# Package Development

Angela Li @civicangela
Samantha Toet @Samantha_Toet

**Workshop materials:** bit.ly/cville_pkg

# Get to know your R installation

Where is R installed on your computer?

**R.home()**

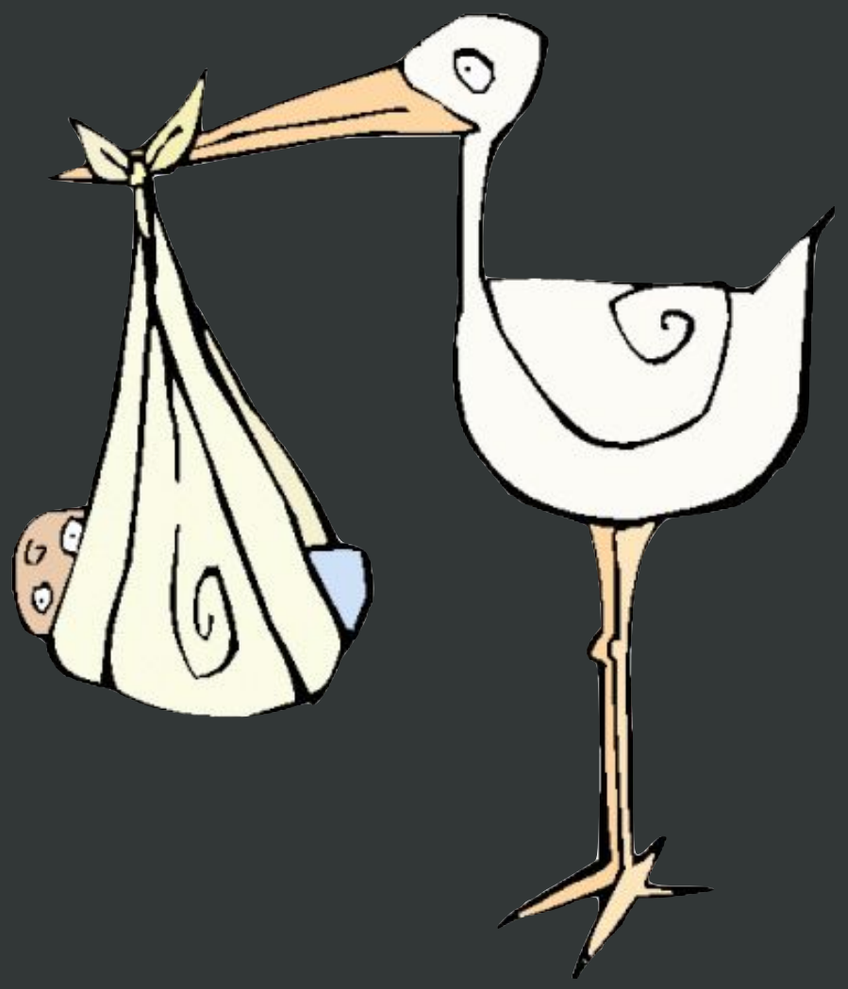
What files are in your R installation?

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

Which version of R are you using?

```
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 (power) useRs maintain multiple libraries

E.g., Jenny Bryan puts add-on packages in a user-level library:
/Users/jenny/resources/R/library

```
installed.packages()
```

How many packages do you have installed?

How do the packages break down by Priority (and what does that mean, anyway)?

# Startup files

`.Rprofile`

Code that runs at startup
Load workflow packages
Set options
Use in moderation!

`.Renviron`

Set lib paths
Set env vars

# Make sure you have recent versions of these packages

