Package 'pkgstats'

October 11, 2026

Title Metrics of R Packages

Version 0.2.5

Description Static code analyses for R packages using the external code-tagging libraries 'ctags' and 'gtags'. Static analyses enable packages to be analysed very quickly, generally a couple of seconds at most. The package also provides access to a database generating by applying the main function to the full 'CRAN' archive, enabling the statistical properties of any package to be compared with all other 'CRAN' packages.

License GPL-3

URL https://docs.ropensci.org/pkgstats/, https://github.com/ropensci-review-tools/pkgstats

BugReports https://github.com/ropensci-review-tools/pkgstats/issues

Imports brio, checkmate, dplyr, fs, igraph, memoise, methods, readr, roxygen2, rprojroot, sys, withr

Suggests callr, curl, hms, httr2, jsonlite, knitr, parallel, piggyback, pkgbuild, quarto, Rcpp, rmarkdown, testthat (>= 3.0.0), tibble, toml

LinkingTo cpp11

VignetteBuilder knitr, quarto

Config/testthat/edition 3

Config/ropensci/maintainer staff

Encoding UTF-8

Language en-US

NeedsCompilation yes

SystemRequirements ctags, global

Config/roxygen2/version 8.1.0

Author Mark Padgham [aut, cre] (ORCID: <https://orcid.org/0000-0003-2172-5265>), Michael Sumner [ctb] (ORCID: <https://orcid.org/0000-0002-2471-7511>), Jeffrey Hollister [ctb] (ORCID:

<https://orcid.org/0000-0002-9254-9740>),
Egor Kotov [ctb] (ORCID: <https://orcid.org/0000-0001-6690-5345>)

Maintainer Mark Padgham <mark.padgham@email.com>

Repository CRAN

Date/Publication 2026-10-11 11:20:02 UTC

Contents

ctags_install 2
ctags_test 3
desc_stats 4
dl_pkgstats_data 4
extract_tarball 5
loc_stats 6
pkgstats 6
pkgstats_cran_current_from_full 7
pkgstats_fns_from_archive 8
pkgstats_fns_update 9
pkgstats_fn_names 11
pkgstats_from_archive 11
pkgstats_summary 13
pkgstats_update 14
plot_network 15
rd_stats 16
tags_data 16

Index **18**

`ctags_install` *Install 'ctags' from a clone of the 'git' repository*

Description

'ctags' is installed with this package on both Windows and macOS systems; this is an additional function to install from source on Unix systems.

Usage

```
ctags_install(bin_dir = NULL, sudo = TRUE)
```

Arguments

`bin_dir`	This parameter only has an effect on *nix-type operating systems (such as Linux), on which it's a prefix to pass to the `autoconf` configure command defining location to install the binary, with default of `/usr/local`.
`sudo`	Set to `FALSE` if `sudo` is not available, in which case a value for `bin_dir` will also have to be explicitly specified, and be a location where a binary is able to be installed without `sudo` privileges.

Value

Nothing; the function will fail if installation fails, otherwise returns nothing.

See Also

Other tags: ctags_test(), tags_data()

Examples

```
## Not run:
ctags_install (bin_dir = "/usr/local") # default Linux location.

## End(Not run)
```

`ctags_test` *test a 'ctags' installation*

Description

This uses the example from https://github.com/universal-ctags/ctags/blob/master/man/ctags-lang-r.7.rst.in and also checks the GNU global installation.

Usage

```
ctags_test(quiet = TRUE, noerror = FALSE)
```

Arguments

quiet	If TRUE, display on screen whether or not 'ctags' is correctly installed.
noerror	If FALSE (default), this function will error if either 'ctags' or 'gtags' are not installed. If TRUE, the function will complete without erroring, and issue appropriate messages regarding required but non-installed system libraries.

Value

'TRUE' or 'FALSE' respectively indicating whether or not 'ctags' is correctly installed.

See Also

Other tags: ctags_install(), tags_data()

Examples

```
# The function errors if not ctags or gtags found.

ctags_okay <- !is.null (tryCatch (
    ctags_test (),
    error = function (e) NULL
))
```

desc_stats *Statistics from DESCRIPTION files*

Description

Statistics from DESCRIPTION files

Usage

```
desc_stats(path)
```

Arguments

path	Directory to source code of package being analysed

Value

A data.frame with one row and 16 columns extracting various information from the 'DESCRIPTION' file, include websites, tallies of different kinds of authors and contributors, and package dependencies.

