Package ‘packageRank’

September 15, 2021

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Computation and Visualization of Package Download Counts and Percentiles</td>
</tr>
<tr>
<td>Version</td>
<td>0.5.0</td>
</tr>
<tr>
<td>Date</td>
<td>2021-09-14</td>
</tr>
<tr>
<td>Maintainer</td>
<td>Peter Li <a href="mailto:lindbrook@gmail.com">lindbrook@gmail.com</a></td>
</tr>
<tr>
<td>Description</td>
<td>Compute and visualize the cross-sectional and longitudinal number and rank percentile of package downloads from RStudio's CRAN mirror.</td>
</tr>
<tr>
<td>URL</td>
<td><a href="https://github.com/lindbrook/packageRank">https://github.com/lindbrook/packageRank</a></td>
</tr>
<tr>
<td>BugReports</td>
<td><a href="https://github.com/lindbrook/packageRank/issues">https://github.com/lindbrook/packageRank/issues</a></td>
</tr>
<tr>
<td>Depends</td>
<td>R (&gt;= 3.5)</td>
</tr>
<tr>
<td>License</td>
<td>GPL (&gt;= 2)</td>
</tr>
<tr>
<td>Encoding</td>
<td>UTF-8</td>
</tr>
<tr>
<td>Language</td>
<td>en-US</td>
</tr>
<tr>
<td>LazyData</td>
<td>true</td>
</tr>
<tr>
<td>RoxygenNote</td>
<td>7.1.2</td>
</tr>
<tr>
<td>Imports</td>
<td>cranlogs, data.table (&gt;= 1.12.2), ggplot2, grDevices, memoise, pkgsearch, RCurl, R.utils, rversions, stats, sugrrants, tools, utils</td>
</tr>
<tr>
<td>Suggests</td>
<td>knitr, rmarkdown</td>
</tr>
<tr>
<td>NeedsCompilation</td>
<td>no</td>
</tr>
<tr>
<td>Author</td>
<td>Peter Li [aut, cre]</td>
</tr>
<tr>
<td>Repository</td>
<td>CRAN</td>
</tr>
<tr>
<td>Date/Publication</td>
<td>2021-09-15 04:20:02 UTC</td>
</tr>
</tbody>
</table>
### R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>annualDownloads</td>
<td>3</td>
</tr>
<tr>
<td>archivePackages</td>
<td>4</td>
</tr>
<tr>
<td>bioconductorDownloads</td>
<td>4</td>
</tr>
<tr>
<td>bioconductorRank</td>
<td>5</td>
</tr>
<tr>
<td>blog.data</td>
<td>6</td>
</tr>
<tr>
<td>countryDistribution</td>
<td>7</td>
</tr>
<tr>
<td>countryPackage</td>
<td>8</td>
</tr>
<tr>
<td>countsRanks</td>
<td>9</td>
</tr>
<tr>
<td>cranDownloads</td>
<td>9</td>
</tr>
<tr>
<td>cranInflationPlot</td>
<td>10</td>
</tr>
<tr>
<td>cranMirrors</td>
<td>10</td>
</tr>
<tr>
<td>cranPackages</td>
<td>11</td>
</tr>
<tr>
<td>cranPackageSize</td>
<td>11</td>
</tr>
<tr>
<td>currentTime</td>
<td>12</td>
</tr>
<tr>
<td>downloadsCountry</td>
<td>12</td>
</tr>
<tr>
<td>fetchCranLog</td>
<td>13</td>
</tr>
<tr>
<td>filteredDownloads</td>
<td>13</td>
</tr>
<tr>
<td>inflationPlot</td>
<td>14</td>
</tr>
<tr>
<td>inflationPlot2</td>
<td>14</td>
</tr>
<tr>
<td>ipCount</td>
<td>15</td>
</tr>
<tr>
<td>ipDownloads</td>
<td>15</td>
</tr>
<tr>
<td>ipFilter</td>
<td>16</td>
</tr>
<tr>
<td>ipPackage</td>
<td>16</td>
</tr>
<tr>
<td>localTime</td>
<td>17</td>
</tr>
<tr>
<td>logDate</td>
<td>18</td>
</tr>
<tr>
<td>logPostInfo</td>
<td>18</td>
</tr>
<tr>
<td>monthlyLog</td>
<td>19</td>
</tr>
<tr>
<td>packageArchive</td>
<td>19</td>
</tr>
<tr>
<td>packageCountry</td>
<td>20</td>
</tr>
<tr>
<td>packageCRAN</td>
<td>21</td>
</tr>
<tr>
<td>packageDistribution</td>
<td>21</td>
</tr>
<tr>
<td>packageHistory</td>
<td>22</td>
</tr>
<tr>
<td>packageLog</td>
<td>23</td>
</tr>
<tr>
<td>packageMRAN</td>
<td>24</td>
</tr>
<tr>
<td>packageRank</td>
<td>24</td>
</tr>
<tr>
<td>packageVersionPercent</td>
<td>25</td>
</tr>
<tr>
<td>plot.annualDownloads</td>
<td>26</td>
</tr>
<tr>
<td>plot.bioconductorDownloads</td>
<td>26</td>
</tr>
<tr>
<td>plot.bioconductorRank</td>
<td>28</td>
</tr>
<tr>
<td>plot.countryDistribution</td>
<td>28</td>
</tr>
<tr>
<td>plot.countsRanks</td>
<td>29</td>
</tr>
<tr>
<td>plot.cranDownloads</td>
<td>29</td>
</tr>
<tr>
<td>plot.packageDistribution</td>
<td>30</td>
</tr>
<tr>
<td>plot.packageRank</td>
<td>31</td>
</tr>
<tr>
<td>plot.packageVersionPercent</td>
<td>32</td>
</tr>
<tr>
<td>plot.weeklyDownloads</td>
<td>32</td>
</tr>
</tbody>
</table>
annualDownloads

