Package ‘packageRank’

May 8, 2020

Type Package

Title Computation and Visualization of Package Download Counts and Percentiles

Version 0.3.5

Date 2020-05-07

Maintainer Peter Li <lindbrook@gmail.com>

Description Compute and visualize the cross-sectional and longitudinal number and rank percentile of package downloads from RStudio's CRAN mirror.

URL https://github.com/lindbrook/packageRank

BugReports https://github.com/lindbrook/packageRank/issues

Depends R (>= 3.4)

License GPL (>= 2)

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.1.0

Imports cranlogs, data.table (>= 1.12.2), ggplot2, grDevices, memoise, pkgsearch, RCurl, R.utils, rversions, stats

Suggests knitr, rmarkdown

NeedsCompilation no

Author Peter Li [aut, cre]

Repository CRAN

Date/Publication 2020-05-08 10:40:03 UTC

R topics documented:

`archivePackages` *Packages in CRAN archive.*

Description

Scrape https://cran.r-project.org/src/contrib/Archive/.

Usage

```
archivePackages(include.date = FALSE, multi.core = TRUE, dev.mode = FALSE)
```

Arguments

`include.date`	Logical. Return data frame with package name and last publication date.
`multi.core`	Logical or Numeric. `TRUE` uses `parallel::detectCores()`. `FALSE` uses one, single core. You can also specify the number logical cores.
`dev.mode`	Logical. Development mode uses parallel::parLapply().

`bioconductorDownloads` *Annual/monthly package downloads from Bioconductor.*

Description

Annual/monthly package downloads from Bioconductor.

Usage

```
bioconductorDownloads(packages = NULL, from = NULL, to = NULL,
  when = NULL, observation = "month")
```

Arguments

`packages`	Character. Vector of package names.
`from`	Start date as `yyyy-mm` or `yyyy`.
`to`	End date as `yyyy-mm` or `yyyy`.
`when`	`"last-year"`, or `"year-to-date"` or `"ytd"`.
`observation`	"year" or "month".

Examples

```
# all packages
bioconductorDownloads()

# entire history
bioconductorDownloads(packages = "clusterProfiler")

# year-to-date
bioconductorDownloads(packages = "clusterProfiler", when = "ytd")
bioconductorDownloads(packages = "clusterProfiler", when = "year-to-date")

# last 12 months
bioconductorDownloads(packages = "clusterProfiler", when = "last-year")
```

```
# from 2015 to current year
bioconductorDownloads(packages = "clusterProfiler", from = 2015)

# 2010 through 2015 (yearly)
bioconductorDownloads(packages = "clusterProfiler", from = 2010, to = 2015, observation = "year")

# selected year (yearly)
bioconductorDownloads(packages = "clusterProfiler", from = 2015, to = 2015)

# selected year (monthly)
bioconductorDownloads(packages = "clusterProfiler", from = "2015-01", to = "2015-12")

# June 2014 through March 2015
bioconductorDownloads(packages = "clusterProfiler", from = "2014-06", to = "2015-03")
```

bioconductorRank *Package download counts and rank percentiles.*

Description

From bioconductor

Usage

```
bioconductorRank(packages = "monocle", date = "2019-01", count = "download")
```

Arguments

packages	Character. Vector of package name(s).
date	Character. Date. yyyy-mm
count	Character. "ip" or "download".

Value

An R data frame.

Examples

```
bioconductorRank(packages = "cicero", date = "2019-09")
```

`blog.data` *Blog post data.*

Description

```
archive.pkg_ver
archive.pkg_ver.filtered
cran.pkg_ver
cran.pkg_ver.filtered
dl.ct
dl.ct2
pkg.ct
pkg.ct2
oct.data
cholera.data
ggplot2.data
VR.data
smpl
smpl.histories
smpl.archive
smpl.archive.histories
ccode.ct
crosstab_2019_10_01
percentiles
```

Usage

```
blog.data
```

Format

A list with 19 elements.

countryPackage *Tabulate a country's package downloads.*

Description

From RStudio's CRAN Mirror http://cran-logs.rstudio.com/

Usage

```
countryPackage(country = "US", date = Sys.Date() - 1, memoization = TRUE,
  sort = TRUE)
```

Arguments

country	Character. country abbreviation.
date	Character. Date. yyyy-mm-dd.
memoization	Logical. Use memoization when downloading logs.
sort	Logical. Sort by download count.

countsRanks *Counts v. Rank Percentiles for 'cholera' for First Week of March 2020.*

Description

Document code for blog graph.

