

Package ‘pkgdiff’

March 13, 2025

Type Package

Title Identifies Package Differences

Version 0.1.9

Maintainer David Bosak <dbosak01@gmail.com>

Description Identifies differences between versions of a package. Specifically, the functions help determine if there are breaking changes from one package version to the next. The package also includes a stability assessment, to help you determine the overall stability of a package, or even an entire repository.

License CC0

Encoding UTF-8

URL <https://pkgdiff.r-sassy.org>, <https://github.com/dbosak01/pkgdiff>

BugReports <https://github.com/dbosak01/pkgdiff/issues>

RoxygenNote 7.3.2

Depends R (>= 4.0.0)

Imports rvest, common, crayon, utils, cranlogs

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author David Bosak [aut, cre],
Brian Varney [ctb],
Kevin Putschko [ctb]

Repository CRAN

Date/Publication 2025-03-13 21:10:05 UTC

Contents

pkg_cache	2
pkg_diff	3
pkg_info	6
pkg_repo	9
pkg_stability	10
pkg_versions	15
print.pcache	16
print.pdiff	17
print.pdiff_score	18
print.pinfo	19
print.prepo	21
print.rbreak	22
print.rstability	22
repo_breakages	23
repo_stability	25
Index	28

pkg_cache	<i>Queries the Package Cache</i>
-----------	----------------------------------

Description

The `pkg_cache` function queries the package cache, and returns information on which packages are included in the cache. The function also returns the last version of the package cached.

Usage

```
pkg_cache(pkgs = NULL)
```

Arguments

`pkgs` A vector of package names to retrieve cache information about. Default is `NULL`, which means to return all packages in the cache.

Details

The **pkgdiff** cache is used to speed up **pkgdiff** functions. Information about each package is pre-processed and stored in the cache on Github. The functions then retrieve this pre-processed information instead of pulling packages down from CRAN. To learn more about the cache, refer to `vignette('pkgdiff-cache')`.

Value

An data frame showing the package name and latest package version stored in the cache. If the package is not stored in the cache, the package version will be `NA`. The "Last Update" time stamp is the last day and time the cache was updated.

See Also

Other pdiff: [pkg_diff\(\)](#), [pkg_info\(\)](#), [pkg_stability\(\)](#), [pkg_versions\(\)](#), [print.pcache\(\)](#), [print.pdiff\(\)](#), [print.pdiff_score\(\)](#), [print.pinfo\(\)](#)

Examples

```
# View single package
pkg_cache("dplyr")
# # A package cache object
# - Last Update: 2025-02-25 14:58 UTC
# - Packages:
#   Package Version
# 1  dplyr  1.1.4

# View multiple packages
pkgs <- c("dplyr", "tidyr", "stringr")
pkg_cache(pkgs)
# # A package cache object
# - Last Update: 2025-02-25 14:58 UTC
# - Packages:
#   Package Version
# 1  dplyr  1.1.4
# 2  tidyr  1.3.1
# 3  stringr 1.5.1

# Get all packages
res <- pkg_cache()

# View first 10
res[1:10, ]
# A package cache object
# - Last Update: 2025-02-25 14:58 UTC
# - Packages:
#   Package Version
# 1      A3  1.0.0
# 2  abbreviate  0.1
# 3   abc.data  1.1
# 4      abc  2.2.2
# 5  ABCanalysis 1.2.1
# 6      abd  0.2-8
# 7   abind  1.4-8
# 8   acepack 1.6.1
# 9   actuar  3.3-5
# 10     ada  2.0-5
```

Description

The `pkg_diff` function is used to compare two different versions of a package. The function returns an object that contains the differences. Differences include any functions added or removed. The object may be printed directly, or stored in a variable and examined programmatically.

Usage

```
pkg_diff(pkg, v1 = "current", v2 = "latest")
```

Arguments

<code>pkg</code>	The package name.
<code>v1</code>	The earlier package version. Pass the version value as a quoted string. Default is "current", which means the function will look up the currently installed version.
<code>v2</code>	The later package version. Pass the version value as a quoted string. The default is "latest", which is the latest version of the package available on CRAN.

Details

The difference operation only compares functions that are exported from the package. It does not compare internal functions or function documentation. Exported functions are identified in the package namespace.

The `pkg_diff` function can compare any two versions of a package. They do not need to be consecutive. They must, however, exist in the CRAN archive.

If the package versions are included in the **pkgdiff** cache, the function will pull them from there. Otherwise, they will be pulled from the CRAN mirror. The function does not access any packages on the local system.

Value

An object of class "pdiff", which contains information regarding the differences between two versions of a package. The object includes a list of functions that were added, a list of function parameters that were added, a list of functions removed, and a list of function parameters that were removed. The object also contains some general information like the package versions examined, and a TRUE or FALSE flag indicating whether there were any breaking changes.

See Also

Other pdiff: [pkg_cache\(\)](#), [pkg_info\(\)](#), [pkg_stability\(\)](#), [pkg_versions\(\)](#), [print.pcache\(\)](#), [print.pdiff\(\)](#), [print.pdiff_score\(\)](#), [print.pinfo\(\)](#)

Examples

```
# View package stability
pkg_stability("rsample")
# # A stability score: rsample package
# - Age: 7.58 years
# - Score: 80.3
```

```

# - Assessment: Somewhat Unstable
# - Version Range: 0.0.1/1.2.1
# - Release Range: 2017-07-08/2024-03-25
# - Release Count: 16
# - Breaking Releases: 6
# - Data:
#   Package Version      FileName      Release   Size  AF AP RF RP BC TF
# 1  rsample    1.2.1 rsample_1.2.1.tar.gz 2024-03-25 320.9K  1 0 0 0 0 401
# 2  rsample    1.2.0 rsample_1.2.0.tar.gz 2023-08-23  321K 43 6 1 3 1 400
# 3  rsample    1.1.1 rsample_1.1.1.tar.gz 2022-12-07  318K 37 5 0 0 0 358
# 4  rsample    1.1.0 rsample_1.1.0.tar.gz 2022-08-08  306K 72 1 0 0 0 321
# 5  rsample    1.0.0 rsample_1.0.0.tar.gz 2022-06-24  268K  1 0 1 0 1 249
# 6  rsample    0.1.1 rsample_0.1.1.tar.gz 2021-11-08  274K  4 1 0 1 1 249
# 7  rsample    0.1.0 rsample_0.1.0.tar.gz 2021-05-08  274K  0 5 0 0 0 245
# 8  rsample    0.0.9 rsample_0.0.9.tar.gz 2021-02-17  269K 19 0 6 0 1 245
# 9  rsample    0.0.8 rsample_0.0.8.tar.gz 2020-09-23  261K 59 0 0 0 0 232
#10  rsample    0.0.7 rsample_0.0.7.tar.gz 2020-06-04  248K 101 1 0 0 0 173
#11  rsample    0.0.6 rsample_0.0.6.tar.gz 2020-03-31  299K  7 2 0 0 0 72
#12  rsample    0.0.5 rsample_0.0.5.tar.gz 2019-07-13  297K  4 5 0 0 0 65
#13  rsample    0.0.4 rsample_0.0.4.tar.gz 2019-01-07  254K  0 0 4 0 1 61
#14  rsample    0.0.3 rsample_0.0.3.tar.gz 2018-11-20  254K  7 0 1 0 1 65
#15  rsample    0.0.2 rsample_0.0.2.tar.gz 2017-11-12  339K 24 0 0 0 0 59
#16  rsample    0.0.1 rsample_0.0.1.tar.gz 2017-07-08  180K 35 35 0 0 0 35

# Examine differences between versions
pkg_diff("rsample", "1.1.1", "1.2.0")
## A difference object: rsample package
# - Comparing: v1.1.1/v1.2.0
# - Breaking Changes: TRUE
# - Added Functions:
#   - analysis.default()
#   - analysis.initial_validation_split()
#   - analysis.rsplit()
#   - analysis.val_split()
#   - assessment.default()
#   - assessment.initial_validation_split()
#   - assessment.rsplit()
#   - assessment.val_split()
#   - dim.initial_validation_split()
#   - group_initial_validation_split()
#   - initial_validation_split()
#   - initial_validation_time_split()
#   - int_bca.bootstraps()
#   - int_pctl.bootstraps()
#   - int_t.bootstraps()
#   - print.initial_validation_split()
#   - testing.default()
#   - testing.initial_validation_split()
#   - testing.rsplit()
#   - testing.val_split()
#   - training.default()
#   - training.initial_validation_split()
#   - training.rsplit()