plotDownloadsCountry ........................................... 33
plotTopCountryCodes ........................................... 33
print.bioc conductorDownloads ................................ 33
print.bioc conductorRank ....................................... 34
print.cranDownloads ........................................... 34
print.packageDistribution ...................................... 35
print.packageRank ............................................... 35
sequenceFilter .................................................. 36
sizeFilter ....................................................... 36
smallFilter ...................................................... 37
summary.bioc conductorDownloads ......................... 37
summary.bioc conductorRank .................................. 38
summary.cranDownloads ....................................... 38
summary.packageRank .......................................... 39
topCountryCodes ............................................... 39
tripletFilter .................................................... 40
utc ............................................................. 40
utc0 ............................................................. 40
versionPlot ...................................................... 41
weeklyDownloads ............................................... 41

Index

annualDownloads  Count Total CRAN Download.

Description

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

Usage

annualDownloads(start.yr = 2013, end.yr = 2020, multi.core = TRUE)

Arguments

start.yr Numeric or Integer.
end.yr Numeric or Integer.
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.
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, unit.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".

unit.observation  "year" or "month".
bioconductorRank

Examples

```r
## Not run:
# 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,
                       unit.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")

## End(Not run)
```

---

bioconductorRank  

Package download counts and rank percentiles.

Description

From bioconductor

Usage

```r
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

```r
## Not run:
bioconductorRank(packages = "cicero", date = "2019-09")

## End(Not run)
```

---

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  
top.n.oct2019  
top.n.jul2020  
download.country  
october.downloads  
july.downloads
**countryDistribution**

- cran.pkgs.oct
- arch.pkgs.oct
- cran.pkgs.jul
- arch.pkgs.jul
- pkg.history

**Usage**

blog.data

**Format**

A list with 29 elements.

---

**countryDistribution**  
*Tabulate package downloads by country.*

**Description**

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

**Usage**

```r
countryDistribution(date = NULL, all.filters = FALSE, ip.filter = FALSE, triplet.filter = FALSE, small.filter = FALSE, sequence.filter = FALSE, size.filter = FALSE, memoization = TRUE, multi.core = TRUE)
```

**Arguments**

- **date**  
  Character. Date. "yyyy-mm-dd". NULL uses latest available log.

- **all.filters**  
  Logical. Master switch for filters.

- **ip.filter**  
  Logical.

- **triplet.filter**  
  Logical.

- **small.filter**  
  Logical. TRUE filters out downloads less than 1000 bytes.

- **sequence.filter**  
  Logical.

- **size.filter**  
  Logical.

- **memoization**  
  Logical. Use memoization when downloading logs.

- **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.

**Value**

An R data frame.
countryPackage

Tabulate a country's package downloads.

Description

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

Usage

countryPackage(country = "HK", date = NULL, all.filters = FALSE, 
ip.filter = FALSE, triplet.filter = FALSE, small.filter = FALSE, 
sequence.filter = FALSE, size.filter = FALSE, sort = TRUE, 
memoization = TRUE, multi.core = TRUE)

Arguments

country Character. Country abbreviation.
date Character. Date. "yyyy-mm-dd". NULL uses latest available log.
all.filters Logical. Master switch for filters.
ip.filter Logical.
triplet.filter Logical.
small.filter Logical.
sequence.filter Logical. Set to FALSE.
size.filter Logical. Set to FALSE.
sort Logical. Sort by download count.
memoization Logical. Use memoization when downloading logs.
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.

Note

"US" outlier 10 min with all filters!
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.
## Not run:

```r
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)
```

## End(Not run)

---

**cranInflationPlot**  
**CRAN inflation plot.**

### Description

Document code.

### Usage

```r
cranInflationPlot(dataset = "october")
```

### Arguments

- `dataset`  
  Character. "october" or "july" for October 2019 or July 2020.

---

**cranMirrors**  
**Scrape CRAN Mirrors data.**

### Description

https://cran.r-project.org/mirrors.html

### Usage

