

# Package ‘remotes’

April 10, 2019

**Title** R Package Installation from Remote Repositories, Including 'GitHub'

**Version** 2.0.4

**Description** Download and install R packages stored in 'GitHub', 'BitBucket', or plain 'subversion' or 'git' repositories. This package provides the 'install\_\*' functions in 'devtools'.  
Indeed most of the code was copied over from 'devtools'.

**License** GPL (>= 2)

**URL** <https://github.com/r-lib/remotes#readme>

**BugReports** <https://github.com/r-lib/remotes/issues>

**Imports** methods, stats, tools, utils

**Suggests** brew, callr, curl, covr, git2r (>= 0.23.0), knitr, mockery, pkgbuild (>= 1.0.1), pingr, rmarkdown, rprojroot, testthat, withr

**Depends** R (>= 3.0.0)

**VignetteBuilder** knitr

**RoxygenNote** 6.1.1

**SystemRequirements** Subversion for install\_svn, git for install\_git

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Jim Hester [aut, cre],  
Gábor Csárdi [aut],  
Hadley Wickham [aut],  
Winston Chang [aut],  
RStudio [cph],  
Martin Morgan [aut],  
Dan Tenenbaum [aut],  
Mango Solutions [cph]

**Maintainer** Jim Hester <james.hester@rstudio.com>

**Repository** CRAN

**Date/Publication** 2019-04-10 16:32:54 UTC

**R topics documented:**

download_version . . . . .	2
github_pull . . . . .	3
install_bioc . . . . .	3
install_bitbucket . . . . .	5
install_cran . . . . .	7
install_deps . . . . .	8
install_dev . . . . .	9
install_git . . . . .	10
install_github . . . . .	11
install_gitlab . . . . .	13
install_local . . . . .	14
install_svn . . . . .	16
install_url . . . . .	17
install_version . . . . .	18
package_deps . . . . .	20
parse-git-repo . . . . .	21
update_packages . . . . .	22
<b>Index</b>	<b>24</b>

---

download_version	<i>Download a specified version of a CRAN package</i>
------------------	---

---

**Description**

It downloads the package to a temporary file, and returns the name of the file.

**Usage**

```
download_version(package, version = NULL, repos = getOption("repos"),
  type = getOption("pkgType"), ...)
```

**Arguments**

package	package name
version	If the specified version is NULL or the same as the most recent version of the package, this function simply calls <code>utils::install.packages()</code> . Otherwise, it looks at the list of archived source tarballs and tries to install an older version instead.
repos	character vector, the base URL(s) of the repositories to use, e.g., the URL of a CRAN mirror such as "https://cloud.r-project.org". For more details on supported URL schemes see <a href="#">url</a> . Can be NULL to install from local files, directories or URLs: this will be inferred by extension from pkgs if of length one.

type character, indicating the type of package to download and install. Will be "source" except on Windows and some macOS builds: see the section on 'Binary packages' for those.

... Other arguments passed on to `utils::install.packages()`.

**Value**

Name of the downloaded file.

---

github\_pull *GitHub references*

---

**Description**

Use as ref parameter to `install_github()`. Allows installing a specific pull request or the latest release.

**Usage**

```
github_pull(pull)
```

```
github_release()
```

**Arguments**

pull The pull request to install

**See Also**

[install\\_github\(\)](#)

---

install\_bioc *Install a package from a Bioconductor repository*

---

**Description**

This function requires git to be installed on your system in order to be used.

**Usage**

```
install_bioc(repo, mirror = getOption("BioC_git",
  download_url("git.bioconductor.org/packages")), git = c("auto",
  "git2r", "external"), dependencies = NA, upgrade = c("default",
  "ask", "always", "never"), force = FALSE, quiet = FALSE,
  build = TRUE, build_opts = c("--no-resave-data", "--no-manual",
  "--no-build-vignettes"), repos = getOption("repos"),
  type = getOption("pkgType"), ...)
```

**Arguments**

repo	Repository address in the format [username:password@][release/]repo[#commit]. Valid values for the release are 'devel', 'release' (the default if none specified), or numeric release numbers (e.g. '3.3').
mirror	The bioconductor git mirror to use
git	Whether to use the git2r package, or an external git client via system. Default is git2r if it is installed, otherwise an external git installation.
dependencies	Which dependencies do you want to check? Can be a character vector (selecting from "Depends", "Imports", "LinkingTo", "Suggests", or "Enhances"), or a logical vector.  TRUE is shorthand for "Depends", "Imports", "LinkingTo" and "Suggests". NA is shorthand for "Depends", "Imports" and "LinkingTo" and is the default. FALSE is shorthand for no dependencies (i.e. just check this package, not its dependencies).
upgrade	One of "default", "ask", "always", or "never". "default" respects the value of the R_REMOTES_UPGRADE environment variable if set, and falls back to "ask" if unset. "ask" prompts the user for which out of date packages to upgrade. For non-interactive sessions "ask" is equivalent to "always". TRUE and FALSE are also accepted and correspond to "always" and "never" respectively.
force	Force installation, even if the remote state has not changed since the previous install.
quiet	If TRUE, suppress output.
build	If TRUE build the package before installing.
build_opts	Options to pass to R CMD build, only used when build is TRUE.
repos	A character vector giving repositories to use.
type	Type of package to update.
...	Other arguments passed on to <code>utils::install.packages()</code> .

