

Packages and R code

eRum 2018

Forwards Teaching Team



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

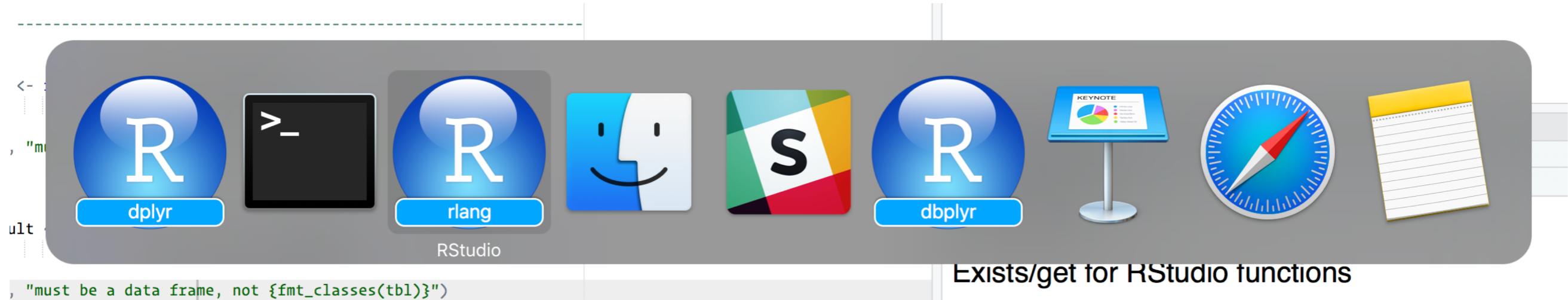
Defines reusable components

No side-effects

RStudio projects

Why use RStudio projects?

