Package: versions (via r-universe)

August 23, 2024

Type Package

Title Query and Install Specific Versions of Packages on CRAN

Version 0.4

Date 2017-04-25

Author Nick Golding

Maintainer Nick Golding <nick.golding.research@gmail.com>

Description Installs specified versions of R packages hosted on CRAN and provides functions to list available versions and the versions of currently installed packages. These tools can be used to help make R projects and packages more reproducible. 'versions' fits in the narrow gap between the 'devtools' install_version() function and the 'checkpoint' package. devtools::install_version() installs a stated package version from source files stored on the CRAN archives. However CRAN does not store binary versions of packages so Windows users need to have RTools installed and Windows and OSX users get longer installation times. 'checkpoint' uses the Revolution Analytics MRAN server to install packages (from source or binary) as they were available on a given date. It also provides a helpful interface to detect the packages in use in a directory and install all of those packages for a given date. 'checkpoint' doesn't provide install.packages-like functionality however, and that's what 'versions' aims to do, by querying MRAN. As MRAN only goes back to 2014-09-17, 'versions' can't install packages archived before this date.

License BSD_3_clause + file LICENSE

LazyData TRUE

BugReports https://github.com/goldingn/versions/issues

RoxygenNote 6.0.1

Suggests testthat

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

RemoteUrl https://github.com/goldingn/versions

RemoteRef HEAD

RemoteSha de231cb523aa9134a1bd4e947c8204a7a08f3daa

Contents

`versions-package` *versions: Query and Install Specific Versions of Packages on CRAN*

Description

Installs specified versions of R packages hosted on CRAN and provides functions to list available versions and the versions of currently installed packages. These tools can be used to help make R projects and packages more reproducible. `versions` fits in the narrow gap between the devtools `install_version` function and the `checkpoint` package.

`devtools::install_version` installs a stated package version from source files stored on the CRAN archives. However CRAN does not store binary versions of packages so Windows users need to have RTools installed and Windows and OSX users get longer installation times.

`checkpoint` uses the Revolution Analytics MRAN server to install packages (from source or binary) as they were available on a given date. It also provides a helpful interface to detect the packages in use in a directory and install all of those packages for a given date. `checkpoint` doesn't provide `install.packages`-like functionality however, and that's what `versions` aims to do, by querying MRAN.

As MRAN only goes back to 2014-09-17, `versions` can't install packages from before this date.

The available functions are:

- `available.versions`
- `install.versions`
- `install.dates`
- `installed.versions`

Details

The URL for MRAN may change from time to time. As of `versions` 0.4, the URL is `https://cran.microsoft.com/snapshot`, and this is what the package uses. If the MRAN server URL changes before `versions` can be updated, users can point versions to the new URL via the option 'versions.mran'. Ie. `options(versions.mran = "<some/new/url>")`

Examples

```
## Not run:

# list the available versions of checkpoint
available.versions('checkpoint')

# install a specific version
install.versions('checkpoint', '0.3.9')

# check the installed version
installed.versions('versions')

# install checkpoint as of a specific date
install.dates('checkpoint', '2014-12-25')


## End(Not run)
```

`available.versions` *available.versions*

Description

List all of the past versions of the named packages ever uploaded to CRAN (and therefore in the CRAN source archives), their publication dates and whether they can be installed from MRAN via `install.versions` or `install.dates`.

Usage

```
available.versions(pkgs)
```

Arguments

`pkgs`	character vector of the names of packages for which to query available versions

Value

a list of dataframes, each giving the versions and publication dates for the corresponding elements of `pkgs` as well as whether they can be installed from MRAN

Examples

```
## Not run:

# available versions of checkpoint
available.versions('checkpoint')
```

```
# available versions of checkpoint and devtools
available.versions(c('checkpoint', 'devtools'))


## End(Not run)
```

`install.dates` *install.dates*

Description

Download and install the latest versions of packages hosted on CRAN as of a specific date from the MRAN server.

Usage

```
install.dates(pkgs, dates, lib, ...)
```

Arguments

`pkgs`	character vector of the names of packages that should be downloaded and installed
`dates`	character or Date vector of the dates for which to install the latest versions of `pkgs`. If a character vector, it must be in the format 'yyyy-mm-dd', e.g. '2014-09-17'. If this has the same length as `pkgs` versions will correspond to those packages. If this has length one the same version will be used for all packages. If it has any other length an error will be thrown. Dates before 2014-09-17 will cause an error as MRAN does not archive before that date.
`lib`	character vector giving the library directories where to install the packages. Recycled as needed. If missing, defaults to the first element of `.libPaths()`.
`...`	other arguments to be passed to `install.packages`. The arguments `repos` and `contriburl` (at least) will be ignored as the function uses the MRAN server to retrieve package versions.

Examples

```
## Not run:

# install yesterday's version of checkpoint
install.dates('checkpoint', Sys.Date() - 1)

# install yesterday's versions of checkpoint and devtools
install.dates(c('checkpoint', 'devtools'), Sys.Date() - 1)

# install yesterday's version of checkpoint and the day before's devtools
install.dates(c('checkpoint', 'devtools'), Sys.Date() - 1:2)
```

```
## End(Not run)
```

`install.versions` *install.versions*

Description

Download and install named versions of packages hosted on CRAN from the MRAN server.

Usage

```
install.versions(pkgs, versions, lib, ...)
```

Arguments

`pkgs`	character vector of the names of packages that should be downloaded and installed
`versions`	character vector of the versions of packages to be downloaded and installed. If this has the same length as `pkgs` versions will correspond to those packages. If this has length one the same version will be used for all packages. If it has any other length an error will be thrown.
`lib`	character vector giving the library directories where to install the packages. Recycled as needed. If missing, defaults to the first element of `.libPaths()`.
`...`	other arguments to be passed to `install.packages`. The arguments `repos` and `contriburl` (at least) will be ignored as the function uses the MRAN server to retrieve package versions.

Examples

```
## Not run:

# install an earlier version of checkpoint
install.versions('checkpoint', '0.3.3')

# install earlier versions of checkpoint and devtools
install.versions(c('checkpoint', 'devtools'), c('0.3.3', '1.6.1'))


## End(Not run)
```

`installed.versions` *installed.versions*

Description

List the installed versions of packages in a library directory

Usage

```
installed.versions(pkgs, lib)
```

Arguments

`pkgs`	character vector of the names of packages for which to query the installed versions
`lib`	character vector of length one giving the library directory containing the packages to query. If missing, defaults to the first element of `.libPaths()`.

Value

a named character vector of version numbers corresponding to `pkgs`, with names giving the package names. If a packakge could not be found in `lib`, an NA will be returned.

Examples

```
# the versions of versions
installed.versions('versions')

# apply to multiple packages
installed.versions(c('stats', 'versions'))

# add a package that doesn't exist or isn't installed
# (returns NA for that one)
installed.versions(c('stats', 'versions', 'notapackage'))
```

Index