default

plot.title.position,plot.caption.position Alignment of the plot title/subtitle and caption. The setting for plot.title.position applies to both the title and the subtitle. A value of "panel" (the default) means that titles and/or caption are aligned to the plot panels. A value of "plot" means that titles

and/or caption are aligned to the entire plot (minus any space for margins and plot tag).

`plot.subtitle` plot subtitle (text appearance) (`element_text()`; inherits from `title`) left-aligned by default

`plot.caption` caption below the plot (text appearance) (`element_text()`; inherits from `title`) right-aligned by default

`plot.tag` upper-left label to identify a plot (text appearance) (`element_text()`; inherits from `title`) left-aligned by default

`plot.tag.position` The position of the tag as a string ("topleft", "top", "topright", "left", "right", "bottomleft", "bottom", "bottomright") or a coordinate. If a coordinate, can be a numeric vector of length 2 to set the x,y-coordinate relative to the whole plot. The coordinate option is unavailable for `plot.tag.location = "margin"`.

`plot.tag.location` The placement of the tag as a string, one of `"panel"`, `"plot"` or `"margin"`. Respectively, these will place the tag inside the panel space, anywhere in the plot as a whole, or in the margin around the panel space.

`plot.margin` margin around entire plot (`unit` with the sizes of the top, right, bottom, and left margins)

`strip.background,strip.background.x,strip.background.y` background of facet labels (`element_rect()`; inherits from `rect`). Horizontal facet background (`strip.background.x`) & vertical facet background (`strip.background.y`) inherit from `strip.background` or can be specified separately

`strip.clip` should strip background edges and strip labels be clipped to the extend of the strip background? Options are `"on"` to clip, `"off"` to disable clipping or `"inherit"` (default) to take the clipping setting from the parent viewport.

`strip.placement` placement of strip with respect to axes, either "inside" or "outside". Only important when axes and strips are on the same side of the plot.

`strip.text,strip.text.x,strip.text.y,strip.text.x.top,strip.text.x.bottom,strip.text` facet labels (`element_text()`; inherits from `text`). Horizontal facet labels (`strip.text.x`) & vertical facet labels (`strip.text.y`) inherit from `strip.text` or can be specified separately. Facet strips have dedicated position-dependent theme elements (`strip.text.x.top`, `strip.text.x.bottom`, `strip.text.y.left`, `strip.text.y.right`) that inherit from `strip.text.x` and `strip.text.y`, respectively. As a consequence, some theme stylings need to be applied to the position-dependent elements rather than to the parent elements

`strip.switch.pad.grid` space between strips and axes when strips are switched (`unit`)

`strip.switch.pad.wrap` space between strips and axes when strips are switched (`unit`)

`complete` set this to `TRUE` if this is a complete theme, such as the one returned by `theme_grey()`. Complete themes behave differently when added to a ggplot object. Also, when setting `complete = TRUE` all elements will be set to inherit from blank elements.

	validate TRUE to run validate_element(), FALSE to bypass checks.
theme	Name of the theme to use. One of "ghibli_mononoke", "islamic_samarquand", "pomological_green", "pomological_red", "nord_blue", "swiss_red", "purple", "doc" or "oc_exams"
mode	One of light or dark

Value

A ggplot2 theme

See Also

ggeo::ggeotheme()

Examples

```
cars |>
  ggplot2::ggplot(ggplot2::aes(x = speed, y = dist)) +
  ggplot2::geom_point() +
  ggeo_knit_theme()
```

---

gvz_book *Book pdf*

Description

Book pdf

Usage

```
gvz_book(..., metadata = c())
```

Arguments

...	Arguments passed on to gvz_render_pdf_book
	template_path Path of the latex template
metadata	Additional pandoc metadata

Examples

```
# Not run: should be used as Rmd format in yaml frontmatter
#
# format: geovizir::gvz_book
```

gvz_book_resources *Get the path for the book resources*

Description

Get the path for the book resources

Usage

```
gvz_book_resources()
```

Value

A full path with the book resources, defaults to project root

Examples

```
# Not run: should be used in a book project
gvz_book_resources()
```

gvz_bs4_book *Book website*

Description

Book website

Usage

```
gvz_bs4_book(...)
```

Arguments

...	Arguments passed on to bookdown::bs4_book theme A named list or bslib::bs_theme() object. The default, bs4_book_theme(), resets the base font size to 1rem to make reading easier and uses a primary colour with greater contrast against the background. repo Either link to repository where book is hosted, used to generate view source and edit buttons or a list with repository base link, default branch, subdir and icon (see "Specifying the repository" in https://bookdown.org/yihui/bookdown/html.html#bootstrap4-style).

`template` Pandoc template to use for rendering. Pass `"default"` to use the bookdown default template; pass a path to use a custom template. The default template should be sufficient for most use cases. For advanced user only, in case you want to develop a custom template, we highly recommend to start from the default template: https://github.com/rstudio/bookdown/blob/master/inst/templates/bs4_book.html. Otherwise, some feature may not work anymore.

