

Package ‘sketchy’

April 20, 2024

Type Package

Title Create Custom Research Compendiums

Version 1.0.3

Maintainer Marcelo Araya-Salas <marcelo.araya@ucr.ac.cr>

Description Provides functions to create and manage research compendiums for data analysis. Research compendiums are a standard and intuitive folder structure for organizing the digital materials of a research project, which can significantly improve reproducibility. The package offers several compendium structure options that fit different research project as well as the ability of duplicating the folder structure of existing projects or implementing custom structures. It also simplifies the use of version control.

License GPL (>= 2)

Imports knitr, stringi, crayon, packrat, utils, git2r, xaringanExtra, rmarkdown, remotes, cli

Depends R (>= 3.5.0)

LazyData TRUE

URL <https://github.com/maRce10/sketchy>

BugReports <https://github.com/maRce10/sketchy/issues>

NeedsCompilation no

Suggests testthat, formatR

RoxygenNote 7.3.1

Repository CRAN

Language en-US

Author Marcelo Araya-Salas [aut, cre]
(<<https://orcid.org/0000-0003-3594-619X>>),
Andrea Yure Arriaga Madrigal [aut]

Date/Publication 2024-04-20 15:12:33 UTC

R topics documented:

add_to_gitignore	2
compendiums	3
load_packages	4
make_compendium	5
print_skeleton	7
sketchy	8

Index	9
--------------	----------

add_to_gitignore	<i>Add entries to gitignore</i>
------------------	---------------------------------

Description

add_to_gitignore adds entries to gitignore based on file extension or file size

Usage

```
add_to_gitignore(add.to.gitignore = FALSE, cutoff = NULL, extension = NULL, path = ".")
```

Arguments

add.to.gitignore	Logical to control if files are added to 'gitignore' or just printed on the console.
cutoff	Numeric. Defines the file size (in MB) cutoff used to find files (i.e. only files above the threshold would returned). 99 (MB) is recommended when hosting projects at github as the current file size limit is 100 MB.
extension	Character string to define the file extension of the files to be searched for.
path	Path to the project directory. Default is current directory.

Details

The function can be used to avoid conflicts when working with large files or just avoid adding non-binary files to remote repositories. It mostly aims to simplify spotting/excluding large files. Note that file names can be manually added to the '.gitignore' file using a text editor.

Value

Prints the name of the files matching the searching parameters. If add.to.ignore = TRUE the files matching the search parameters ('cutoff' and/or 'extension') would be added 'gitignore' (a file used by git to exclude files form version control, including adding them to github).

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Willink, B., Arriaga, A. (2020), sketchy: research compendiums for data analysis in R. R package version 1.0.2.

See Also

[compendiums](#), [make_compendium](#)

Examples

```
{
data(compendiums)

make_compendium(name = "my_compendium", path = tempdir(),
  format = "basic", force = TRUE)

# save a file
write.csv(iris, file.path(tempdir(), "my_compendium", "iris.csv"))

# add the file to gitignore
add_to_gitignore(add.to.gitignore = TRUE,
  path = file.path(tempdir(), "my_compendium"), extension = "csv")
}
```

compendiums

List with compendium skeletons

Description

compendiums is a list containing the format of 14 different project folder skeletons. For each format 3 elements are provided: ‘\$skeleton’ (folder structure), ‘\$comments’ and ‘\$info’ (reference to the original source).

Usage

```
data(compendiums)
```

Format

A list with 14 compendium formats:

basic basic sketchy format

figures similar to basic, but including output/figures folders

project_template following Kenton White’s [ProjectTemplate](#)

pakillo following Francisco Rodriguez-Sanchez’ [template](#)

boettiger following Carl Boettiger’s [blog](#)

wilson following Wilson et al. (2017) format
small_compendium following Marwick et al (2018) small compendium format
medium_compendium following Marwick et al (2018) medium compendium format
large_compendium following Marwick et al (2018) large compendium format
vertical following Vuorre *et al.* (2018) **R package vertical**
rrtools following Marwick (2018) (**R package rrtools**)
rdir following folder structure described on at a r-dir blog post (although seems like it was removed)
workflow following Blischak *et al.* (2019) **R package workflow**
sketchy same skeleton than 'basic' but including a custom Rmarkdown for documenting data analyses

References

Blischak, J. D., Carbonetto, P., & Stephens, M. 2019. *Creating and sharing reproducible research code the workflow way*. F1000Research, 8.
 Marwick, B. 2018. *rrtools: Creates a reproducible research compendium*.
 Marwick, B., Boettiger, C., & Mullen, L. 2018. *Packaging data analytical work reproducibly using R (and friends)*. The American Statistician, 72(1), 80-88.
 Vuorre, Matti, and Matthew J. C. Crump. 2020. *Sharing and Organizing Research Products as R Packages*. PsyArXiv. January 15.
 Wilson G, Bryan J, Cranston K, Kitzes J, Nederbragt L. & Teal, T. K.. 2017. *Good enough practices in scientific computing*. PLOS Computational Biology 13(6): e1005510.

load_packages	<i>Install and load packages</i>
---------------	----------------------------------

Description

load_packages installs and loads packages from different repositories.

