Package ‘deepdep’

October 13, 2022

**Title**  Visualise and Explore the Deep Dependencies of R Packages

**Version**  0.4.1

**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

**RoxygenNote**  7.1.2

**Depends**  R (>= 3.2.0)

**Imports**  cranlogs, httr, jsonlite

**Suggests**  BiocManager, covr, devtools, ggplot2, ggraph, graphlayouts, igrigraph, knitr, miniCRAN, plyr, rmarkdown, scales, shiny, shinyessloaders, 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], Laura Bakala [aut], Dirk Eddelbuettel [ctb]
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

```r
deepdep(
  package,
  depth = 1,
  downloads = FALSE,
  bioc = FALSE,
  local = FALSE,
  dependency_type = "strong"
)
```
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, a subset of c("Imports", "Depends", "LinkingTo", "Suggests", "Enhances"). Other possibilities are: character string "all", a shorthand for the whole vector; character string "most" for the same vector without "Enhances"; character string "strong" (default) for the first three elements of that vector. Works analogously to package_dependencies.

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)
```
**deepdep_shiny**  
*Run Shiny app*

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

**Usage**
```r
deepep_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**

**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,  # A character. Name of the package that is on CRAN, Bioconductor repository or locally installed. See bioc and local arguments.
  downloads = TRUE,  # A logical. If TRUE add package downloads data. By default it's TRUE.
  bioc = FALSE,  # 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 = FALSE,  # A logical value. If TRUE only data of locally installed packages will be used (without API usage).
  dependency_type = "strong"  # A character vector. Types of the dependencies that should be sought, a subset of c("Imports", "Depends", "LinkingTo", "Suggests", "Enhances"). Other possibilities are: character string "all", a shorthand for the whole vector; character string "most" for the same vector without "Enhances"; character string "strong" (default) for the first three elements of that vector. Works analogously to `package_dependencies`.
)
```

**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, a subset of c("Imports", "Depends", "LinkingTo", "Suggests", "Enhances"). Other possibilities are: character string "all", a shorthand for the whole vector; character string "most" for the same vector without "Enhances"; character string "strong" (default) for the first three elements of that vector. Works analogously to `package_dependencies`. 
get_description

Value
An object of package_dependencies class.

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

library(deepdep)

description <- get_description("ggplot2")
description

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

get_downloads Scrap the download data of the package

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

Usage
get_downloads(package)

Arguments

package A character. Name of the package that is on CRAN.

Value
An object of package_downloads class.

Examples

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,
  show_stamp = TRUE,
  declutter = FALSE,
  ...
)
```

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

  declutter = FALSE,
  ... 
)

## S3 method for class 'deepdep'
plot_dependencies(
  x,
  type = "circular",
  same_level = FALSE,
  reverse = FALSE,
  label_percentage = 1,
  show_version = FALSE,
  show_downloads = FALSE,
  show_stamp = TRUE,
  declutter = FALSE,
  ...
)

Arguments

x A deepdep object or a character package name.
type A character. Possible values are circular and tree.
same_level A logical. If TRUE links between dependencies on the same level will be added. By default it's FALSE.
reverse A logical. If TRUE links between dependencies pointing from deeper level to more shallow level will be added. By default it's FALSE.
label_percentage A numeric value between 0 and 1. A fraction of labels to be displayed. By default it's 1 (all labels displayed).
show_version A logical. If TRUE required version of package will be displayed below package name. Defaults to FALSE.
show_downloads A logical. If TRUE total number of downloads of packages will be displayed below package names. Defaults to FALSE.
show_stamp A logical. If TRUE (the default) the package version and plot creation time will be added
declutter A logical. If TRUE then all layers beyond the first one ignore non-strong dependencies (i.e. "Suggests" and "Enhances"). This visualizes the so-called "hard costs of weak suggests".
...
Other arguments passed to the deepdep function.

Value

A ggplot2, gg, ggraph, deepdep_plot class object.
Examples

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)

plot_downloads

\textit{Plot download count of CRAN packages.}

Description

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

Usage

\texttt{plot_downloads(x, \ldots)}

## Default S3 method:
\texttt{plot_downloads(x, \ldots)}

## S3 method for class 'deepdep'
\texttt{plot_downloads(x, from = Sys.Date() - 365, to = Sys.Date(), \ldots)}

## S3 method for class 'package_dependencies'
\texttt{plot_downloads(x, from = Sys.Date() - 365, to = Sys.Date(), \ldots)}

## S3 method for class 'package_downloads'
\texttt{plot_downloads(x, from = Sys.Date() - 365, to = Sys.Date(), \ldots)}

## S3 method for class 'character'
\texttt{plot_downloads(x, from = Sys.Date() - 365, to = Sys.Date(), \ldots)}
Arguments

\(~x~\) A character vector. Names of the packages that are on CRAN.

\(~\ldots\) Ignored.

\(~\text{from}~\) A \text{Date} class object. From which date plot the data. By default it’s one year back.

\(~\text{to}~\) A \text{Date} class object. To which date plot the data. By default it’s now.

Value

A \text{ggplot2} class object.

Examples

library(deepdep)

plot_downloads("htmltools")

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

print.available_packages

\textit{Print function for an object of available\_packages class}

Description

Print function for an object of available\_packages class

Usage

\texttt{## S3 method for class \textquote{\texttt{available\_packages}}}

\texttt{print(x, \ldots)}

Arguments

\(~x~\) An object of available\_packages class.

\(~\ldots\) other
Examples

```r
library(deepdep)

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

print.deepdep

Print function for an object of deepdep class

Description

Print function for an object of deepdep class

Usage

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

Arguments

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

Examples

```r
library(deepdep)

dd <- deepdep("stringr")
dd
```

print.package_dependencies

Print function for an object of package_dependencies class

Description

Print function for an object of package_dependencies class
print.package_description

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

Print function for an object of package_description class

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
S3 method for class 'package_downloads'

print(x, ...)

Arguments

x An object of package_downloads class.

... other

Examples

library(deepdep)

desc <- get_downloads("stringr")
desc
Index

deepdep, 2, 9
deepdep_shiny, 4

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

package_dependencies, 3, 5
plot_dependencies, 8
plot_downloads, 10
print.available_packages, 11
print.deepdep, 12
print.package_dependencies, 12
print.package_description, 13
print.package_downloads, 14