`footnotes_inline` By default, footnotes will be set inline and shown on hover. Set to `FALSE` to keep footnotes at the bottom of the page with links.

Examples

```
# Not run: should be used as Rmd format in yaml frontmatter
#
# format: geovizir::gvz_bs4_book
```

`gvz_document` *Standard pdf document*

Description

Standard pdf document

Usage

```
gvz_document(..., metadata = c())
```

Arguments

`...`	Arguments passed on to `gvz_render_pdf_document` `template_path` Path of the latex template
`metadata`	Additional pandoc metadata

Examples

```
# Not run: should be used as Rmd format in yaml frontmatter
#
# format: geovizir::gvz_document
```

gvz_doc_theme *Document ggeo theme*

Description

Document ggeo theme

Usage

```
gvz_doc_theme(...)
```

Arguments

...	Arguments passed on to ggeo::ggeotheme
theme	Name of the theme to use. One of "ghibli_mononoke", "islamic_samarquand", "pomological_green", "pomological_red", "nord_blue", "swiss_red", "purple", "doc" or "oc_exams"
main	One of main, main_latex or main_exa
plot	One of plot, plot_latex or plot_exa
mode	One of light or dark
base	A ggplot2 theme

Value

A ggplot2 theme

See Also

ggeo::ggeotheme()

Examples

```
cars |>
  ggplot2::ggplot(ggplot2::aes(x = speed, y = dist)) +
  ggplot2::geom_point() +
  gvz_doc_theme()
```

gvz_global_opts_chunk *Global chunk options*

Description

Set useful global chunk options.

Usage

```
gvz_global_opts_chunk()
```

Examples

```
gvz_global_opts_chunk()
```

gvz_html_or_pdf *Select output depending on format*

Description

Select output depending on format

Usage

```
gvz_html_or_pdf(html_output, pdf_output)
```

Arguments

html_output	Output for html format
pdf_output	Output for pdf format

Value

The selected output

Examples

```
# Not run: for use in Rmd/Quarto documents
gvz_html_or_pdf("The <b>HTML</b> output", "The \\textbf{LaTex} output")
```

gvz_knit_child_matu_oraux *Knit child for MF oral exams*

Description

Knit child for MF oral exams

Usage

```
gvz_knit_child_matu_oraux(...)
```

Arguments

...	Data that is passed to the fragment Must have *ID*, *Name*, *Subject* and *Question*

Value

Knitted document

gvz_latex_table *Generate latex code for table from tibble*

Description

Generate latex code for table from tibble

Usage

```
gvz_latex_table(tbl, md_cols = c())
```

Arguments

tbl	A tibble
md_cols	Columns to treat as markdown

Value

A collapsed string to use `asis`

Examples

```
# Not run: use in Rmd/Quarto document
table |>
  gvz_latex_table()
```

gvz_letter *LDDR pdf letter*

Description

LDDR pdf letter

Usage

```
gvz_letter(..., metadata = c())
```

Arguments

`...`	Arguments passed on to `gvz_letter_standard`
`metadata`	Additional pandoc metadata

Examples

```
# Not run: should be used as Rmd format in yaml frontmatter
#
# format: geovizir::gvz_letter
```

gvz_matu *Examen matu LDDR*

Description

Examen matu LDDR

Usage

```
gvz_matu(..., metadata = c())
```

Arguments

`...`	Arguments passed on to `gvz_render_pdf_document`
	`template_path` Path of the latex template
`metadata`	Additional pandoc metadata

Examples

```
# Not run: should be used as Rmd format in yaml frontmatter
#
# format: geovizir::gvz_matu
```

gvz_matu_oraux *Oraux matu fédérale*

Description

Oraux matu fédérale

Usage

```
gvz_matu_oraux(..., metadata = c())
```

Arguments

...	Arguments passed on to gvz_render_pdf_document
	template_path Path of the latex template
metadata	Additional pandoc metadata

Examples

```
# Not run: should be used as Rmd format in yaml frontmatter
#
# format: geovizir::gvz_matu_oraux
```

gvz_md_to_latex *Helper to convert from Markdown to LaTeX*

Description

Helper to convert from Markdown to LaTeX

Usage

```
gvz_md_to_latex(content)
```

Arguments

content	String to convert

Value

A string in LaTeX

Examples

```
gvz_md_to_latex("Some text with **bold** and *italic* markup")
```

gvz_quarto_get_presentation_yaml *Gets presentation yaml file from GeoKey*

Description

Gets presentation yaml file from GeoKey

Usage

```
gvz_quarto_get_presentation_yaml(path, prefix)
```

Arguments

path	Path where the yaml file gets downloaded to
prefix	Prefix for the path on GeoKey

Examples

```
# Not run: only work in quarto projects

gvz_quarto_get_presentation_yaml("Presentations/", "DF")
```

gvz_quarto_make_presentation_yaml *Make a simple yaml file for all presentations*

Description

Make a simple yaml file for all presentations

Usage

```
gvz_quarto_make_presentation_yaml()
```

Examples

```
# Not run: only work in quarto projects

gvz_quarto_make_presentation_yaml()
```

gvz_quarto_setup *Quarto chunk options setup*

Description

Set useful chunk options in Quarto documents.

