Package ‘deepdep’

June 7, 2021

Title  Visualise and Explore the Deep Dependencies of R Packages

Version  0.2.5.3

Description  Provides tools for exploration of R package dependencies.
    The main deepdep() function allows to acquire deep dependencies of any package and plot them in an elegant way.
    It also adds some popularity measures for the packages e.g. in the form of download count through the 'cranlogs' package.
    Other data acquire functions are: get_dependencies(), get_downloads() and get_description().
    The deepdep_shiny() function runs shiny application that helps to produce a nice ‘deepdep’ plot.

License  GPL-3

Encoding  UTF-8

LazyData  false

RoxygenNote  7.1.1

Depends  R (>= 3.2.0)

Imports  cranlogs, ggforce, ggplot2, ggraph, graphlayouts, httr, igraph, jsonlite, scales

Suggests  BiocManager, shiny, shinycssloaders, covr, devtools, knitr, miniCRAN, plyr, rmarkdown, spelling, stringi, testthat (>= 2.1.0), vcr

VignetteBuilder  knitr

URL  https://dominikrafacz.github.io/deepdep/,
    https://github.com/DominikRafacz/deepdep

BugReports  https://github.com/DominikRafacz/deepdep/issues

Language  en-GB

NeedsCompilation  no

Author  Dominik Rafacz [aut, cre] (<https://orcid.org/0000-0003-0925-1909>),
        Hubert Baniecki [aut],
        Szymon Maksymiuk [aut],
        Mateusz Bakala [aut]
Maintainer Dominik Rafacz <dominikrafacz@gmail.com>

Repository CRAN

Date/Publication 2021-06-07 11:10:02 UTC

R topics documented:

- deepdep ........................................... 2
- deepdep_shiny .................................... 4
- get_available_packages .......................... 4
- get_dependencies ................................. 5
- get_description .................................. 6
- get_downloads .................................... 7
- plot_dependencies ............................... 8
- plot_downloads ................................... 10
- print.available_packages ....................... 11
- print.deepdep ................................... 12
- print.package_dependencies .................... 12
- print.package_description ..................... 13
- print.package_downloads ....................... 13

Index 15

---

deepdep

Acquire the dependencies of the package on any depth level

## Description

This function is an ultimate wrapper for `get_dependencies`. It inherits all of the arguments and allows to recursively search for the dependencies at the higher level of depth.

## Usage

```
deepdep(
  package,
  depth = 1,
  downloads = FALSE,
  bioc = FALSE,
  local = FALSE,
  dependency_type = c("Depends", "Imports")
)
```
**Arguments**

- **package**: A character. Name of the package that is on CRAN, Bioconductor repository or locally installed. See bioc and local arguments.

- **depth**: An integer. Maximum depth level of the dependency. By default it’s 1.

- **downloads**: A logical. If TRUE add dependency downloads data. By default it’s FALSE.

- **bioc**: A logical value. If TRUE the Bioconductor dependencies data will be taken from the Bioconductor repository. For this option to work properly, BiocManager package needs to be installed.

- **local**: A logical value. If TRUE only data of locally installed packages will be used (without API usage).

- **dependency_type**: A character vector. Types of the dependencies that should be sought. Possibilities are: "Imports", "Depends", "Suggests", "Enhances", "LinkingTo". By default it’s "Depends", "Imports".

**Value**

An object of deepdep class.

**See Also**

- `get_dependencies`

**Examples**

``` r
library(deepdep)

dd_downloads <- deepdep("ggplot2")
head(dd_downloads)

dd_2 <- deepdep("ggplot2", depth = 2, downloads = TRUE)
plot_dependencies(dd_2, "circular")

dd_local <- deepdep("deepdep", local = TRUE)
plot_dependencies(dd_local)
```
**get_available_packages**

---

**deepdep_shiny**  
*Run Shiny app*

**Description**

This function runs shiny app that helps to produce nice deepdep plot.

**Usage**

```r
deepdep_shiny()
```

---

**get_available_packages**  
*Get the list of available packages*

**Description**

Get names of packages that you have locally installed or that are available to be installed.

**Usage**

```r
get_available_packages(bioc = FALSE, local = FALSE, reset_cache = FALSE)
```

**Arguments**

- `bioc`  
  A logical value. If TRUE the Bioconductor dependencies data will be taken from the Bioconductor repository. For this option to work properly, `BiocManager` package needs to be installed.

- `local`  
  A logical value. If TRUE only data of locally installed packages will be used (without API usage).

- `reset_cache`  
  A logical value. If TRUE the cache will be cleared before obtaining the list of packages.