Usage

```
countsRanks(package = "cholera", size.filter = FALSE)
```

Arguments

package	Character.
size.filter	Logical.

cranDownloads *Daily package downloads from the RStudio CRAN mirror.*

Description

Enhanced implementation of cranlogs::cran_downloads().

Usage

```
cranDownloads(packages = NULL, when = NULL, from = NULL, to = NULL,
  check.package = TRUE, dev.mode = FALSE)
```

Arguments

`packages`	A character vector, the packages to query, or `NULL` for a sum of downloads for all packages. Alternatively, it can also be `"R"`, to query downloads of R itself. `"R"` cannot be mixed with packages.
`when`	`last-day`, `last-week` or `last-month`. If this is given, then `from` and `to` are ignored.
`from`	Start date as `yyyy-mm-dd`, `yyyy-mm` or `yyyy`.
`to`	End date as `yyyy-mm-dd`, `yyyy-mm` or `yyyy`.
`check.package`	Logical. Validate and "spell check" package.
`dev.mode`	Logical. Use validatePackage0() to scrape CRAN.

Examples

```
cranDownloads(packages = "HistData")
cranDownloads(packages = "HistData", when = "last-week")
cranDownloads(packages = "HistData", when = "last-month")

# January 7 - 31, 2019
cranDownloads(packages = "HistData", from = "2019-01-07", to = "2019-01-31")

# February through March 2019
cranDownloads(packages = "HistData", from = "2019-02", to = "2019-03")

# 2020 year-to-date
cranDownloads(packages = "HistData", from = 2020)
```

`fetchCranLog` *Fetch CRAN Logs.*

Description

Fetch CRAN Logs.

Usage

```
fetchCranLog(date, memoization)
```

Arguments

`date`	Character. Date. yyyy-mm-dd.
`memoization`	Logical. Use memoization when downloading logs.

`fetchLog` *fread() to data.frame.*

Description

fread() to data.frame.

Usage

```
fetchLog(x)
```

Arguments

`x`	Character. URL

Note

mFetchLog() is memoized version.

fixDate_2012 *Re-map filenames (dates) for 2012 download logs.*

Description

Correct for mis-labeled filenames for 2012 logs at RStudio's CRAN Mirror http://cran-logs.rstudio.com/.

Usage

```
fixDate_2012(date = "2012-12-31")
```

Arguments

`date`	Character. Date. "yyyy-mm-dd".

Value

A one unit R date or character vector.

Note

This date problem does not affect `cranDownlaods()`.

inflationPlot *Inflation plots of effects of "small" downloads and prior versions for October 2019: 'cholera', 'ggplot2', and 'VR'.*

Description

Document code for blog graph.

Usage

```
inflationPlot(package = "cholera", filter = "size", legend.loc = "topleft")
```

Arguments

`package`	Character.
`filter`	Character. Size, version, or size and version
`legend.loc`	Character. Location of legend.

packageArchive *Extract version history from Archive.*

Description

Extract version history from Archive.

Usage

```
packageArchive(package = "cholera")
```

Arguments

package	Character. Package name.

Value

An R data frame or NULL.

Examples

```
packageArchive(package = "HistData")
packageArchive(package = "adjustedcranlogs")  # No archived versions.
```

packageCountry *Package download counts by country.*

Description

From RStudio's CRAN Mirror http://cran-logs.rstudio.com/

Usage

```
packageCountry(packages = NULL, date = Sys.Date() - 1,
  memoization = TRUE, sort = TRUE, na.rm = FALSE)
```

Arguments

packages	Character. Vector of package name(s).
date	Character. Date. yyyy-mm-dd.
memoization	Logical. Use memoization when downloading logs.
sort	Logical. Sort by download count.
na.rm	Logical. Remove NAs.

packageCRAN *Extract package version history from CRAN.*

Description

Date and version of most recent publication.

Usage

```
packageCRAN(package = "cholera")
```

Arguments

package	Character. Package name.

Value

An R data frame or NULL.