Usage

```
gvz_quarto_setup()
```

Examples

```
gvz_quarto_setup()
```

gvz_render_diagrams *Render all tikz diagram in PDF format*

Description

Render all tikz diagram in PDF format

Usage

```
gvz_render_diagrams(source, out)
```

Arguments

source	Source file with tikz chunks
out	Path for diagram's to export to

Examples

```
# Not run: needs a file with tikz chunks
gvz_render_diagrams("file.Rmd", "out/directory/")
```

gvz_render_multiple *Render multiple files*

Description

Render multiple files

Usage

```
gvz_render_multiple(data, template, output_dir, merge = FALSE)
```

Arguments

data	A tibble Each row will generate a new document, the data in the columns is passed as variables to be used in the template.
template	A quarto template
output_dir	The directory to render the files to
merge	If a string, merges the multiple document in document with the corresponding name. Default is false.

Value

Generate pdf files

Examples

```
# Read a file containing the data used in the template
readr::read_csv("data_file.csv") |>
  gvz_render_multiple("template.qmd", "./")
```

gvz_reveal_theme *Theme function for reveal presentations*

Description

Theme function for reveal presentations

Usage

```
gvz_reveal_theme(...)
```

Arguments

`...`	Arguments passed on to `ggeo::ggeotheme`
	`theme` Name of the theme to use. One of "ghibli_mononoke", "islamic_samarquand", "pomological_green", "pomological_red", "nord_blue", "swiss_red", "purple", "doc" or "oc_exams"
	`main` One of `main`, `main_latex` or `main_exa`
	`plot` One of `plot`, `plot_latex` or `plot_exa`
	`mode` One of `light` or `dark`
	`base` A ggplot2 theme

Value

A ggplot2 theme

Examples

```
cars |>
  ggplot2::ggplot(ggplot2::aes(x = speed, y = dist)) +
  ggplot2::geom_point() +
  gvz_reveal_theme()
```

`gvz_ski` *Ski camp pdf document*

Description

Ski camp pdf document

Usage

```
gvz_ski(..., metadata = c())
```

Arguments

`...`	Arguments passed on to `gvz_render_pdf_document`
	`template_path` Path of the latex template
`metadata`	Additional pandoc metadata

Examples

```
# Not run: should be used as Rmd format in yaml frontmatter
#
# format: geovizir::gvz_ski
```

gvz_test *Test pdf*

Description

Test pdf

Usage

```
gvz_test(..., metadata = c())
```

Arguments

`...`	Arguments passed on to `gvz_render_pdf_document` `template_path` Path of the latex template
`metadata`	Additional pandoc metadata

Examples

```
# Not run: should be used as Rmd format in yaml frontmatter
#
# format: geovizir::gvz_test
```

gvz_test_folder *Test with cover page pdf*

Description

Test with cover page pdf

Usage

```
gvz_test_folder(..., metadata = c())
```

Arguments

`...`	Arguments passed on to `gvz_render_pdf_document` `template_path` Path of the latex template
`metadata`	Additional pandoc metadata

Examples

```
# Not run: should be used as Rmd format in yaml frontmatter
#
# format: geovizir::gvz_test_folder
```

knit_all *Knit all Rmd files in the Project*

Description

Knit all Rmd files in the Project

Usage

```
knit_all(path = here::here())
```

Arguments

path	The path to start at

Examples

```
# Not run: use in Rmd project
knit_all()
```

knit_letters *Custom knit function for multiple letters*

Description

Custom knit function for multiple letters

Usage

```
knit_letters(input, ...)
```

Arguments

input	The input file
...	Unused, for compatibility in yaml frontmatter

Examples

```
# Not run: this function is for use in Rmd yaml frontmatter
#
# knit: geovizir::knit_letters
```

knit_quiet *Custom Knit function for RStudio*

Description

Custom Knit function for RStudio

Usage

```
knit_quiet(input, ...)
```

Arguments

input	Input file
...	Unused, for compatibility in yaml frontmatter

Examples

```
# Not run: this function is for use in Rmd yaml frontmatter
#
# knit: geovizir::knit_quiet
```

tikz *Get the path for the tikz template stub*

Description

Get the path for the tikz template stub

Usage

```
tikz()
```

Value

A full path to the tikz template stub

Examples

```
tikz()
```

Index