

Develop Packages in R

Part 1: Make code into R package

April 2019

Isabella Gollini

Isabella.Gollini@ucd.ie

[@IsabellaGollini](#)

Bruna Wundervald [@bwundervald](#)

Chiara Cotroneo [@selenocysteina](#)

Jo Nieć [@joannaniec](#)

[@RLadiesDublin](#)



<https://rladies.org/>

<https://forwards.github.io>

<https://github.com/forwards/workshops/>

Preliminaries

The material is mostly drawn from



<http://r-pkgs.had.co.nz>

<https://amzn.com/1491910399>

The bad news:
It's going to be
frustrating





The good news:
Frustration is
typical and
temporary

Why make a package?

- You want to **test**
- You want to **generalise**
- You want to **document**
- You want to **share**

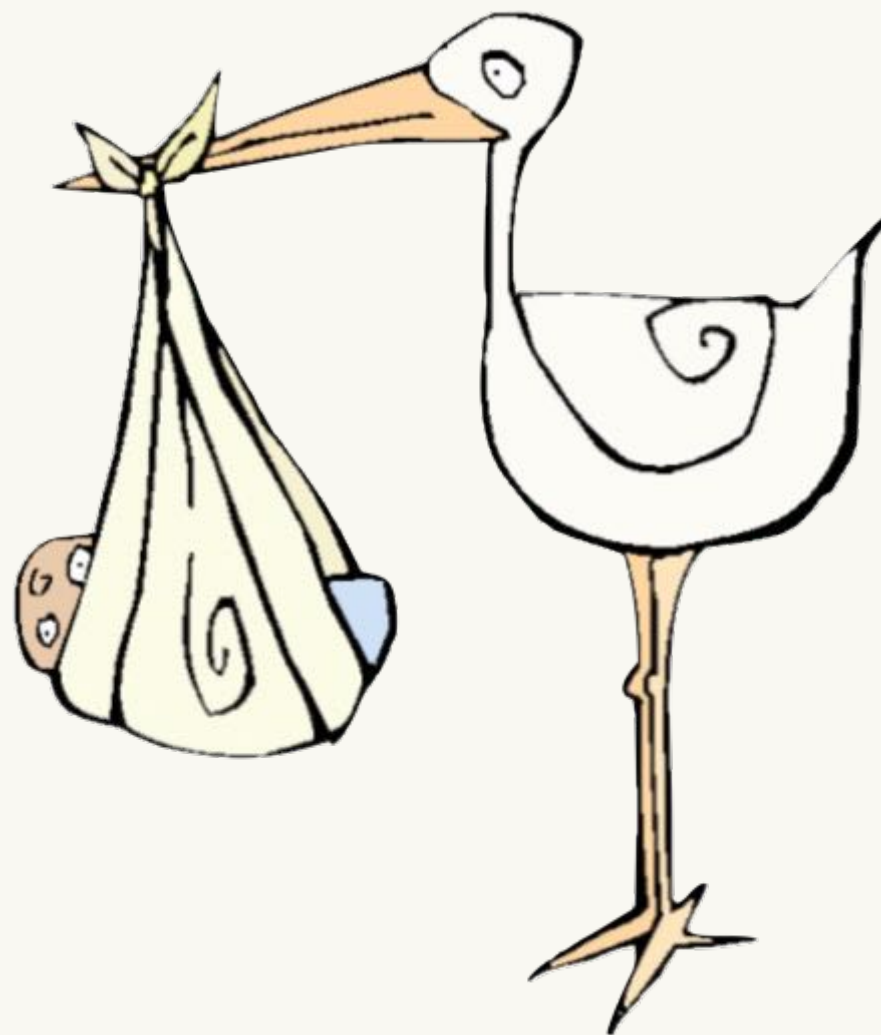
Get to know your R installation

```
R.home ( )
```

```
list.files (R.home ( ) )
```

```
R.version
```


Where do R packages come from?



