Package: repro (via r-universe)

September 9, 2024

Type Package

Title Automated Setup of Reproducible Workflows and their Dependencies

Version 0.1.0

Description Set up components for reproducible workflows, quickly and painlessly. Apply best practises from [Peikert & Brandmaier (2019)](<https://psyarxiv.com/8xzqy/>), [Van Lissa et. al 2020](<https://osf.io/zcvbs/>) or [The Turing Way Community](<doi.org:10.5281/zenodo.3233986>) easily in your own projects. Based upon the great [`usethis`-package](<https://github.com/r-lib/usethis>).

License GPL-3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.1

Depends R (>= 3.6.0)

Imports usethis (>= 1.5.1), xfun, fs, glue, stringr, yaml, rstudioapi, rmarkdown, credentials, gh (>= 1.2.1)

URL https://github.com/aaronpeikert/repro

BugReports https://github.com/aaronpeikert/repro/issues

Suggests testthat (>= 2.1.0), covr, curl, withr, rlang, knitr, cli, stringi, remotes, gert

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

RemoteUrl https://github.com/aaronpeikert/repro

RemoteRef HEAD

RemoteSha 54666ed2cd4f6a774ffe23542b00838408fa810b

Contents

`automate` *Automate the use of Docker & Make*

Description

`automate()` & friends use yaml metadata from RMarkdowns to create `Dockerfile`'s and `Makefile`'s. It should be clear which is created by `automate_docker()` & which by `automate_make()`.

Usage

```
automate(path = ".")

automate_make(path = ".")

automate_publish(path = ".")

automate_docker(path = ".")
```

Arguments

`path`	Where should we look for RMarkdowns?

See Also

`automate_load_packages()`, `automate_load_data()`, `automate_load_scripts()`

automate_load *Access repro YAML Metadata from within the document*

Description

- `automate_load_packages()` loads all packages listed in YAML via `library()`
- `automate_load_scripts()` registeres external scripts via `knitr::read_chunk()`
- `automate_load_data()` reads in the data from the yaml with abitrary functions

Usage

```
automate_load_packages()

automate_load_scripts()

automate_load_data(data, func, ...)
```

Arguments

`data`	How is the entry in the YAML called? It will be the name of the object.
`func`	Which function should be used to read in the data? Its first argument must be the path to the file.
`...`	Further arguments supplied to `func`.

Value

`automate_load_packages()` & `automate_load_scripts()` do not return anything. `automate_load_data()` returns the data.

check *Check System Dependencies*

Description

Check if a dependency is installed and if not it recommends how to install it depending on the operating system. Most importantly it checks for `git`, `make` & `docker`. And just for convenience of the installation it checks on OS X for `Homebrew` and on Windows for `Chocolately`.

Usage

```
check_docker()

check_make()

check_git()

check_brew()

check_choco()

check_ssh(install = getOption("repro.install"))

check_github_token(install = getOption("repro.install"))

check_github_token_access()

check_github_ssh()

check_github(auth_method = "token")

check_renv(install = getOption("repro.install"))

check_targets(install = getOption("repro.install"))

check_worcs(install = getOption("repro.install"))
```

Arguments

install	Should something be installed? Defaults to "ask", but can be TRUE/FALSE.
auth_method	How do you want to authenticate with GitHub? Either "token" (default) or "ssh".

See Also

Other checkers: check_package(), has_package(), has, uses

---

check_package *Check if package exists*

Description

Check if package exists

Usage

```
check_package(pkg, install = getOption("repro.install"), github = NULL)
```

Arguments

pkg	Which package are we talking about?
install	Should we install the package in case its missing?
github	A github username/package from which the package is installed. If NULL (the default) CRAN (or whatever repo) you have set is used.

See Also

Other checkers: check, has_package(), has, uses

---

current_hash *Report Current Hash*

Description

Find out what the currently checked out commit is and report its respective hash.

Usage

```
current_hash(length = 7L, backend = NULL)
```

Arguments

length	The length to which the hash is cut.
backend	Can be either of NULL, "gert", or "git". If NULL Git is used with preference and gert as a fallback.

---

docker *Use Docker*

Description

Add or modify the Dockerfile in the current project.