```
packageVersion()
```

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

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

# How are they the same?

Iterate early and often!

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

# How is developing a package different from writing a script?

Write functions, not “top-level” code.

Dependencies are different,

no library() calls.

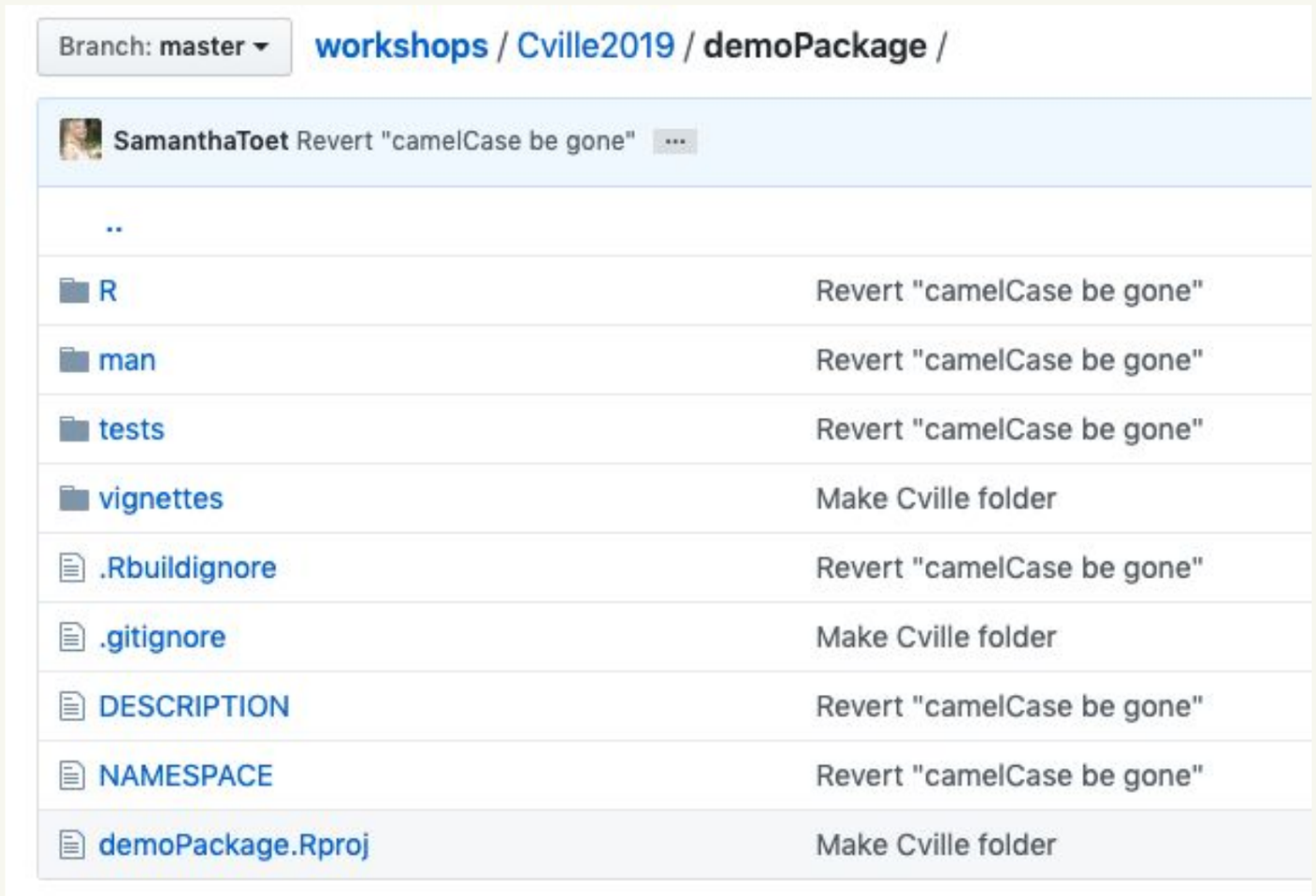
Install & Restart (or simulate that),

don’t source().

# Your turn

Look at our demo package on github here:

1. Where's the R code? What is it mostly comprised of?
2. What do you think is in the DESCRIPTION file? How about the NAMESPACE file?
3. MAN-what?



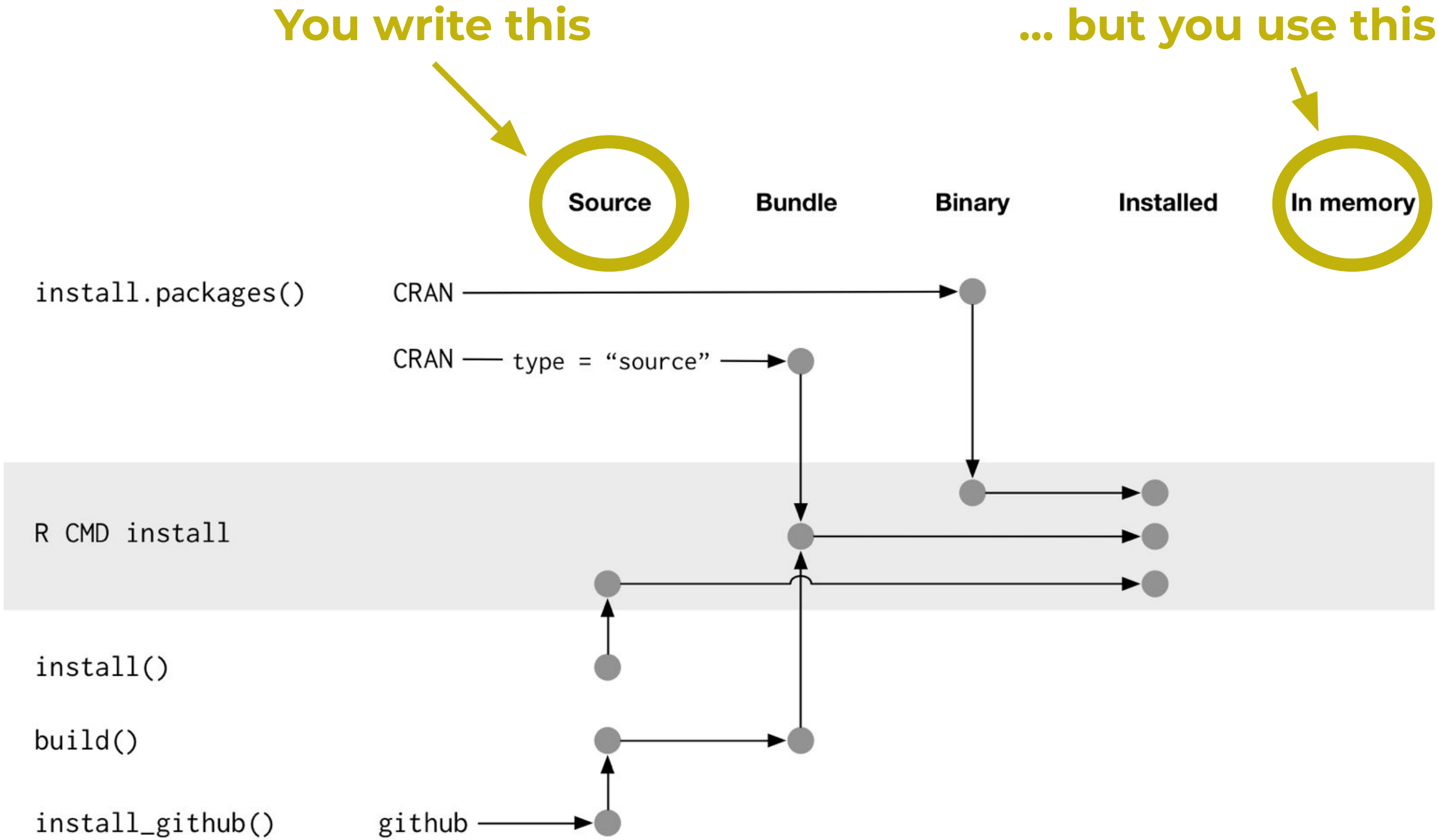



**Figure from Hadley Wickham’s book, R packages**

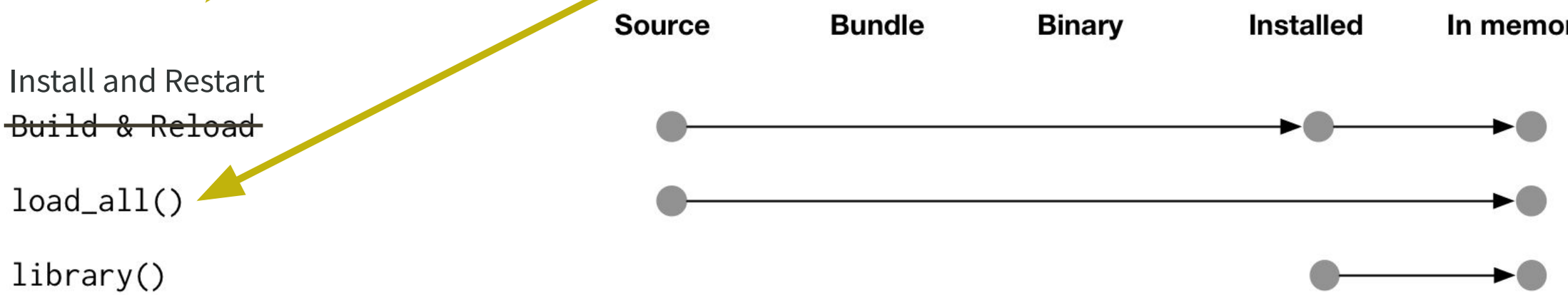
**http://r-pkgs.had.co.nz**

**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**




**Normal day-to-day usage**

**Figure from Hadley Wickham's book, R packages**

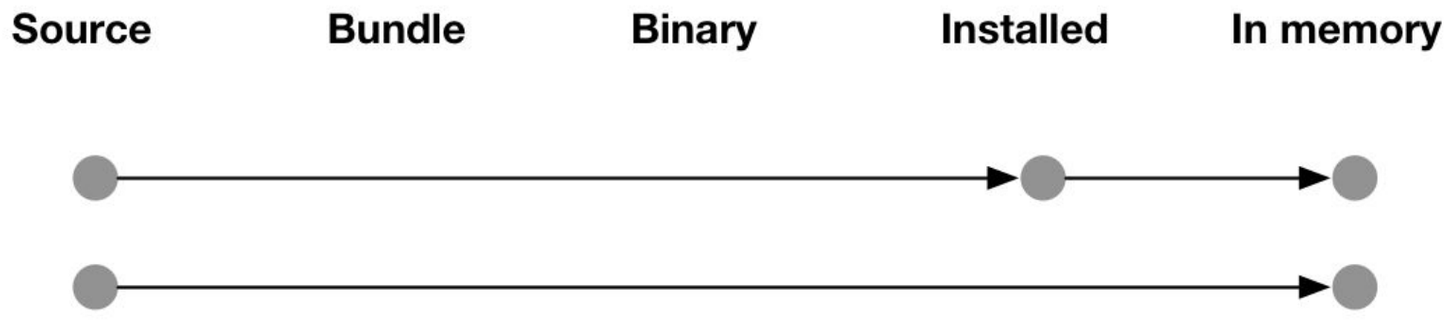
