

# Package Development Day 1

Code ▾

Joyce Robbins

## What are packages?

- Collections of code

## Where are they stored?

- CRAN
- GitHub
- BioConductor
- Your hard drive

## How to install

- from CRAN:

Hide

```
install.packages("assertthat")
```

```
trying URL 'https://cran.rstudio.com/bin/macosx/contrib/4.1/assertthat_0.2.1.tgz'
Content type 'application/x-gzip' length 52388 bytes (51 KB)
=====
downloaded 51 KB
```

```
The downloaded binary packages are in
/var/folders/_h/qwz_0t7x3s7598nys0bgnj_c0000gn/T//RtmpUED6YH/downloaded_packages
```

Hide

```
install.packages("RLadiesnyc")
```

```
Warning in install.packages :
  package 'RLadiesnyc' is not available for this version of R

A version of this package for your version of R might be available elsewhere,
see the ideas at
https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages
```

- from GitHub:

Hide

```
devtools::install_github("jtr13/ggformat")
```

Skipping install of 'ggformat' from a github remote, the SHA1 (73d14902) has not changed since last install.

Use `force = TRUE` to force installation

Hide

```
devtools::install_github("jtr13/ggformat", force = TRUE)
```

Downloading GitHub repo jtr13/ggformat@HEAD

```
checking for file '/private/var/folders/_h/qwz_0t7x3s7598nys0bgnj_c0000gn/T/RtmpUED6YH/remotes5cf3058d848/jtr13-ggformat-73d1490/DESCRIPTION' ...
```

```
✓ checking for file '/private/var/folders/_h/qwz_0t7x3s7598nys0bgnj_c0000gn/T/RtmpUED6YH/remotes5cf3058d848/jtr13-ggformat-73d1490/DESCRIPTION'
```

```
- preparing 'ggformat':
```

```
checking DESCRIPTION meta-information ...
```

```
✓ checking DESCRIPTION meta-information
```

```
- checking for LF line-endings in source and make files and shell scripts
```

```
- checking for empty or unneeded directories  
Omitted 'LazyData' from DESCRIPTION
```

```
- building 'ggformat_0.1.0.tar.gz'
```

```
* installing *source* package 'ggformat' ...
```

```
** using staged installation
```

```
** R
```

```
** inst
```

```
** byte-compile and prepare package for lazy loading
```

```
** help
```

```
*** installing help indices
```

```
** building package indices
```

```
** testing if installed package can be loaded from temporary location
```

```
** testing if installed package can be loaded from final location
```

```
** testing if installed package keeps a record of temporary installation path
```

```
* DONE (ggformat)
```

Many packages are available both on CRAN and GitHub:

“dev version”

Hide

```
devtools::install_github("tidyverse/forcats")
```

Downloading GitHub repo tidyverse/forcats@HEAD

```
checking for file '/private/var/folders/_h/qwz_0t7x3s7598nys0bgnj_c0000gn/T/RtmpUED6YH/remotes5cf42df9294/tidyverse-forcats-b4dade0/DESCRIPTION' ...
```

```
✓ checking for file '/private/var/folders/_h/qwz_0t7x3s7598nys0bgnj_c0000gn/T/RtmpUED6YH/remotes5cf42df9294/tidyverse-forcats-b4dade0/DESCRIPTION'
```

```
- preparing 'forcats':
```

```
checking DESCRIPTION meta-information ...
```

```
✓ checking DESCRIPTION meta-information
```

```
- checking for LF line-endings in source and make files and shell scripts
```

```
- checking for empty or unneeded directories
```

```
- building 'forcats_0.5.1.9000.tar.gz'
```

```
* installing *source* package 'forcats' ...
```

```
** using staged installation
```

```
** R
```

```
** data
```

```
*** moving datasets to lazyload DB
```

```
** byte-compile and prepare package for lazy loading
```

```
** help
```

```
*** installing help indices
```

```
*** copying figures
```

```
** building package indices
```

```
** installing vignettes
```

```
** testing if installed package can be loaded from temporary location
```

```
** testing if installed package can be loaded from final location
```

```
** testing if installed package keeps a record of temporary installation path
```

```
* DONE (forcats)
```