CRAN and GitHub, mostly

```
install.packages("foo")
```

```
library(devtools)
```

```
install_github("jane/foo")
```

Where do R packages live on your computer?



R packages live in a library



Get to know your R
library(ies?)

`.Library`

The default library



`.libPaths()`

All the libraries

R knows about



`.Library == .libPaths()`

For many users, these are same

Other users maintain multiple libraries

E.g., You can put add-on packages in a user-level library:

`/Users/isabella/resources/R/library`

Make sure you have recent versions of
these packages

```
old.packages()  
packageVersion()
```

```
install.packages("devtools")  
install.packages("pkgdown")  
install.packages("roxygen2")  
install.packages("testthat")  
install.packages("tidyverse")  
install.packages("usethis")
```


My first package

Your turn

Create a package on your Desktop:

```
usethis::create_package("~/Desktop/mypackage")
```

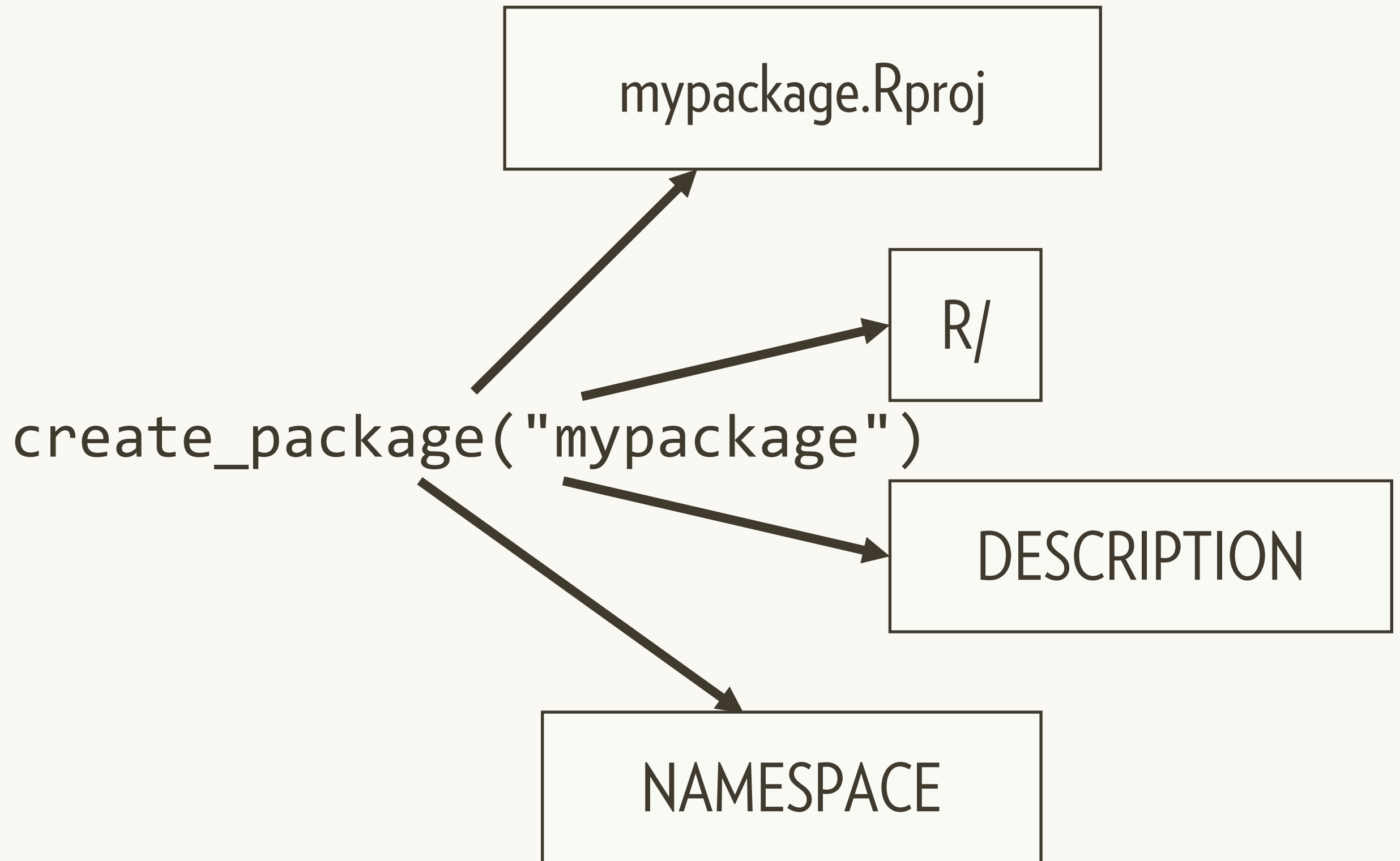
Or on RStudio.cloud (create within main project)

```
usethis::create_package("mypackage")
```