**Details**

It is vectorised so you can install multiple packages with a single command.

**See Also**

Other package installation: [install\\_bitbucket](#), [install\\_cran](#), [install\\_dev](#), [install\\_github](#), [install\\_gitlab](#), [install\\_git](#), [install\\_local](#), [install\\_svn](#), [install\\_url](#), [install\\_version](#)

**Examples**

```
## Not run:
install_bioc("SummarizedExperiment")
install_bioc("release/SummarizedExperiment")
install_bioc("3.3/SummarizedExperiment")
install_bioc("SummarizedExperiment#abc123")
```

```
install_bioc("user:password@release/SummarizedExperiment")
install_bioc("user:password@devel/SummarizedExperiment")
install_bioc("user:password@SummarizedExperiment#abc123")

## End(Not run)
```

---

install\_bitbucket      *Install a package directly from bitbucket*

---

## Description

This function is vectorised so you can install multiple packages in a single command.

## Usage

```
install_bitbucket(repo, ref = "master", subdir = NULL,
  auth_user = bitbucket_user(), password = bitbucket_password(),
  host = "api.bitbucket.org/2.0", dependencies = NA,
  upgrade = c("default", "ask", "always", "never"), force = FALSE,
  quiet = FALSE, build = TRUE, build_opts = c("--no-resave-data",
  "--no-manual", "--no-build-vignettes"), repos = getOption("repos"),
  type = getOption("pkgType"), ...)
```

## Arguments

repo	Repository address in the format username/repo[/subdir][@ref #pull]. Alternatively, you can specify subdir and/or ref using the respective parameters (see below); if both is specified, the values in repo take precedence.
ref	Desired git reference; could be a commit, tag, or branch name. Defaults to master.
subdir	subdirectory within repo that contains the R package.
auth_user	your account username if you're attempting to install a package hosted in a private repository (and your username is different to username). Defaults to the BITBUCKET_USER environment variable.
password	your password. Defaults to the BITBUCKET_PASSWORD environment variable. See details for further information on setting up a password.
host	GitHub API host to use. Override with your GitHub enterprise hostname, for example, "github.hostname.com/api/v3".
dependencies	Which dependencies do you want to check? Can be a character vector (selecting from "Depends", "Imports", "LinkingTo", "Suggests", or "Enhances"), or a logical vector.  TRUE is shorthand for "Depends", "Imports", "LinkingTo" and "Suggests". NA is shorthand for "Depends", "Imports" and "LinkingTo" and is the default. FALSE is shorthand for no dependencies (i.e. just check this package, not its dependencies).

upgrade	One of "default", "ask", "always", or "never". "default" respects the value of the R_REMOTES_UPGRADE environment variable if set, and falls back to "ask" if unset. "ask" prompts the user for which out of date packages to upgrade. For non-interactive sessions "ask" is equivalent to "always". TRUE and FALSE are also accepted and correspond to "always" and "never" respectively.
force	Force installation, even if the remote state has not changed since the previous install.
quiet	If TRUE, suppress output.
build	If TRUE build the package before installing.
build_opts	Options to pass to R CMD build, only used when build is TRUE.
repos	A character vector giving repositories to use.
type	Type of package to update.
...	Other arguments passed on to <code>utils::install.packages()</code> .

### Details

To install from a private repo, or more generally, access the Bitbucket API with your own credentials, you will need to get an access token. You can create an access token following the instructions found in the [Bitbucket App Passwords documentation](#). The App Password requires read-only access to your repositories and pull requests. Then store your password in the environment variable BITBUCKET\_PASSWORD (e.g. `evelynwaugh:swor dofhonour`)

Note that on Windows, authentication requires the "libcurl" download method. You can set the default download method via the `download.file.method` option:

```
options(download.file.method = "libcurl")
```

In particular, if unset, RStudio sets the download method to "wininet". To override this, you might want to set it to "libcurl" in your R profile, see [base::Startup](#). The caveat of the "libcurl" method is that it does *not* set the system proxies automatically, see "Setting Proxies" in [utils::download.file\(\)](#).

### See Also

Bitbucket API docs: <https://confluence.atlassian.com/bitbucket/use-the-bitbucket-cloud-rest-apis-22272.html>

Other package installation: [install\\_bioc](#), [install\\_cran](#), [install\\_dev](#), [install\\_github](#), [install\\_gitlab](#), [install\\_git](#), [install\\_local](#), [install\\_svn](#), [install\\_url](#), [install\\_version](#)

### Examples

```
## Not run:
install_bitbucket("sulab/mygene.r@default")
install_bitbucket("djenavarro/lsr")

## End(Not run)
```

---

install_cran	<i>Attempts to install a package from CRAN.</i>
--------------	---

---

## Description

This function is vectorised on pkgs so you can install multiple packages in a single command.