Usage

```
use_docker(
  rver = NULL,
  stack = "verse",
  date = Sys.Date(),
  file = "Dockerfile",
  open = TRUE
)

use_dockerignore(file, open = TRUE)
```

Arguments

rver	Which r version to use, defaults to current version.
stack	Which stack to use, possible values are c("r-ver", "rstudio", "tidyverse", "verse", "geospatial").
date	Which date should be used for package instalation, defaults to today.
file	Which file to save to
open	Open the newly created file for editing? Happens in RStudio, if applicable, or via utils::file.edit() otherwise.

---

docker_windows_path *Generate the weird format for docker on windows.*

Description

Generate the weird format for docker on windows.

Usage

```
docker_windows_path(path = NULL, todo = interactive())
```

Arguments

path	Some windows path. If NULL the current directory.
todo	Should an actionable advice be given. Defaults to TRUE.

---

has *Check System Dependencies*

Description

Corresponding functions to check, but intended for programatic use.

Usage

```
has_make(silent = TRUE, force_logical = TRUE)

has_git(silent = TRUE, force_logical = TRUE)

has_docker(silent = TRUE, force_logical = TRUE)

has_docker_running(silent = TRUE, force_logical = TRUE)

has_choco(silent = TRUE, force_logical = TRUE)
```

```
has_brew(silent = TRUE, force_logical = TRUE)

has_ssh(silent = TRUE, force_logical = TRUE)

has_github_token(silent = TRUE, force_logical = TRUE)

has_github_token_access(silent = TRUE, force_logical = TRUE)

has_github_ssh(silent = TRUE, force_logical = TRUE)

has_github(silent = TRUE, force_logical = TRUE)

has_renv(silent = TRUE, force_logical = TRUE)

has_targets(silent = TRUE, force_logical = TRUE)

has_worcs(silent = TRUE, force_logical = TRUE)

has_gert(silent = TRUE, force_logical = TRUE)
```

Arguments

silent	Should a message be printed that informs the user?
force_logical	Should the return value be passed through isTRUE?

See Also

Other checkers: check_package(), check, has_package(), uses

---

has_package *Check if package exists*

Description

Check if package exists

Usage

```
has_package(pkg, silent = TRUE, force_logical = TRUE)
```

Arguments

pkg	Which package are we talking about?
silent	Should a message be printed that informs the user?
force_logical	Should the return value be passed through isTRUE?

See Also

Other checkers: check_package(), check, has, uses

---

make *Use Make*

Description

Add a (GNU-)Makefile(s) with special emphasis on the use of containers.

Usage

```
use_make(
  docker = FALSE,
  publish = FALSE,
  singularity = FALSE,
  torque = FALSE,
  open = TRUE
)

use_make_docker(file, use_docker = TRUE, dockerignore = TRUE, open = FALSE)

use_make_singularity(file, use_singularity = TRUE, open = FALSE)

use_make_publish(file, open = FALSE)
```

Arguments

docker	If true or a path a setup is created that can partially send make commands to a Docker container.
publish	Should the Makefile_publish also be created?
singularity	If true or a path a setup is created that can partially send make commands to a Singularity container (which requires the Dockerimage)
torque	If true a or a path setup is created that can partially send make comands to a TORQUE job scheduler. Especially usefull in combination with a Singularity container.
open	Open the newly created file for editing? Happens in RStudio, if applicable, or via utils::file.edit() otherwise.
file	Path to the file that is to be created.
use_docker	If true use_docker() is called.
dockerignore	If true a .dockerignore file is created.
use_singularity	If true use_singularity() is called.

`reproduce` *Find entrypoint for analysis and suggest how to reproduce it.*

Description

`reproduce()` inspects the files of a project and suggest a way to reproduce the project.

Usage

```
reproduce(fun, ..., path = ".", cache = FALSE, silent = FALSE)

reproduce_make(path, cache = FALSE, silent = FALSE)
```

Arguments

`fun`	a function that inspects `dir` and advises on how to reproduce the analysis. Defaults to reproduce_funs.
`...`	more functions like `fun`.
`path`	Were should I look for entrypoints?
`cache`	Default is `FALSE`. Some entrypoints have a cache, which you probably do not want to use in a reproduction.
`silent`	Should a message be presented?