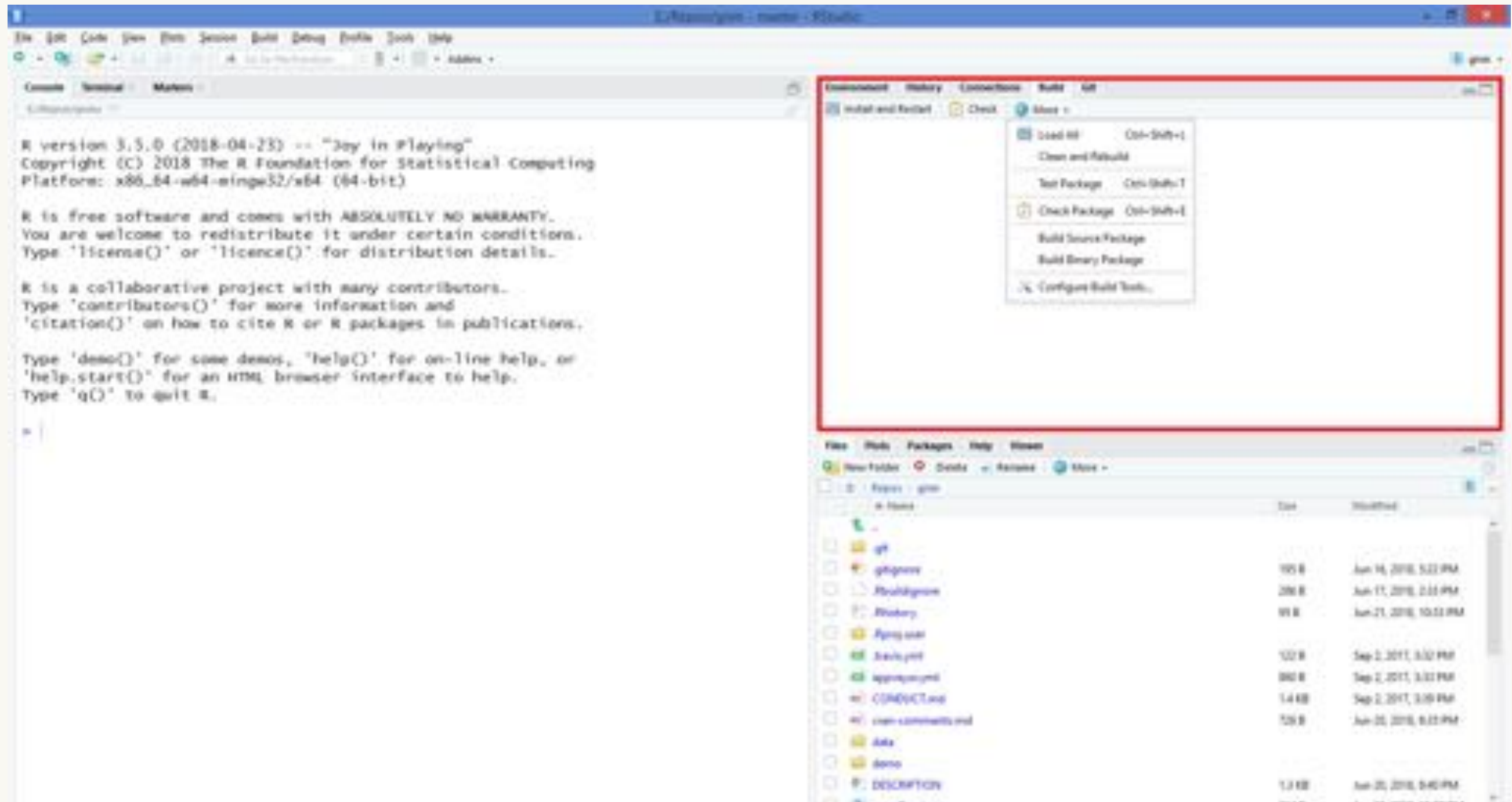
This will create a new R project and start a new session.