## Usage

```
install_cran(pkgs, repos = getOption("repos"),
             type = getOption("pkgType"), dependencies = NA,
             upgrade = c("default", "ask", "always", "never"), force = FALSE,
             quiet = FALSE, build = TRUE, build_opts = c("--no-resave-data",
             "--no-manual", "--no-build-vignettes"), ...)
```

## Arguments

pkgs	Character vector of packages to install.
repos	A character vector giving repositories to use.
type	Type of package to update.
dependencies	Which dependencies do you want to check? Can be a character vector (selecting from "Depends", "Imports", "LinkingTo", "Suggests", or "Enhances"), or a logical vector. TRUE is shorthand for "Depends", "Imports", "LinkingTo" and "Suggests". NA is shorthand for "Depends", "Imports" and "LinkingTo" and is the default. FALSE is shorthand for no dependencies (i.e. just check this package, not its dependencies).
upgrade	One of "default", "ask", "always", or "never". "default" respects the value of the R_REMOTES_UPGRADE environment variable if set, and falls back to "ask" if unset. "ask" prompts the user for which out of date packages to upgrade. For non-interactive sessions "ask" is equivalent to "always". TRUE and FALSE are also accepted and correspond to "always" and "never" respectively.
force	Force installation, even if the remote state has not changed since the previous install.
quiet	If TRUE, suppress output.
build	If TRUE build the package before installing.
build_opts	Options to pass to R CMD build, only used when build is TRUE.
...	Other arguments passed on to <code>utils::install.packages()</code> .

## See Also

Other package installation: [install\\_bioc](#), [install\\_bitbucket](#), [install\\_dev](#), [install\\_github](#), [install\\_gitlab](#), [install\\_git](#), [install\\_local](#), [install\\_svn](#), [install\\_url](#), [install\\_version](#)

**Examples**

```
## Not run:
install_cran("ggplot2")
install_cran(c("httpuv", "shiny"))

## End(Not run)
```

---

<code>install_deps</code>	<i>Install package dependencies if needed.</i>
---------------------------	--

---

**Description**

Install package dependencies if needed.

**Usage**

```
install_deps(pkgdir = ".", dependencies = NA,
             repos = getOption("repos"), type = getOption("pkgType"),
             upgrade = c("default", "ask", "always", "never"), quiet = FALSE,
             build = TRUE, build_opts = c("--no-resave-data", "--no-manual",
             "--no-build-vignettes"), ...)
```

**Arguments**

<code>pkgdir</code>	path to a package directory, or to a package tarball.
<code>dependencies</code>	Which dependencies do you want to check? Can be a character vector (selecting from "Depends", "Imports", "LinkingTo", "Suggests", or "Enhances"), or a logical vector. TRUE is shorthand for "Depends", "Imports", "LinkingTo" and "Suggests". NA is shorthand for "Depends", "Imports" and "LinkingTo" and is the default. FALSE is shorthand for no dependencies (i.e. just check this package, not its dependencies).
<code>repos</code>	A character vector giving repositories to use.
<code>type</code>	Type of package to update.
<code>upgrade</code>	One of "default", "ask", "always", or "never". "default" respects the value of the <code>R_REMOTES_UPGRADE</code> environment variable if set, and falls back to "ask" if unset. "ask" prompts the user for which out of date packages to upgrade. For non-interactive sessions "ask" is equivalent to "always". TRUE and FALSE are also accepted and correspond to "always" and "never" respectively.
<code>quiet</code>	If TRUE, suppress output.
<code>build</code>	If TRUE build the package before installing.
<code>build_opts</code>	Options to pass to R CMD build, only used when build is TRUE.
<code>...</code>	additional arguments passed to <code>utils::install.packages()</code> .

**Examples**

```
## Not run: install_deps(".")
```



---

install_dev	<i>Install the development version of a package</i>
-------------	---

---

### Description

`install_dev()` retrieves the package DESCRIPTION from the CRAN mirror and looks in the 'URL' and 'BugReports' fields for GitHub, GitLab or Bitbucket URLs. It then calls the appropriate `install_()` function to install the development package.

### Usage

```
install_dev(package, cran_url = getOption("repos")[["CRAN"]], ...)
```

### Arguments

package	The package name to install.
cran_url	The URL of the CRAN mirror to use, by default based on the 'repos' option. If unset uses 'https://cloud.r-project.org'.
...	Additional arguments passed to <a href="#">install_github()</a> , <a href="#">install_gitlab()</a> , or <a href="#">install_bitbucket()</a> functions.

### See Also

Other package installation: [install\\_bioc](#), [install\\_bitbucket](#), [install\\_cran](#), [install\\_github](#), [install\\_gitlab](#), [install\\_git](#), [install\\_local](#), [install\\_svn](#), [install\\_url](#), [install\\_version](#)

### Examples

```
## Not run:  
# From GitHub  
install_dev("dplyr")  
  
# From GitLab  
install_dev("iemiscdata")  
  
# From Bitbucket  
install_dev("argparser")  
  
## End(Not run)
```

---

install_git	<i>Install a package from a git repository</i>
-------------	--

---

### Description

It is vectorised so you can install multiple packages with a single command. You do not need to have the git2r package, or an external git client installed.