See Also

Other stats: loc_stats(), pkgstats(), pkgstats_summary(), rd_stats()

Examples

```
f <- system.file ("extdata", "pkgstats_9.9.tar.gz", package = "pkgstats")
# have to extract tarball to call function on source code:
path <- extract_tarball (f)
desc_stats (path)
```

dl_pkgstats_data *Download latest version of 'pkgstats' data*

Description

Download latest version of 'pkgstats' data

Usage

```
dl_pkgstats_data(current = TRUE, path = tempdir(), quiet = FALSE)
```

Arguments

current	If 'FALSE', download data for all CRAN packages ever released, otherwise (default) download data only for current CRAN packages.
path	Local path to download file.
quiet	If FALSE, display progress information on screen.

Value

(Invisibly) A data.frame of pkgstats results, one row for each package.

See Also

Other archive: pkgstats_cran_current_from_full(), pkgstats_fns_from_archive(), pkgstats_fns_update(), pkgstats_from_archive(), pkgstats_update()

---

extract_tarball *Extract tarball of a package into temp directory and return path to extracted package*

Description

Extract tarball of a package into temp directory and return path to extracted package

Usage

```
extract_tarball(tarball, exdir = fs::path_temp())
```

Arguments

tarball	Full path to local tarball of an R package.
exdir	Directory into which tarballs are to be extracted.

Value

Path to extracted version of package (in tempdir()).

See Also

Other misc: pkgstats_fn_names()

Examples

```
f <- system.file ("extdata", "pkgstats_9.9.tar.gz", package = "pkgstats")
path <- extract_tarball (f)
```

loc_stats *Internal calculation of Lines-of-Code Statistics*

Description

Internal calculation of Lines-of-Code Statistics

Usage

```
loc_stats(path)
```

Arguments

path	Directory to source code of package being analysed

Value

A list of statistics for each of three directories, 'R', 'src', and 'inst/include', each one having 5 statistics of total numbers of lines, numbers of empty lines, total numbers of white spaces, total numbers of characters, and indentation used in files in that directory.

Note

NA values are returned for directories which do not exist.

See Also

Other stats: desc_stats(), pkgstats(), pkgstats_summary(), rd_stats()

Examples

```
f <- system.file ("extdata", "pkgstats_9.9.tar.gz", package = "pkgstats")
# have to extract tarball to call function on source code:
path <- extract_tarball (f)
loc_stats (path)
```

pkgstats *Analyse statistics of one R package*

Description

Analyse statistics of one R package

Usage

```
pkgstats(path = ".")
```

Arguments

path	Either a path to a local source repository, or a local '.tar.gz' file, containing code for an R package.

Value

List of statistics and data on function call networks (or object relationships in other languages). Includes the following components:

1. loc: Summary of Lines-of-Code in all package directories
2. vignettes: Numbers of vignettes and "demo" files
3. data_stats: Statistics of numbers and sizes of package data files
4. desc: Summary of contents of 'DESCRIPTION' file
5. translations: List of translations into other (human) languages (where provides)
6. objects: A `data.frame` of all functions in R, and all other objects (functions, classes, structures, global variables, and more) in all other languages
7. network: A `data.frame` of object references within and between all languages; in R these are function calls, but may be more abstract in other languages.
8. external_calls: A `data.frame` of all calls make to all functions from all other R packages, including base and recommended as well as contributed packages.

See Also

Other stats: `desc_stats()`, `loc_stats()`, `pkgstats_summary()`, `rd_stats()`

Examples

```
# 'path' can be path to a package tarball:
f <- system.file ("extdata", "pkgstats_9.9.tar.gz", package = "pkgstats")

s <- pkgstats (f)
# or to a source directory:
path <- extract_tarball (f)
s <- pkgstats (path)
```

---

pkgstats_cran_current_from_full

Reduce `data.frame` *of full CRAN archive data to current packages only.*

Description

Reduce `data.frame` of full CRAN archive data to current packages only.

Usage

```
pkgstats_cran_current_from_full(prev_results, results_file = NULL)
```

Arguments

prev_results	Result of previous call to this function, if available. Submitting previous results will ensure that only newer packages not present in previous result will be analysed, with new results simply appended to previous results. This parameter can also specify a file to be read with `readRDS()`.
results_file	Can be used to specify the name or full path of a `.Rds` file to which results should be saved once they have been generated. The '.Rds' extension will be automatically appended, and any other extensions will be ignored.

Value

A `data.frame` object with one row for each package containing summary statistics generated from the pkgstats_summary function.

