Packages & R code

December 2017

Hadley Wickham,
Jenny Bryan,
Di Cook

Motivation

“Workflow: you should have one”

— *Jenny Bryan*

A package is a set of conventions that (with the right tools) makes your life easier

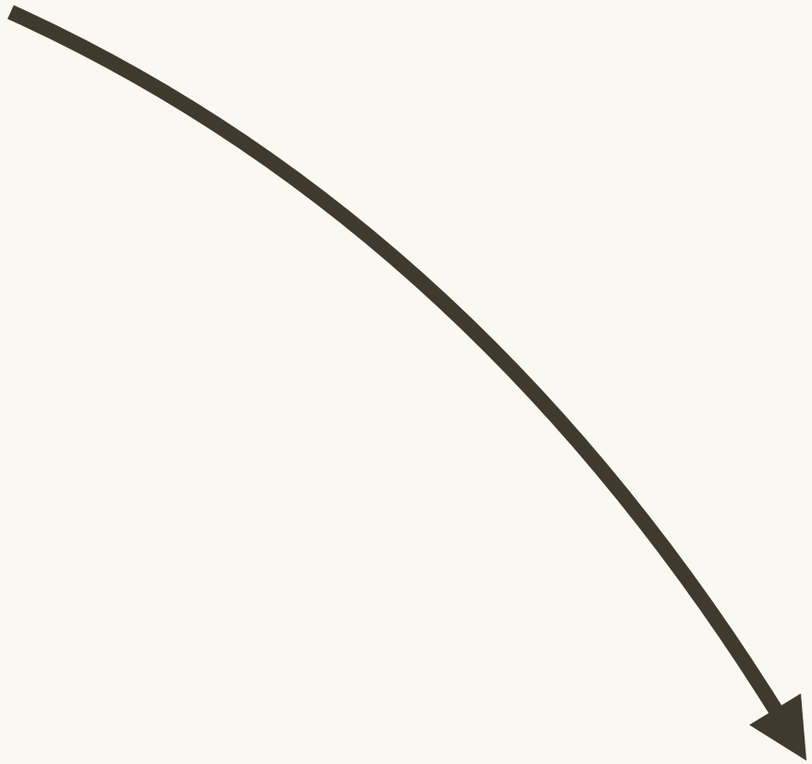
“Seriously, it doesn’t have to be about sharing your code (although that is an added benefit!). It is about saving yourself time.”

— *Hilary Parker*

Script

One off data analysis

Primarily side-effects



Package

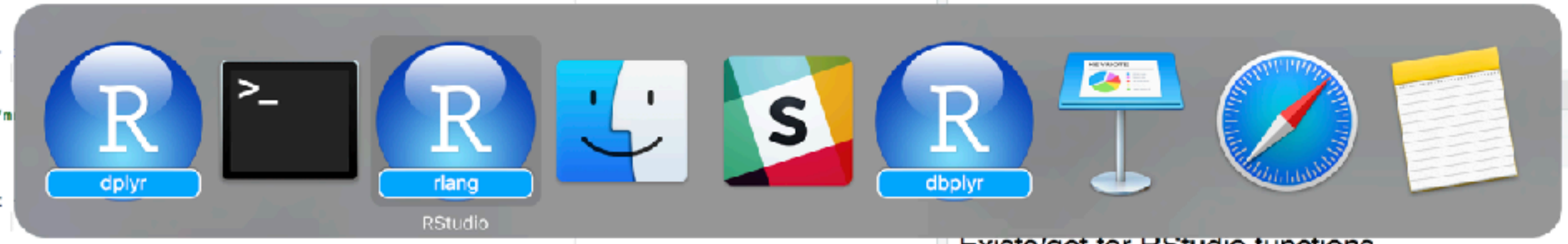
Defines reusable components

No side-effects

RStudio projects

Why use RStudio projects?

