Installing packages in R

```
install.packages(pkgs, lib, repos = getOption("repos"),
                 contriburl = contrib.url(repos, type),
                 method, available = NULL, destdir = NULL,
                 dependencies = NA, type = getOption("pkgType"),
                 configure.args = getOption("configure.args"),
                 configure.vars = getOption("configure.vars"),
                 clean = FALSE, Ncpus = getOption("Ncpus", 1L),
                 verbose = getOption("verbose"),
                 libs_only = FALSE, INSTALL_opts, quiet = FALSE,
                 keep_outputs = FALSE, ...)
```

install.packages {utils}

R Documentation

Install Packages from Repositories or Local Files

Description

Download and install packages from CRAN-like repositories or from local files.

Usage

```
install.packages(pkgs, lib, repos = getOption("repos"),
                 contriburl = contrib.url(repos, type),
                 method, available = NULL, destdir = NULL,
                 dependencies = NA, type = getOption("pkgType"),
                 configure.args = getOption("configure.args"),
                 configure.vars = getOption("configure.vars"),
                 clean = FALSE, Ncpus = getOption("Ncpus", 1L),
                 verbose = getOption("verbose"),
                 libs_only = FALSE, INSTALL_opts, quiet = FALSE,
                 keep_outputs = FALSE, ...)
```

Arguments

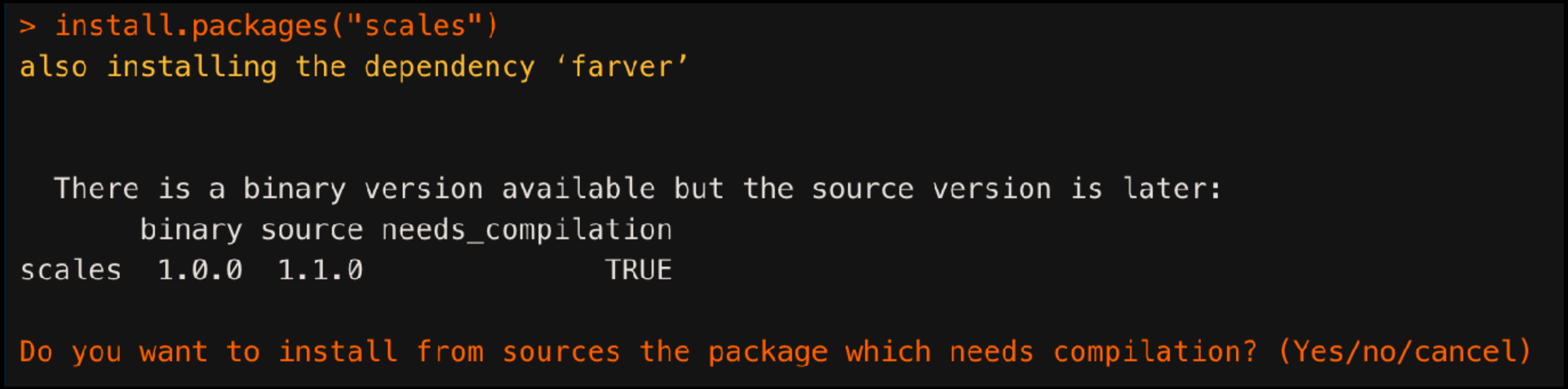
pkgs

character vector of the names of packages whose current versions should be downloaded from the repositories.