3 reasons



Work on multiple projects simultaneously and independently

Manage working directories

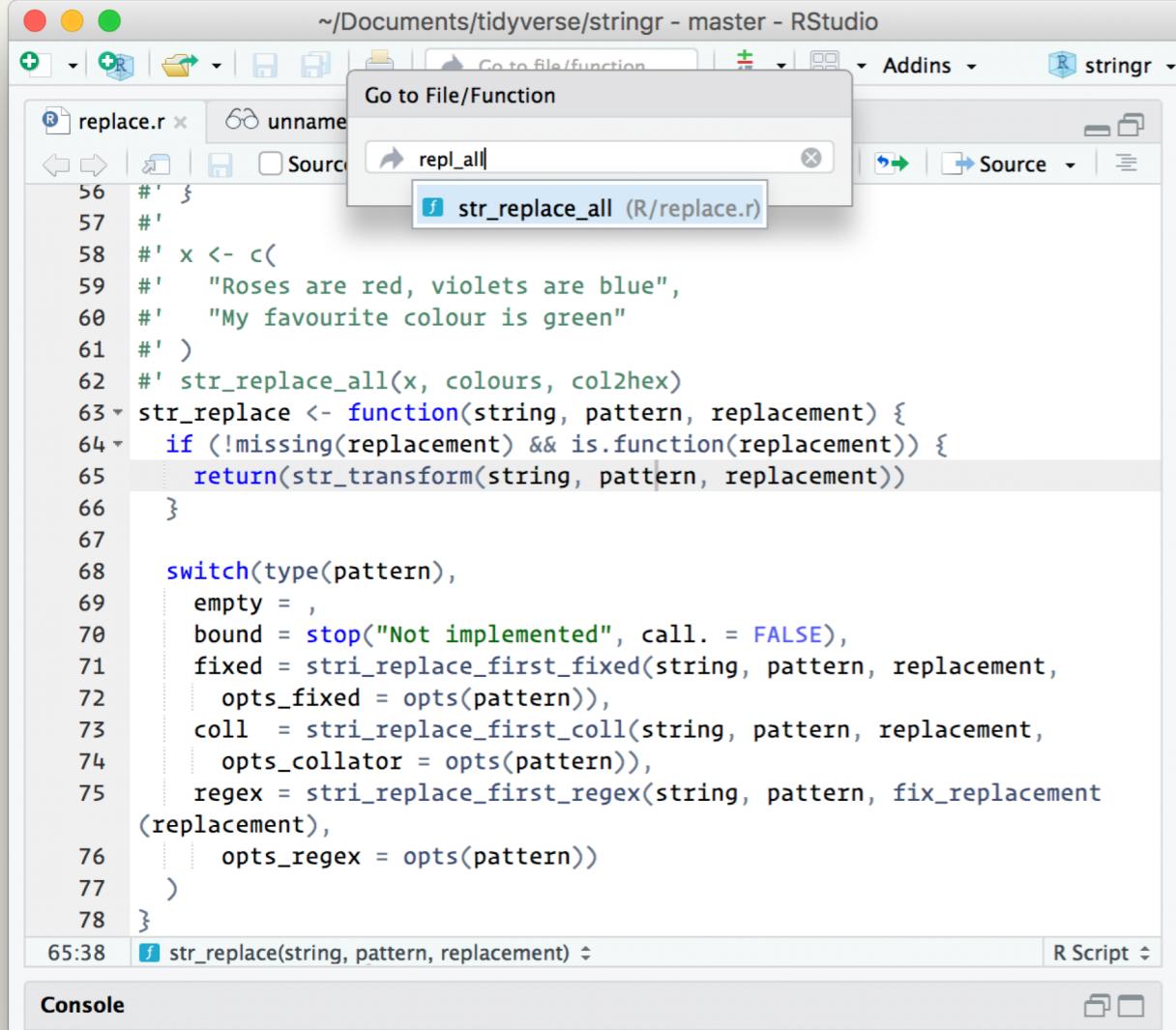
If the first line of your #rstats script is

```
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

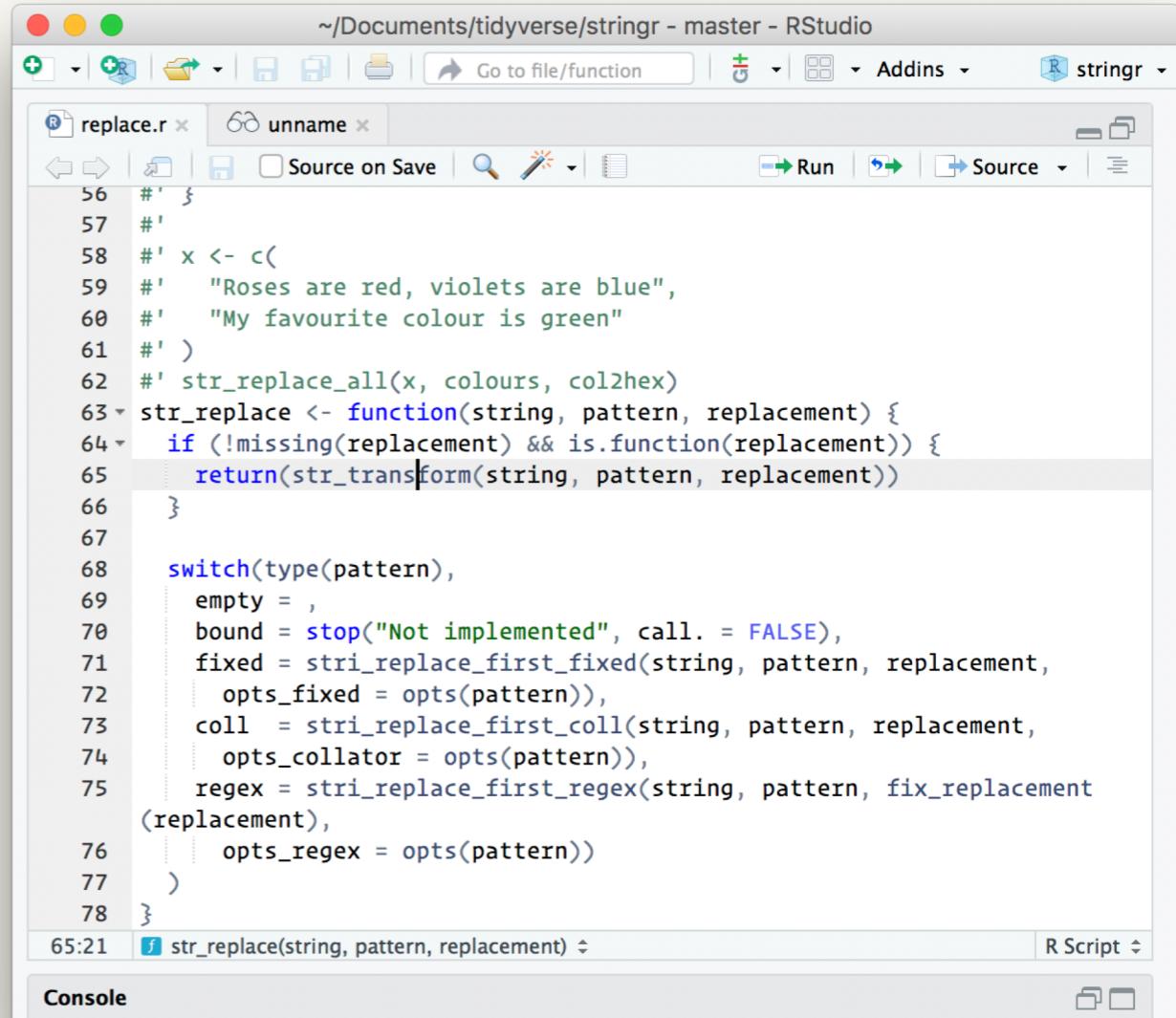


A screenshot of the RStudio interface. The title bar says "~/Documents/tidyverse/stringr - master - RStudio". The main area shows an R script named "replace.r". A modal dialog titled "Go to File/Function" is open, with the search input field containing "repl_all". Below the input field, a list shows "str_replace_all (R/replace.r)". The script code includes several lines of R code related to string manipulation.