### Usage

```
install_git(url, subdir = NULL, ref = NULL, branch = NULL,
  credentials = NULL, git = c("auto", "git2r", "external"),
  dependencies = NA, upgrade = c("default", "ask", "always", "never"),
  force = FALSE, quiet = FALSE, build = TRUE,
  build_opts = c("--no-resave-data", "--no-manual",
    "--no-build-vignettes"), repos = getOption("repos"),
  type = getOption("pkgType"), ...)
```

### Arguments

url	Location of package. The url should point to a public or private repository.
subdir	A sub-directory within a git repository that may contain the package we are interested in installing.
ref	Name of branch, tag or SHA reference to use, if not HEAD.
branch	Deprecated, synonym for ref.
credentials	A git2r credentials object passed through to clone. Supplying this argument implies using git2r with git.
git	Whether to use the git2r package, or an external git client via system. Default is git2r if it is installed, otherwise an external git installation.
dependencies	Which dependencies do you want to check? Can be a character vector (selecting from "Depends", "Imports", "LinkingTo", "Suggests", or "Enhances"), or a logical vector. TRUE is shorthand for "Depends", "Imports", "LinkingTo" and "Suggests". NA is shorthand for "Depends", "Imports" and "LinkingTo" and is the default. FALSE is shorthand for no dependencies (i.e. just check this package, not its dependencies).
upgrade	One of "default", "ask", "always", or "never". "default" respects the value of the R_REMOTES_UPGRADE environment variable if set, and falls back to "ask" if unset. "ask" prompts the user for which out of date packages to upgrade. For non-interactive sessions "ask" is equivalent to "always". TRUE and FALSE are also accepted and correspond to "always" and "never" respectively.
force	Force installation, even if the remote state has not changed since the previous install.
quiet	If TRUE, suppress output.

build	If TRUE build the package before installing.
build_opts	Options to pass to R CMD build, only used when build is TRUE.
repos	A character vector giving repositories to use.
type	Type of package to update.
...	Other arguments passed on to <code>utils::install.packages()</code> .

### See Also

Other package installation: [install\\_bioc](#), [install\\_bitbucket](#), [install\\_cran](#), [install\\_dev](#), [install\\_github](#), [install\\_gitlab](#), [install\\_local](#), [install\\_svn](#), [install\\_url](#), [install\\_version](#)

### Examples

```
## Not run:
install_git("git://github.com/hadley/stringr.git")
install_git("git://github.com/hadley/stringr.git", ref = "stringr-0.2")

## End(Not run)
```

---

install_github	<i>Attempts to install a package directly from GitHub.</i>
----------------	--

---

### Description

This function is vectorised on repo so you can install multiple packages in a single command.

### Usage

```
install_github(repo, ref = "master", subdir = NULL,
  auth_token = github_pat(), host = "api.github.com",
  dependencies = NA, upgrade = c("default", "ask", "always", "never"),
  force = FALSE, quiet = FALSE, build = TRUE,
  build_opts = c("--no-resave-data", "--no-manual",
    "--no-build-vignettes"), repos = getOption("repos"),
  type = getOption("pkgType"), ...)
```

### Arguments

repo	Repository address in the format <code>username/repo[/subdir][@ref #pull]</code> . Alternatively, you can specify <code>subdir</code> and/or <code>ref</code> using the respective parameters (see below); if both is specified, the values in <code>repo</code> take precedence.
ref	Desired git reference. Could be a commit, tag, or branch name, or a call to <a href="#">github_pull()</a> . Defaults to "master".
subdir	subdirectory within repo that contains the R package.

auth_token	To install from a private repo, generate a personal access token (PAT) in <a href="https://github.com/settings/tokens">https://github.com/settings/tokens</a> and supply to this argument. This is safer than using a password because you can easily delete a PAT without affecting any others. Defaults to the GITHUB_PAT environment variable.
host	GitHub API host to use. Override with your GitHub enterprise hostname, for example, "github.hostname.com/api/v3".
dependencies	Which dependencies do you want to check? Can be a character vector (selecting from "Depends", "Imports", "LinkingTo", "Suggests", or "Enhances"), or a logical vector. TRUE is shorthand for "Depends", "Imports", "LinkingTo" and "Suggests". NA is shorthand for "Depends", "Imports" and "LinkingTo" and is the default. FALSE is shorthand for no dependencies (i.e. just check this package, not its dependencies).
upgrade	One of "default", "ask", "always", or "never". "default" respects the value of the R_REMOTES_UPGRADE environment variable if set, and falls back to "ask" if unset. "ask" prompts the user for which out of date packages to upgrade. For non-interactive sessions "ask" is equivalent to "always". TRUE and FALSE are also accepted and correspond to "always" and "never" respectively.
force	Force installation, even if the remote state has not changed since the previous install.
quiet	If TRUE, suppress output.
build	If TRUE build the package before installing.
build_opts	Options to pass to R CMD build, only used when build is TRUE.
repos	A character vector giving repositories to use.
type	Type of package to update.
...	Other arguments passed on to <code>utils::install.packages()</code> .

### Details

If the repository uses submodules a command-line git client is required to clone the submodules.

### See Also

[github\\_pull\(\)](#)

Other package installation: [install\\_bioc](#), [install\\_bitbucket](#), [install\\_cran](#), [install\\_dev](#), [install\\_gitlab](#), [install\\_git](#), [install\\_local](#), [install\\_svn](#), [install\\_url](#), [install\\_version](#)

### Examples

```
## Not run:
install_github("klutometis/roxygen")
install_github("wch/ggplot2")
install_github(c("rstudio/httpuv", "rstudio/shiny"))
install_github(c("hadley/httr@v0.4", "klutometis/roxygen#142",
  "mfrasca/r-logging/pkg"))
```

```
# To install from a private repo, use auth_token with a token
# from https://github.com/settings/tokens. You only need the
# repo scope. Best practice is to save your PAT in env var called
# GITHUB_PAT.
install_github("hadley/private", auth_token = "abc")

## End(Not run)
```

---

install_gitlab	<i>Install a package from GitLab</i>
----------------	--------------------------------------

---

## Description

This function is vectorised on `repo` so you can install multiple packages in a single command. Like other remotes the repository will skip installation if `force == FALSE` (the default) and the remote state has not changed since the previous installation.