```

```

# - training.val_split()
# - validation()
# - validation.default()
# - validation.initial_validation_split()
# - validation.val_split()
# - validation_set()
# - vec_cast.data.frame.validation_set()
# - vec_cast.tbl_df.validation_set()
# - vec_cast.validation_set.data.frame()
# - vec_cast.validation_set.tbl_df()
# - vec_cast.validation_set.validation_set()
# - vec_ptype_abbr.group_initial_validation_split()
# - vec_ptype_abbr.initial_validation_split()
# - vec_ptype_abbr.validation_set()
# - vec_ptype2.data.frame.validation_set()
# - vec_ptype2.tbl_df.validation_set()
# - vec_ptype2.validation_set.data.frame()
# - vec_ptype2.validation_set.tbl_df()
# - vec_ptype2.validation_set.validation_set()
# - vec_restore.validation_set()
# - Added Parameters:
# - int_pct1(): ...
# - int_t(): ...
# - testing(): ...
# - tidy.nested_cv(): unique_ind
# - tidy.rset(): unique_ind
# - training(): ...
# - Removed Functions:
# - gather()
# - Removed Parameters:
# - int_bca(): statistics, alpha, .fn
# - int_pct1(): statistics, alpha
# - int_t(): statistics, alpha

```

pkg_info

Get Information for a Package

Description

Combines information related to a package. Objects of this class are used by several other functions in the **pkgdiff** system. This class also makes a compact storage format for pre-processed package information.

Usage

```
pkg_info(pkg, ver = "current", cache = TRUE)
```

Arguments

pkg	The package name as a quoted string. This parameter is required.
ver	The version of the package to retrieve information for. Pass the version as a quoted string. Special values are "current" and "latest". The value "current" is the current version of the package running on the machine. The value "latest" is the latest version of the package from CRAN.
cache	Whether to retrieve the info from the Github cache, or from CRAN. If TRUE, the function will first search the cache, and return the info if available. If the info is not available in the Github cache, or the cache parameter is set to FALSE, the info will be retrieved from CRAN.

Details

Package information is unique for each version of a package. Since a package can have multiple versions, you must select which version you wish to return information for. By default, the version returned is the current version of the package on the current machine. You may also specify a previous version number from the CRAN archive. Another option is to specify the latest version on CRAN. See the "ver" parameter for additional details on how to select these different package versions.

The package info object contains a list of functions and function parameters associated with the specified version of the package. You may access this list using the `$functions` item name.

Most other information contained in the info object is retrieved from the package description file. One exception is the downloads per month. This information is retrieved from CRAN logs.

In addition, the package cache status is appended to the info object. The package cache status indicates whether the package info has been stored in the **pkgdiff** Github cache. Packages that have been stored in the cache enjoy faster results from **pkgdiff** functions.

Note that `pkg_info` and other **pkgdiff** functions only work with contributed CRAN packages. They do not work with Base R packages.

Value

A package information object of class "pinfo". This object contains a set of general information about the package, such as the version, release date, maintainer, title, etc. Most of this information comes from the package description file. In addition, the info object also contains a list of functions in the package and their parameters.

If the package has been archived on CRAN, info will be returned, but the "Archived" flag will be set to TRUE. If the package is not found on CRAN, the function will emit a message and return NULL.

See Also

Other pdiff: [pkg_cache\(\)](#), [pkg_diff\(\)](#), [pkg_stability\(\)](#), [pkg_versions\(\)](#), [print.pcache\(\)](#), [print.pdiff\(\)](#), [print.pdiff_score\(\)](#), [print.pinfo\(\)](#)

Examples

```

# View package info
pkg_info("glue")
# A package info object: glue package
# - Version: v1.7.0
# - Release Date: 2024-01-09
# - Title: Interpreted String Literals
# - Maintainer: Jennifer Bryan <jenny@posit.co>
# - License: MIT + file LICENSE
# - Description: An implementation of interpreted string literals, inspired by
# Python's Literal String Interpolation
# <https://www.python.org/dev/peps/pep-0498/> and Docstrings
# <https://www.python.org/dev/peps/pep-0257/> and Julia's Triple-Quoted
# String Literals
# <https://docs.julialang.org/en/v1.3/manual/strings/#Triple-Quoted-String-Literals-1>.
# - Depends: R (>= 3.6)
# - Imports: methods
# - Suggests: crayon, DBI (>= 1.2.0), dplyr, knitr, magrittr, rlang,
# rmarkdown, RSQLite, testthat (>= 3.2.0), vctrs (>= 0.3.0),
# waldo (>= 0.3.0), withr
# - Downloads/Month: 1463244
# - Repository: CRAN
# - Cached: TRUE
# - Functions: 24

# Get info object
res <- pkg_info("glue")

# Extract package version
res$Version
# [1] "1.7.0"

# Extract maintainer
res$Maintainer
# [1] "Jennifer Bryan <jenny@posit.co>"

# Extract function list
res$Functions
# $`.glue`
# [1] "x" "i" "...
#
# $`[[.glue`
# [1] "x" "i" "...
#
# $`+.glue`
# [1] "e1" "e2"
#
# $as.character.glue
# [1] "x" "...
#
# $as_glue
# [1] "x" "...

```



```

#
# $as_glue.character
# [1] "x"  "...
#
# $as_glue.default
# [1] "x"  "...
#
# $as_glue.glue
# [1] "x"  "...
#
# $backtick
# [1] "x"
#
# $double_quote
# [1] "x"
#
# $glue
# [1] "...      ".sep"      ".envir"
# [4] ".open"    ".close"    ".na"
# [7] ".null"    ".comment"  ".literal"
# [10] ".transformer" ".trim"
# ...