See Also

Other archive: `dl_pkgstats_data()`, `pkgstats_fns_from_archive()`, `pkgstats_fns_update()`, `pkgstats_from_archive()`, `pkgstats_update()`

---

pkgstats_fns_from_archive

Trawl a local CRAN archive to extract function names only from all packages

---

Description

Trawl a local CRAN archive to extract function names only from all packages

Usage

```
pkgstats_fns_from_archive(
  path,
  archive = FALSE,
  prev_results = NULL,
  results_file = NULL,
  chunk_size = 1000L,
  num_cores = 1L,
  results_path = fs::path_temp()
)
```

Arguments

path	Path to local archive of R packages, either as source directories, or '.tar.gz' files such as in a CRAN mirror.
archive	If TRUE, extract statistics for all packages in the /Archive sub-directory, otherwise only statistics for main directory (that is, current packages only).
prev_results	Result of previous call to this function, if available. Submitting previous results will ensure that only newer packages not present in previous result will be analysed, with new results simply appended to previous results. This parameter can also specify a file to be read with readRDS().
results_file	Can be used to specify the name or full path of a .Rds file to which results should be saved once they have been generated. The '.Rds' extension will be automatically appended, and any other extensions will be ignored.
chunk_size	Divide large archive trawl into chunks of this size, and save intermediate results to local files. These intermediate files can be combined to generate a single prev_results file, to enable jobs to be stopped and re-started without having to recalculate all results. These files will be named pkgstats-results-N.Rds, where "N" incrementally numbers each file.
num_cores	Number of machine cores to use in parallel, defaulting to single-core processing.
results_path	Path to save intermediate files generated by the chunk_size parameter described above.

Value

A data.frame object with one row for each function in each package and the following columns:

- Package name
- Package version
- Function name

See Also

Other archive: dl_pkgstats_data(), pkgstats_cran_current_from_full(), pkgstats_fns_update(), pkgstats_from_archive(), pkgstats_update()

---

pkgstats_fns_update *Update function names data from previous data and newly updated CRAN packages only.*

Description

Update function names data from previous data and newly updated CRAN packages only.

Usage

```
pkgstats_fns_update(
  prev_results = NULL,
  results_file = NULL,
  chunk_size = 1000L,
  num_cores = 1L,
  results_path = tempdir()
)
```

Arguments

prev_results	Result of previous call to this function, if available. Submitting previous results will ensure that only newer packages not present in previous result will be analysed, with new results simply appended to previous results. This parameter can also specify a file to be read with readRDS().
results_file	Can be used to specify the name or full path of a .Rds file to which results should be saved once they have been generated. The '.Rds' extension will be automatically appended, and any other extensions will be ignored.
chunk_size	Divide large archive trawl into chunks of this size, and save intermediate results to local files. These intermediate files can be combined to generate a single prev_results file, to enable jobs to be stopped and re-started without having to recalculate all results. These files will be named pkgstats-results-N.Rds, where "N" incrementally numbers each file.
num_cores	Number of machine cores to use in parallel, defaulting to single-core processing.
results_path	Path to save intermediate files generated by the chunk_size parameter described above.

Value

A data.frame object with one row for each function in each package and the following columns:

- Package name
- Package version
- Function name

See Also

Other archive: dl_pkgstats_data(), pkgstats_cran_current_from_full(), pkgstats_fns_from_archive(), pkgstats_from_archive(), pkgstats_update()

pkgstats_fn_names *Extract names of all functions for one R package*

Description

Extract names of all functions for one R package

Usage

```
pkgstats_fn_names(path)
```

Arguments

path	Either a path to a local source repository, or a local '.tar.gz' file, containing code for an R package.

Value

A `data.frame` with three columns:

- package: Name of package
- version: Package version
- fn_name: Name of function

See Also

Other misc: `extract_tarball()`

Examples

```
# 'path' can be path to a package tarball:
f <- system.file ("extdata", "pkgstats_9.9.tar.gz", package = "pkgstats")
path <- extract_tarball (f)
s <- pkgstats_fn_names (path)
```

pkgstats_from_archive *Trawl a local CRAN archive and extract statistics from all packages*

Description

Trawl a local CRAN archive and extract statistics from all packages

Usage

```
pkgstats_from_archive(
  path,
  archive = TRUE,
  prev_results = NULL,
  results_file = NULL,
  chunk_size = 1000L,
  num_cores = 1L,
  save_full = FALSE,
  save_ex_calls = FALSE,
  results_path = fs::path_temp()
)
```