- from your hard drive:

Hide

```
devtools::install("~/ggformat")
```

```

checking for file '/Users/joycerobbins/ggformat/DESCRIPTION' ...
✓ checking for file '/Users/joycerobbins/ggformat/DESCRIPTION'

- preparing 'ggformat':

✓ checking DESCRIPTION meta-information

- checking for LF line-endings in source and make files and shell scripts

- checking for empty or unneeded directories
Omitted 'LazyData' from DESCRIPTION

- building 'ggformat_0.0.0.9000.tar.gz'

Running /Library/Frameworks/R.framework/Resources/bin/R CMD INSTALL \
/var/folders/_h/qwz_0t7x3s7598nys0bgnj_c0000gn/T//RtmpUED6YH/ggformat_0.0.0.9000.tar.gz \
--install-tests
* installing to library '/Library/Frameworks/R.framework/Versions/4.1/Resources/library'
* installing *source* package 'ggformat' ...
** using staged installation
** R
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded from temporary location
** testing if installed package can be loaded from final location
** testing if installed package keeps a record of temporary installation path
* DONE (ggformat)

```

Regardless of the method, the packages are *installed* – see Packages →

## What is a “source” package?

Hide

```
fs::dir_tree("~/ggformat")
```

```
~/ggformat
├── DESCRIPTION
├── LICENSE
├── NAMESPACE
├── R
│   └── ggformat.R
├── Readme.Rmd
├── Readme.md
├── ggformat.Rproj
├── ggplot2template.png
├── inst
│   ├── rstudio
│   └── addins.dcf
├── man
│   └── FormatCode.Rd
└── orderwords.txt
```

[Hide](#)

```
list.files("~/ggformat")
```

```
[1] "DESCRIPTION"      "ggformat.Rproj"    "ggplot2template.png"
[4] "inst"             "LICENSE"           "man"
[7] "NAMESPACE"        "orderwords.txt"    "R"
[10] "Readme.md"        "Readme.Rmd"
```

## Why use projects?

So you never have to use `getwd()` or `setwd()` again. You're always just in the right place.

## Loading vs. installing

Usually we use `library()` to load packages into memory. While developing a package we use `devtools::load_all()`.

- It doesn't show up in the list of packages (or at least the version under development doesn't.)
- The functions are in memory and will disappear if we restart R.

## Creating a package

We need some essential files: `DESCRIPTION` and a folder called `R` with `.R` files.

The easier way to create the structure is with `create_package()`:

```
library(devtools)
create_package("~/covidtime")
```

[Hide](#)

```

✓ Creating '/Users/joycerobbins/covidtime/'
✓ Setting active project to '/Users/joycerobbins/covidtime'
✓ Creating 'R/'
✓ Writing 'DESCRIPTION'
Package: covidtime
Title: What the Package Does (One Line, Title Case)
Version: 0.0.0.9000
Authors@R (parsed):
  * First Last <first.last@example.com> [aut, cre] (YOUR-ORCID-ID)
Description: What the package does (one paragraph).
License: `use_mit_license()`, `use_gpl3_license()` or friends to
  pick a license
Encoding: UTF-8
Roxygen: list(markdown = TRUE)
RoxygenNote: 7.1.2
✓ Writing 'NAMESPACE'
✓ Writing 'covidtime.Rproj'
✓ Adding '^covidtime\\.Rproj$' to '.Rbuildignore'
✓ Adding '.Rproj.user' to '.gitignore'
✓ Adding '^\\.Rproj\\.user$' to '.Rbuildignore'
✓ Opening '/Users/joycerobbins/covidtime/' in new RStudio session
✓ Setting active project to '<no active project>'