```
56 #' 
57 #' 
58 #' x <- c(
59 #'   "Roses are red, violets are blue",
60 #'   "My favourite colour is green"
61 #' )
62 #' str_replace_all(x, colours, col2hex)
63 str_replace <- function(string, pattern, replacement) {
64   if (!missing(replacement) && is.function(replacement)) {
65     return(str_transform(string, pattern, replacement))
66   }

67   switch(type(pattern),
68     empty = ,
69     bound = stop("Not implemented", call. = FALSE),
70     fixed = stri_replace_first_fixed(string, pattern, replacement,
71       opts_fixed = opts(pattern)),
72     coll = stri_replace_first_coll(string, pattern, replacement,
73       opts_collator = opts(pattern)),
74     regex = stri_replace_first_regex(string, pattern, fix_replacement
(replacement),
75       opts_regex = opts(pattern)))
76   )
77 }
78 }
```

Ctrl + . = find functions/files



A screenshot of the RStudio interface. The title bar says "~/Documents/tidyverse/stringr - master - RStudio". The main area shows an R script named "replace.r". The "Source" tab is selected. The toolbar has a "Go to file/function" button highlighted with a red box. The script code is identical to the one in the first screenshot.

```
56 #' 
57 #' 
58 #' x <- c(
59 #'   "Roses are red, violets are blue",
60 #'   "My favourite colour is green"
61 #' )
62 #' str_replace_all(x, colours, col2hex)
63 str_replace <- function(string, pattern, replacement) {
64   if (!missing(replacement) && is.function(replacement)) {
65     return(str_transform(string, pattern, replacement))
66   }

