Package 'groundhog'

October 13, 2022

Title Version-Control for CRAN, GitHub, and GitLab Packages

Version 2.1.0

Description Make R scripts reproducible, by ensuring that every time a given script is run, the same version of the used packages are loaded (instead of whichever version the user running the script happens to have installed). This is achieved by using the command groundhog.library() instead of the base command library(), and including a date in the call. The date is used to call on the same version of the package every time (the most recent version available at that date). Load packages from CRAN, GitHub, or Gitlab.

URL https://groundhogr.com/, https://github.com/CredibilityLab/groundhog

BugReports https://github.com/CredibilityLab/groundhog/issues

Depends utils

Imports methods

Suggests git2r, remotes

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.0

NeedsCompilation no

Author Uri Simonsohn [aut, cre] (<https://orcid.org/0000-0002-8601-7211>), Hugo Gruson [ctb, aut] (<https://orcid.org/0000-0002-4094-1476>)

Maintainer Uri Simonsohn <urisohn@gmail.com>

Repository CRAN

Date/Publication 2022-10-02 21:30:02 UTC

R topics documented:

`cross.toc` *Show toc table with multiple packages*

Description

Show toc table with multiple packages

Usage

```
cross.toc(pkgs, date1 = "1970-1-1", date2 = Sys.Date())
```

Arguments

`pkgs`	character vector containing the package names.
`date1, date2`	date range to consider (in the format "%Y-%m-%d").

Value

A `data.frame` with 3 columns:

Version The package version number

Published The date at which the specific version was published

Package The package name

See Also

`toc()` for the same function for a single package.

Examples

```
## Not run:
cross.toc(c("magrittr", "R"))
cross.toc(c("magrittr", "rlang"), date1 = "2012-02-01", date2 = "2020-02-01")

## End(Not run)
```

get.groundhog.folder *Get current local path to groundhog folder*

Description

Get current local path to groundhog folder

Usage

```
get.groundhog.folder()
```

Value

the path to the groundhog folder, the meta-library where `groundhog.library()` downloads and stores packages that can be loaded

Note

you can change the location of this folder with the command `set.groundhog.folder("path")`.

See Also

`set.groundhog.folder()`

Examples

```
## Not run:
get.groundhog.folder()

## End(Not run)
```

get.snowball *Generates dataframe with all dependencies needed to install a package, in the order they will be loaded*

Description

Generates dataframe with all dependencies needed to install a package, in the order they will be loaded

Usage

```
get.snowball(pkg, date, include.suggests = FALSE, force.install = FALSE)
```

Arguments

pkg	character string, name of target package to load (and install if needed),
date	character string (yyyy-mm-dd), or date value, with the date which determines the version of the package, and all dependencies, to be loaded (and installed if needed).
include.suggests	logical, defaults to FALSE. When set to TRUE, includes dependencies classified in the DESCRIPTION file as suggested.
force.install	logical, defaults to FALSE. When set to TRUE, the column installed in the generated snowball is set FALSE for all packages, causing them to be installed even if already installed.

Value

a dataframe with all packages that need to be installed, their version , whether they are installed, where to obtain them if not locally available (CRAN vs MRAN), which date to use for MRAN, installation time from source (in seconds), and local path for storage

Examples

```
## Not run:
get.snowball("rio", "2020-07-12")

## End(Not run)
```

---

groundhog.library *Install & load CRAN, GitHub, and GitLab packages as current on given date*

Description

Groundhog maintains a separate local package library where it stores version-controlled packages, with multiple versions of the same package saved side-by-side. The date argument in the groundhog.library() function determines the version of the package that is loaded (the most recently available version on that date). #' If that version of the package is not already installed in the local groundhog library, #' it is automatically installed. groundhog.library() thus substitutes both library() and install.packages(). No changes to how R manages packages are made (e.g., no change to #' .libPaths(), to .Rprofile, or to R Studio global settings). Therefore, to discontinue relying on groundhog, all you do is go back to #' executing the install.packages() and library() #' functions, instead of the groundhog.library() function.