## Usage

```
install_gitlab(repo, auth_token = gitlab_pat(), host = "gitlab.com",
  dependencies = NA, upgrade = c("default", "ask", "always", "never"),
  force = FALSE, quiet = FALSE, build = TRUE,
  build_opts = c("--no-resave-data", "--no-manual",
    "--no-build-vignettes"), repos = getOption("repos"),
  type = getOption("pkgType"), ...)
```

## Arguments

<code>repo</code>	Repository address in the format <code>username/repo[/subdir][@ref]</code> .
<code>auth_token</code>	To install from a private repo, generate a personal access token (PAT) in <a href="https://gitlab.com/profile/personal_access_tokens">https://gitlab.com/profile/personal_access_tokens</a> and supply to this argument. This is safer than using a password because you can easily delete a PAT without affecting any others. Defaults to the <code>GITLAB_PAT</code> environment variable.
<code>host</code>	GitLab API host to use. Override with your GitLab enterprise hostname, for example, <code>gitlab.hostname.com</code> .
<code>dependencies</code>	Which dependencies do you want to check? Can be a character vector (selecting from "Depends", "Imports", "LinkingTo", "Suggests", or "Enhances"), or a logical vector.  TRUE is shorthand for "Depends", "Imports", "LinkingTo" and "Suggests". NA is shorthand for "Depends", "Imports" and "LinkingTo" and is the default. FALSE is shorthand for no dependencies (i.e. just check this package, not its dependencies).

upgrade	One of "default", "ask", "always", or "never". "default" respects the value of the R_REMOTES_UPGRADE environment variable if set, and falls back to "ask" if unset. "ask" prompts the user for which out of date packages to upgrade. For non-interactive sessions "ask" is equivalent to "always". TRUE and FALSE are also accepted and correspond to "always" and "never" respectively.
force	Force installation, even if the remote state has not changed since the previous install.
quiet	If TRUE, suppress output.
build	If TRUE build the package before installing.
build_opts	Options to pass to R CMD build, only used when build is TRUE.
repos	A character vector giving repositories to use.
type	Type of package to update.
...	Other arguments passed on to <code>utils::install.packages()</code> .

### See Also

Other package installation: [install\\_bioc](#), [install\\_bitbucket](#), [install\\_cran](#), [install\\_dev](#), [install\\_github](#), [install\\_git](#), [install\\_local](#), [install\\_svn](#), [install\\_url](#), [install\\_version](#)

### Examples

```
## Not run:
install_gitlab("jimhester/covr")

## End(Not run)
```

---

install_local	<i>Install a package from a local file</i>
---------------	--

---

### Description

This function is vectorised so you can install multiple packages in a single command.

### Usage

```
install_local(path = ".", subdir = NULL, dependencies = NA,
  upgrade = c("default", "ask", "always", "never"), force = FALSE,
  quiet = FALSE, build = !is_binary_pkg(path),
  build_opts = c("--no-resave-data", "--no-manual",
    "--no-build-vignettes"), repos = getOption("repos"),
  type = getOption("pkgType"), ...)
```

**Arguments**

path	path to local directory, or compressed file (tar, zip, tar.gz tar.bz2, tgz2 or tbz)
subdir	subdirectory within url bundle that contains the R package.
dependencies	Which dependencies do you want to check? Can be a character vector (selecting from "Depends", "Imports", "LinkingTo", "Suggests", or "Enhances"), or a logical vector.  TRUE is shorthand for "Depends", "Imports", "LinkingTo" and "Suggests". NA is shorthand for "Depends", "Imports" and "LinkingTo" and is the default. FALSE is shorthand for no dependencies (i.e. just check this package, not its dependencies).
upgrade	One of "default", "ask", "always", or "never". "default" respects the value of the R_REMOTES_UPGRADE environment variable if set, and falls back to "ask" if unset. "ask" prompts the user for which out of date packages to upgrade. For non-interactive sessions "ask" is equivalent to "always". TRUE and FALSE are also accepted and correspond to "always" and "never" respectively.
force	Force installation, even if the remote state has not changed since the previous install.
quiet	If TRUE, suppress output.
build	If TRUE build the package before installing.
build_opts	Options to pass to R CMD build, only used when build is TRUE.
repos	A character vector giving repositories to use.
type	Type of package to update.
...	Other arguments passed on to <code>utils::install.packages()</code> .

**See Also**

Other package installation: [install\\_bioc](#), [install\\_bitbucket](#), [install\\_cran](#), [install\\_dev](#), [install\\_github](#), [install\\_gitlab](#), [install\\_git](#), [install\\_svn](#), [install\\_url](#), [install\\_version](#)

**Examples**

```
## Not run:
dir <- tempfile()
dir.create(dir)
pkg <- download.packages("testthat", dir, type = "source")
install_local(pkg[, 2])

## End(Not run)
```

install\_svn

*Install a package from a SVN repository***Description**

This function requires svn to be installed on your system in order to be used.