Usage

```
load_packages(packages, quiet = FALSE, upgrade.deps = FALSE)
```

Arguments

packages Character vector with the names of the packages to be installed. The vector names indicate the repositories from which packages will be installed. If no name is included CRAN will be used as the default repository. Available repositories are: 'cran', 'github', 'gitlab', 'bitbucket' and 'bioconductor'. Note that for 'github', 'gitlab' and 'bitbucket' the string must include the user name in the form 'user/package'.

quite	Logical argument to control if package startup messages are printed. Default is FALSE (messages are printed).
upgrade.deps	Logical argument to control if package dependencies are upgraded. Default is FALSE.

Details

The function installs and loads packages from different repositories.

Value

No object is returned.

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Willink, B., Arriaga, A. (2020), sketchy: research compendiums for data analysis in R. R package version 1.0.2.

See Also

[compendiums](#), [make_compendium](#)

Examples

```
## Not run:  
load_packages(packages = c("kableExtra", bioconductor = "ggtree",  
github = "maRce10/Rraven"), quite = TRUE)  
  
## End(Not run)
```

make_compendium	<i>Generate folder structures for research compendiums</i>
-----------------	--

Description

make_compendium generates the folder structure of a research compendium.

Usage

```
make_compendium(name = "research_compendium", path = ".", force = FALSE,  
format = "basic", packrat = FALSE,  
git = FALSE, clone = NULL, readme = TRUE, Rproj = FALSE)
```

Arguments

name	character string: the research compendium directory name. No special characters should be used. Default is "research_compendium".
path	Path to put the project directory in. Default is current directory.
force	Logical controlling whether existing folders with the same name are used for setting the folder structure. The function will never overwrite existing files or folders.
format	A character vector of length 1 with the name of the built-in compendiums available in the example object 'compendiums' (see compendiums for available formats). Default is 'basic'. Alternatively, it can be a character vector with 2 or more elements with the names of the folders and subfolders to be included.
packrat	Logical to control if packrat is initialized (<code>packrat::init()</code>) when creating the compendium. Default is FALSE.
git	Logical to control if a git repository is initialized (<code>git2r::init()</code>) when creating the compendium. Default is FALSE.
clone	Path to a directory containing a folder structure to be cloned. Default is NULL. If provided 'format' is ignored. Folders starting with <code>^\.git ^\.Rproj.user ^\.\.Rcheck</code> will be ignored.
readme	Logical. Controls if a readme file (in Rmd format) is added to the project. The file has predefined fields for documenting objectives and current status of the project. Default is TRUE.
Rproj	Logical. If TRUE a R project is created (i.e. a .Rproj file is saved in the main project directory).

Details

The function takes predefined folder structures to generate the directory skeleton of a research compendium.

Value

A folder skeleton for a research compendium. In addition the structure of the compendium is printed in the console. If the compendium format includes a "manuscript" or "doc(s)" folder the function saves a manuscript template in Rmarkdown format ("manuscript.Rmd"), a BibTex file ("example_library.bib", for showing how to add citations) and APA citation style file ("apa.csl") inside that folder.

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Willink, B., Arriaga, A. (2020), sketchy: research compendiums for data analysis in R. R package version 1.0.2.

Marwick, B., Boettiger, C., & Mullen, L. (2018). Packaging Data Analytical Work Reproducibly Using R (and Friends). *American Statistician*, 72(1), 80-88.

Alston, J., & Rick, J. (2020). A Beginners Guide to Conducting Reproducible Research.

See Also

[compendiums](#), [print_skeleton](#)

Examples

```
{
  data(compendiums)

  make_compendium(name = "mycompendium", path = tempdir(), format = "basic",
  force = TRUE)
}
```

print_skeleton	<i>Print folder structures</i>
----------------	--------------------------------

Description

print_skeleton prints the folder structure of a research compendium.

Usage

```
print_skeleton(path = ".", comments = NULL, folders = NULL)
```

Arguments

path	path to the directory to be printed. Default is current directory.
comments	A character string with the comments to be added to each folder in the graphical representation of the folder skeleton printed on the console.
folders	A character vector including the name of the sub-directories of the project.

Details

The function prints the folder structure of an existing project.

Value

The folder skeleton is printed in the console.

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Willink, B., Arriaga, A. (2020), sketchy: research compendiums for data analysis in R. R package version 1.0.2.

See Also

[compendiums](#), [make_compendium](#)

Examples

```
{
  data(compendiums)

  make_compendium(name = "my_other_compendium", path = tempdir(), format = "basic")

  print_skeleton(path = file.path(tempdir(), "mycompendium"))
}
```

sketchy

sketchy: create custom research compendiums

Description

‘sketchy’ is intended to facilitate the use of research compendiums for data analysis in the R environment. Standard research compendiums provide a easily recognizable means for organizing digital materials, allowing other researchers to inspect, reproduce, and build upon that research.

Details

The main features of the package are:

- Creation of (customized) folder structure
- Simplify the inclusion of big data files with version control software and online collaborative platforms (e.g. github)

Author(s)

Marcelo Araya-Salas, Beatriz Willink & Andrea Arriaga

Maintainer: Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

See Also

Useful links:

- <https://github.com/maRce10/sketchy>
- Report bugs at <https://github.com/maRce10/sketchy/issues>

Index

* datasets

- compendiums, 3
- add_to_gitignore, 2
- compendiums, 3, 3, 5–8
- load_packages, 4
- make_compendium, 3, 5, 5, 8
- print_skeleton, 7, 7
- sketchy, 8
- sketchy-package (sketchy), 8