```

## YOUR TURN

See Day1\_lab.md (Day1\_lab.md)

## Using the package in another project / script

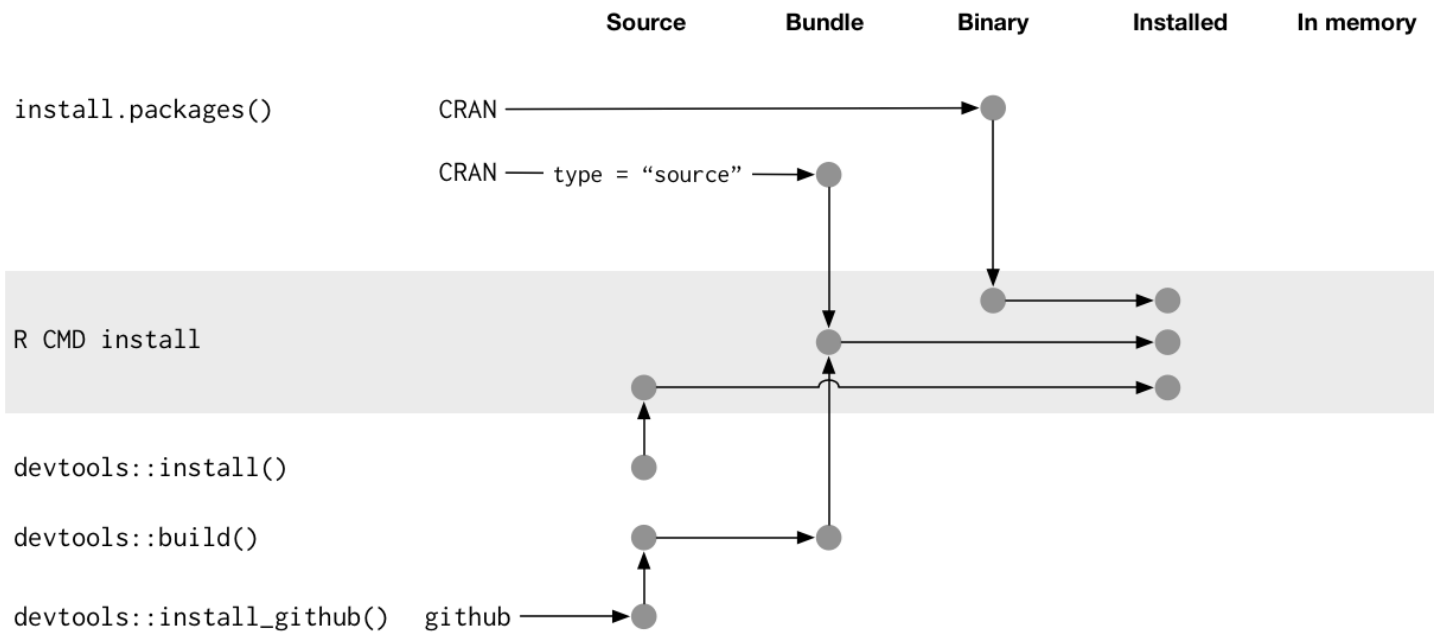
Before we can use the package elsewhere we need to:

- add function documentation by clicking “Code... Insert Roxygen Skeleton” (or at a minimum include `@export` )
- edit DESCRIPTION
- document with `devtools::document()`
- install the package with `devtools::install()`

## YOUR TURN

See Day1\_lab.md (Day1\_lab.md)

## Package states



Source: <https://r-pkgs.org/package-structure-state.html#installed-package> (<https://r-pkgs.org/package-structure-state.html#installed-package>)

Key points:

- `devtools::install()` : source -> installed
- `install.packages()` : binary -> installed
- Developers submit *bundles* to CRAN, CRAN builds and distributes *binaries*

## Closer look at package files

DEMO