```

pkg_repo

Retrieve Package Versions from a Repository

Description

The `pkg_repo` function combines information related to a package repository. The function retrieves all the packages in a repository for a specified version of R. Results will list the package name and version.

Usage

```
pkg_repo(pkgs = NULL, ver = "current", libpaths = NULL)
```

Arguments

<code>pkgs</code>	A vector of package names used to subset the repository list. Default is <code>NULL</code> , which means all packages in the repository will be returned.
<code>ver</code>	The R version of the repository. Pass the R version as a quoted string. Default is "current", meaning the current version of the repository. The value "latest" will return the latest versions on CRAN.
<code>libpaths</code>	A vector of paths specifying the locations of the repositories to query. Default is <code>NULL</code> , meaning the function will use the default R locations on the machine. These default paths are identified by the Base R function <code>.libPaths()</code> .

Value

An object of class "repo". The object will contain a data frame of R packages with corresponding version numbers. This table may then be passed to [repo_breakages](#) or [repo_stability](#).

If a package is not found in the repo, a row will still be returned for the package. However, the version will appear as NA.

See Also

Other `repo`: [print.repo\(\)](#), [print.rbreak\(\)](#), [print.rstability\(\)](#), [repo_breakages\(\)](#), [repo_stability\(\)](#)

Examples

```
# Create vector of packages
pkgs <- c("common", "dplyr", "rvest", "stringr")

# Retrieve latest versions
pkg_repo(pkgs, ver = "latest")
# # A package repo object
# - Repo Version: latest
# - Packages:
#   Package Version
# 1 common 1.1.3
# 2 dplyr 1.1.4
# 3 rvest 1.0.4
# 4 stringr 1.5.1
```

pkg_stability

Calculate Stability Score for a Package

Description

The `pkg_stability` function calculates a stability score for a specified package. The score is calculated as the percentage of releases with no breaking changes. A breaking change is defined as either a removed function or removed function parameter. The object also returns the raw data from which the stability score is calculated. This data can be useful for review or even custom analytics.

Usage

```
pkg_stability(pkg, releases = NULL, months = NULL)
```

Arguments

<code>pkg</code>	The name of the package.
<code>releases</code>	An integer indicating the number of releases to collect stability data for. For example, <code>releases = 10</code> will return stability data for the last 10 releases of the package. Default is <code>NULL</code> , which means the function will return data for all releases.

months An integer indicating the number of months back to collect stability data for. For example, `months = 24` will collect stability data for the previous 2 years. Default is `NULL`, meaning there is no limitation on the number of release months, and the function will collect data from all releases.

Details

The `pkg_stability` function is used to get an overall feel for the stability of a package. The stability score and other information returned by the function are based on data retrieved from CRAN. The function compares each version of a package and determines if any functions or function parameters have been removed from the previous release. If a release removes functions or parameters contained in the previous release, it is flagged as a "breaking release" or "breaking change".

The stability score is calculated as the percentage of non-breaking releases. For example, if a package has 10 releases, and one breaking release, the stability score will be .9, or 90%.

The stability assessment is a categorization of the stability score. The aim of the assessment is to provide a general evaluation of the package, whether it is stable or unstable. The assessment has 5 categories:

- **Perfect:** Stability score of 100. Package has never had a breaking release in its entire history.
- **Very Stable:** Stability score between 95 and 100. Package has had a breaking release, but very rarely: less than 1 in 20 releases.
- **Stable:** Stability score between 90 and 95. The package has had some breaking releases, but it is still rather rare: less than 1 in 10 releases.
- **Somewhat unstable:** Stability score between 80 and 90. The package sometimes has a breaking release, at a rate of about 1 in 5.
- **Unstable:** Stability score below 80. The package breaks more frequently than every 5th release. This frequency of breaking releases is considered unstable.

As can be seen above, the stability assessment is weighted heavily toward the high end. This categorization is intentionally designed to encourage package stability, and discourage the number of breaking changes.

Value

An stability score object of class "pdiff_score". The object contains the stability score, plus other useful information such as the release and version ranges, the number of releases, and number of breaking releases. All of these items can be accessed using dollar sign (\$) syntax.

Stability Data

Stability calculations are based on a data frame of stability information gathered from each package release. The data frame has one row per release. Each row contains comparison information against the prior release. The columns are organized as follows:

- **Package:** The package name.
- **Version:** The version number of the release.
- **FileName:** The file name of the package.
- **Release:** The release date of the package.

- **Size**: The size of the package file on disk.
- **AF**: The number of functions added from the previous release.
- **AP**: The number of functions that had parameters added from the previous release.
- **RF**: The number of functions removed from the previous release.
- **RP**: The number of functions that had parameters removed from the previous release.
- **BC**: Whether the release had any breaking changes. Breaking changes are removed functions or removed parameters. Values are zero or one. The value one (1) means the release had breaking changes. The value zero (0) means the release had no breaking changes.
- **TF**: The total number of functions in the package.

The above data will be included in the function print out. It can also be accessed on the returned object from the "data" list item, i.e. `obj$data`.

Disclaimers

Note that a "breaking release" does not currently factor in the number of functions deprecated. One deprecated function counts the same as 10 deprecated functions within a release. Current methodology also does not differentiate between removed functions and removed parameters. They are both considered "breaking", and counted equally.

The methodology also does not factor in changed functions. The reason is simply that it is much more difficult to identify whether or not a changed function is breaking. That is to say, you may still have breaking changes that are not identified by the **pkgdiff** package.

Performance

Performance of the `pkg_stability` function can vary greatly depending on the package selected. The information for some packages has been cached on Github, and can be retrieved very quickly. For packages that have not been cached, the function must download each version of the package from the CRAN archive, and compare consecutive versions. This process can take considerable time. The most popular packages on CRAN have been cached. If there is a package you query frequently that has not been cached, please submit an issue to the **pkgdiff** issue log on Github. In the issue description simply request that the package be added to the cache.

For more information on the package cache, see `vignette("pkgdiff-cache")`.

For more information about how the stability score and assessment are calculated, see `vignette("pkgdiff-stability")`.

