

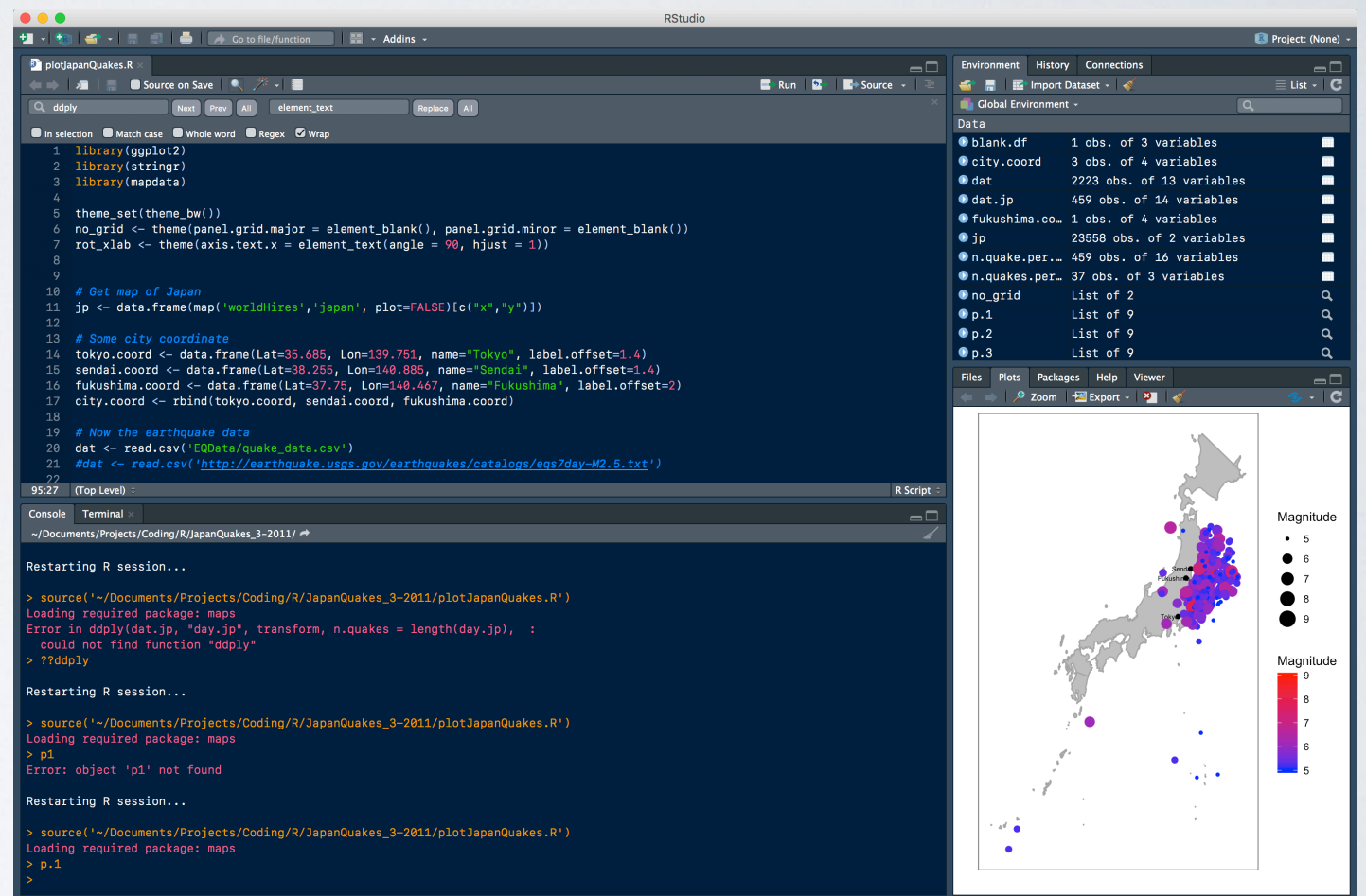


GETTING STARTED WITH RSTUDIO

Ryan Benz • R Ladies IRVINE • Oct 3, 2018

YOU'LL SPEND A LOT OF TIME IN RSTUDIO — MAKE IT PLEASANT TO USE!

- RStudio is like an artist's paint brush or musician's instrument
- Tune it to fit your needs and preferences
- Learn the basics, then dig into the details
- RStudio has a TON of features to make your life easier



WHAT IS RSTUDIO?