Arguments

path	Path to local archive of R packages, either as source directories, or '.tar.gz' files such as in a CRAN mirror.
archive	If TRUE, extract statistics for all packages in the /Archive sub-directory, otherwise only statistics for main directory (that is, current packages only).
prev_results	Result of previous call to this function, if available. Submitting previous results will ensure that only newer packages not present in previous result will be analysed, with new results simply appended to previous results. This parameter can also specify a file to be read with readRDS().
results_file	Can be used to specify the name or full path of a .Rds file to which results should be saved once they have been generated. The '.Rds' extension will be automatically appended, and any other extensions will be ignored.
chunk_size	Divide large archive trawl into chunks of this size, and save intermediate results to local files. These intermediate files can be combined to generate a single prev_results file, to enable jobs to be stopped and re-started without having to recalculate all results. These files will be named pkgstats-results-N.Rds, where "N" incrementally numbers each file.
num_cores	Number of machine cores to use in parallel, defaulting to single-core processing.
save_full	If TRUE, full pkgstats results are saved for each package to files in results_path.
save_ex_calls	If TRUE, the results of the external_calls component are saved for each package to files in results_path (only if save_full = FALSE).
results_path	Path to save intermediate files generated by the chunk_size parameter described above.

Value

A data.frame object with one row for each package containing summary statistics generated from the pkgstats_summary function.

See Also

Other archive: dl_pkgstats_data(), pkgstats_cran_current_from_full(), pkgstats_fns_from_archive(), pkgstats_fns_update(), pkgstats_update()

Examples

```
# Create fake archive directory with single tarball:
f <- system.file ("extdata", "pkgstats_9.9.tar.gz", package = "pkgstats")
tarball <- basename (f)

archive_path <- file.path (tempdir (), "archive")
if (!dir.exists (archive_path)) {
    dir.create (archive_path)
}
path <- file.path (archive_path, tarball)
file.copy (f, path)
tarball_path <- file.path (archive_path, "tarballs")
dir.create (tarball_path, recursive = TRUE)
file.copy (path, file.path (tarball_path, tarball))
out <- pkgstats_from_archive (tarball_path)
```

---

pkgstats_summary *Condense the output of* pkgstats *to summary statistics only*

Description

Condense the output of pkgstats to summary statistics only

Usage

```
pkgstats_summary(s = NULL)
```

Arguments

s	Output of pkgstats, containing full statistical data on one package. Default of NULL returns a single row with NA values (used in pkgstats_from_archive).

Value

Summarised version of s, as a single row of a standardised data.frame object

Note

Variable names in the summary object use the following abbreviations:

- "loc" = Lines-of-Code
- "fn" = Function
- "n_fns" = Number of functions
- "npars" = Number of parameters
- "doclines" = Number of documentation lines
- "nedges" = Number of edges in function call network, as a count of *unique* edges, which may be less than the size of the network object returned by pkgstats, because that may include multiple calls between identical function pairs.

- "n_clusters" = Number of connected clusters within the function call network.
- "centrality" used as a prefix for several statistics, along with "dir" or "undir" for centrality calculated on networks respectively constructed with directed or undirected edges; "mn" or "md" for respective measures of mean or median centrality, and "no0" for measures excluding edges with zero centrality.

See Also

Other stats: desc_stats(), loc_stats(), pkgstats(), rd_stats()

Examples

```
f <- system.file ("extdata", "pkgstats_9.9.tar.gz", package = "pkgstats")

p <- pkgstats (f)
s <- pkgstats_summary (p)
```

---

pkgstats_update *Update pkgstats' data on GitHub release*

Description

This function is intended for internal rOpenSci use only. Usage by any unauthorized users will error and have no effect unless run with upload = FALSE, in which case updated data will be created in the sub-directory "pkgstats-results" of R's current temporary directory.

Usage

```
pkgstats_update(upload = TRUE)
```

Arguments

upload	If TRUE, upload updated results to GitHub release.

Value

Local path to directory containing updated results.

See Also

Other archive: dl_pkgstats_data(), pkgstats_cran_current_from_full(), pkgstats_fns_from_archive(), pkgstats_fns_update(), pkgstats_from_archive()

plot_network *Plot interactive visualisation of object-relationship network of package.*

Description

Plot interactive visualisation of object-relationship network of package.

Usage

```
plot_network(s, plot = TRUE, vis_save = NULL)
```

Arguments

s	Package statistics obtained from pkgstats function.
plot	If TRUE, plot the network which opens an interactive browser pane.
vis_save	Name of local file in which to save html file of network visualisation (will override plot to FALSE).

Value

(Invisibly) A pkgstats_network object: a list of the nodes and edges used to construct the network, along with the self-contained html document itself.