**Usage**

```
install_svn(url, subdir = NULL, args = character(0), revision = NULL,
  dependencies = NA, upgrade = c("default", "ask", "always", "never"),
  force = FALSE, quiet = FALSE, build = TRUE,
  build_opts = c("--no-resave-data", "--no-manual",
    "--no-build-vignettes"), repos = getOption("repos"),
  type = getOption("pkgType"), ...)
```

**Arguments**

url	Location of package. The url should point to a public or private repository.
subdir	A sub-directory within a svn repository that contains the package we are interested in installing.
args	A character vector providing extra options to pass on to svn.
revision	svn revision, if omitted updates to latest
dependencies	Which dependencies do you want to check? Can be a character vector (selecting from "Depends", "Imports", "LinkingTo", "Suggests", or "Enhances"), or a logical vector. TRUE is shorthand for "Depends", "Imports", "LinkingTo" and "Suggests". NA is shorthand for "Depends", "Imports" and "LinkingTo" and is the default. FALSE is shorthand for no dependencies (i.e. just check this package, not its dependencies).
upgrade	One of "default", "ask", "always", or "never". "default" respects the value of the R_REMOTES_UPGRADE environment variable if set, and falls back to "ask" if unset. "ask" prompts the user for which out of date packages to upgrade. For non-interactive sessions "ask" is equivalent to "always". TRUE and FALSE are also accepted and correspond to "always" and "never" respectively.
force	Force installation, even if the remote state has not changed since the previous install.
quiet	If TRUE, suppress output.
build	If TRUE build the package before installing.
build_opts	Options to pass to R CMD build, only used when build is TRUE.
repos	A character vector giving repositories to use.
type	Type of package to update.
...	Other arguments passed on to <code>utils::install.packages()</code> .



**Details**

It is vectorised so you can install multiple packages with a single command.

**See Also**

Other package installation: [install\\_bioc](#), [install\\_bitbucket](#), [install\\_cran](#), [install\\_dev](#), [install\\_github](#), [install\\_gitlab](#), [install\\_git](#), [install\\_local](#), [install\\_url](#), [install\\_version](#)

**Examples**

```
## Not run:
install_svn("https://github.com/hadley/stringr/trunk")
install_svn("https://github.com/hadley/htr/branches/oauth")

## End(Not run)
```

---

install\_url

*Install a package from a url*


---

**Description**

This function is vectorised so you can install multiple packages in a single command.

**Usage**

```
install_url(url, subdir = NULL, dependencies = NA,
  upgrade = c("default", "ask", "always", "never"), force = FALSE,
  quiet = FALSE, build = TRUE, build_opts = c("--no-resave-data",
  "--no-manual", "--no-build-vignettes"), repos = getOption("repos"),
  type = getOption("pkgType"), ...)
```

**Arguments**

url	location of package on internet. The url should point to a zip file, a tar file or a bziped/gzipped tar file.
subdir	subdirectory within url bundle that contains the R package.
dependencies	Which dependencies do you want to check? Can be a character vector (selecting from "Depends", "Imports", "LinkingTo", "Suggests", or "Enhances"), or a logical vector. TRUE is shorthand for "Depends", "Imports", "LinkingTo" and "Suggests". NA is shorthand for "Depends", "Imports" and "LinkingTo" and is the default. FALSE is shorthand for no dependencies (i.e. just check this package, not its dependencies).
upgrade	One of "default", "ask", "always", or "never". "default" respects the value of the R_REMOTES_UPGRADE environment variable if set, and falls back to "ask" if unset. "ask" prompts the user for which out of date packages to upgrade. For non-interactive sessions "ask" is equivalent to "always". TRUE and FALSE are also accepted and correspond to "always" and "never" respectively.

force	Force installation, even if the remote state has not changed since the previous install.
quiet	If TRUE, suppress output.
build	If TRUE build the package before installing.
build_opts	Options to pass to R CMD build, only used when build is TRUE.
repos	A character vector giving repositories to use.
type	Type of package to update.
...	Other arguments passed on to <code>utils::install.packages()</code> .

**See Also**

Other package installation: [install\\_bioc](#), [install\\_bitbucket](#), [install\\_cran](#), [install\\_dev](#), [install\\_github](#), [install\\_gitlab](#), [install\\_git](#), [install\\_local](#), [install\\_svn](#), [install\\_version](#)

**Examples**

```
## Not run:
install_url("https://github.com/hadley/stringr/archive/master.zip")

## End(Not run)
```

---

install_version	<i>Install specified version of a CRAN package.</i>
-----------------	---

---

**Description**

If you are installing an package that contains compiled code, you will need to have an R development environment installed. You can check if you do by running `devtools::has_devel` (you need the `devtools` package for this).