What other files and directories are created?

What happens we run `create_package()`?



RStudio helpers: build (& git) menus



DESCRIPTION

Package: mypackage

Version: 0.0.0.9000

Title: What the Package Does (One Line, Title Case)

Description: What the package does (one paragraph).

Authors@R: person("First", "Last", ,
"first.last@example.com", c("aut", "cre"))

License: What license it uses

Encoding: UTF-8

LazyData: true

ByteCompile: true

Development workflow

How is developing a
package
same / different
from developing a script?

How same?

Iterate early and often!

Change it, try it, change it, try it, ***ad nauseum***

How different?

Write functions, not “top-level” code.

Dependencies are different,

no `library()` calls.

Install & Restart (or simulate that),

don't `source()`.

You write this

... but you use this

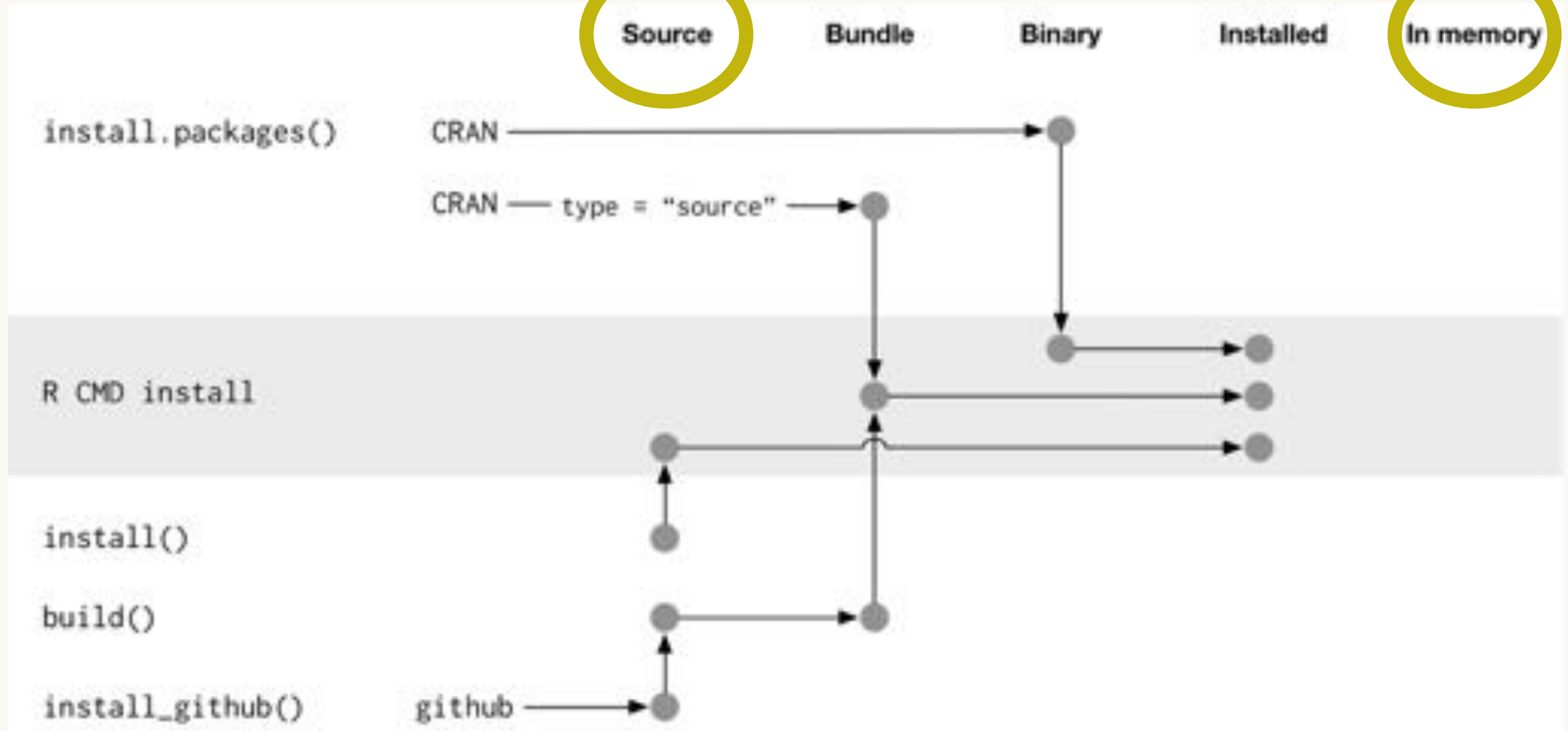


Figure from Hadley Wickham's book, R packages

<http://r-pkgs.had.co.nz>

<https://github.com/hadley/r-pkgs/blob/master/diagrams/installation.png>

How do packages get into memory?

Development: every couple hours or days?