67   switch(type(pattern),
68     empty = ,
69     bound = stop("Not implemented", call. = FALSE),
70     fixed = stri_replace_first_fixed(string, pattern, replacement,
71       opts_fixed = opts(pattern)),
72     coll = stri_replace_first_coll(string, pattern, replacement,
73       opts_collator = opts(pattern)),
74     regex = stri_replace_first_regex(string, pattern, fix_replacement
(replacement),
75       opts_regex = opts(pattern)))
76   )
77 }
78 }
```

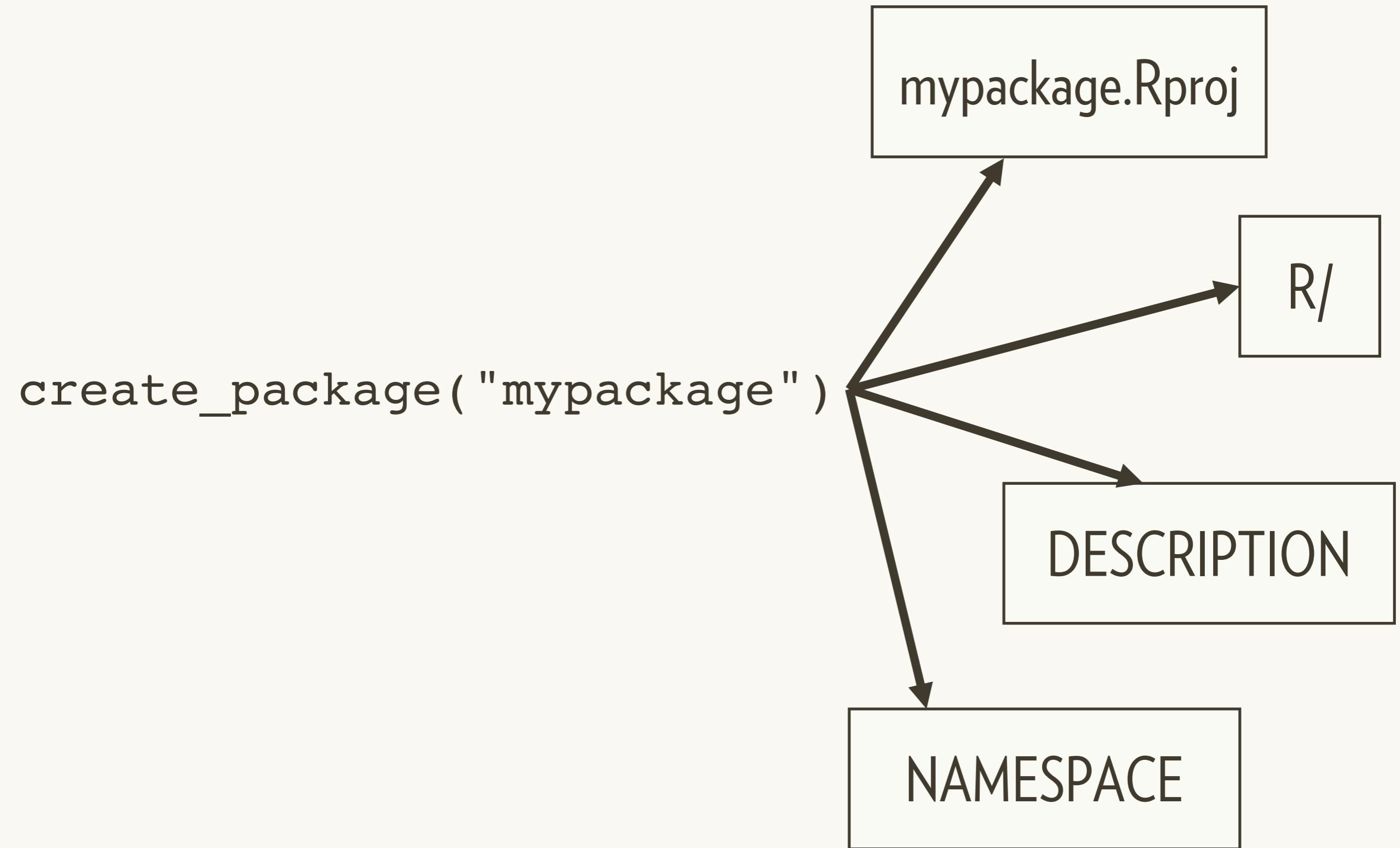
F2 = jump to definition

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)
```