**http://r-pkgs.had.co.nz**

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

Install and Restart

~~Build & Reload~~

`load_all()`

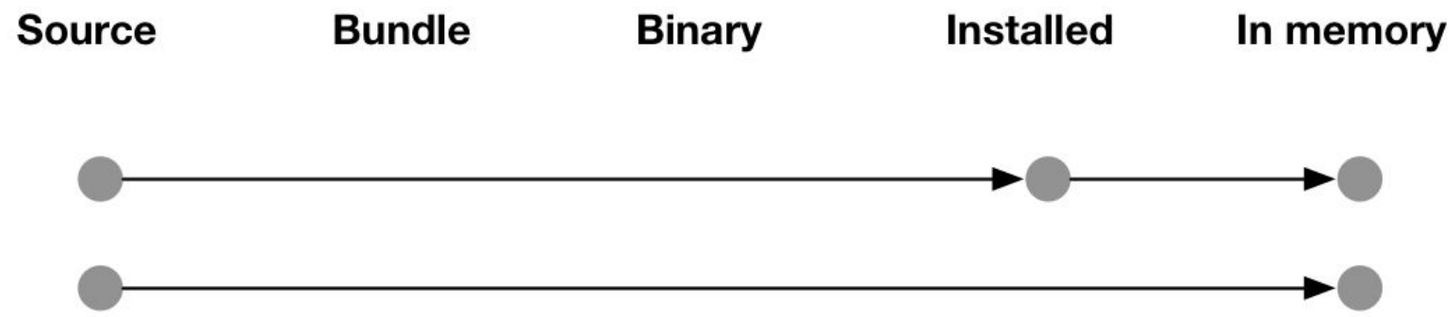



`devtools::load_all()` is to package development

as

interactive “stepping through” code is to script development

Install and Restart
~~Build & Reload~~
`load_all()`




RStudio's Install & Restart is to package development as `source()` or RStudio's "Source" or `Rscript foo.R` is to script development

Use devtools::load_all( ) a lot!

(And “Install and Restart” sparingly)

Let’s do this!