3 reasons



Work on multiple projects simultaneously and independently

Manage working directories

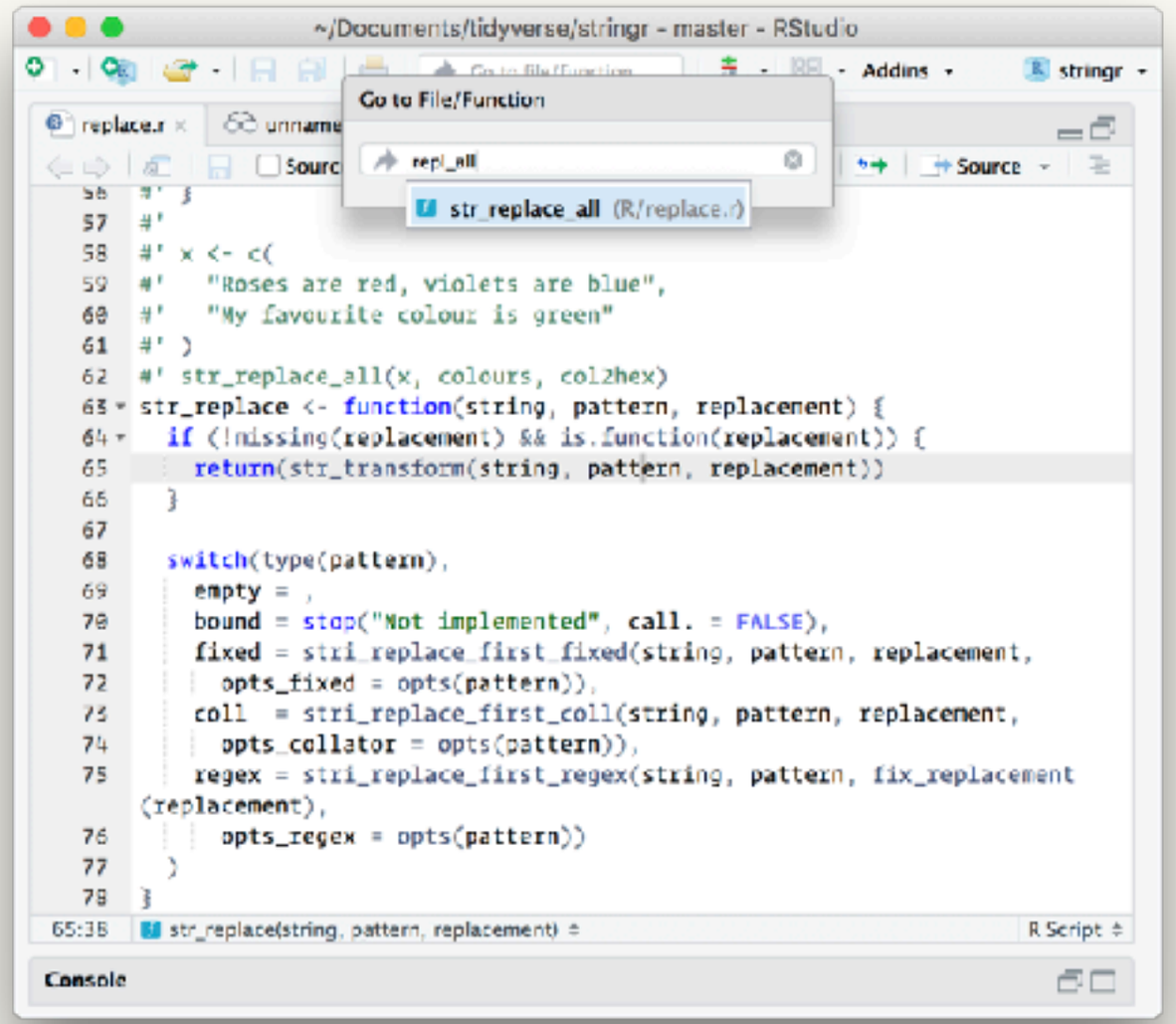
If the first line of your #rstats script is

```
setwd("C:\Users\jenny\path\that\only\I\have")
```