```r
cranMirrors(mirror.description = FALSE)
```

### Arguments

- `mirror.description`  
  Logical. Mirror details.
**cranPackages**

Scrape CRAN package information.

**Description**

Current version, date and size (source and binary).

**Usage**

```r
cranPackages(binary = FALSE, bytes = FALSE, multi.core = TRUE)
```

**Arguments**

- `binary` Logical. Compute size of binary files.
- `bytes` Logical. Compute approximate numeric file size in bytes.
- `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.

**Value**

An R data frame.

---

**cranPackageSize**

Scrape package data from CRAN.

**Description**

Version, date and size (source file) of most recent publication.

**Usage**

```r
cranPackageSize(package = "cholera", check.package = TRUE, size = TRUE, r.ver = "4.0", bytes = TRUE, multi.core = TRUE)
```

**Arguments**

- `package` Character. Package name.
- `check.package` Logical. Validate and "spell check" package.
- `size` Logical. Include size of source file.
- `r.ver` Character. Current R version; used in directory path.
- `bytes` Logical. Compute approximate file size (bytes).
- `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.

**Value**

An R data frame or NULL.
**currentTime**

*Compute Current Time in Selected Time Zone.*

**Description**

Compute Current Time in Selected Time Zone.

**Usage**

```r
currentTime(tz = "Australia/Sydney")
```

**Arguments**

- `tz` Character. Local time zone. See OlsonNames() or use Sys.timezone().

---

**downloadsCountry**

*Compute Downloads by Country Code.*

**Description**

Compute Downloads by Country Code.

**Usage**

```r
downloadsCountry(month_cran_log, multi.core = TRUE)
```

**Arguments**

- `month_cran_log` Object.
- `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.
fetchCranLog  

*Fetch CRAN Logs.*

**Description**

Fetch CRAN Logs.

**Usage**

```r
fetchCranLog(date, memoization = FALSE, dev.mode = FALSE)
```

**Arguments**

- `date`: Character. Date. yyyy-mm-dd.
- `memoization`: Logical. Use memoization when downloading logs.
- `dev.mode`: Logical. Use Base R code.

---

**filteredDownloads**  

*Filtered package downloads from the RStudio CRAN mirror (prototype).*

**Description**

ip, triplet, small, sequence and size filters.

**Usage**

```r
filteredDownloads(packages = "HistData", date = NULL, all.filters = TRUE, 
ip.filter = FALSE, triplet.filter = FALSE, small.filter = FALSE, 
sequence.filter = FALSE, size.filter = FALSE, check.package = TRUE, 
memoization = TRUE, multi.core = TRUE)
```

**Arguments**

- `packages`: Character. Vector of package name(s).
- `date`: Character. Date. "yyy-mm-dd". NULL uses latest available log.
- `all.filters`: Logical. Master switch for filters.
- `ip.filter`: Logical.
- `triplet.filter`: Logical.
- `small.filter`: Logical. TRUE filters out downloads less than 1000 bytes.
- `sequence.filter`: Logical.
- `size.filter`: Logical.
check.package Logical. Validate and "spell check" package.

memoization Logical. Use memoization when downloading logs.

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.

---

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.

---

inflationPlot2

Inflation plots of effects of "small" downloads on aggregate CRAN downloads for October 2019 and July 2020.

---

Description

Document code.

Usage

inflationPlot2(dataset = "october", filter = "small", wed = FALSE, subtitle = TRUE, legend.loc = "topleft")

Arguments

dataset Character. "october" or "july" for October 2019 or July 2020.

filter Character. "small", "ip", or "ip.small".

wed Logical.

subtitle Logical.

legend.loc Character. Location of legend.
ipCount

Count number of IP addresses.

Description

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

Usage

ipCount(date = NULL, memoization = TRUE, sort = TRUE)

Arguments

date Character. Date. "yyyy-mm-dd". NULL uses latest available log.
memoization Logical. Use memoization when downloading logs.
sort Logical. Sort by download count.

ipDownloads

Unique package download counts by IP address.

Description

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

Usage

ipDownloads(date = NULL, memoization = TRUE)

Arguments

date Character. Date. "yyyy-mm-dd". NULL uses latest available log.
memoization Logical. Use memoization when downloading logs.
ipFilter

Filter Out A-Z Campaigns from IPs with many unique package downloads.

Description

Uses run length encoding, rle(), and k-means clustering, stats::kmeans().

Usage

ipFilter(cran_log, campaigns = TRUE, rle.depth = 100, case.sensitive = FALSE, multi.core = TRUE)

Arguments

cran_log  Object. Package log entries.
campaigns Logical. Filter A-Z campaigns when checking IPs with high unique package download counts.
rle.depth Numeric. Ceiling for number of rows of run length encoding. Fewer rows means longer runs.
case.sensitive Logical.
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.

ipPackage

Tabulate an IP's package downloads.

Description

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

Usage

ipPackage(ip = 10, date = NULL, all.filters = FALSE, ip.filter = FALSE, triplet.filter = FALSE, small.filter = FALSE, sequence.filter = FALSE, size.filter = FALSE, sort = TRUE, memoization = TRUE, multi.core = TRUE)
localTime

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip</td>
<td>Numeric. ip_id.</td>
</tr>
<tr>
<td>date</td>
<td>Character. Date. &quot;yyyy-mm-dd&quot;. NULL uses latest available log.</td>
</tr>
<tr>
<td>all.filters</td>
<td>Logical. Master switch for filters.</td>
</tr>
<tr>
<td>ip.filter</td>
<td>Logical.</td>
</tr>
<tr>
<td>triplet.filter</td>
<td>Logical.</td>
</tr>
<tr>
<td>small.filter</td>
<td>Logical. TRUE filters out downloads less than 1000 bytes.</td>
</tr>
<tr>
<td>sequence.filter</td>
<td>Logical.</td>
</tr>
<tr>
<td>size.filter</td>
<td>Logical.</td>
</tr>
<tr>
<td>sort</td>
<td>Logical. Sort by download count.</td>
</tr>
<tr>
<td>memoization</td>
<td>Logical. Use memoization when downloading logs.</td>
</tr>
<tr>
<td>multi.core</td>
<td>Logical or Numeric. TRUE uses parallel::detectCores(). FALSE uses one, single core. You can also specify the number logical cores. Mac and Unix only.</td>
</tr>
</tbody>
</table>

Note

ip = 10 is a tw top-level domain on 2020-07-09.

localTime

Compute Local Time from Coordinated Universal Time (UTC/GMT).

Description

Compute Local Time from Coordinated Universal Time (UTC/GMT).

Usage

localTime(date = "2021-1-1", time = "12:00", tz = Sys.timezone())

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>Character. Date &quot;yyyy-mm-dd&quot;.</td>
</tr>
<tr>
<td>time</td>
<td>Character. Local time &quot;hh:mm&quot; or &quot;hh:mm:ss&quot;.</td>
</tr>
<tr>
<td>tz</td>
<td>Character. Local time zone. See OlsonNames() or use Sys.timezone().</td>
</tr>
</tbody>
</table>
logDate

*Compute Effective CRAN Log Date Based on Local and UTC Time (prototype).*

**Description**

RStudio CRAN Mirror Logs for previous day are posted at 17:00:00 UTC.

**Usage**