Examples

```
packageCRAN(package = "HistData")
packageCRAN(package = "VR") # No version on CRAN (archived)
```

packageDistribution *Package Download Distribution.*

Description

Package Download Distribution.

Usage

```
packageDistribution(package = "HistData", date = Sys.Date() - 1,
  size.filter = FALSE, memoization = TRUE, check.package = TRUE,
  dev.mode = FALSE)
```

Arguments

`package`	Character. Vector of package name(s).
`date`	Character. Date. "yyyy-mm-dd".
`size.filter`	Logical or Numeric. If Logical, TRUE filters out downloads less than 1000 bytes. If Numeric, a positive value sets the minimum download size (in bytes) to consider; a negative value sets the maximum download size to consider.
`memoization`	Logical. Use memoization when downloading logs.
`check.package`	Logical. Validate and "spell check" package.
`dev.mode`	Logical. Use validatePackage0() to scrape CRAN.

`packageHistory` *Extract package version history CRAN and Archive.*

Description

Date and version of all publications.

Usage

```
packageHistory(package = "cholera", short.date = TRUE)
```

Arguments

`package`	Character. Package name.
`short.date`	Logical

`packageHistory0` *Extract package version history CRAN and Archive (scrape CRAN).*

Description

Date and version of most recent publication.

Usage

```
packageHistory0(package = "cholera")
```

Arguments

`package`	Character. Package name.

`packageInfo` *Extract package information from CRAN.*

Description

Extract package information from CRAN.

Usage

```
packageInfo(multi.core = TRUE, platform = "win", r.ver = "release",
  source = TRUE)
```

Arguments

`multi.core` Logical or Numeric. `TRUE` uses `parallel::detectCores()`. `FALSE` uses one, single core. You can also specify the number logical cores. Mac and Unix only.

`platform` Character.

`r.ver` Character.

`source` Logical.

`packageLog` *Get Package Download Logs.*

Description

From RStudio's CRAN Mirror http://cran-logs.rstudio.com/

Usage

```
packageLog(packages = NULL, date = Sys.Date() - 1, filter = FALSE,
  memoization = TRUE)
```

Arguments

`packages` Character. Vector of package name(s).

`date` Character. Date.

`filter` Logical or Numeric. If Logical, TRUE filters out downloads less than 1000 bytes. If Numeric, a positive value (bytes) sets the minimum download size to consider; a negative value sets the maximum download size to consider.

`memoization` Logical. Use memoization when downloading logs.

Value

An R data frame.

packageRank *Package download counts and rank percentiles.*

Description

From RStudio's CRAN Mirror http://cran-logs.rstudio.com/

Usage

```
packageRank(packages = "HistData", date = Sys.Date() - 1,
  size.filter = TRUE, memoization = TRUE, check.package = TRUE,
  dev.mode = FALSE)
```

Arguments

packages	Character. Vector of package name(s).
date	Character. Date. "yyyy-mm-dd".
size.filter	Logical or Numeric. If Logical, TRUE filters out downloads less than 1000 bytes. If Numeric, a positive value sets the minimum download size (in bytes) to consider; a negative value sets the maximum download size to consider.
memoization	Logical. Use memoization when downloading logs.
check.package	Logical. Validate and "spell check" package.
dev.mode	Logical. Use validatePackage0() to scrape CRAN.

Value

An R data frame.

Examples

```
packageRank(packages = "HistData", date = "2020-01-01")
packageRank(packages = c("h2o", "Rcpp", "rstan"), date = "2020-01-01")
```

plot.bioconductorDownloads *Plot method for bioconductorDownloads().*

Description

Plot method for bioconductorDownloads().

Usage

```
## S3 method for class 'bioconductorDownloads'
plot(x, graphics = NULL,
  count = "download", points = "auto", smooth = FALSE, smooth.f = 2/3,
  se = FALSE, log_count = FALSE, ...)
```

Arguments

`x`	object.
`graphics`	Character. NULL, "base" or "ggplot2".
`count`	Character. "download" or "ip".
`points`	Character of Logical. Plot points. "auto", TRUE, FALSE. "auto" for bioconductorDownloads(observation = "month") with 24 or fewer months, points are plotted.
`smooth`	Logical. Add stats::lowess smoother.
`smooth.f`	Numeric. smoother span.
`se`	Logical. Works only with graphics = "ggplot2".
`log_count`	Logical. Logarithm of package downloads.
`...`	Additional plotting parameters.