See Also

Other pdiff: `pkg_cache()`, `pkg_diff()`, `pkg_info()`, `pkg_versions()`, `print.pcache()`, `print.pdiff()`, `print.pdiff_score()`, `print.pinfo()`

Examples

```
# View package stability
pkg_stability("curl")
# # A stability score: curl package
# - Age: 10.33 years
# - Score: 98.0
```

```

# - Assessment: Very Stable
# - Version Range: 0.2/6.2.1
# - Release Range: 2014-11-20/2025-02-19
# - Release Count: 51
# - Breaking Releases: 1
# - Data:
# Package Version      FileName      Release      Size AF AP RF RP BC TF
# 1      curl      6.2.1 curl_6.2.1.tar.gz 2025-02-19 911.7K 0 0 0 0 0 0 46
# 2      curl      6.2.0 curl_6.2.0.tar.gz 2025-01-23 911.3K 0 1 0 0 0 0 46
# 3      curl      6.1.0 curl_6.1.0.tar.gz 2025-01-06 911K 0 1 0 0 0 0 46
# 4      curl      6.0.1 curl_6.0.1.tar.gz 2024-11-14 911K 0 0 0 0 0 0 46
# 5      curl      6.0.0 curl_6.0.0.tar.gz 2024-11-05 911K 1 1 0 1 1 1 46
# 6      curl      5.2.3 curl_5.2.3.tar.gz 2024-09-20 700K 0 0 0 0 0 0 45
# 7      curl      5.2.2 curl_5.2.2.tar.gz 2024-08-26 701K 0 0 0 0 0 0 45
# 8      curl      5.2.1 curl_5.2.1.tar.gz 2024-03-02 700K 0 0 0 0 0 0 45
# 9      curl      5.2.0 curl_5.2.0.tar.gz 2023-12-08 699K 1 0 0 0 0 0 45
# 10     curl      5.1.0 curl_5.1.0.tar.gz 2023-10-02 697K 0 1 0 0 0 0 44
# 11     curl      5.0.2 curl_5.0.2.tar.gz 2023-08-14 697K 0 0 0 0 0 0 44
# 12     curl      5.0.1 curl_5.0.1.tar.gz 2023-06-08 697K 0 1 0 0 0 0 44
# 13     curl      5.0.0 curl_5.0.0.tar.gz 2023-01-12 666K 1 1 0 0 0 0 44
# 14     curl      4.3.3 curl_4.3.3.tar.gz 2022-10-06 655K 0 0 0 0 0 0 43
# 15     curl      4.3.2 curl_4.3.2.tar.gz 2021-06-23 775K 0 0 0 0 0 0 43
# 16     curl      4.3.1 curl_4.3.1.tar.gz 2021-04-30 775K 0 0 0 0 0 0 43
# 17     curl      4.3      curl_4.3.tar.gz 2019-12-02 658K 0 0 0 0 0 0 43
# 18     curl      4.2      curl_4.2.tar.gz 2019-09-24 652K 0 0 0 0 0 0 43
# 19     curl      4.1      curl_4.1.tar.gz 2019-09-16 652K 0 1 0 0 0 0 43
# 20     curl      4.0      curl_4.0.tar.gz 2019-07-22 370K 3 0 0 0 0 0 43
# 21     curl      3.3      curl_3.3.tar.gz 2019-01-10 363K 2 0 0 0 0 0 40
# 22     curl      3.2      curl_3.2.tar.gz 2018-03-28 358K 0 0 0 0 0 0 38
# 23     curl      3.1      curl_3.1.tar.gz 2017-12-12 358K 0 0 0 0 0 0 38
# 24     curl      3.0      curl_3.0.tar.gz 2017-10-06 403K 1 2 0 0 0 0 38
# 25     curl      2.8.1 curl_2.8.1.tar.gz 2017-07-22 400K 0 1 0 0 0 0 37
# 26     curl      2.7      curl_2.7.tar.gz 2017-06-26 397K 1 1 0 0 0 0 37
# 27     curl      2.6      curl_2.6.tar.gz 2017-04-27 396K 0 1 0 0 0 0 36
# 28     curl      2.5      curl_2.5.tar.gz 2017-04-14 396K 1 0 0 0 0 0 36
# 29     curl      2.4      curl_2.4.tar.gz 2017-03-24 394K 2 0 0 0 0 0 35
# 30     curl      2.3      curl_2.3.tar.gz 2016-11-24 391K 0 0 0 0 0 0 33
# 31     curl      2.2      curl_2.2.tar.gz 2016-10-21 389K 0 0 0 0 0 0 33
# 32     curl      2.1      curl_2.1.tar.gz 2016-09-22 388K 0 0 0 0 0 0 33
# 33     curl      2.0      curl_2.0.tar.gz 2016-09-17 389K 10 0 0 0 0 0 33
# 34     curl      1.2      curl_1.2.tar.gz 2016-08-13 287K 0 0 0 0 0 0 23
# 35     curl      1.1      curl_1.1.tar.gz 2016-07-26 277K 0 0 0 0 0 0 23
# 36     curl      1.0      curl_1.0.tar.gz 2016-07-24 277K 0 0 0 0 0 0 23
# 37     curl      0.9.7 curl_0.9.7.tar.gz 2016-04-10 258K 0 0 0 0 0 0 23
# 38     curl      0.9.6 curl_0.9.6.tar.gz 2016-02-17 256K 1 0 0 0 0 0 23
# 39     curl      0.9.5 curl_0.9.5.tar.gz 2016-01-23 255K 1 0 0 0 0 0 22
# 40     curl      0.9.4 curl_0.9.4.tar.gz 2015-11-20 240K 0 1 0 0 0 0 21
# 41     curl      0.9.3 curl_0.9.3.tar.gz 2015-08-25 240K 0 0 0 0 0 0 21
# 42     curl      0.9.2 curl_0.9.2.tar.gz 2015-08-08 239K 1 0 0 0 0 0 21
# 43     curl      0.9.1 curl_0.9.1.tar.gz 2015-07-04 237K 0 0 0 0 0 0 20
# 44     curl      0.9      curl_0.9.tar.gz 2015-06-19 236K 0 0 0 0 0 0 20
# 45     curl      0.8      curl_0.8.tar.gz 2015-06-06 235K 0 0 0 0 0 0 20
# 46     curl      0.7      curl_0.7.tar.gz 2015-05-22 236K 0 0 0 0 0 0 20

```

```

# 47  curl    0.6  curl_0.6.tar.gz 2015-05-19 235K 14 2 0 0 0 20
# 48  curl    0.5  curl_0.5.tar.gz 2015-02-01  16K  0 0 0 0 0 6
# 49  curl    0.4  curl_0.4.tar.gz 2015-01-08  16K  4 0 0 0 0 6
# 50  curl    0.3  curl_0.3.tar.gz 2014-12-01  7.5K 1 0 0 0 0 2
# 51  curl    0.2  curl_0.2.tar.gz 2014-11-20  5.5K 1 1 0 0 0 1

```

```

# View package stability
pkg_stability("ggplot2")
# # A stability score: ggplot2 package
# - Age: 17.75 years
# - Score: 82.5
# - Assessment: Somewhat Unstable
# - Version Range: 0.5/3.5.1
# - Release Range: 2007-06-01/2024-04-23
# - Release Count: 50
# - Breaking Releases: 29
# - Data:

```

#	Package	Version	FileName	Release	Size	AF	AP	RF	RP	BC	TF
# 1	ggplot2	3.5.1	ggplot2_3.5.1.tar.gz	2024-04-23	3.4M	0	4	0	0	0	711
# 2	ggplot2	3.5.0	ggplot2_3.5.0.tar.gz	2024-02-23	3.4M	37	92	26	6	1	711
# 3	ggplot2	3.4.4	ggplot2_3.4.4.tar.gz	2023-10-12	3.0M	0	0	0	0	0	700
# 4	ggplot2	3.4.3	ggplot2_3.4.3.tar.gz	2023-08-14	2.1M	0	0	0	0	0	700
# 5	ggplot2	3.4.2	ggplot2_3.4.2.tar.gz	2023-04-03	3.0M	3	0	0	1	1	700
# 6	ggplot2	3.4.1	ggplot2_3.4.1.tar.gz	2023-02-10	3.0M	2	1	0	0	0	697
# 7	ggplot2	3.4.0	ggplot2_3.4.0.tar.gz	2022-11-04	3.0M	28	15	0	0	0	695
# 8	ggplot2	3.3.6	ggplot2_3.3.6.tar.gz	2022-05-03	2.9M	0	0	0	1	1	667
# 9	ggplot2	3.3.5	ggplot2_3.3.5.tar.gz	2021-06-25	2.9M	0	0	0	0	0	667
# 10	ggplot2	3.3.4	ggplot2_3.3.4.tar.gz	2021-06-16	2.9M	8	16	0	1	1	667
# 11	ggplot2	3.3.3	ggplot2_3.3.3.tar.gz	2020-12-30	2.9M	0	0	0	0	0	659
# 12	ggplot2	3.3.2	ggplot2_3.3.2.tar.gz	2020-06-19	2.9M	17	5	0	1	1	659
# 13	ggplot2	3.3.1	ggplot2_3.3.1.tar.gz	2020-05-29	2.9M	0	0	0	0	0	642
# 14	ggplot2	3.3.0	ggplot2_3.3.0.tar.gz	2020-03-05	2.9M	68	47	0	1	1	642
# 15	ggplot2	3.2.1	ggplot2_3.2.1.tar.gz	2019-08-11	3.1M	0	0	0	0	0	574
# 16	ggplot2	3.2.0	ggplot2_3.2.0.tar.gz	2019-06-16	3.0M	3	7	3	0	1	574
# 17	ggplot2	3.1.1	ggplot2_3.1.1.tar.gz	2019-04-07	2.7M	0	2	0	0	0	574
# 18	ggplot2	3.1.0	ggplot2_3.1.0.tar.gz	2018-10-25	2.7M	5	8	0	0	0	574
# 19	ggplot2	3.0.0	ggplot2_3.0.0.tar.gz	2018-07-03	2.7M	89	53	6	4	1	569
# 20	ggplot2	2.2.1	ggplot2_2.2.1.tar.gz	2016-12-30	2.1M	0	0	0	0	0	486
# 21	ggplot2	2.2.0	ggplot2_2.2.0.tar.gz	2016-11-11	2.1M	22	27	21	2	1	486
# 22	ggplot2	2.1.0	ggplot2_2.1.0.tar.gz	2016-03-01	1.5M	11	6	0	2	1	485
# 23	ggplot2	2.0.0	ggplot2_2.0.0.tar.gz	2015-12-18	1.5M	159	94	103	33	1	474
# 24	ggplot2	1.0.1	ggplot2_1.0.1.tar.gz	2015-03-17	2.2M	0	0	0	0	0	418
# 25	ggplot2	1.0.0	ggplot2_1.0.0.tar.gz	2014-05-21	2.2M	12	5	0	0	0	418
# 26	ggplot2	0.9.3.1	ggplot2_0.9.3.1.tar.gz	2013-03-02	2.2M	0	1	0	0	0	406
# 27	ggplot2	0.9.3	ggplot2_0.9.3.tar.gz	2012-12-05	2.2M	4	4	0	0	0	406
# 28	ggplot2	0.9.2.1	ggplot2_0.9.2.1.tar.gz	2012-09-11	2.2M	0	0	0	0	0	402
# 29	ggplot2	0.9.2	ggplot2_0.9.2.tar.gz	2012-09-04	2.2M	26	16	8	6	1	402
# 30	ggplot2	0.9.1	ggplot2_0.9.1.tar.gz	2012-05-08	9.1M	13	11	0	3	1	384
# 31	ggplot2	0.9.0	ggplot2_0.9.0.tar.gz	2012-03-01	1.6M	282	4	149	5	1	371
# 32	ggplot2	0.8.9	ggplot2_0.8.9.tar.gz	2010-12-23	2.0M	4	1	2	0	1	238
# 33	ggplot2	0.8.8	ggplot2_0.8.8.tar.gz	2010-07-05	2.0M	6	0	2	0	1	236
# 34	ggplot2	0.8.7	ggplot2_0.8.7.tar.gz	2010-03-02	2.0M	2	0	0	0	0	232
# 35	ggplot2	0.8.6	ggplot2_0.8.6.tar.gz	2010-02-18	2.0M	0	0	0	0	0	230

```

# 36 ggplot2 0.8.5 ggplot2_0.8.5.tar.gz 2009-12-16 2.0M 9 2 4 0 1 230
# 37 ggplot2 0.8.4 ggplot2_0.8.4.tar.gz 2009-12-09 2.0M 3 0 2 0 1 225
# 38 ggplot2 0.8.3 ggplot2_0.8.3.tar.gz 2009-04-20 2.0M 51 2 6 2 1 224
# 39 ggplot2 0.8.2 ggplot2_0.8.2.tar.gz 2009-02-25 2.0M 11 1 10 0 1 179
# 40 ggplot2 0.8.1 ggplot2_0.8.1.tar.gz 2008-12-14 1.9M 19 3 3 3 1 178
# 41 ggplot2 0.8 ggplot2_0.8.tar.gz 2008-11-21 1.9M 25 6 2 2 1 162
# 42 ggplot2 0.7 ggplot2_0.7.tar.gz 2008-10-05 1.9M 45 7 26 3 1 139
# 43 ggplot2 0.6 ggplot2_0.6.tar.gz 2008-04-03 1.9M 30 5 125 2 1 120
# 44 ggplot2 0.5.7 ggplot2_0.5.7.tar.gz 2008-01-11 1.8M 16 3 3 0 1 215
# 45 ggplot2 0.5.6 ggplot2_0.5.6.tar.gz 2007-10-20 1.8M 36 0 5 0 1 202
# 46 ggplot2 0.5.5 ggplot2_0.5.5.tar.gz 2007-09-01 1.8M 1 0 0 2 1 171
# 47 ggplot2 0.5.4 ggplot2_0.5.4.tar.gz 2007-07-08 1.8M 3 0 1 0 1 170
# 48 ggplot2 0.5.2 ggplot2_0.5.2.tar.gz 2007-06-18 1.8M 2 0 0 0 0 168
# 49 ggplot2 0.5.1 ggplot2_0.5.1.tar.gz 2007-06-10 1.8M 11 1 5 0 1 166
# 50 ggplot2 0.5 ggplot2_0.5.tar.gz 2007-06-01 7.1M 160 160 0 0 0 160

```

pkg_versions

Retrieves All Versions of a Package

Description

The `pkg_versions` function queries the CRAN archive, and returns a data frame of package version information for all releases of the package. This function may be used to understand what versions of a package are available. It is often used in conjunction with `pkg_info` to get a general overview of a package.

Usage

```
pkg_versions(pkg)
```

Arguments

`pkg` A package name to return versions for. Pass the package name as a quoted string. This parameter is required.

Value

A data frame with one row per package release, showing the version and date of each release. If the package is not found on CRAN, a message will be generated and the function will return `NULL`.