```r
call = logDate(date = NULL, check.url = TRUE, repository = "CRAN", tz = Sys.timezone(), upload.time = "17:00", warning.msg = TRUE)
```

**Arguments**

- **date** Character. Date of desired log "yyyy-mm-dd". NULL returns date of latest available log.
- **check.url** Logical.
- **repository** Character. "CRAN" or "MRAN". RStudio CRAN mirror log or Microsoft MRAN snapshot.
- **tz** Character. Time zone. See OlsonNames().
- **upload.time** Character. UTC upload time for logs "hh:mm" or "hh:mm:ss".
- **warning.msg** Logical. TRUE uses warning() if the function returns the date of the previous available log.

**Value**

An R date object.

logPostInfo

*Compute Date and Time of Latest Available Log.*

**Description**

GMT and Local Posting Times.

**Usage**

```r
call = logPostInfo(tz = Sys.timezone())
```

**Arguments**

- **tz** Character. Local time zone. See OlsonNames() or use Sys.timezone().
**monthlyLog**

Get CRAN logs for selected month.

**Description**

Compute list of log files, 'lst', for packageVersionPercent().

**Usage**

```r
monthlyLog(yr.mo = "2020-07")
```

**Arguments**

- `yr.mo` Character. "yyyy-mm".

**Note**

This is computationally intensive; you're downloading 30 odd files that are each around 50 MB in size (and creating a ~1.5 GB file)! Parallelization not practical; multiple attempts to connect to website causes problems. Truncates in-progress/future dates to yesterday’s date. Automatically takes care of leap days (e.g., monthlyLog("2020-02").

**packageArchive**

Scrape package data from Archive.

**Description**

Scrape package data from Archive.

**Usage**