**Details**

Function uses caching - only the first usage scraps information from servers. Those objects are then saved locally in temporary file and further usages loads needed data from the file.

Arguments `bioc` and `local` cannot be TRUE simultaneously. If neither `local` nor `bioc` are TRUE, vector contains all packages available currently on CRAN. If `bioc` is TRUE, vector contains all packages available currently on CRAN and via Bioconductor. If `local` is TRUE, vector contains all of the packages that are currently installed.

**Value**

A character vector.
**get_dependencies**

*Acquire the dependencies of the package*

**Examples**

```r
library(deepdep)

av <- get_available_packages()
head(av)
```

**Description**

This function uses `get_description` and `get_downloads` to acquire the dependencies of the package (with their downloads).

**Usage**

```r
get_dependencies(  
  package,  
  downloads = TRUE,  
  bioc = FALSE,  
  local = FALSE,  
  dependency_type = c("Depends", "Imports")
)
```

**Arguments**

- `package` A character. Name of the package that is on CRAN, Bioconductor repository or locally installed. See `bioc` and `local` arguments.
- `downloads` A logical. If `TRUE` add package downloads data. By default it's `TRUE`.
- `bioc` A logical value. If `TRUE` the Bioconductor dependencies data will be taken from the Bioconductor repository. For this option to work properly, `BiocManager` package needs to be installed.
- `local` A logical value. If `TRUE` only data of locally installed packages will be used (without API usage).
- `dependency_type` A character vector. Types of the dependencies that should be sought. Possibilities are: "Imports", "Depends", "Suggests", "Enhances", "LinkingTo". By default it's "Depends", "Imports".

**Value**

An object of `package_dependencies` class.
get_description

See Also

get_description get_downloads

Examples

library(deepdep)

dependencies <- get_dependencies("htmltools", downloads = FALSE)
dependencies

dependencies_local <- get_dependencies("deepdep", downloads = FALSE, local = TRUE)
dependencies_local

get_description Scrap the DESCRIPTION file and CRAN metadata of the package

Description

This function uses api of CRAN Data Base to scrap the DESCRIPTION file and CRAN metadata of the package. It caches the results to speed the computation process.

Usage

get_description(package, bioc = FALSE, local = FALSE, reset_cache = FALSE)

Arguments

package A character. Name of the package that is on CRAN, Bioconductor repository or locally installed. See bioc and local arguments.

bioc A logical value. If TRUE the Bioconductor dependencies data will be taken from the Bioconductor repository. For this option to work properly, BiocManager package needs to be installed.

local A logical value. If TRUE only data of locally installed packages will be used (without API usage).

reset_cache A logical value. If TRUE the cache will be cleared before obtaining the list of packages.

Value

An object of package_description class.
get_downloads

Examples

```r
library(deepdep)

description <- get_description("ggplot2")
description

description_local <- get_description("deepdep", local = TRUE)
description_local
```

---

<table>
<thead>
<tr>
<th>get_downloads</th>
<th>Scrap the download data of the package</th>
</tr>
</thead>
</table>

Description

This function uses API of CRAN Logs to scrap the download logs of the package.

Usage

```r
get_downloads(package)
```

Arguments

```r
table
```

Value

An object of package_downloads class.

Examples

```r
library(deepdep)

downloads <- get_downloads("ggplot2")
downloads
```
**plot_dependencies**

*Main plot function for a deepdep object*

**Description**

Visualize dependency data from a deepdep object using ggplot2 and ggraph packages. Several tree-like layouts are available.

**Usage**

```r
plot_dependencies(
  x,
  type = "circular",
  same_level = FALSE,
  reverse = FALSE,
  label_percentage = 1,
  show_version = FALSE,
  show_downloads = FALSE,
  ...
)
```

```r
## Default S3 method:
plot_dependencies(
  x,
  type = "circular",
  same_level = FALSE,
  reverse = FALSE,
  label_percentage = 1,
  show_version = FALSE,
  show_downloads = FALSE,
  ...
)
```

```r
## S3 method for class 'character'
plot_dependencies(
  x,
  type = "circular",
  same_level = FALSE,
  reverse = FALSE,
  label_percentage = 1,
  show_version = FALSE,
  show_downloads = FALSE,
  ...
)
```

```r
## S3 method for class 'deepdep'
plot_dependencies(
```

```r
```
\texttt{plot\_dependencies}

\begin{verbatim}
x,  
  type = "circular",
  same_level = FALSE,
  reverse = FALSE,
  label_percentage = 1,
  show_version = FALSE,
  show_downloads = FALSE,