Note

Edge thicknesses are scaled to centrality within the package function call network. Node sizes are scaled to numbers of times each function is called from all other functions within a package. The visualisation is rendered with a small bundle of **D3**-based JavaScript shipped with this package (see system.file ("js", package = "pkgstats")), so no internet connection or additional R packages are required to view it.

Examples

```
f <- system.file ("extdata", "pkgstats_9.9.tar.gz", package = "pkgstats")

p <- pkgstats (f)
net <- plot_network (p, plot = FALSE)

# use default 'plot = TRUE' to automatically open network in browser.
```

rd_stats *Stats from '.Rd' files*

Description

Stats from '.Rd' files

Usage

```
rd_stats(path)
```

Arguments

path	Directory to source code of package being analysed

Value

A data.frame of function names and numbers of parameters and lines of documentation for each, along with mean and median numbers of characters used to document each parameter.

See Also

Other stats: desc_stats(), loc_stats(), pkgstats(), pkgstats_summary()

Examples

```
f <- system.file ("extdata", "pkgstats_9.9.tar.gz", package = "pkgstats")
# have to extract tarball to call function on source code:
path <- extract_tarball (f)
rd_stats (path)
```

tags_data *use ctags and gtags to parse call data*

Description

use ctags and gtags to parse call data

Usage

```
tags_data(path, has_tabs = NULL, pkg_name = NULL)
```

Arguments

path	Path to local repository
has_tabs	A logical flag indicating whether or not the code contains any tab characters. This can be determined from loc_stats, which has a tabs column. If not given, that value will be extracted from internally calling that function.
pkg_name	Only used for external_call_network, to label package-internal calls.

Value

A list of three items:

- "network" A data.frame of relationships between objects, generally as calls between functions in R, but other kinds of relationships in other source languages. This is effectively an edge-based network representation, and the data frame also include network metrics for each edge, calculated through representing the network in both directed (suffix "_dir") and undirected (suffix "_undir") forms.
- "objects" A data.frame of statistics on each object (generally functions in R, and other kinds of objects in other source languages), including the kind of object, the language, numbers of lines-of-code, parameters, and lines of documentation, and a binary flag indicating whether or not R functions accept "three-dots" parameters (...).
- "external_calls" A data.frame of every call from within every R function to any external R package, including base and recommended packages. The location of each calls is recorded, along with the external function and package being called.

See Also

Other tags: ctags_install(), ctags_test()

Examples

```
f <- system.file ("extdata", "pkgstats_9.9.tar.gz", package = "pkgstats")
# have to extract tarball to call function on source code:
path <- extract_tarball (f)

tags <- tags_data (path)
```

Index

∗ **archive**
dl_pkgstats_data, 4
pkgstats_cran_current_from_full, 7
pkgstats_fns_from_archive, 8
pkgstats_fns_update, 9
pkgstats_from_archive, 11
pkgstats_update, 14
∗ **misc**
extract_tarball, 5
pkgstats_fn_names, 11
∗ **output**
plot_network, 15
∗ **stats**
desc_stats, 4
loc_stats, 6
pkgstats, 6
pkgstats_summary, 13
rd_stats, 16
∗ **tags**
ctags_install, 2
ctags_test, 3
tags_data, 16

ctags_install, 2
ctags_install(), *3*, *17*
ctags_test, 3
ctags_test(), *3*, *17*

desc_stats, 4
desc_stats(), *6, 7*, *14*, *16*
dl_pkgstats_data, 4
dl_pkgstats_data(), *8–10*, *12*, *14*

extract_tarball, 5
extract_tarball(), *11*

loc_stats, 6, *17*
loc_stats(), *4*, *7*, *14*, *16*

pkgstats, 6, *12, 13*, *15*
pkgstats(), *4*, *6*, *14*, *16*
pkgstats_cran_current_from_full, 7
pkgstats_cran_current_from_full(), *5*, *9*, *10*, *12*, *14*
pkgstats_fn_names, 11
pkgstats_fn_names(), *5*
pkgstats_fns_from_archive, 8
pkgstats_fns_from_archive(), *5*, *8*, *10*, *12*, *14*
pkgstats_fns_update, 9
pkgstats_fns_update(), *5*, *8, 9*, *12*, *14*
pkgstats_from_archive, 11, *13*
pkgstats_from_archive(), *5*, *8–10*, *14*
pkgstats_summary, *8*, *12*, 13
pkgstats_summary(), *4*, *6, 7*, *16*
pkgstats_update, 14
pkgstats_update(), *5*, *8–10*, *12*
plot_network, 15

rd_stats, 16
rd_stats(), *4*, *6, 7*, *14*

tags_data, 16
tags_data(), *3*