- **RStudio**: an awesome **I**ntegrated **D**evelopment **E**nvironment (IDE) for doing “stuff” with R
- RStudio \neq R (the language/environment)
- You don't need to use RStudio to use R, but its features and power will probably make you want to

A QUICK TOUR

The screenshot displays the RStudio environment with several key components:

- Text Editor:** The main workspace shows an R script named `plotJapanQuakes.R`. The code includes library imports (`ggplot2`, `stringr`, `mapdata`), theme settings, data loading for Japan's coordinates and earthquake data, and a `ddply` operation. A red box labeled "Text Editor" is overlaid on the script.
- Environment, History:** The top-right pane lists objects in the global environment, such as `blank.df`, `city.coord`, `dat`, `dat.jp`, `fukushima.co...`, `jp`, `n.quake.per...`, `n.quakes.per...`, `no_grid`, `p.1`, `p.2`, and `p.3`. A red box labeled "Env, Hist" is overlaid on this pane.
- Files, Plots, Help, ...:** The bottom-right pane shows a map of Japan with earthquake locations plotted as points. The size of the points represents magnitude (5 to 9), and the color represents magnitude (5 to 9). A red box labeled "Files, Plots, Help, ..." is overlaid on this pane.
- R Console:** The bottom-left pane shows the R session output, including messages like "Restarting R session...", "Loading required package: maps", and error messages such as "Error in ddply(dat.jp, 'day.jp', transform, n.quakes = length(day.jp), : could not find function 'ddply'" and "Error: object 'p1' not found". A red box labeled "R Console" is overlaid on this pane.

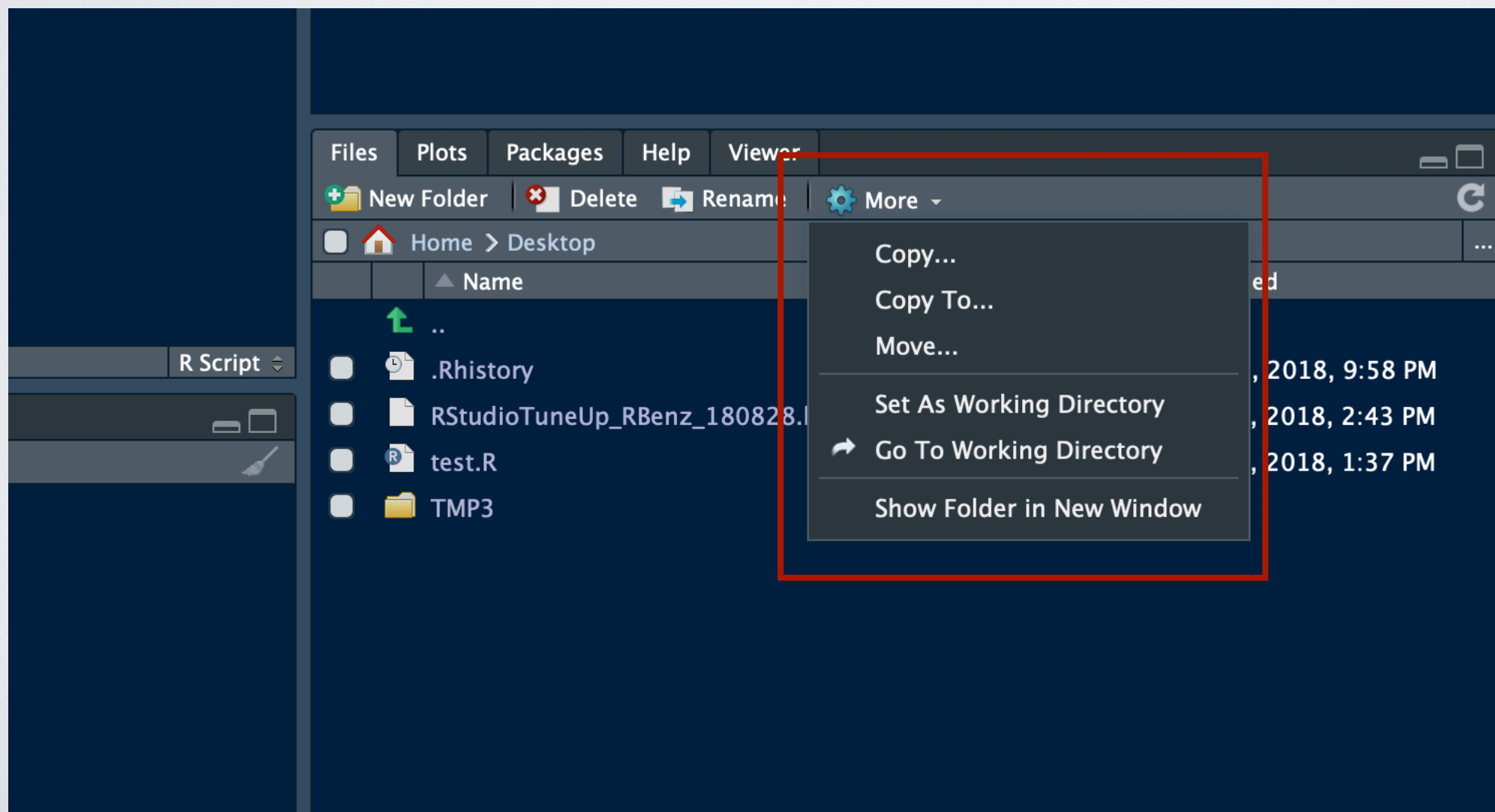
SUGGESTION: SPEND MOST OF YOUR TIME IN THE EDITOR