Examples

```
plot(bioconductorDownloads())
plot(bioconductorDownloads(packages = "graph"))
plot(bioconductorDownloads(packages = "graph", from = 2010, to = 2015))
plot(bioconductorDownloads(packages = "graph", from = "2014-06", to = "2015-03"))
plot(bioconductorDownloads(packages = c("graph", "IRanges", "S4Vectors"), from = 2018))
```

`plot.bioconductorRank` *Plot method for bioconductorRank().*

Description

Plot method for bioconductorRank().

Usage

```
## S3 method for class 'bioconductorRank'
plot(x, graphics = NULL, log_count = TRUE, ...)
```

Arguments

`x`	An object of class "bioconductor_rank" created by `bioconductorRank()`.
`graphics`	Character. "base" or "ggplot2".
`log_count`	Logical. Logarithm of package downloads.
`...`	Additional plotting parameters.

Value

A base R or ggplot2 plot.

`plot.countsRanks` *Plot method for countsRanks().*

Description

Plot method for countsRanks().

Usage

```
## S3 method for class 'countsRanks'
plot(x, ...)
```

Arguments

`x`	object.
`...`	Additional plotting parameters.

`plot.cranDownloads` *Plot method for cranDownloads().*

Description

Plot method for cranDownloads().

Usage

```
## S3 method for class 'cranDownloads'
plot(x, graphics = "auto", points = "auto",
  log.count = FALSE, smooth = FALSE, se = FALSE, f = 1/3,
  package.version = FALSE, r.version = FALSE, population.plot = FALSE,
  multi.plot = FALSE, same.xy = TRUE, legend.loc = "topleft",
  dev.mode = FALSE, ...)
```

Arguments

`x`	object.
`graphics`	Character. "auto", "base" or "ggplot2".
`points`	Character of Logical. Plot points. "auto", TRUE, FALSE.
`log.count`	Logical. Logarithm of package downloads.
`smooth`	Logical. Add smoother.
`se`	Logical. Works only with graphics = "ggplot2".
`f`	Numeric. stats::lowess() smoother window. For use with graphics = "base" only.
`package.version`	Logical. Add latest package release dates.
`r.version`	Logical. Add R release dates.
`population.plot`	Logical. Plot population plot.
`multi.plot`	Logical.
`same.xy`	Logical. Use same scale for multiple packages when graphics = "base".
`legend.loc`	Character.
`dev.mode`	Logical. Use packageHistory0() to scrape CRAN.
`...`	Additional plotting parameters.

Value

A base R or ggplot2 plot.

Examples

```
plot(cranDownloads(packages = c("Rcpp", "rlang", "data.table")))
plot(cranDownloads(packages = c("Rcpp", "rlang", "data.table"), when = "last-month"))
plot(cranDownloads(packages = "R", from = "2020-01-01", to = "2020-01-01"))
plot(cranDownloads(packages = "R", from = 2020))
```

`plot.packageDistribution` *Plot method for packageDistribution().*

Description

Plot method for packageDistribution().

Usage

```
## S3 method for class 'packageDistribution'
plot(x, ...)
```

Arguments

`x`	An object of class "packageDistribution" created by `packageDistribution()`.
`...`	Additional plotting parameters.

`plot.packageRank` *Plot method for packageRank().*

Description

Plot method for packageRank().

Usage

```
## S3 method for class 'packageRank'
plot(x, graphics = NULL, log_count = TRUE, ...)
```

Arguments

`x`	An object of class "packageRank" created by `packageRank()`.
`graphics`	Character. "base" or "ggplot2".
`log_count`	Logical. Logarithm of package downloads.
`...`	Additional plotting parameters.

Value

A base R or ggplot2 plot.

Examples

```
plot(packageRank(packages = "HistData", date = "2020-01-01"))
plot(packageRank(packages = c("h2o", "Rcpp", "rstan"), date = "2020-01-01"))
```

`populationPlot` *Visualize a Package's Downloads Relative to "All" CRAN packages over Time.*

Description

Uses a stratified random sample cohort of packages plus top ten packages.