Usage

```
groundhog.library(
  pkg,
  date,
```

```
  quiet.install = TRUE,
  include.suggests = FALSE,
  ignore.deps = c(),
  force.source = FALSE,
  force.install = FALSE,
  tolerate.R.version = ""
)
```

Arguments

pkg	character string or vector with name of target package(s). Single package names need not be in quotes.
date	character string (yyyy-mm-dd), or date value, with the date which determines the version of the package, and all dependencies, to be loaded (and installed if needed). #'The most recent #'date accepted is 2 days prior to when the code is executed.
quiet.install	logical, defaults to TRUE. When set to FALSE, displays output generated by `install.packages()` when installing from source
include.suggests	logical, defaults to FALSE. When set to TRUE, loads dependencies classified in the DESCRIPTION file as suggested.
ignore.deps	an optional character vector containing dependencies which may be already loaded in the R session and even if the loaded version does not match the version implied by the entered date, groundhog.library() will proceed and ignore #' this conflict. If one version of a package is loaded, and a different is needed for #' groundhog, the default behavior is to stop the request and ask the user to restart #' the R session to unload all packages. This will bypass that requirement.
force.source	logical (defaults to FALSE). When set to TRUE, will not attempt installing binary from CRAN or MRAN and instead download source file and install it.
force.install	logical (defaults to FALSE). When set to TRUE, will
tolerate.R.version	optional character string containing an R version which `groundhog.library()` will not throw an error for using, even if the date entered corresponds to a more recent major R release.

Details

For more information about groundhog check out groundhogr.com

Value

a character vector containing all active packages for the session, with their version number, under the format pkg_vrs.

Examples

```
## Not run:
groundhog.library("magrittr", "2022-04-01")
```

```
pkgs <- c('pwr','metafor')
groundhog.library(pkgs, "2022-04-01")

# When running an existing  script that relied on `library()` to load packages,
# you can wrap the library calls in double-quotes, loading the packages with
# groundhog:

  groundhog.library(
       "
        library('pwr')
        library('metafor')
        library('tidyr')
        library('rio')
        library('this.path')
       "
       ,'2022-04-01')

#Allow using R 3.6.3 despite entering a date that corresponds to R >=4.0.0
  groundhog.library('rio', '2022-04-11', tolerate.R.version='3.6.3')


## End(Not run)
```

`meta.groundhog` *Load a specific version of groundhog, as available on a given date*

Description

Load a specific version of groundhog, as available on a given date

Usage

```
meta.groundhog(date)
```

Arguments

`date`	character string (yyyy-mm-dd), or date value, with the date which determines the version of groundhog to load

Examples

```
## Not run:
#Load groundhog as available on 2021-03-12 (v1.3.2)
meta.groundhog("2021-03-12")

## End(Not run)
```

mran.is.up *Re-enable groundhog.library() to try MRAN binaries*

Description

When an install from MRAN fails, groundhog does not try MRAN again for 5 hours assuming the server is down. You can over-rule this preventive measure by running this function.

Usage

```
mran.is.up()
```

set.groundhog.folder *Set groundhog folder location*

Description

Set groundhog folder location

Usage

```
set.groundhog.folder(path)
```

Arguments

path	Character. The path to the groundhog folder containing the library where packages are downloaded and installed.

Value

(invisibly) TRUE upon success.

See Also

get.groundhog.folder()

Examples

```
## Not run:
set.groundhog.folder("~/.R_groundhog")

## End(Not run)
```

toc *Show CRAN publication dates for all versions of a given package*

Description

Show CRAN publication dates for all versions of a given package

Usage

```
toc(pkg, dependencies = FALSE)
```

Arguments

pkg	(required) package name
dependencies	logical (defaults to FALSE). Should the output contain package dependencies (Imports, Depends and Suggests) for pkg.

Value

a data.frame where each row corresponds to one version of pkg, a date column contains the publication date, and when dependecies=TRUE, columns show package dependencies over time as well.

Examples

```
## Not run:
toc("R")
toc("magrittr")
toc("rio",dependencies = TRUE)

## End(Not run)
```

Index