**Usage**

```
install_version(package, version = NULL, dependencies = NA,
  upgrade = c("default", "ask", "always", "never"), force = FALSE,
  quiet = FALSE, build = FALSE, build_opts = c("--no-resave-data",
  "--no-manual", "--no-build-vignettes"), repos = getOption("repos"),
  type = "source", ...)
```

**Arguments**

package	package name
version	If the specified version is NULL or the same as the most recent version of the package, this function simply calls <code>utils::install.packages()</code> . Otherwise, it looks at the list of archived source tarballs and tries to install an older version instead.

dependencies	<p>logical indicating whether to also install uninstalled packages which these packages depend on/link to/import/suggest (and so on recursively). Not used if repos = NULL. Can also be a character vector, a subset of c("Depends", "Imports", "LinkingTo", "Suggests").</p> <p>Only supported if lib is of length one (or missing), so it is unambiguous where to install the dependent packages. If this is not the case it is ignored, with a warning.</p> <p>The default, NA, means c("Depends", "Imports", "LinkingTo").</p> <p>TRUE means to use c("Depends", "Imports", "LinkingTo", "Suggests") for pkgs and c("Depends", "Imports", "LinkingTo") for added dependencies: this installs all the packages needed to run pkgs, their examples, tests and vignettes (if the package author specified them correctly).</p> <p>In all of these, "LinkingTo" is omitted for binary packages.</p>
upgrade	<p>One of "default", "ask", "always", or "never". "default" respects the value of the R_REMOTES_UPGRADE environment variable if set, and falls back to "ask" if unset. "ask" prompts the user for which out of date packages to upgrade. For non-interactive sessions "ask" is equivalent to "always". TRUE and FALSE are also accepted and correspond to "always" and "never" respectively.</p>
force	<p>Force installation, even if the remote state has not changed since the previous install.</p>
quiet	<p>logical: if true, reduce the amount of output.</p>
build	<p>If TRUE build the package before installing.</p>
build_opts	<p>Options to pass to R CMD build, only used when build is TRUE.</p>
repos	<p>character vector, the base URL(s) of the repositories to use, e.g., the URL of a CRAN mirror such as "https://cloud.r-project.org". For more details on supported URL schemes see <a href="#">url</a>.</p> <p>Can be NULL to install from local files, directories or URLs: this will be inferred by extension from pkgs if of length one.</p>
type	<p>character, indicating the type of package to download and install. Will be "source" except on Windows and some macOS builds: see the section on 'Binary packages' for those.</p>
...	<p>Other arguments passed on to <code>utils::install.packages()</code>.</p>

**Author(s)**

Jeremy Stephens

**See Also**

Other package installation: [install\\_bioc](#), [install\\_bitbucket](#), [install\\_cran](#), [install\\_dev](#), [install\\_github](#), [install\\_gitlab](#), [install\\_git](#), [install\\_local](#), [install\\_svn](#), [install\\_url](#)

---

package\_deps

*Find all dependencies of a CRAN or dev package.*


---

### Description

Find all the dependencies of a package and determine whether they are ahead or behind CRAN. A `print()` method identifies mismatches (if any) between local and CRAN versions of each dependent package; an `update()` method installs outdated or missing packages from CRAN.

### Usage

```
package_deps(packages, dependencies = NA, repos = getOption("repos"),
             type = getOption("pkgType"))

local_package_deps(pkgdir = ".", dependencies = NA)

dev_package_deps(pkgdir = ".", dependencies = NA,
                 repos = getOption("repos"), type = getOption("pkgType"))

## S3 method for class 'package_deps'
update(object, dependencies = NA,
       upgrade = c("default", "ask", "always", "never"), force = FALSE,
       quiet = FALSE, build = TRUE, build_opts = c("--no-resave-data",
       "--no-manual", "--no-build-vignettes"), repos = getOption("repos"),
       type = getOption("pkgType"), ...)
```

### Arguments

<code>packages</code>	A character vector of package names.
<code>dependencies</code>	Which dependencies do you want to check? Can be a character vector (selecting from "Depends", "Imports", "LinkingTo", "Suggests", or "Enhances"), or a logical vector. TRUE is shorthand for "Depends", "Imports", "LinkingTo" and "Suggests". NA is shorthand for "Depends", "Imports" and "LinkingTo" and is the default. FALSE is shorthand for no dependencies (i.e. just check this package, not its dependencies).
<code>repos</code>	A character vector giving repositories to use.
<code>type</code>	Type of package to update.
<code>pkgdir</code>	path to a package directory, or to a package tarball.
<code>object</code>	A <code>package_deps</code> object.
<code>upgrade</code>	One of "default", "ask", "always", or "never". "default" respects the value of the <code>R_REMOTES_UPGRADE</code> environment variable if set, and falls back to "ask" if unset. "ask" prompts the user for which out of date packages to upgrade. For non-interactive sessions "ask" is equivalent to "always". TRUE and FALSE are also accepted and correspond to "always" and "never" respectively.

force	Force installation, even if the remote state has not changed since the previous install.
quiet	If TRUE, suppress output.
build	If TRUE build the package before installing.
build_opts	Options to pass to R CMD build, only used when build is TRUE.
...	Additional arguments passed to install_packages.

### Value

A data.frame with columns:

package	The dependent package's name,
installed	The currently installed version,
available	The version available on CRAN,
diff	An integer denoting whether the locally installed version of the package is newer (1), the same (0) or older (-1) t

### Examples

```
## Not run:
package_deps("devtools")
# Use update to update any out-of-date dependencies
update(package_deps("devtools"))

## End(Not run)
```

---

parse-git-repo	<i>Parse a remote git repo specification</i>
----------------	--

---

### Description

A remote repo can be specified in two ways:

**as a URL** `parse_github_url()` handles HTTPS and SSH remote URLs and various GitHub browser URLs

**via a shorthand** `parse_repo_spec()` handles this concise form: `[username/]repo[/subdir][#pull|@ref|@*release]`

### Usage

```
parse_repo_spec(repo)
```

```
parse_github_repo_spec(repo)
```

```
parse_github_url(repo)
```

### Arguments

repo                    Character scalar, the repo specification.