```r
packageArchive(package = "cholera", check.package = TRUE, size = FALSE)
```

**Arguments**

- `package` Character. Package name.
- `check.package` Logical. Validate and "spell check" package.
- `size` Logical. Include size of source file.

**Value**

An R data frame or NULL.
packageCountry

Package download counts by country.

Description

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

Usage

packageCountry(packages = "cholera", date = NULL, all.filters = FALSE, ip.filter = FALSE, triplet.filter = FALSE, small.filter = FALSE, sequence.filter = FALSE, size.filter = FALSE, sort = TRUE, na.rm = FALSE, memoization = TRUE, check.package = TRUE)

Arguments

- **packages**: Character. Vector of package name(s).
- **date**: Character. Date. "yyyy-mm-dd". NULL uses latest available log.
- **all.filters**: Logical. Master switch for filters.
- **ip.filter**: Logical.
- **triplet.filter**: Logical.
- **small.filter**: Logical.
- **sequence.filter**: Logical.
- **size.filter**: Logical.
- **sort**: Logical. Sort by download count.
- **na.rm**: Logical. Remove NAs.
- **memoization**: Logical. Use memoization when downloading logs.
- **check.package**: Logical. Validate and "spell check" package.
packageCRAN

Scrape package data from CRAN.

Description

Version, date and size (source file) of most recent publication.

Usage

packageCRAN(package = "cholera", check.package = TRUE, size = FALSE)

Arguments

package Character. Package name.
check.package Logical. Validate and "spell check" package.
size Logical. Include size of source file.

Value

An R data frame or NULL.

Examples

## Not run:
packageCRAN(package = "HistData")
packageCRAN(package = "VR") # No version on CRAN (archived)

## End(Not run)

packageDistribution

Package Download Distribution.

Description

Package Download Distribution.

Usage

packageDistribution(package = "HistData", date = NULL,
all.filters = FALSE, ip.filter = FALSE, triplet.filter = FALSE,
small.filter = FALSE, sequence.filter = FALSE, size.filter = FALSE,
memoization = TRUE, check.package = TRUE, multi.core = TRUE)
packageHistory

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>package</td>
<td>Character. Vector of package name(s).</td>
</tr>
<tr>
<td>date</td>
<td>Character. Date. &quot;yyyy-mm-dd&quot;. NULL uses latest available log.</td>
</tr>
<tr>
<td>all.filters</td>
<td>Logical. Master switch for filters.</td>
</tr>
<tr>
<td>ip.filter</td>
<td>Logical.</td>
</tr>
<tr>
<td>triplet.filter</td>
<td>Logical.</td>
</tr>
<tr>
<td>small.filter</td>
<td>Logical. TRUE filters out downloads less than 1000 bytes.</td>
</tr>
<tr>
<td>sequence.filter</td>
<td>Logical.</td>
</tr>
<tr>
<td>size.filter</td>
<td>Logical.</td>
</tr>
<tr>
<td>memoization</td>
<td>Logical. Use memoization when downloading logs.</td>
</tr>
<tr>
<td>check.package</td>
<td>Logical. Validate and &quot;spell check&quot; package.</td>
</tr>
<tr>
<td>multi.core</td>
<td>Logical or Numeric. TRUE uses parallel::detectCores(). FALSE uses one, single core. You can also specify the number logical cores. Mac and Unix only.</td>
</tr>
</tbody>
</table>

packageHistory

Extract package version history CRAN and Archive.

Description

Date and version of all publications.

Usage

packageHistory(package = "cholera", check.package = TRUE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>package</td>
<td>Character. Package name.</td>
</tr>
<tr>
<td>check.package</td>
<td>Logical. Validate and &quot;spell check&quot; package.</td>
</tr>
</tbody>
</table>
packageLog

Get Package Download Logs.

Description

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

Usage

packageLog(packages = "cholera", date = NULL, all.filters = FALSE, ip.filter = FALSE, triplet.filter = FALSE, small.filter = FALSE, sequence.filter = FALSE, size.filter = FALSE, memoization = TRUE, check.package = TRUE, clean.output = FALSE, multi.core = TRUE)

Arguments

packages Character. Vector of package name(s).
date Character. Date. "yyyy-mm-dd". NULL uses latest available log.
all.filters Logical. Master switch for filters.
ip.filter Logical.
triplet.filter Logical.
small.filter Logical. TRUE filters out downloads less than 1000 bytes.
sequence.filter Logical.
size.filter Logical.
memoization Logical. Use memoization when downloading logs.
check.package Logical. Validate and "spell check" package.
clean.output Logical. NULL row names.
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.

Value

An R data frame.
packageMRAN  
*Extract package data from MRAN (prototype).*

**Description**

Binary or source size.

**Usage**

```r
packageMRAN(package = "cholera", date = NULL, check.package = TRUE, multi.core = TRUE)
```

**Arguments**

- **package**  
  Character. Package name.
- **date**  
  Character. NULL uses latest available log.
- **check.package**  
  Logical. Validate and "spell check" package.
- **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.

**Note**

Depending on when synchronization occurred, you may need to add 3 or 4 days to CRAN publication date, see packageHistory(), to find the package or version you’re looking for.

packageRank  
*Package download counts and rank percentiles (prototype).*

**Description**

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

**Usage**