See Also

Other `pdiff`: `pkg_cache()`, `pkg_diff()`, `pkg_info()`, `pkg_stability()`, `print.pcache()`, `print.pdiff()`, `print.pdiff_score()`, `print.pinfo()`

Examples

```
# View package versions
pkg_versions("rsample")
#   Package Version      FileName   Release   Size
# 1  rsample  1.2.1 rsample_1.2.1.tar.gz 2024-03-25 320.9K
# 2  rsample  1.2.0 rsample_1.2.0.tar.gz 2023-08-23  321K
# 3  rsample  1.1.1 rsample_1.1.1.tar.gz 2022-12-07  318K
# 4  rsample  1.1.0 rsample_1.1.0.tar.gz 2022-08-08  306K
# 5  rsample  1.0.0 rsample_1.0.0.tar.gz 2022-06-24  268K
# 6  rsample  0.1.1 rsample_0.1.1.tar.gz 2021-11-08  274K
# 7  rsample  0.1.0 rsample_0.1.0.tar.gz 2021-05-08  274K
# 8  rsample  0.0.9 rsample_0.0.9.tar.gz 2021-02-17  269K
# 9  rsample  0.0.8 rsample_0.0.8.tar.gz 2020-09-23  261K
#10  rsample  0.0.7 rsample_0.0.7.tar.gz 2020-06-04  248K
#11  rsample  0.0.6 rsample_0.0.6.tar.gz 2020-03-31  299K
#12  rsample  0.0.5 rsample_0.0.5.tar.gz 2019-07-13  297K
#13  rsample  0.0.4 rsample_0.0.4.tar.gz 2019-01-07  254K
#14  rsample  0.0.3 rsample_0.0.3.tar.gz 2018-11-20  254K
#15  rsample  0.0.2 rsample_0.0.2.tar.gz 2017-11-12  339K
#16  rsample  0.0.1 rsample_0.0.1.tar.gz 2017-07-08  180K
```

```
print.pcache
```

```
Print a Package Cache Object
```

Description

Print routine for a package cache object of class "pcache".

Usage

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

Arguments

```
x          The package cache to print.
...       Follow-on parameters to the print function.
```

Value

The package cache object, invisibly.

See Also

Other pdiff: [pkg_cache\(\)](#), [pkg_diff\(\)](#), [pkg_info\(\)](#), [pkg_stability\(\)](#), [pkg_versions\(\)](#), [print.pdiff\(\)](#), [print.pdiff_score\(\)](#), [print.pinfo\(\)](#)

Examples

```
# Create vector of packages
pkgs <- c("ggplot2", "patchwork", "gt")

# Capture cache versions
res <- pkg_cache(pkgs)

# Print cache versions
print(res)
# # A package cache object
# - Last Update: 2025-03-01 20:51 UTC
# - Packages:
#   Package Version
# 1 ggplot2 3.5.1
# 2 patchwork 1.3.0
# 3 gt 0.11.1
```

print.pdiff

Print a Package Difference Object

Description

Print routine for a package difference object of class "pdiff".

Usage

```
## S3 method for class 'pdiff'
print(x, ..., verbose = TRUE)
```

Arguments

x	The package difference to print
...	Follow-on parameters to the print function
verbose	Whether to print all added and removed functions and function parameters. Default is TRUE. If FALSE, the function will print removed functions and parameters, but only print counts of added functions and parameters.

Value

The package difference object, invisibly.

See Also

Other pdiff: [pkg_cache\(\)](#), [pkg_diff\(\)](#), [pkg_info\(\)](#), [pkg_stability\(\)](#), [pkg_versions\(\)](#), [print.pcache\(\)](#), [print.pdiff_score\(\)](#), [print.pinfo\(\)](#)

Examples

```

# View versions
pkg_versions("rvest")
#   Package Version      FileName   Release Size
# 1  rvest   1.0.4 rvest_1.0.4.tar.gz 2024-02-12 113K
# 2  rvest   1.0.3 rvest_1.0.3.tar.gz 2022-08-20  92K
# 3  rvest   1.0.2 rvest_1.0.2.tar.gz 2021-10-17  92K
# 4  rvest   1.0.1 rvest_1.0.1.tar.gz 2021-07-26  92K
# 5  rvest   1.0.0 rvest_1.0.0.tar.gz 2021-03-09  92K
# 6  rvest   0.3.6 rvest_0.3.6.tar.gz 2020-07-25 2.7M
# 7  rvest   0.3.5 rvest_0.3.5.tar.gz 2019-11-09 1.1M
# 8  rvest   0.3.4 rvest_0.3.4.tar.gz 2019-05-15 1.6M
# 9  rvest   0.3.3 rvest_0.3.3.tar.gz 2019-04-11 1.6M
#10  rvest   0.3.2 rvest_0.3.2.tar.gz 2016-06-17 1.5M
#11  rvest   0.3.1 rvest_0.3.1.tar.gz 2015-11-11 1.5M
#12  rvest   0.3.0 rvest_0.3.0.tar.gz 2015-09-23 1.5M
#13  rvest   0.2.0 rvest_0.2.0.tar.gz 2015-01-01 5.2M
#14  rvest   0.1.0 rvest_0.1.0.tar.gz 2014-11-22 5.1M

# Find differences in latest version
res <- pkg_diff("rvest", "1.0.3", "1.0.4")

# View differences
print(res)
# # A difference object: rvest package
# - Comparing: v1.0.3/v1.0.4
# - Breaking Changes: FALSE
# - Added Functions:
#   - html_element.LiveHTML()
#   - html_elements.LiveHTML()
#   - html_table.LiveHTML()
#   - LiveHTML()
#   - read_html_live()

```

```
print.pdiff_score      Print a Stability Score Object
```

Description

Print routine for a package stability score object of class "pdiff_score".

Usage

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

Arguments

```
x          The package stability score to print
...        Follow-on parameters to the print function
```

Value

The stability score object, invisibly.

See Also

Other pdiff: [pkg_cache\(\)](#), [pkg_diff\(\)](#), [pkg_info\(\)](#), [pkg_stability\(\)](#), [pkg_versions\(\)](#), [print.pcache\(\)](#), [print.pdiff\(\)](#), [print.pinfo\(\)](#)