**Value**

List with members: username, repo, subdir ref, pull, release, some which will be empty.

**Examples**

```

parse_repo_spec("metacran/crandb")
parse_repo_spec("jimhester/covr#47")      ## pull request
parse_repo_spec("jeroen/curl@v0.9.3")    ## specific tag
parse_repo_spec("tidyverse/dplyr@*release") ## shorthand for latest release
parse_repo_spec("r-lib/remotes@550a3c7d3f9e1493a2ba") ## commit SHA
parse_repo_spec("r-lib/remotes@550a3c7d3f9e1493a2ba") ## commit SHA
parse_repo_spec("igraph=igraph/rigraph") ## Different package name from repo name

parse_github_url("https://github.com/jeroen/curl.git")
parse_github_url("git@github.com:metacran/crandb.git")
parse_github_url("https://github.com/jimhester/covr")
parse_github_url("https://github.example.com/user/repo.git")
parse_github_url("git@github.example.com:user/repo.git")

parse_github_url("https://github.com/r-lib/remotes/pull/108")
parse_github_url("https://github.com/r-lib/remotes/tree/name-of-branch")
parse_github_url("https://github.com/r-lib/remotes/commit/1234567")
parse_github_url("https://github.com/r-lib/remotes/releases/latest")
parse_github_url("https://github.com/r-lib/remotes/releases/tag/1.0.0")

```

---

update_packages	<i>Update packages that are missing or out-of-date.</i>
-----------------	---

---

**Description**

Works similarly to `utils::install.packages()` but doesn't install packages that are already installed, and also upgrades out dated dependencies.

**Usage**

```

update_packages(packages = TRUE, dependencies = NA,
  upgrade = c("default", "ask", "always", "never"), force = FALSE,
  quiet = FALSE, build = TRUE, build_opts = c("--no-resave-data",
  "--no-manual", "--no-build-vignettes"), repos = getOption("repos"),
  type = getOption("pkgType"), ...)

```

**Arguments**

packages	Character vector of packages to update.
dependencies	Which dependencies do you want to check? Can be a character vector (selecting from "Depends", "Imports", "LinkingTo", "Suggests", or "Enhances"), or a logical vector.

TRUE is shorthand for "Depends", "Imports", "LinkingTo" and "Suggests". NA is shorthand for "Depends", "Imports" and "LinkingTo" and is the default. FALSE is shorthand for no dependencies (i.e. just check this package, not its dependencies).

upgrade	One of "default", "ask", "always", or "never". "default" respects the value of the R_REMOTES_UPGRADE environment variable if set, and falls back to "ask" if unset. "ask" prompts the user for which out of date packages to upgrade. For non-interactive sessions "ask" is equivalent to "always". TRUE and FALSE are also accepted and correspond to "always" and "never" respectively.
force	Force installation, even if the remote state has not changed since the previous install.
quiet	If TRUE, suppress output.
build	If TRUE build the package before installing.
build_opts	Options to pass to R CMD build, only used when build is TRUE.
repos	A character vector giving repositories to use.
type	Type of package to update.
...	Other arguments passed on to <code>utils::install.packages()</code> .

**See Also**

[package\\_deps\(\)](#) to see which packages are out of date/ missing.

**Examples**

```
## Not run:  
update_packages("ggplot2")  
update_packages(c("plyr", "ggplot2"))  
  
## End(Not run)
```

# Index

`base::Startup`, 6

`dev_package_deps` (`package_deps`), 20

`download_version`, 2

`github_pull`, 3

`github_pull()`, 11, 12

`github_release` (`github_pull`), 3

`install_bioc`, 3, 6, 7, 9, 11, 12, 14, 15, 17–19

`install_bitbucket`, 4, 5, 7, 9, 11, 12, 14, 15, 17–19

`install_bitbucket()`, 9

`install_cran`, 4, 6, 7, 9, 11, 12, 14, 15, 17–19

`install_deps`, 8

`install_dev`, 4, 6, 7, 9, 11, 12, 14, 15, 17–19

`install_git`, 4, 6, 7, 9, 10, 12, 14, 15, 17–19

`install_github`, 4, 6, 7, 9, 11, 11, 14, 15, 17–19

`install_github()`, 3, 9

`install_gitlab`, 4, 6, 7, 9, 11, 12, 13, 15, 17–19

`install_gitlab()`, 9

`install_local`, 4, 6, 7, 9, 11, 12, 14, 14, 17–19

`install_svn`, 4, 6, 7, 9, 11, 12, 14, 15, 16, 18, 19

`install_url`, 4, 6, 7, 9, 11, 12, 14, 15, 17, 17, 19

`install_version`, 4, 6, 7, 9, 11, 12, 14, 15, 17, 18, 18

`local_package_deps` (`package_deps`), 20

`package_deps`, 20

`package_deps()`, 23

`parse-git-repo`, 21

`parse_github_repo_spec` (`parse-git-repo`), 21

`parse_github_url` (`parse-git-repo`), 21

`parse_repo_spec` (`parse-git-repo`), 21

`update.package_deps` (`package_deps`), 20

`update_packages`, 22

`url`, 2, 19

`utils::download.file()`, 6

`utils::install.packages()`, 2–4, 6–8, 11, 12, 14–16, 18, 19, 22, 23