Usage

```
populationPlot(x, graphics = NULL, log.count = TRUE, smooth = TRUE,
  sample.smooth = TRUE, f = 1/3, sample.pct = 5, multi.core = TRUE)
```

Arguments

`x` object.

`graphics` Character. NULL, "base" or "ggplot2".

`log.count` Logical. Logarithm of package downloads.

`smooth` Logical. Add smoother.

`sample.smooth` Logical. Add smoother.

`f` Numeric. stats::lowess() smoother window. For use with graphics = "base" only.

`sample.pct` Numeric. Percent of packages to sample.

`multi.core` Logical or Numeric. `TRUE` uses `parallel::detectCores()`. `FALSE` uses one, single core. You can also specify the number logical cores to use. Note that due to performance considerations, the number of cores defaults to one on Windows.

`print.bioconductorDownloads` *Print method for bioconductorDownloads().*

Description

Print method for bioconductorDownloads().

Usage

```
## S3 method for class 'bioconductorDownloads'
print(x, ...)
```

Arguments

`x` object.

`...` Additional parameters.

print.bioconductorRank *Print method for bioconductorRank().*

Description

Print method for bioconductorRank().

Usage

```
## S3 method for class 'bioconductorRank'
print(x, ...)
```

Arguments

x	An object of class "bioconductor_rank" created by `bioconductorRank()`
...	Additional parameters.

print.cranDownloads *Print method for cranDownloads().*

Description

Print method for cranDownloads().

Usage

```
## S3 method for class 'cranDownloads'
print(x, ...)
```

Arguments

x	object.
...	Additional parameters.

print.packageDistribution

Print method for packageDistribution().

Description

Print method for packageDistribution().

Usage

```
## S3 method for class 'packageDistribution'
print(x, ...)
```

Arguments

`x`	An object of class "packageDistribution" created by `packageDistribution()`
`...`	Additional parameters.

print.packageRank

Print method for packageRank().

Description

Print method for packageRank().

Usage

```
## S3 method for class 'packageRank'
print(x, ...)
```

Arguments

`x`	An object of class "packageRank" created by `packageRank()`
`...`	Additional parameters.

`resolveDate` *Resolve date.*

Description

Check date format and validate date.

Usage

```
resolveDate(date, type = "from")
```

Arguments

`date`	Character. `"yyyy-mm-dd"`, `"yyyy-mm"`, `"yyyy"` or `yyyy` (numeric).
`type`	Character. Type of date "to" or "from".

`summary.bioconductorDownloads` *Summary method for bioconductorDownloads().*

Description

Summary method for bioconductorDownloads().

Usage

```
## S3 method for class 'bioconductorDownloads'
summary(object, ...)
```

Arguments

`object`	Object.
`...`	Additional parameters.

summary.bioconductorRank

Summary method for bioconductorRank().

Description

Summary method for bioconductorRank().

Usage

```
## S3 method for class 'bioconductorRank'
summary(object, ...)
```

Arguments

`object`	Object. An object of class "bioconductor_rank" created by `bioconductorRank()`
`...`	Additional parameters.

Note

This is useful for directly accessing the data frame.

summary.cranDownloads

Summary method for cranDownloads().

Description

Summary method for cranDownloads().

Usage

```
## S3 method for class 'cranDownloads'
summary(object, ...)
```

Arguments

`object`	Object.
`...`	Additional parameters.

Note

This is useful for directly accessing the data frame.

summary.packageRank *Summary method for packageRank().*

Description

Summary method for packageRank().

Usage

```
## S3 method for class 'packageRank'
summary(object, ...)
```

Arguments

`object`	Object. An object of class "packageRank" created by `packageRank()`
`...`	Additional parameters.

Note

This is useful for directly accessing the data frame.

validatePackage *Check for valid package names.*

Description

Check for valid package names.

Usage

```
validatePackage(packages)
```

Arguments

`packages`	Character. Vector of package name(s).

validatePackage0 *Check for valid package names (scrape CRAN).*

Description

Check for valid package names (scrape CRAN).

Usage

```
validatePackage0(packages, check.archive = TRUE)
```

Arguments

`packages`	Character. Vector of package name(s).
`check.archive`	Logical. Include archive when validating package. This is computationally expensive because it scrapes https://cran.r-project.org/src/contrib/Archive/.

Index