Examples

```
# View package stability
res <- pkg_stability("rsample")

# Print stability scores
print(res)
# # A stability score: rsample package
# - Age: 7.67 years
# - Score: 80.7
# - Assessment: Somewhat Unstable
# - Version Range: 0.0.1/1.2.1
# - Release Range: 2017-07-08/2024-03-25
# - Release Count: 16
# - Breaking Releases: 6
# - Data:
#   Package Version      FileName      Release      Size  AF AP RF RP BC TF
# 1 rsample 1.2.1 rsample_1.2.1.tar.gz 2024-03-25 321K  1 0 0 0 0 401
# 2 rsample 1.2.0 rsample_1.2.0.tar.gz 2023-08-23 321K 43 6 1 3 1 400
# 3 rsample 1.1.1 rsample_1.1.1.tar.gz 2022-12-07 318K 37 5 0 0 0 358
# 4 rsample 1.1.0 rsample_1.1.0.tar.gz 2022-08-08 306K 72 1 0 0 0 321
# 5 rsample 1.0.0 rsample_1.0.0.tar.gz 2022-06-24 268K  1 0 1 0 1 249
# 6 rsample 0.1.1 rsample_0.1.1.tar.gz 2021-11-08 274K  4 1 0 1 1 249
# 7 rsample 0.1.0 rsample_0.1.0.tar.gz 2021-05-08 274K  0 5 0 0 0 245
# 8 rsample 0.0.9 rsample_0.0.9.tar.gz 2021-02-17 269K 19 0 6 0 1 245
# 9 rsample 0.0.8 rsample_0.0.8.tar.gz 2020-09-23 261K 59 0 0 0 0 232
#10 rsample 0.0.7 rsample_0.0.7.tar.gz 2020-06-04 248K 101 1 0 0 0 173
#11 rsample 0.0.6 rsample_0.0.6.tar.gz 2020-03-31 299K  7 2 0 0 0 72
#12 rsample 0.0.5 rsample_0.0.5.tar.gz 2019-07-13 297K  4 5 0 0 0 65
#13 rsample 0.0.4 rsample_0.0.4.tar.gz 2019-01-07 254K  0 0 4 0 1 61
#14 rsample 0.0.3 rsample_0.0.3.tar.gz 2018-11-20 254K  7 0 1 0 1 65
#15 rsample 0.0.2 rsample_0.0.2.tar.gz 2017-11-12 339K 24 0 0 0 0 59
#16 rsample 0.0.1 rsample_0.0.1.tar.gz 2017-07-08 180K 35 35 0 0 0 35
```

print.pinfo

Print a Package Info Object

Description

Print routine for a package info object of class "pinfo".

Usage

```
## S3 method for class 'pinfo'
print(x, ..., verbose = FALSE)
```

Arguments

x	The package info to print.
...	Follow-on parameters to the print function.
verbose	Whether to print all function names or not. Default is FALSE. When FALSE, only a count of the functions will be printed.

Value

The package info object, invisibly.

See Also

Other pdiff: [pkg_cache\(\)](#), [pkg_diff\(\)](#), [pkg_info\(\)](#), [pkg_stability\(\)](#), [pkg_versions\(\)](#), [print.pcache\(\)](#), [print.pdiff\(\)](#), [print.pdiff_score\(\)](#)

Examples

```
# Capture package info
res <- pkg_info("patchwork")

# View package info
print(res)
# # A package info object: patchwork package
# - Version: v1.3.0
# - Release Date: 2024-09-16
# - Title: The Composer of Plots
# - Maintainer: Thomas Lin Pedersen <thomasp85@gmail.com>
# - License: MIT + file LICENSE
# - Description: The 'ggplot2' package provides a strong API for sequentially
# building up a plot, but does not concern itself with composition of multiple
# plots. 'patchwork' is a package that expands the API to allow for
# arbitrarily complex composition of plots by, among others, providing
# mathematical operators for combining multiple plots. Other packages that try
# to address this need (but with a different approach) are 'gridExtra' and
# 'cowplot'.
# - Imports: ggplot2 (>= 3.0.0), gtable, grid, stats, grDevices, utils,
# graphics, rlang (>= 1.0.0), cli, farver
# - Suggests: knitr, rmarkdown, gridGraphics, gridExtra, ragg,
# testthat (>= 2.1.0), vdiffr, covr, png, gt (>= 0.11.0)
# - Downloads/Month: 239555
# - Repository: CRAN
# - Cached: TRUE
# - Functions: 90
```

`print.prepo`*Print a Package Repo Object*

Description

Print routine for a package repo object of class "prepo".

Usage

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

Arguments

`x` The package repo to print.
`...` Follow-on parameters to the print function.

Value

The package repo object, invisibly.

See Also

Other prepo: [pkg_repo\(\)](#), [print.rbreak\(\)](#), [print.rstability\(\)](#), [repo_breakages\(\)](#), [repo_stability\(\)](#)

Examples

```
# Create vector of packages  
pkgs <- c("tidymodels", "rsample", "parsnip", "recipes", "workflows")  
  
# Retrieve latest versions  
res <- pkg_repo(pkgs, ver = "latest")  
  
# Print results  
print(res)  
# # A package repo object  
# - Repo Version: latest  
# - Packages:  
#   Package Version  
# 1  parsnip  1.3.0  
# 2  recipes  1.1.1  
# 3  rsample  1.2.1  
# 4  tidymodels  1.3.0  
# 5  workflows  1.2.0
```

print.rbreak *Print a Repo Breakages Object*

Description

Print routine for a repo breakages object of class "rbreak".

Usage

```
## S3 method for class 'rbreak'
print(x, ..., verbose = FALSE)
```

Arguments

x	The repo breakages object to print.
...	Follow-on parameters to the print function.
verbose	If FALSE, prints only the difference removals. If TRUE, prints both additions and removals. Default is FALSE.

Value

The repo breakage object, invisibly.

See Also

Other prepo: [pkg_repo\(\)](#), [print.prepo\(\)](#), [print.rstability\(\)](#), [repo_breakages\(\)](#), [repo_stability\(\)](#)

print.rstability *Print a Repo Stability Object*

Description

Print routine for a repot stability object of class "rstability".

Usage

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

Arguments

x	The repo stability object to print.
...	Follow-on parameters to the print function.

Value

The repo stability object, invisibly.

See Also

Other prepo: [pkg_repo\(\)](#), [print.prepo\(\)](#), [print.rbreak\(\)](#), [repo_breakages\(\)](#), [repo_stability\(\)](#)

Examples

```
# Create vector of packages
vct <- c("curl", "dplyr", "rvest", "tidymodels")

# Get stability scores
res <- repo_stability(vct)

# View stability scores
print(res)
## A repo stability object
# - Run Datetime: 2025-03-01 19:19 UTC
# - Summary:
#   Package   FV   LV   FR   LR TR BR Score   Assessment
# 1   curl    0.2 6.2.1 2014-11-20 2025-02-19 51 1 98.0   Very Stable
# 2   dplyr   0.1 1.1.4 2014-01-16 2023-11-17 45 20 87.5 Somewhat Unstable
# 3   rvest   0.1.0 1.0.4 2014-11-22 2024-02-12 14 4 93.2   Stable
# 4 tidymodels 0.0.1 1.3.0 2018-07-27 2025-02-21 14 0 100.0   Perfect
```

repo_breakages

Identify Breakages for a Repository

Description

The `repo_breakages` function generates a data frame of breakage information for multiple packages. This function can be used to identify breakages for a small set of packages, or even an entire repository.

Usage

```
repo_breakages(r1 = "current", r2 = "latest")
```

Arguments

- `r1` A data frame that identifies the source repository packages and versions. The default value is "current", which means the function will use the current versions of all packages in the current R repository.
- `r2` A data frame that identifies the target repository packages and versions. The default value is "latest", which means the function will use the latest versions of all packages on CRAN.

Details

The `repo_breakages` function aims to help with repository upgrades. The purpose is to identify packages that will break if you upgrade your repository.