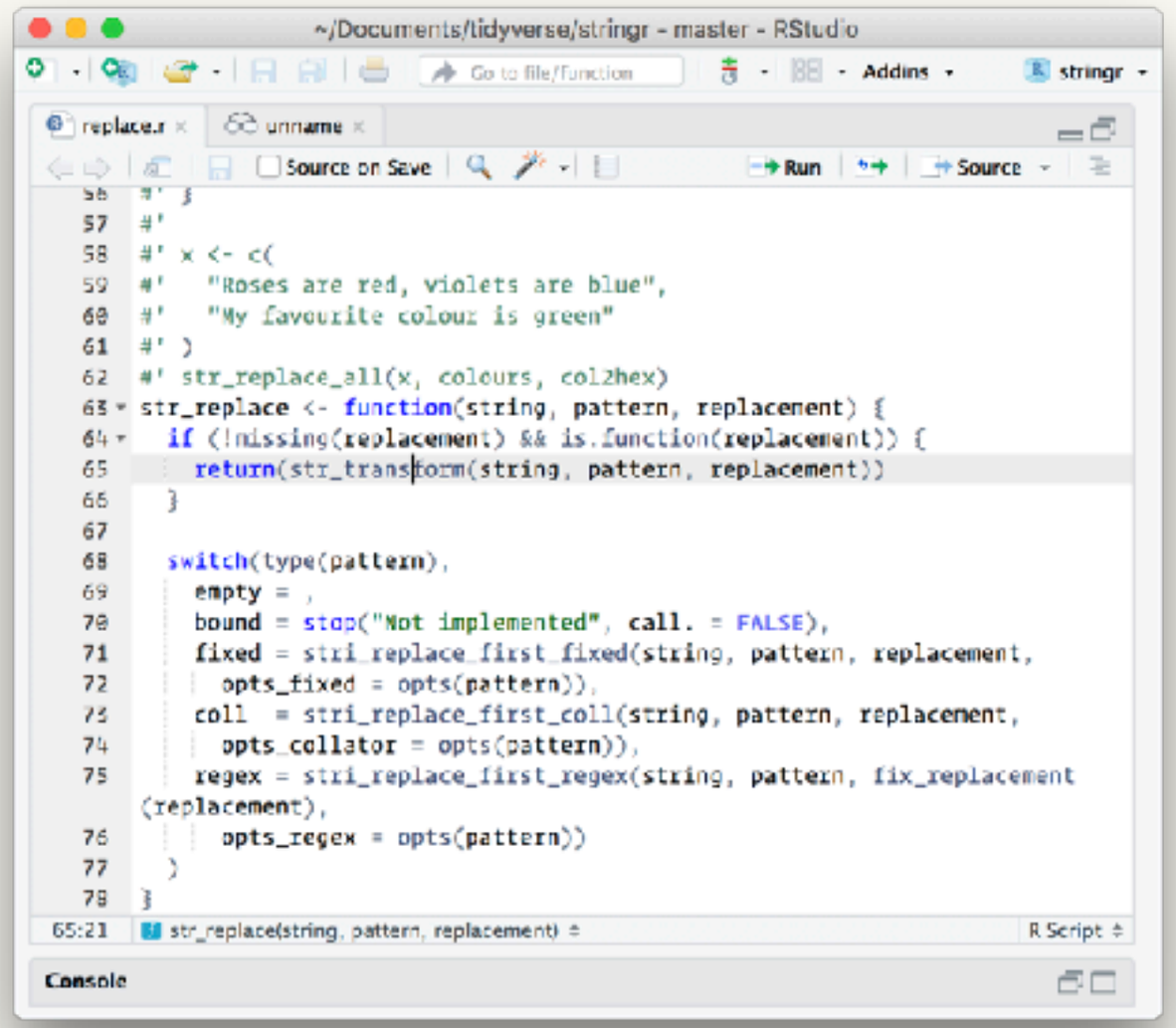
I will come into your lab and SET YOUR COMPUTER ON FIRE 🔥.

— Mash-up of rage tweets by @jennybc and @tpoi.