```r
packageRank(packages = "HistData", date = NULL, all.filters = FALSE, ip.filter = FALSE, triplet.filter = FALSE, small.filter = FALSE, sequence.filter = FALSE, size.filter = FALSE, memoization = TRUE, check.package = TRUE, multi.core = TRUE)
```
packageVersionPercent

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>packages</td>
<td>Character. Vector of package name(s).</td>
</tr>
<tr>
<td>date</td>
<td>Character. Date. &quot;yyyy-mm-dd&quot;. NULL uses latest available log.</td>
</tr>
<tr>
<td>all.filters</td>
<td>Logical. Master switch for filters.</td>
</tr>
<tr>
<td>ip.filter</td>
<td>Logical.</td>
</tr>
<tr>
<td>triplet.filter</td>
<td>Logical.</td>
</tr>
<tr>
<td>small.filter</td>
<td>Logical. TRUE filters out downloads less than 1000 bytes.</td>
</tr>
<tr>
<td>sequence.filter</td>
<td>Logical.</td>
</tr>
<tr>
<td>size.filter</td>
<td>Logical.</td>
</tr>
<tr>
<td>memoization</td>
<td>Logical. Use memoization when downloading logs.</td>
</tr>
<tr>
<td>check.package</td>
<td>Logical. Validate and &quot;spell check&quot; package.</td>
</tr>
<tr>
<td>multi.core</td>
<td>Logical or Numeric. TRUE uses parallel::detectCores(). FALSE uses one, single core. You can also specify the number logical cores. Mac and Unix only.</td>
</tr>
</tbody>
</table>

Value

An R data frame.

Examples

```r
## Not run:
packageRank(packages = "HistData", date = "2020-01-01")
packageRank(packages = c("h2o", "Rcpp", "rstan"), date = "2020-01-01")

## End(Not run)
```

packageVersionPercent  Compute data for versionPlot().

Description

packageRank::blog.data or recompute random sample of packages.

Usage

```r
packageVersionPercent(lst, yr.mo = "2020-07", multi.core = TRUE)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lst</td>
<td>Object. List of CRAN download logs data frames. Use monthlyLog().</td>
</tr>
<tr>
<td>yr.mo</td>
<td>Character. &quot;yyyy-mo&quot;. packageVersionsPercent(NULL, yr.mo)</td>
</tr>
<tr>
<td>multi.core</td>
<td>Logical or Numeric. TRUE uses parallel::detectCores(). FALSE uses one, single core. You can also specify the number logical cores. Mac and Unix only.</td>
</tr>
</tbody>
</table>
Examples

```r
## Not run:
# To resample and recompute, set lst to NULL, specify a yr.mo:
packageVersionPercent(NULL, yr.mo = "2020-07")

Otherwise, you must provide a pre-computed lst of logs.

## End(Not run)
```

## S3 method for class 'annualDownloads'
```r
plot(x, statistic = "count", pool.obs = FALSE,
     log.y = TRUE, nrow = 3, smooth = TRUE, span = 3/4, ...
```

Arguments

- `x` object.
- `statistic` Character. "count" or "percent".
- `pool.obs` Logical.
- `log.y` Logical. Base 10 logarithm of y-axis.
- `nrow` Numeric. Number of rows for ggplot2 facets.
- `smooth` Logical. Add smoother. 2/3 is built-in default.
- `span` Numeric. Smoothing parameter for geom_smooth(); c.f. stats::loess(span).
- `...` Additional plotting parameters.
### Usage

```r
## S3 method for class 'bioconductorDownloads'
plot(x, graphics = NULL,
     count = "download", cumulative = FALSE, points = "auto",
     smooth = FALSE, f = 2/3, span = 3/4, se = FALSE, log.count = FALSE,
     r.version = FALSE, same.xy = TRUE, multi.plot = FALSE,
     legend.loc = "topleft", ...)
```

### Arguments

- `x`: object.
- `graphics`: Character. NULL, "base" or "ggplot2".
- `count`: Character. "download" or "ip".
- `cumulative`: Logical. Use cumulative counts.
- `points`: Character of Logical. Plot points. "auto", TRUE, FALSE. "auto" for bioconductorDownloads(unit.observation = "month") with 24 or fewer months, points are plotted.
- `smooth`: Logical. Add stats::lowess smoother.
- `f`: Numeric. smoother window for stats::lowess(). For graphics = "base" only; c.f. stats::lowess(f)
- `se`: Logical. Works only with graphics = "ggplot2".
- `log.count`: Logical. Logarithm of package downloads.
- `r.version`: Logical. Add R release dates.
- `same.xy`: Logical. Use same scale for multiple packages when graphics = "base".
- `multi.plot`: Logical. Plot all data in a single window frame.
- `legend.loc`: Character.
- `...`: Additional plotting parameters.

### Examples

```r
## Not run:
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))
## End(Not run)
```
plot.bioconductorRank  
*Plot method for bioconductorRank().*

**Description**

Plot method for bioconductorRank().