The function accepts two data frames as input. Each data frame should have two columns: "Package" and "Version". The "r1" data frame is for your current repo. The "r2" data frame is for the upgrade repo.

When executed, `repo_breakages` will then match up the source and target versions, run a difference operation for each package, and look for breaking changes.

The result will be a table of package names, the source and target version, and whether or not there are any breakages. If there are breakages, the difference object for that package will be added to a "details" list. This difference object can show you exactly which functions or function parameters were removed.

The `repo_breakages` function is one of primary motivations for the **pkgdiff** package. This mass-comparison ability did not exist in any other R package prior to **pkgdiff**.

Value

A list containing a summary table and a list of difference objects. The summary table will contain one row for each package. The columns show the package name, source and target versions, and whether there are any breaking changes between the specified versions. If breaking changes are found, the difference object for that package will be included in the details list.

See Also

Use the [pkg_repo](#) function to help gather version information for each package set. Also see [repo_stability](#) to collect stability information on multiple packages.

Other prepo: [pkg_repo\(\)](#), [print.prepo\(\)](#), [print.rbreak\(\)](#), [print.rstability\(\)](#), [repo_stability\(\)](#)

Examples

```
# Create package vector
pkgs <- c("curl", "dplyr", "crayon", "stringr")

# Backdated versions
r1 <- data.frame(Package = pkgs,
                 Version = c("5.2.1", "1.1.4", "1.5.2", "1.5.0"))

# More recent versions
r2 <- data.frame(Package = pkgs,
                 Version = c("6.2.1", "1.1.4", "1.5.3", "1.5.1"))

# Find any breaking changes
# res <- repo_breakages(r1, r2) # - Commented out to pass CRAN checks
# Comparing curl v5.2.1/v6.2.1
# Comparing crayon v1.5.2/v1.5.3
# Comparing stringr v1.5.0/v1.5.1

# View results
# res # - Commented out to pass CRAN checks
```



```

# # A repo breakages object
# - Run Datetime: 2025-03-01 17:47 UTC
# - Summary:
#   Package Version1 Version2 Breakages
# 1   curl    5.2.1    6.2.1    TRUE
# 2  dplyr    1.1.4    1.1.4    FALSE
# 3  crayon   1.5.2    1.5.3    FALSE
# 4 stringr  1.5.0    1.5.1    FALSE
# - Details:
#   # A difference object: curl package
#   - Comparing: v5.2.1/v6.2.1
#   - Breaking Changes: TRUE
#   - Added Functions: 1
#   - Added Parameters: 3
#   - Removed Parameters:
#     - multi_download(): timeout

```

repo_stability

Generate Stability Scores for a Repository

Description

The function generates stability scores for a vector of packages. If passing a large number of packages, be prepared for the function to run for a considerable amount of time.

Usage

```
repo_stability(pkgs, releases = NULL, months = NULL)
```

Arguments

pkgs	A vector of package names.
releases	An integer number of releases to assess. The function will then limit the scope of the assessment to the specified number of releases.
months	An integer number of months from the current date. The function will then limit the scope of the assessment to the specified number of months.

Details

To assess stability for a package, **pkgdiff** has to compare all releases of a package, and identify breaking changes between each release. This comparison takes time, especially for packages that have been active for many years.

Therefore, the first step of the function is to compare all releases for each package in the input vector. The function will send messages to the console to let you know which package it is currently comparing. These messages will let you know how far the function is through the processing.

Once all the comparisons are completed, the function will return a data frame summary of stability results. The data frame will have one row for each package in the input vector. The data frame will also show some information about the comparison, and give a stability assessment for each package. The data frame columns are as follows:

- **Package:** The package name.
- **FV:** The first version of the package.
- **LV:** The last version of the package.
- **FR:** The first release date of the package.
- **LR:** The last release date of the package.
- **TR:** The total number of releases.
- **BR:** The number of breaking releases.
- **Score:** The stability score.
- **Assessment:** The stability assessment.

To learn how the package stability score and assessment are calculated, see `vignette("pkgdiff-stability")`. Additional information is included in the documentation of the `pkg_stability` function.

If a package is not found in the package cache, the function will download and compare each version of the package on CRAN. This process may increase the processing time. See `vignette("pkg-cache")` and `pkg_cache` for additional information.

If a package does not exist on CRAN, a row for that package will still be returned. However, all data values will be NA. This situation may occur if a package exists on Github, but has not yet been submitted to CRAN. The `pkgdiff` stability functions only work with packages that have been published to CRAN.

Value

A data frame of information regarding the stability of each package in the input vector.

See Also

Other prepo: `pkg_repo()`, `print.prepo()`, `print.rbreak()`, `print.rstability()`, `repo_breakages()`

Examples

```
# Create vector of packages
vct <- c("curl", "dplyr", "rvest", "tidymodels")

# Get stability scores
res <- repo_stability(vct)
# Getting stability score for package 'curl'...
# Getting stability score for package 'dplyr'...
# Getting stability score for package 'rvest'...
# Getting stability score for package 'tidymodels'...

# View stability results
res
# A repo stability object
# - Run Datetime: 2025-03-01 17:53 UTC
# - Summary:
#   Package   FV   LV     FR     LR TR BR Score   Assessment
# 1     curl   0.2 6.2.1 2014-11-20 2025-02-19 51 1 98.0     Very Stable
# 2     dplyr   0.1 1.1.4 2014-01-16 2023-11-17 45 20 87.5 Somewhat Unstable
```

# 3	rvest	0.1.0	1.0.4	2014-11-22	2024-02-12	14	4	93.2	Stable
# 4	tidymodels	0.0.1	1.3.0	2018-07-27	2025-02-21	14	0	100.0	Perfect

Index

* **pdiff**

- pkg_cache, 2
- pkg_diff, 3
- pkg_info, 6
- pkg_stability, 10
- pkg_versions, 15
- print.pcache, 16
- print.pdiff, 17
- print.pdiff_score, 18
- print.pinfo, 19

* **prepo**

- pkg_repo, 9
- print.prepo, 21
- print.rbreak, 22
- print.rstability, 22
- repo_breakages, 23
- repo_stability, 25

pkg_cache, 2, 4, 7, 12, 15–17, 19, 20, 26
pkg_diff, 3, 3, 7, 12, 15–17, 19, 20
pkg_info, 3, 4, 6, 12, 15–17, 19, 20
pkg_repo, 9, 21–24, 26
pkg_stability, 3, 4, 7, 10, 15–17, 19, 20, 26
pkg_versions, 3, 4, 7, 12, 15, 16, 17, 19, 20
print.pcache, 3, 4, 7, 12, 15, 16, 17, 19, 20
print.pdiff, 3, 4, 7, 12, 15, 16, 17, 19, 20
print.pdiff_score, 3, 4, 7, 12, 15–17, 18, 20
print.pinfo, 3, 4, 7, 12, 15–17, 19, 19
print.prepo, 10, 21, 22–24, 26
print.rbreak, 10, 21, 22, 23, 24, 26
print.rstability, 10, 21, 22, 22, 24, 26

repo_breakages, 10, 21–23, 23, 26
repo_stability, 10, 21–24, 25