Enhanced navigation



Ctrl + . = find functions/files



F2 = jump to definition

My first package

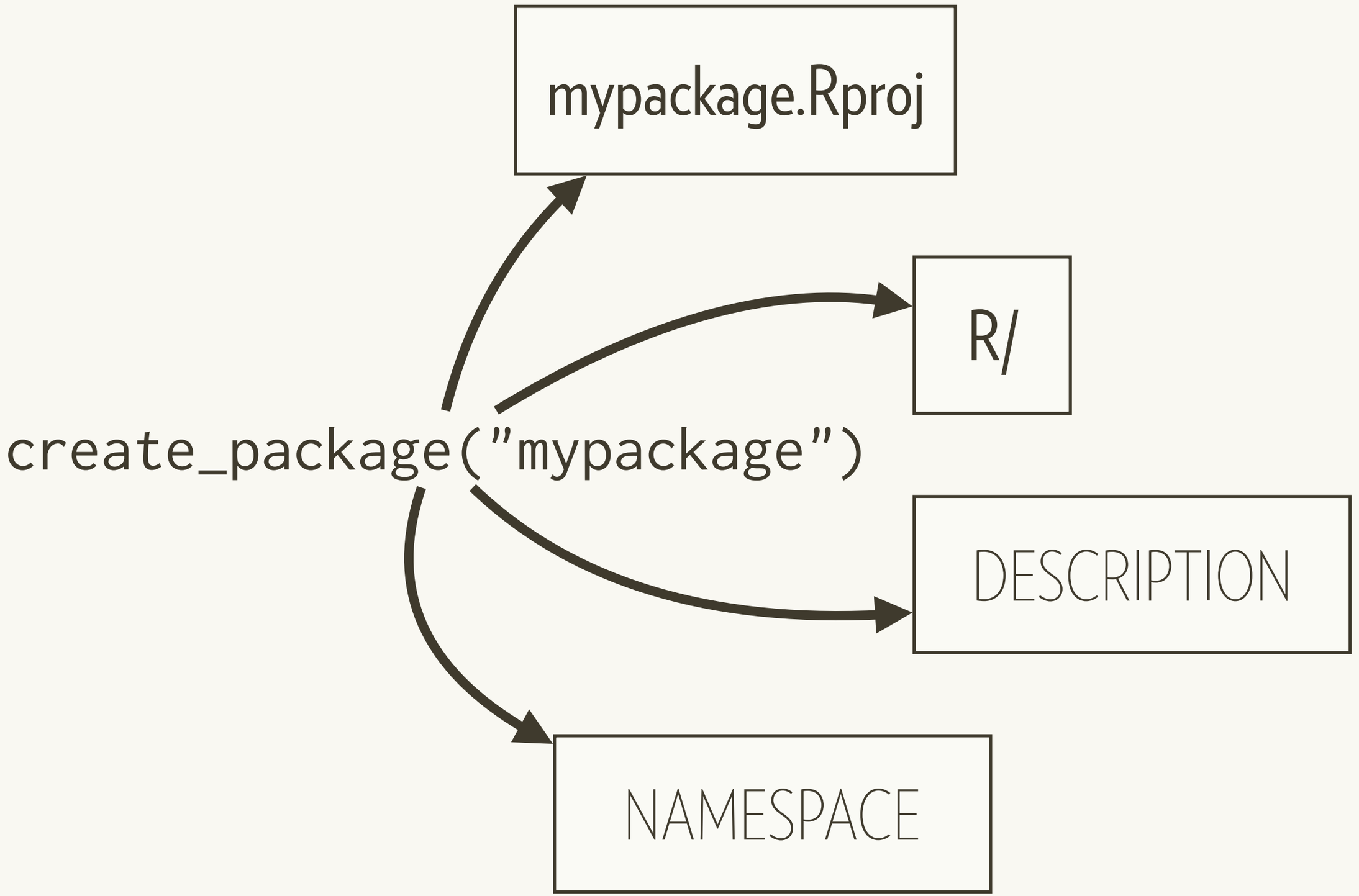
Your turn

```
# Verify that you can create a package with:
usethis::create_package("~/Desktop/mypackage")

# What other files and directories are created?

# You can also create new project using RStudio
# but it has some slight differences that will
# cause hassles today (but not in general)
```

What happens we run create_package()?

