

Package ‘rUM’

July 23, 2025

Title R Templates from the University of Miami

Version 2.1.0

Description This holds some r markdown and quarto templates and a template to create a research project in ``R Studio".

Depends R (>= 3.5)

Imports bookdown, conflicted, glue, gtsummary (>= 2.0.3), here, quarto, readr, rio, rlang, rmarkdown, roxygen2, stringr, table1, tidymodels, tidyverse, usethis

Suggests knitr, tinytex

License MIT + file LICENSE

URL <https://raymondbalise.github.io/rUM/>,
<https://github.com/RaymondBalise/rUM>

BugReports <https://github.com/RaymondBalise/rUM/issues>

Encoding UTF-8

RoxygenNote 7.3.2

VignetteBuilder knitr

NeedsCompilation no

Author Raymond Balise [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-9856-5901>>),
Gabriel Odom [aut] (ORCID: <<https://orcid.org/0000-0003-1341-4555>>),
Kyle Grealis [aut] (ORCID: <<https://orcid.org/0000-0002-9223-8854>>),
Francisco Cardozo [aut] (ORCID:
<<https://orcid.org/0000-0002-1925-4954>>),
Frank Gutierrez [ctb] (ORCID: <<https://orcid.org/0009-0005-9742-6992>>)

Maintainer Raymond Balise <balise@miami.edu>

Repository CRAN

Date/Publication 2025-03-21 12:40:02 UTC

Contents

make_project	2
write_notes	3
write_quarto	4
write_readme	5
write_scss	6
Index	7

make_project	<i>Make an Analysis Project</i>
--------------	---------------------------------

Description

This function makes an R project that includes an analysis.Rmd or analysis.qmd file using the conflicted and tidyverse packages. This project automatically includes an aggressive .gitignore which is designed to help protect against leaking data (with protected health information), a starter bibliography file called "references" (in standard .bib format), and a stock Citation Style Language (.csl) file for the New England Journal of Medicine.

Usage

```
make_project(  
  path,  
  type = c("Quarto (analysis.qmd)", "R Markdown (analysis.Rmd)"),  
  example = FALSE,  
  vignette = FALSE,  
  overwrite = FALSE,  
  openInteractive = TRUE  
)
```

Arguments

path	Path automatically set by research_project.dcf (see ./rstudio/templates/project/)
type	Choose between "Quarto (analysis.qmd)" or "R Markdown (analysis.Rmd)"
example	Will the analysis file include an example table/figure?
vignette	Will the analysis file be saved as a package vignette?
overwrite	Will an existing RStudio project be overwritten? This is needed for for Posit.Cloud. You will be prompted to confirm this option.
openInteractive	Should this new project be opened in a new RStudio window? Defaults to TRUE. NOTE: this option exists to prevent RStudio from opening two duplicate versions of the new project when this function is executed from RStudio menus. MODIFY WITH CAUTION.

Details

Behind the scenes, this function used by `research_project.dcf` when a user selects New project... > New Directory > rUM Research Project Template within the RStudio IDE. See `./rstudio/templates/project/`.

Value

Returns nothing. See description above.

Examples

```
## Not run:
# This makes a project with an example Quarto paper in the project's folder.
make_project(path = "~/test", type = "Quarto (analysis.qmd)",
             example = TRUE, vignette = TRUE)

# make_project() allows abbreviations on the project type: "Q" for Quarto or "R" for R Markdown
make_project(path = "~/test_project", "Q", TRUE, TRUE)

# This makes a project with an example R Markdown paper in the project's folder.
make_project(path = "~/test", type = "R Markdown (analysis.Rmd)",
             example = TRUE, vignette = TRUE)

# This makes a project with an example paper in the project's folder.
make_project(path = "~/test_project", "R", example = TRUE)

## End(Not run)
```

write_notes

Create a project README file

Description

This function streamlines project documentation by creating and managing both `README.md` and `dated_progress_notes.md` files. It provides interactive prompts for existing files and maintains consistent project documentation structure.

Usage

```
write_notes(path = here::here())
```

Arguments

`path` The destination directory for the progress notes file. Defaults to `here::here()`.

Details

The `dated_progress_notes.md` file is initialized with the current date and is designed to help track project milestones chronologically. If the progress notes file already exists, the function will stop and warn the user.

Value

Creates a chronological project progress notes tracker

Examples

```
# Create new progress note file in temporary directory
tmp <- tempdir()
write_notes(path = tmp)
```

write_quarto

Create a New Quarto Document

Description

This function creates a new Quarto document (.qmd file) complete with a useful header.

Usage

```
write_quarto(filename = NULL, path = here::here())
```

Arguments

filename	Character string. The name of the file without the '.qmd' extension. Only letters, numbers, hyphens, and underscores are allowed.
path	Character string. Directory where the file will be created. Defaults to the current project's base directory.

Value

Opens file after creating the Quarto document.

Examples

```
## Not run:
# Create a new Quarto document
write_quarto(filename = "data_cleaning", path = tempdir())

## End(Not run)
```

write_readme	<i>Create a project README file</i>
--------------	-------------------------------------

Description

This function streamlines project documentation by creating and managing a README.md file. It provides interactive prompts for existing files and maintains consistent project documentation structure.

Usage

```
write_readme(path = here::here())
```

Arguments

path	The destination directory for the README file. Defaults to <code>here::here()</code> .
------	--

Details

The README.md template includes structured sections for:

- Project description (study name, principal investigator, author)
- Project setup steps for reproducibility
- File and directory descriptions
- Miscellaneous project notes

If the README file already exists, the function will stop and warn the user. The templates include example documentation that can be modified to suit project needs.

Value

Creates a comprehensive README template for project documentation.

Examples

```
# Create new README in temporary directory
tmp <- tempdir()
write_readme(path = tmp)
```

write_scss

Create a Quarto SCSS file

Description

This function creates the .scss file so that any Quarto project can be easily customized with SCSS styling variables, mixins, and rules. When creating additional SCSS files beyond the default custom.scss, the function will attempt to update the YAML of your Quarto document while preserving any existing SCSS configurations.

Usage

```
write_scss(name = "custom", path = here::here(), add_to_yaml = FALSE)
```

Arguments

name	The name of the scss file without extension. Default name is "custom".
path	The destination directory for the SCSS file. Defaults to here::here().
add_to_yaml	Boolean. Add the new SCSS filename to YAML structure.

Details

The function includes a robust YAML handling mechanism that:

- Preserves existing YAML structure and indentation
- Safely adds new SCSS files without disrupting existing ones
- Provides manual instructions if the YAML structure differs from expected

For more information on customizing Quarto documents with SCSS, please refer to <https://quarto.org/docs/output-formats/html-themes.html#customizing-themes>, <https://quarto.org/docs/output-formats/html-themes-more.html>, and https://github.com/twbs/bootstrap/blob/main/scss/_variables.scss will provide you with over 1500 lines of SCSS variables.

Value

A .scss file to customize Quarto styling. If name is not "custom", the function will also attempt to update the Quarto document's YAML to include the new SCSS file while preserving any existing SCSS configurations.

Examples

```
# Create the default custom.scss in a temporary directory
tmp <- tempdir()
write_scss(name = "custom", path = tmp)

# Add another SCSS file and update YAML in the temporary directory
write_scss(name = "special_theme", path = tmp)
```

Index

`make_project`, [2](#)

`write_notes`, [3](#)

`write_quarto`, [4](#)

`write_readme`, [5](#)

`write_scss`, [6](#)