Development: Every 2 minutes

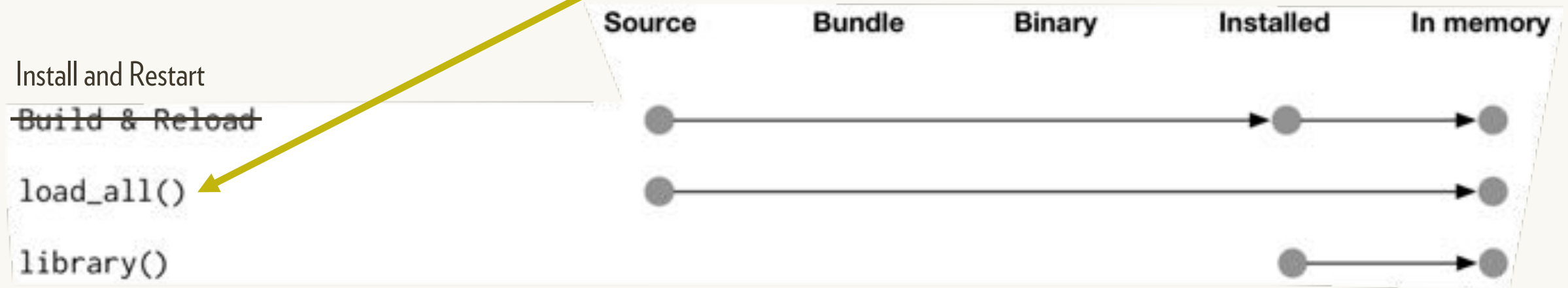
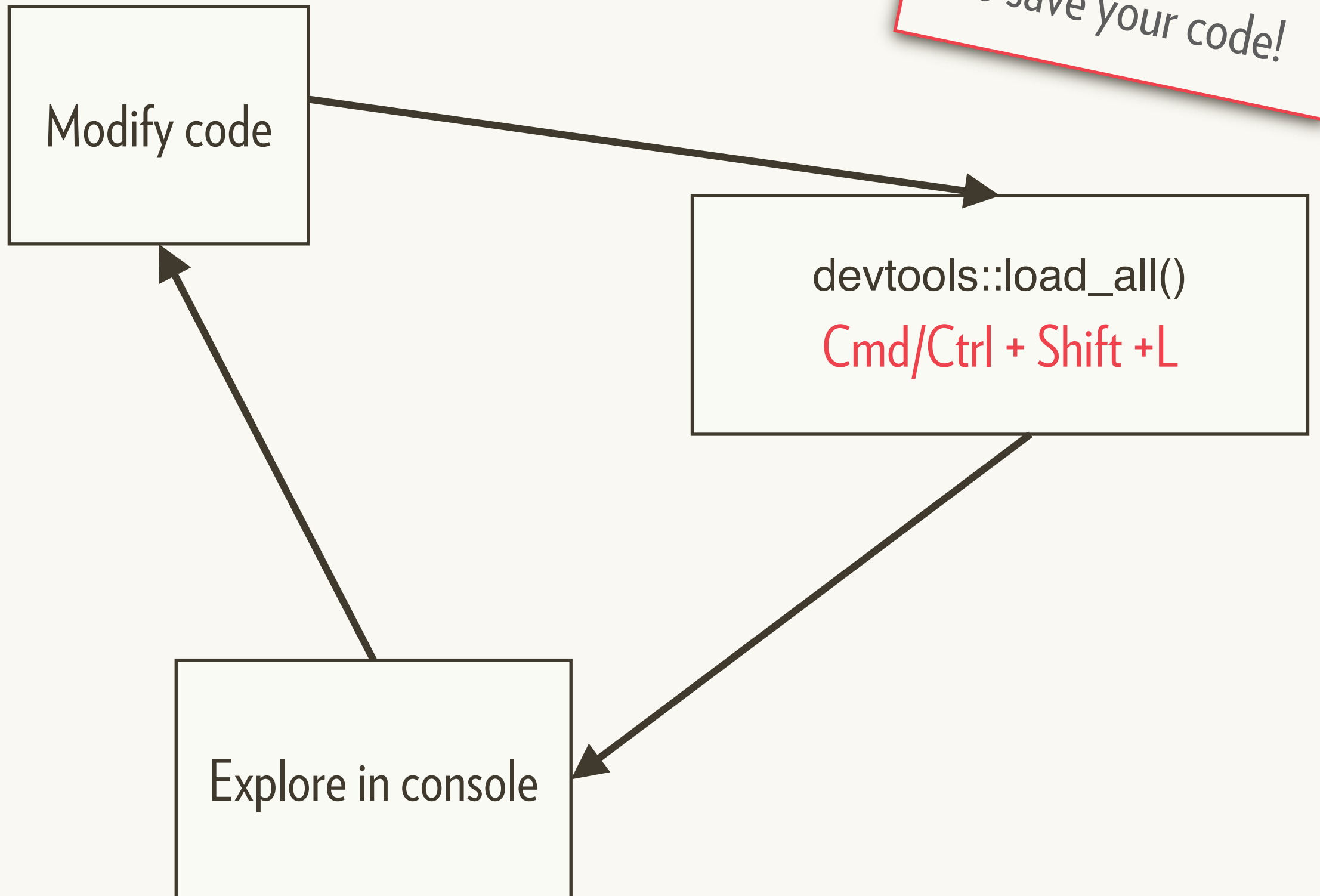


Figure from Hadley Wickham's book, R packages

<http://r-pkgs.had.co.nz>

<https://github.com/hadley/r-pkgs/blob/master/diagrams/loading.png>

Basic workflow



Your turn

[Uses <https://github.com/forwards/mylittlepony>]

- Open `mylittlepony.Rproj`.
- Load all the functions using,
`devtools::load_all()` then run
`rpony(10)`

Uhoh! I have forgotten to include `Fluttershy` in the list of ponies.

- Add her, reload the code, and verify that your change worked.

This work is licensed under the
Creative Commons Attribution-Noncommercial 3.0
United States License.

To view a copy of this license, visit
<http://creativecommons.org/licenses/by-nc/3.0/us/>