Details

`reproduce()` walks through a list of functions that check for a specific entrypoint. As soon as a function returns a possible entrypoint the search stops. If no function is supplied the standard list of reproduce_funs is used.

Value

Returns invisibly the command users should use to reproduce.

See Also

reproduce_funs

reproduce_funs *A list of functions to detect entrypoints.*

Description

At the moment only reproduce_make is available.

Usage

```
reproduce_funs
```

Format

An object of class `list` of length 1.

Details

`reproduce_make` detects make as an entrypoint if there is a Makefile at top level. If it does also encounter a Makefile_Docker somewhere it recognizes the different make instructions.

See Also

reproduce

rerun *Find entrypoint for analysis and suggest how to reproduce it.*

Description

`rerun()` was renamed to `reproduce()`. `rerun()` is deprecated.

Usage

```
rerun(...)
```

Arguments

`...`	passed to `reproduce()`.

See Also

reproduce_funs

template *Create repro template*

Description

Fills a newly created folder with examples of RMarkdown, scripts, and data, which work well with `automate()`.

Usage

```
repro_template(path, ...)

use_repro_template(path, ...)
```

Arguments

`path`	character specifieng target folder. Will be created if it doesn't exist.
`...`	Passed down to the markdown template.

uses *Check Project Configuration*

Description

Check if a project uses certain things, e.g. Make or Docker.

Usage

```
uses_make(silent = FALSE)

uses_docker(silent = FALSE)

uses_gha_docker(silent = FALSE)

uses_gha_publish(silent = FALSE)

uses_make_publish(silent = FALSE)

uses_make_rmds(silent = FALSE)
```

Arguments

`silent`	Defaults to `false`. Should a message be printed that informs the user?

See Also

Other checkers: `check_package()`, `check`, `has_package()`, `has`

use_docker_packages *Add dependencies to Dockerfile*

Description

Adds package dependencies as a new RUN statement to Dockerfile. Sorts packages first into source (cran & github) and then alphabetically.

Usage

```
use_docker_packages(
  packages,
  github = NULL,
  strict = TRUE,
  file = "Dockerfile",
  write = TRUE,
  open = write,
  append = TRUE
)
```

Arguments

packages	Which packages to add.
github	Are there github packages?
strict	Defaults to TRUE, force a specific version for github packages.
file	Where is the 'Dockerfile'?
write	Should the 'Dockerfile' be modified?
open	Should the file be opened?
append	Should the return value be appended to the 'Dockerfile'?

use_gha_docker *Use GitHub Action to Build Dockerimage*

Description

Add an standard actions that builds and publishes an Dockerimage from the `Dockerfile`.

Usage

```
use_gha_docker(file = getOption("repro.gha.docker"), open = TRUE)
```

Arguments

file	Which file to save to.
open	Open the newly created file for editing? Happens in RStudio, if applicable, or via utils::file.edit() otherwise.

use_gha_publish *Use GitHub Action to Publish Results*

Description

Add an standard actions that builds the `publish`-target, and publishes the results on the `gh-pages` branch. Requires a Dockerimage published within the same GitHub repository. See `use_gha_docker()`.

Usage

```
use_gha_publish(file = getOption("repro.gha.publish"), open = TRUE)
```

Arguments

`file`	Which file to save to.
`open`	Open the newly created file for editing? Happens in RStudio, if applicable, or via utils::file.edit() otherwise.

use_template_template *Deal with templates that are themselves whisker templates.*

Description

If a template is itself used again as a whisker template, we need some other escape mechanism. The `repro`-template than uses `[[]]` instead of `{{}}` which are than later reinstated.

Usage

```
use_template_template(
  template,
  save_as = template,
  escape = c(`\\[\\[` = "{{", `\\]\\]` = "}}"),
  data = list(),
  ignore = FALSE,
  open = FALSE,
  package = "repro"
)
```

Arguments

`template`	`usethis::use_template()`
`save_as`	`usethis::use_template()`
`escape`	named vector, name is replacement (default curly), value is placeholder (default square)

data	usethis::use_template()
ignore	usethis::use_template()
open	usethis::use_template()
package	usethis::use_template()

Index