IfÂ repos = NULL, a character vector of file paths,

on windows,

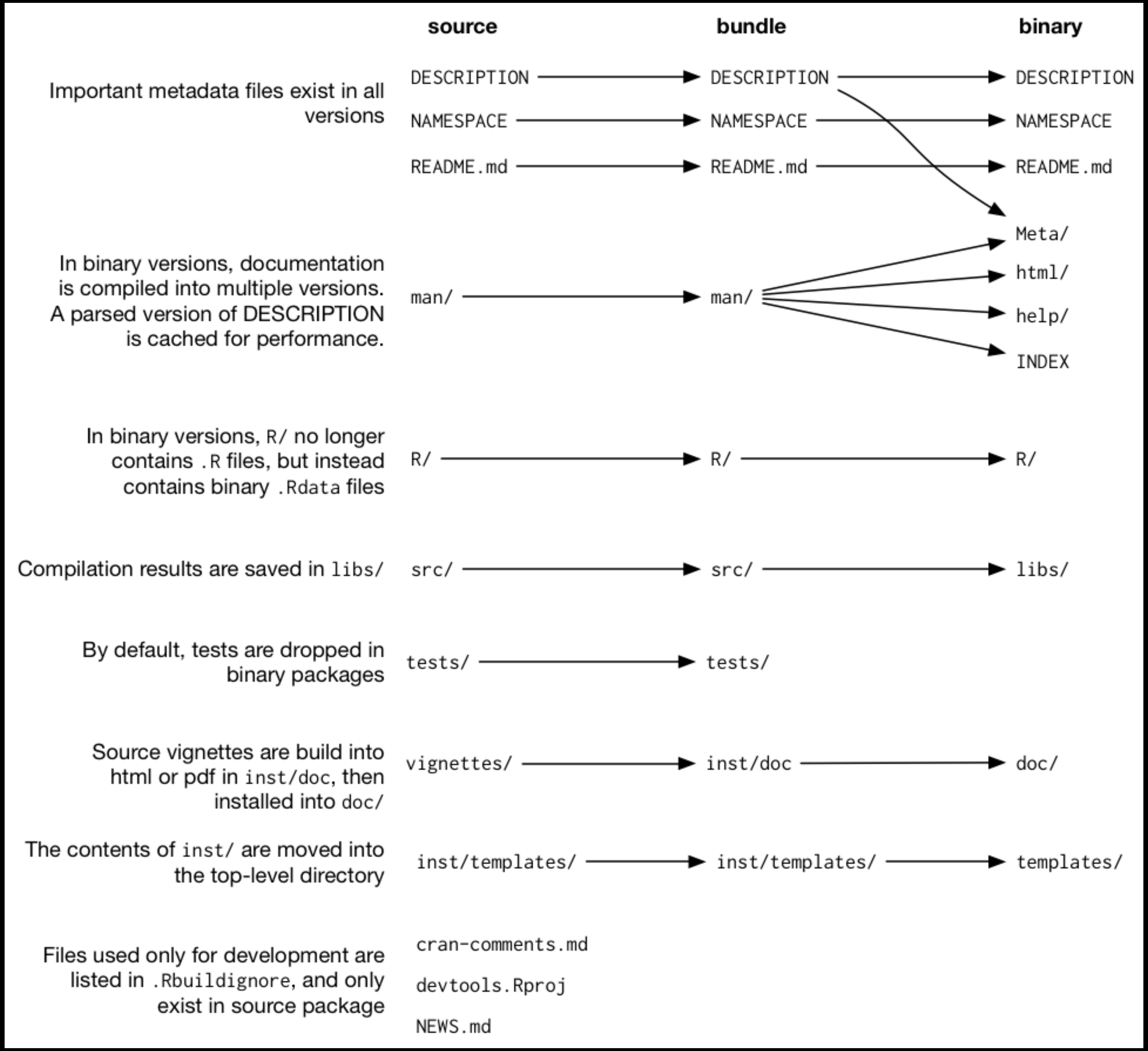
file paths of â€˜.zipâ€™ files containing binary builds of packages. (http://andÂ file://Â URLs are also accepted and the files will be



```
> install.packages("scales")
also installing the dependency ‘farver’


  There is a binary version available but the source version is later:
       binary source needs_compilation
scales  1.0.0  1.1.0              TRUE

Do you want to install from sources the package which needs compilation? (Yes/no/cancel)
```



	source	bundle	binary
Important metadata files exist in all versions	DESCRIPTION	DESCRIPTION	DESCRIPTION
	NAMESPACE	NAMESPACE	NAMESPACE
	README.md	README.md	README.md
In binary versions, documentation is compiled into multiple versions. A parsed version of DESCRIPTION is cached for performance.	man/	man/	Meta/ html/ help/ INDEX
In binary versions, `R/` no longer contains `.R` files, but instead contains binary `.Rdata` files	R/	R/	R/
Compilation results are saved in `libs/`	src/	src/	libs/
By default, tests are dropped in binary packages	tests/	tests/	
Source vignettes are build into html or pdf in `inst/doc`, then installed into `doc/`	vignettes/	inst/doc	doc/
The contents of `inst/` are moved into the top-level directory	inst/templates/	inst/templates/	templates/
Files used only for development are listed in `.Rbuildignore`, and only exist in source package	cran-comments.md devtools.Rproj NEWS.md		

Install binary versus source

```
install.packages('tidyverse')

install.packages('tidyverse', type = 'binary')

install.packages('tidyverse', type = 'source')
```

Default is getOption(“pkgType”), which usually (on macOS and Windows) will search for both binaries and source and offer to install the source version if it is a more recent version than what is available in binary, otherwise install the binary.

Install from a different repo

```
install.packages('tidyverse', repos = 'https://demo.rstudiopm.com/all/__linux__/bionic/latest')
```

Can be very useful for installing binary packages on Linux which RStudio is now compiling, unlike CRAN which does not compile Linux binaries. Binary packages will install much faster because they are pre-compiled.

remotes::install_github() / devtools::install_github()

- There are numerous remotes::install_* functions
- Can install packages directly from their development repository
- Installs as *source*, with all the potential problems mentioned previously
- Sometimes advantageous to get recent bug fixes etc. that have not been released on CRAN yet

Installing package to a non-standard library directory

```
install.packages("tidyverse", lib = "project_lib")
```

Better to use a package manager

- packrat
- renv