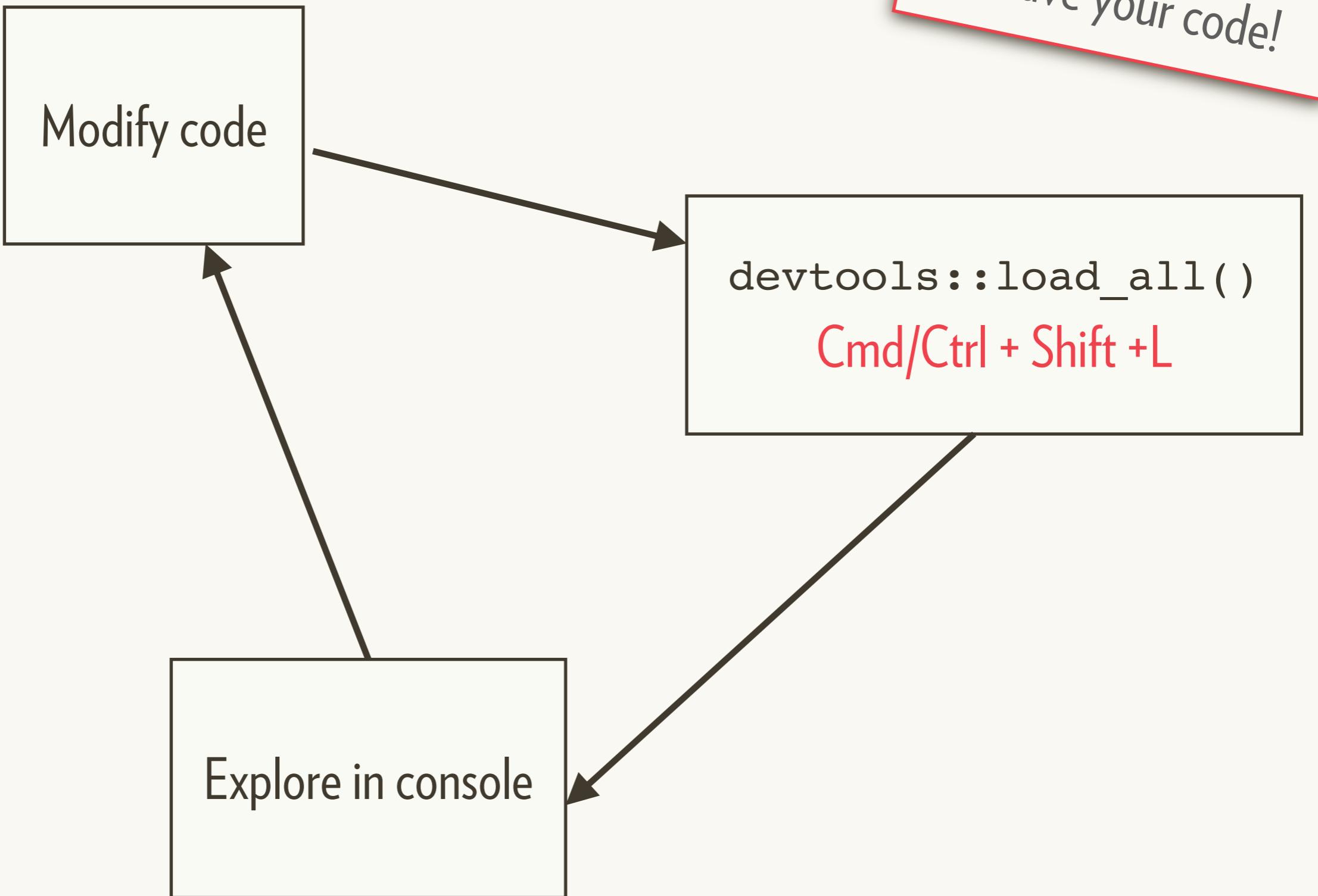
What happens we run `create_package()`?



`package.skeleton()`

Never use this!

Why bother?



Your turn

1. Open `mylittlepackage.Rproj`.
2. Jump to `rpony()` using only the keyboard.
3. Load all the functions, then run `rpony(10)`.

Uhoh! Hadley has 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))
}
```

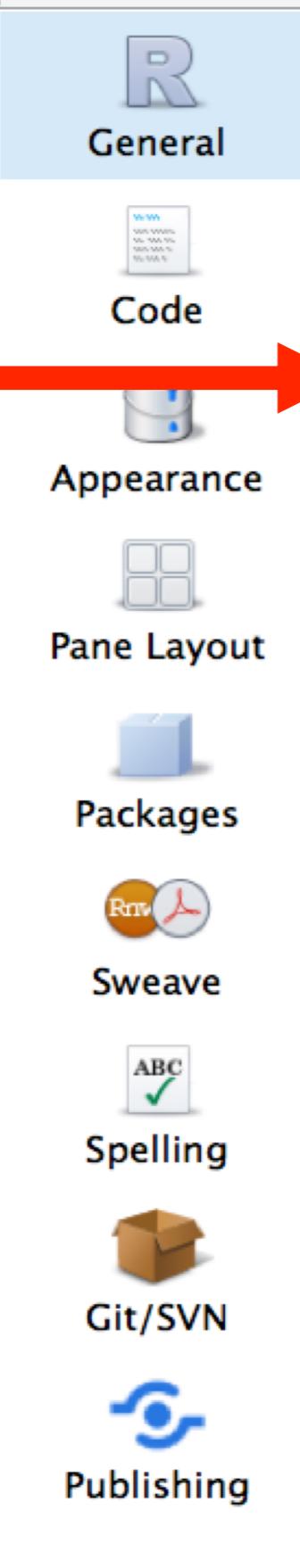
Never include analysis packages here

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

While you're in there, also add

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

Options



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.

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/>