**Usage**

```r
## 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.countryDistribution  
*Plot top 10 package downloads by country domain.*

**Description**

Plot method for `packageDistribution()`.

**Usage**

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

**Arguments**

- `x`: An object of class "countryDistribution" created by `countryDistribution()`.
- `...`: Additional plotting parameters.
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, statistic = "count", graphics = "auto",
     points = "auto", log.count = FALSE, smooth = FALSE, se = FALSE,
     f = 1/3, span = 3/4, package.version = FALSE, r.version = FALSE,
     population.plot = FALSE, population.seed = as.numeric(Sys.Date()),
     multi.plot = FALSE, same.xy = TRUE, legend.loc = "topleft",
     r.total = FALSE, dev.mode = FALSE, unit.observation = "day",
     multi.core = TRUE, ...)

Arguments

\[ x \]
object.

\[ statistic \]
Character. "count" or "cumulative".

\[ graphics \]
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. Smoother window for stats::lowess(). For graphics = "base" only; c.f. stats::lowess(f).

span
Numeric. Smoothing parameter for geom_smooth(); c.f. stats::loess(span).

package.version
Logical. Add latest package release dates.

r.version
Logical. Add R release dates.

population.plot
Logical. Plot population plot.

population.seed
Numeric. Seed for sample in population plot.

multi.plot
Logical.

same.xy
Logical. Use same scale for multiple packages when graphics = "base".

legend.loc
Character.

r.total
Logical.

dev.mode
Logical. Use packageHistory0() to scrape CRAN.

unit.observation
Character. "year", "month", or "day".

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.

... Additional plotting parameters.

Value
A base R or ggplot2 plot.

Examples

```r
## Not run:
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))
```

## End(Not run)

plot.packageDistribution
Plot method for packageDistribution().

Description
Plot method for packageDistribution().
Usage

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

Arguments

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

Description

Plot method for `packageRank()` and `packageRank0()`.

Usage

```r
## 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

```r
## Not run:
plot(packageRank(packages = "HistData", date = "2020-01-01"))
plot(packageRank(packages = c("h2o", "Rcpp", "rstan"), date = "2020-01-01"))
## End(Not run)
```
plot.packageVersionPercent

Plot method for packageVersionPercent().

Description
Plot method for packageVersionPercent().

Usage
## S3 method for class 'packageVersionPercent'
plot(x, ...)

Arguments
x
An object of class "packageVersions" created by packageVersions().

... Additional plotting parameters.

plot.weeklyDownloads

Plot method for annualDownloads().

Description
Plot method for annualDownloads().

Usage
## S3 method for class 'weeklyDownloads'
plot(x, statistic = "percent",
    aggregation = "day", typical.value = "mean", nrow = 3L, ...)

Arguments
x
object.

statistic Character. "count" or "percent".

aggregation Character. "week" or "day".

typical.value Character. "mean" or "median".

nrow Numeric. Number of rows for ggplot2 facets.

... Additional plotting parameters.

Examples
## Not run:
plot(weeklyDownloads())
plot(weeklyDownloads(n = 9), aggregation = "week")

## End(Not run)
### plotDownloadsCountry

**Plot Compute Downloads by Country Code.**

**Description**

Plot Compute Downloads by Country Code.

**Usage**

```r
plotDownloadsCountry()
```

### plotTopCountryCodes

**Plot Top N Downloads by Country Code.**

**Description**

Plot Top N Downloads by Country Code.

**Usage**

```r
plotTopCountryCodes(dataset = "october", second.place = FALSE)
```

**Arguments**

- `dataset` Character.
- `second.place` Logical. Annotate second place country.

### print.bioconductorDownloads

**Print method for bioconductorDownloads().**

**Description**

Print method for bioconductorDownloads().

**Usage**

```r
## 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()
...

Arguments

Additional parameters.

print.cranDownloads

Print method for cranDownloads().

Description

Print method for cranDownloads().

Usage

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

Arguments

x  object.
...

Arguments

Additional parameters.
print.packageDistribution

Description

Print method for packageDistribution().

Usage

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

Arguments

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

print.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.
sequenceFilter  
*Filter downloads of full-sized sequential versions (prototype).*

**Description**
Filter downloads of full-sized sequential versions (prototype).

**Usage**
```
sequenceFilter(pkg.data, arch.pkg.history, download.time = 30)
```

**Arguments**
- **pkg.data** Object.
- **arch.pkg.history** Object.
- **download.time** Numeric. Package download time allowance (seconds).

sizeFilter  
*Filter out size anomalies (prototype).*

**Description**
Logs from RStudio’s CRAN Mirror http://cran-logs rstudio.com/

**Usage**
```
sizeFilter(dat, packages, cores)
```

**Arguments**
- **dat** Object. Package log entries.
- **packages** Character. Vector of package name(s).
- **cores** Integer. Number of cores for parallelization.
smallFilter

Filter out small downloads (prototype).

**Description**

Filter out small downloads (prototype).

**Usage**

```r
goingToTest\(\text{smallFilter}(\text{dat}, \text{threshold} = 1000L)\)
```

**Arguments**

- **dat**: Object. Package log entries.
- **threshold**: Numeric. Bytes.

summary.bioconductorDownloads

Summary method for bioconductorDownloads().

**Description**

Summary method for bioconductorDownloads().

**Usage**

```r
\text{## S3 method for class} \text{`bioconductorDownloads'}
\text{summary(}\text{object, ...}\text{)}
```

**Arguments**

- **object**: Object.
- **...**: Additional parameters.
**summary.bioconductorRank**  
*Summary method for bioconductorRank().*

**Description**  
Summary method for bioconductorRank().

**Usage**  
```r  
## 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**  
```r  
## 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**

```r
## 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.

