

# Package ‘Visualize.CRAN.Downloads’

March 21, 2020

**Type** Package

**Title** Visualize Downloads from 'CRAN' Packages

**Version** 1.0

**Date** 2020-03-19

**Author** Marcelo Ponce [aut, cre]

**Maintainer** Marcelo Ponce <mponce@scinet.utoronto.ca>

**Description** Visualize the trends and historical downloads from packages in the 'CRAN' repository. Data is obtained by using the 'API' to query the database from the 'RStudio' 'CRAN' mirror.

**Imports** graphics, stats, cranlogs, plotly, htmlwidgets

**Suggests** knitr, devtools, roxygen2, testthat

**License** GPL (>= 2)

**URL** <https://github.com/mponce0/Visualize.CRAN.Downloads>

**BugReports** <https://github.com/mponce0/Visualize.CRAN.Downloads/issues>

**RoxygenNote** 6.1.99.9001

**VignetteBuilder** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2020-03-21 15:40:02 UTC

## R topics documented:

interactivePlots . . . . .	2
movingFn . . . . .	3
processPckg . . . . .	3
retrievePckgData . . . . .	4
staticPlots . . . . .	5
summaries . . . . .	6
<b>Index</b>	<b>7</b>

---

interactivePlots	<i>function that generates interactive plots of the package downloads logs from CRAN</i>
------------------	--

---

### Description

function that generates interactive plots of the package downloads logs from CRAN

### Usage

```
interactivePlots(  
  downloads.data,  
  mytitle = paste(downloads.data$package[1], "Package downloads counts"),  
  nbrPlts = 2,  
  month.ln = 30,  
  HTMLfile = paste0("Interactive_DWNLDS_", downloads.data$package[1], ".html"),  
  device = "HTML",  
  dirSave = NULL  
)
```

### Arguments

downloads.data	total downloads from the package
mytitle	optional char argument specifying the title to be displayed
nbrPlts	optional numeric argument specifying number of plots to generate
month.ln	optional numeric argument specifying the length of the month in days
HTMLfile	an optional string argument specifying the name of the file where to save the plots
device	an optional string describing whether the interactive plot will be set to screen or to save in an HTML file
dirSave	specify a valid directory where to save the plot

### Examples

```
packageXdownloads <- retrievePckgData("ggplot")[[1]]  
interactivePlots(packageXdownloads)
```

---

movingFn	<i>generic fn that computes the "fn" on a moving window</i>
----------	---

---

**Description**

generic fn that computes the "fn" on a moving window

**Usage**

```
movingFn(x, fn = mean, period = length(x), direction = "forward")
```

**Arguments**

x	a numeric vector
fn	a function to be applied/computed, default is set to mean()
period	size of the "moving window", default set to the length of the vector
direction	type of moving average to consider: "forward", "centered", "backward"; ie. whether the window computation is ( "centered" / "forward" / "backward" ) wrt the data series

**Value**

a vector with the 'moving operation' applied to the x vector

---

processPckg	<i>main function to analyze a list of packages in a given time frame</i>
-------------	--

---

**Description**

main function to analyze a list of packages in a given time frame

**Usage**

```
processPckg(  
  pkg.lst,  
  t0 = lastyear.date(),  
  t1 = today(),  
  opts = list(),  
  device = "PDF",  
  dirSave = NULL  
)
```

**Arguments**

pckg.lst	list of packages to process
t0	initial date, beginning of the time period given in "YYYY-MM-DD" format
t1	final date, ending of the time period given in "YYYY-MM-DD" format
opts	a list of different options available for customizing the output
device	string to select the output format: 'PDF'/'PNG'/'JPEG' or 'screen'
dirSave	name of a valid directory where to save the file, eg. do not specify this argument or enter "." for using the current working directory

**Examples**

```
# device is set to "screen" so no files are generated and plots will appear on "screen"
# alternative to 'device' are "PDF"/"PNG"/"JPEG"
processPckg("ehelp", device="screen")
processPckg(c("ehelp", "plotly", "ggplot2"), "2001-01-01", device="screen")
processPckg(c("ehelp", "plotly", "ggplot2"), "2001-01-01", opts="nostatic", device="screen")
processPckg(c("ehelp", "plotly", "ggplot2"), "2001-01-01",
opts=c("nostatic", "nocombined", "nointeractive"), device="screen")
```

---

retrievePckgData	<i>function to download the data from the CRAN logs for an specific package</i>
------------------	---

---

**Description**

function to download the data from the CRAN logs for an specific package

**Usage**

```
retrievePckgData(pckg = NULL, t0 = lastyear.date(), t1 = today())
```

**Arguments**

pckg	is the name of the package to look for the downloads data
t0	is the initial date
t1	is the final date

**Value**

a list composed of the stats from the original time frame and the last month

**Examples**

```
retrievePckgData("ehelp")
retrievePckgData("ehelp", "2018-01-01", "2020-01-01")
```

---

staticPlots	<i>function that generates visual trends of the package downloads logs from CRAN, it will generate 4 plots: two histograms, a pulse plot and the main plot is a plot of the downloads as a function of time</i>
-------------	---

---

**Description**

function that generates visual trends of the package downloads logs from CRAN, it will generate 4 plots: two histograms, a pulse plot and the main plot is a plot of the downloads as a function of time

**Usage**

```
staticPlots(
  pkg.stats.total,
  device = "PDF",
  fileName = paste0("DWNLDS_", pkg.stats.total$package[1], ".", tolower(device)),
  dirSave = NULL,
  combinePlts = FALSE,
  noMovAvg = FALSE,
  noConfBands = FALSE,
  cutOff.pts = 250,
  dbg = FALSE
)
```

**Arguments**

pkg.stats.total	total downloads from the package
device	string to select the output format: 'PDF'/'PNG'/'JPEG' or 'screen'
fileName	an optional string argument specifying the name of the file where to save the plots
dirSave	specify a valid directory where to save the plot
combinePlts	a boolean indicating whether the plots generated will be combined into one single figure or not
noMovAvg	a boolean indicating whether moving statistical estimators, such as, the moving average will be displayed
noConfBands	a boolean indicating whether a confidence band will be displayed
cutOff.pts	an integer value indicating the cut-off value to determine whether there would be a subsample for clarity sake in the plots
dbg	internal flag for activating debugging options, i.e. display further information in screen

## Examples

```
packageData <- retrievePckgData("ggplot2")
totalDownloads <- packageData[[1]]
#lastmonthDownloads <- packageData[[2]]
staticPlots(totalDownloads, device="screen")
staticPlots(totalDownloads,combinePlts=TRUE, device="screen")
```

---

summaries

*function to display the summary of the data*

---

## Description

function to display the summary of the data

## Usage

```
summaries(data1, deltaTs = 30)
```

## Arguments

data1	first dataset, eg. total data
deltaTs	a numerical (integer) value, indicating the lenght –in days– for selecting a subset of the original dataset; default value is 1 mont, ie. 30 days

## Examples

```
packageXdownloads <- retrievePckgData("ehelp")[[1]]
summaries(packageXdownloads)
```

# Index

[interactivePlots](#), [2](#)

[movingFn](#), [3](#)

[processPckg](#), [3](#)

[retrievePckgData](#), [4](#)

[staticPlots](#), [5](#)

[summaries](#), [6](#)