  ...
)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{x} \hspace{1cm} A \texttt{deepdep} object or a character package name.
  \item \texttt{type} \hspace{1cm} A character. Possible values are \texttt{circular} and \texttt{tree}.
  \item \texttt{same_level} \hspace{1cm} A logical. If \texttt{TRUE} links between dependencies on the same level will be added. By default it’s \texttt{FALSE}.
  \item \texttt{reverse} \hspace{1cm} A logical. If \texttt{TRUE} links between dependencies pointing from deeper level to more shallow level will be added. By default it’s \texttt{FALSE}.
  \item \texttt{label\_percentage} \hspace{1cm} A numeric value between 0 and 1. A fraction of labels to be displayed. By default it’s 1 (all labels displayed).
  \item \texttt{show\_version} \hspace{1cm} A logical. If \texttt{TRUE} required version of package will be displayed below package name. Defaults to \texttt{FALSE}.
  \item \texttt{show\_downloads} \hspace{1cm} A logical. If \texttt{TRUE} total number of downloads of packages will be displayed below package names. Defaults to \texttt{FALSE}.
  \item \texttt{...} \hspace{1cm} Other arguments passed to the \texttt{deepdep} function.
\end{itemize}

\textbf{Value}

A \texttt{ggplot2,gg,ggraph,deepdep\_plot} class object.

\textbf{Examples}

\begin{verbatim}
library(deepdep)

# use local packages
plot_dependencies(\"deepdep\", depth = 2, local = TRUE)

dd <- deepdep(\"ggplot2\")
plot_dependencies(dd, \"tree\")

dd2 <- deepdep(\"ggplot2\", depth = 2)
plot_dependencies(dd2, \"circular\")

# show grand_total download count
plot_dependencies(\"shiny\", show_downloads = TRUE)
\end{verbatim}
**plot_downloads**  
*Plot download count of CRAN packages.*

**Description**

This function uses API of CRAN Logs to scrap the download logs of the packages and then plots the data. It works on objects of class character (vector), deepdep, package_dependencies and package_downloads.

**Usage**

```
plot_downloads(x, ...)  
## Default S3 method:  
plot_downloads(x, ...)  
## S3 method for class 'deepdep'  
plot_downloads(x, from = Sys.Date() - 365, to = Sys.Date(), ...)  
## S3 method for class 'package_dependencies'  
plot_downloads(x, from = Sys.Date() - 365, to = Sys.Date(), ...)  
## S3 method for class 'package_downloads'  
plot_downloads(x, from = Sys.Date() - 365, to = Sys.Date(), ...)  
## S3 method for class 'character'  
plot_downloads(x, from = Sys.Date() - 365, to = Sys.Date(), ...)  
```

**Arguments**

- `x` A character vector. Names of the packages that are on CRAN.
- `...` Ignored.
- `from` A Date class object. From which date plot the data. By default it’s one year back.
- `to` A Date class object. To which date plot the data. By default it’s now.

**Value**

A ggplot2 class object.
print.available_packages

Examples

library(deepdep)

plot_downloads("htmltools")

dd <- deepdep("ggplot2")
plot_downloads(dd)

print.available_packages

Print function for an object of available_packages class

Description

Print function for an object of available_packages class

Usage

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

Arguments

x An object of available_packages class.

... other

Examples

library(deepdep)

av <- get_available_packages()
head(av)
print.deepdep

Description
Print function for an object of deepdep class

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

Arguments
x  An object of deepdep class.
... other

Examples

library(deepdep)

dd <- deepdep("stringr")
dd

print.package_dependencies

Description
Print function for an object of package_dependencies class

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

Arguments
x  An object of package_dependencies class.
... other
Examples

library(deepdep)

get_dependencies("htmltools", downloads = TRUE)

print.package_description

Description
Print function for an object of package_description class

Usage

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

Arguments

x An object of package_description class.
...
other

Examples

library(deepdep)

description <- get_description("ggplot2")
description

print.package_downloads

Description
Print function for an object of package_downloads class
print.package_downloads

Usage

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

Arguments

- `x` An object of package_downloads class.
- `...` other

Examples

```r
library(deepdep)

desc <- get_downloads("stringr")
desc
```
Index

deepprop, 2, 9
deepdep_shiny, 4

get_available_packages, 4
get_dependencies, 2, 3, 5
get_description, 5, 6, 6
get_downloads, 5, 6, 7

plot_dependencies, 8
plot_downloads, 10
print.available_packages, 11
print.deepdep, 12
print.package_dependencies, 12
print.package_description, 13
print.package_downloads, 13