---

### topCountryCodes

**Compute Top N Downloads by Country Code.**

**Description**

Compute Top N Downloads by Country Code.

**Usage**

```r
topCountryCodes(month_cran_log, top.n = 5L, multi.core = TRUE)
```

**Arguments**

- `month_cran_log` Object.
- `top.n` Integer.
- `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.
tripletFilter  
*Filter out small downloads triplets (prototype).*

**Description**

Logs from RStudio’s CRAN Mirror http://cran-logs.rstudio.com/

**Usage**

`tripletFilter(dat, time.window = 2, multi.core = TRUE)`

**Arguments**

- **dat**: Object. Package log entries.
- **time.window**: Numeric. Seconds.
- **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.

utc  
*Compute Coordinated Universal Time (UTC/GMT) for Your Local Time.*

**Description**

Compute Coordinated Universal Time (UTC/GMT) for Your Local Time.

**Usage**

`utc()`

utc0  
*Compute Coordinated Universal Time (UTC/GMT) for Specified Local Time.*

**Description**

Compute Coordinated Universal Time (UTC/GMT) for Specified Local Time.

**Usage**

`utc0(date = "2020-01-01", time = "12:00:00", tz = "Europe/Vienna")`

**Arguments**

- **date**: Character. Date "yyyy-mm-dd".
- **time**: Character. Local time "hh:mm" or "hh:mm:ss".
- **tz**: Character. Local time zone. See OlsonNames() or use Sys.timezone().
versionPlot  

Version Plot.

Description
Document code for blog graph.

Usage
versionPlot()

weeklyDownloads  Sample Weekly CRAN Downloads Data.

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

Usage
weeklyDownloads(start.yr = 2013, n = 50, multi.core = TRUE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>start.yr</td>
<td>Numeric or Integer. Number of weeks (samples).</td>
</tr>
<tr>
<td>n</td>
<td>Numeric or Integer.</td>
</tr>
<tr>
<td>multi.core</td>
<td>Logical or Numeric. TRUE uses parallel::detectCores(). FALSE uses one, single core. You can also specify the number logical cores. Mac and Unix only.</td>
</tr>
</tbody>
</table>
Index

datasets
- blog.data, 6
  - annualDownloads, 3
  - archivePackages, 4
  - bioconductorDownloads, 4
  - bioconductorRank, 5
  - blog.data, 6
  - countryDistribution, 7
  - countryPackage, 8
  - countsRanks, 9
  - cranDownloads, 9
  - cranInflationPlot, 10
  - cranMirrors, 10
  - cranPackages, 11
  - cranPackageSize, 11
  - currentTime, 12
  - downloadsCountry, 12
  - fetchCranLog, 13
  - filteredDownloads, 13
  - inflationPlot, 14
  - inflationPlot2, 14
  - ipCount, 15
  - ipDownloads, 15
  - ipFilter, 16
  - ipPackage, 16
  - localTime, 17
  - logDate, 18
  - logPostInfo, 18
  - monthlyLog, 19
  - packageArchive, 19
  - packageCountry, 20
  - packageCRAN, 21
  - packageDistribution, 21
  - packageHistory, 22
  - packageLog, 23
  - packageMRAN, 24
  - packageRank, 24
  - packageVersionPercent, 25
  - plot.annualDownloads, 26
  - plot.bioconductorDownloads, 26
  - plot.bioconductorRank, 28
  - plot.countryDistribution, 28
  - plot.countsRanks, 29
  - plot.cranDownloads, 29
  - plot.packageDistribution, 30
  - plot.packageRank, 31
  - plot.packageVersionPercent, 32
  - plot.weeklyDownloads, 32
  - plotDownloadsCountry, 33
  - plotTopCountryCodes, 33
  - print.bioconductorDownloads, 33
  - print.bioconductorRank, 34
  - print.cranDownloads, 34
  - print.packageDistribution, 35
  - print.packageRank, 35
  - sequenceFilter, 36
  - sizeFilter, 36
  - smallFilter, 37
  - summary.bioconductorDownloads, 37
  - summary.bioconductorRank, 38
  - summary.cranDownloads, 38
  - summary.packageRank, 39
  - topCountryCodes, 39
  - tripletFilter, 40
  - utc, 40
  - utc0, 40
  - versionPlot, 41
  - weeklyDownloads, 41