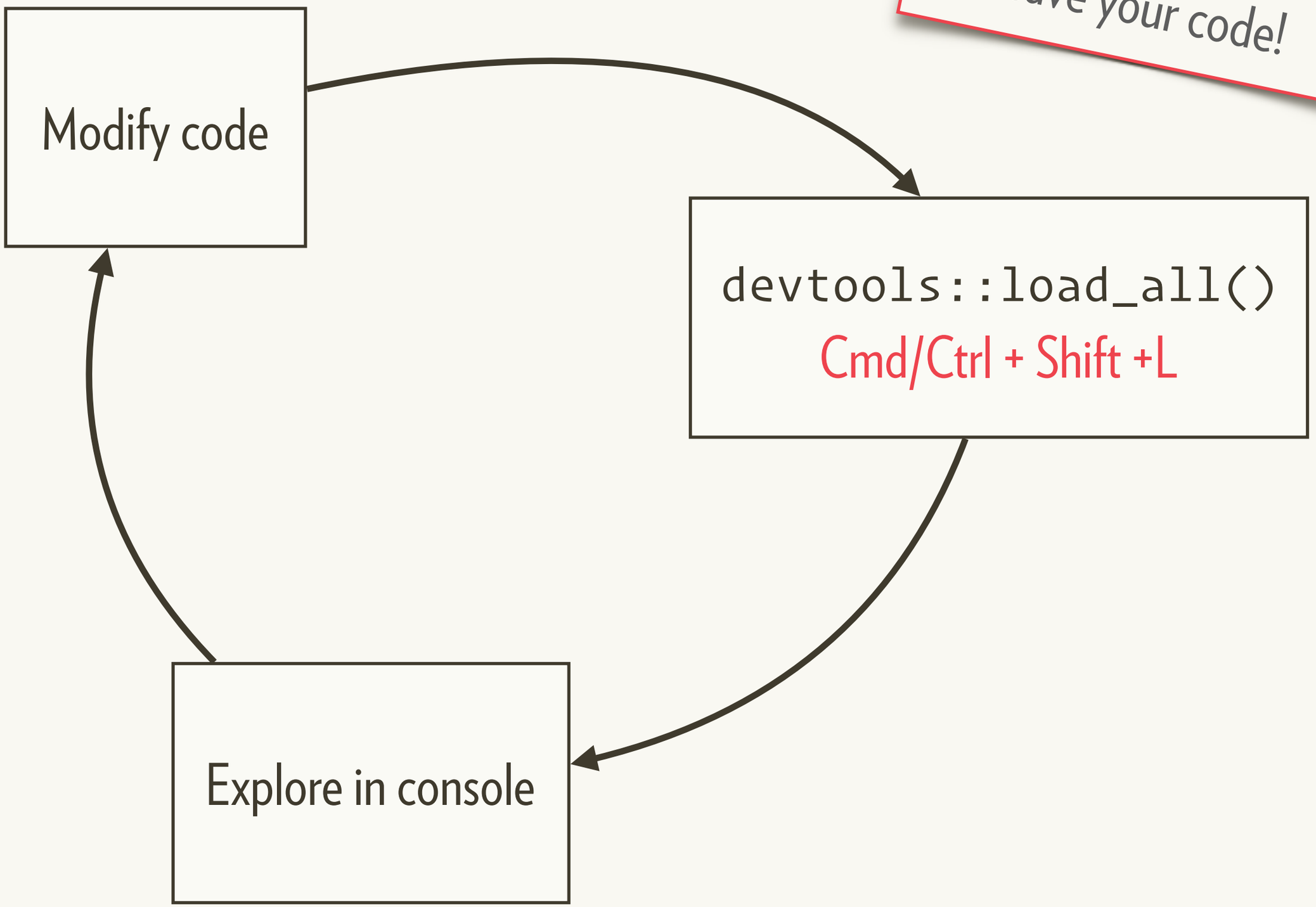


```
package.skeleton()
```

Never use this!

Why bother?

You don't even need to save your code!



Your turn

Open mylittlepackage.Rproj.

Jump to rpony() using only the keyboard.

Load all the functions, 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.

What if you need to create a new file?

```
# There's a usethis helper for that too!
usethis::use_r("file-name")

# Organise files so that related code
# lives together. If you can give a file
# a concise and informative name, it's
# probably about right
```

Workflow setup: your .Rprofile

```
# Setup some code that is run every time
# you start R
# usethis::edit_r_profile()

if (interactive()) {
  suppressMessages(require(devtools))
  suppressMessages(require(usethis))
  suppressMessages(require(testthat))
}
```

Never include analysis packages here

```
if (interactive()) {
  suppressMessages(require(ggplot2))
  suppressMessages(require(dplyr))
}
```

While you're in there, also add

```
options(
  warnPartialMatchArgs = TRUE
  warnPartialMatchDollar = TRUE,
  warnPartialMatchAttr = TRUE
)
```

Options

General

Code

Appearance

Pane Layout

Packages

Sweave

Spelling

Git/SVN

Publishing

Default working directory (when not in a project):

~ Browse...

☑ Restore most recently opened project at startup

☑ Restore previously open source documents at startup

☐ Restore .RData into workspace at startup

Save workspace to .RData on exit: Never

☑ Always save history (even when not saving .RData)

☑ Remove duplicate entries in history

☐ Use debug error handler only when my code contains errors

☐ Automatically expand tracebacks in error inspector

Default text encoding:

UTF-8 Change...

☑ Automatically notify me of updates to RStudio

Combine with **Ctrl/cmd+ shift + F10**

OK Cancel Apply

Your turn

Follow the instructions in previous slides and make sure that you're optimally configured.

Determine the rules for package names via experimentation.

Git + GitHub

Use both!

The 😡💩😱 will pay off

happygitwithr.com

Excuse me, do you have a moment to talk about version control?

https://doi.org/10.7287/peerj.preprints.3159v2