The image shows the RStudio interface with four main components highlighted by red boxes and labels:

- Text Editor:** The central pane showing R code for plotting Japan earthquakes. The code includes library calls for ggplot2, stringr, and mapdata, followed by data loading and plotting commands. A yellow selection box highlights lines 1 through 22.
- Env, Hist:** The Environment and History pane on the right, listing objects in the global environment such as blank.df, city.coord, dat, dat.jp, fukushima.co..., jp, n.quake.per..., n.quakes.per..., no_grid, p.1, p.2, and p.3.
- Files, Plots, Help, ...:** The bottom-right pane showing a map of Japan with earthquake locations plotted as colored circles. A legend on the right indicates earthquake magnitude from 5 to 9.
- R Console:** The bottom-left pane showing the R session output, including messages about restarting the session and loading required packages (maps).

MIND THE PATH

Every R session has an associated *Working Directory*
This is the base directory where you'll read and write files
You'll get lost if you don't know your *Working Directory*



Manually set a
path

Commands

getwd()
setwd()

*but don't use these
in your scripts...
see next slide

THINK ABOUT YOUR WORK IN TERMS OF PROJECTS

Please read Jenny Bryan's
great post on project-
oriented workflows



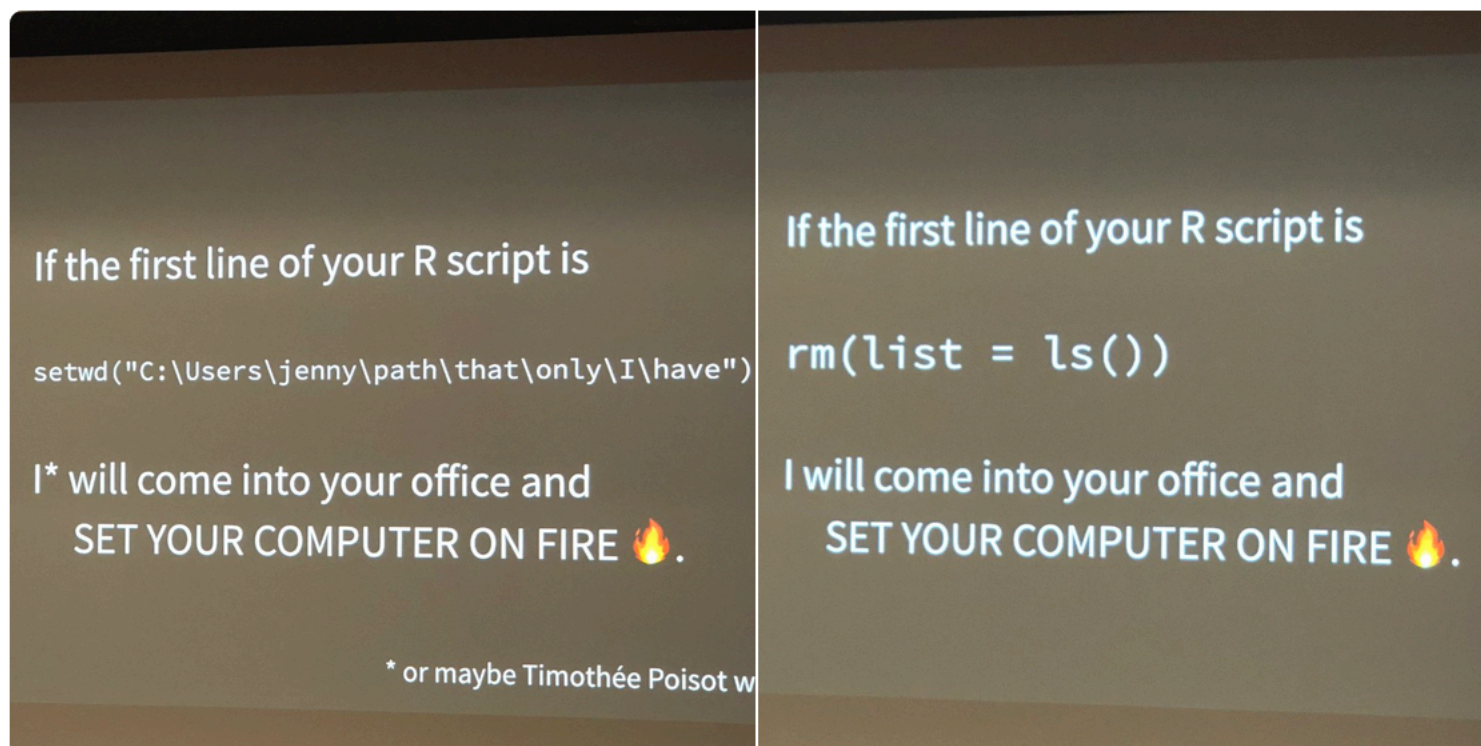
Hadley Wickham ✓

@hadleywickham

Follow




The only two things that make [@JennyBryan](#)
😡😡😡. Instead use projects + `here::here()`
[#rstats](#)

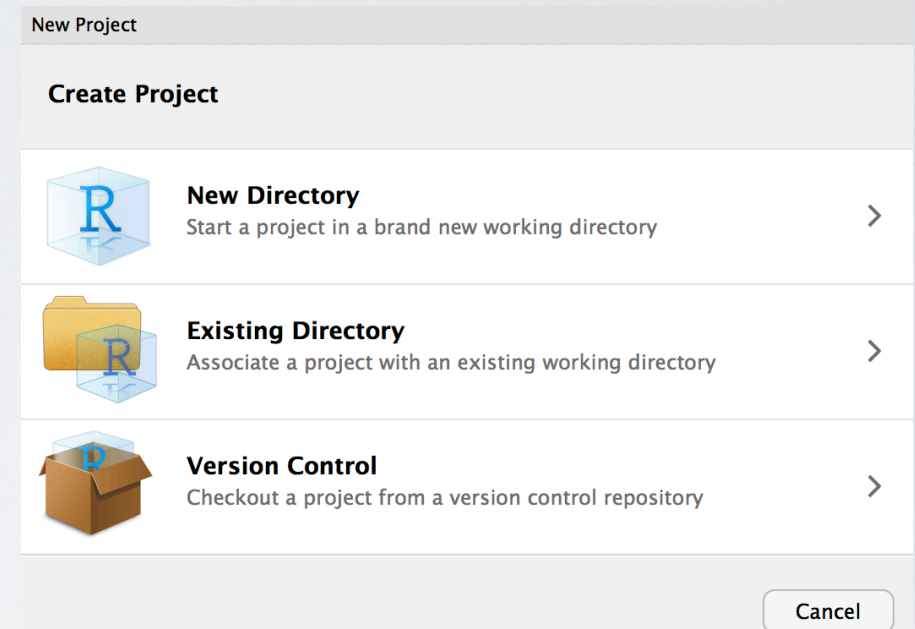


4:50 PM - 10 Dec 2017



BE DELIBERATE WHEN YOU START RSTUDIO

- Create a new RStudio project
File → New Project...
- In the terminal, cd to your working directory and run:
`open -a RStudio .`
- Hadley's cool launcher tip 
- If you do launch RStudio from *Applications*, manually navigate to your project directory before starting to work



EXAMPLE WORKFLOW #1

1. Open RStudio

2. Create a new project

File → New Project... → New Directory → ...

3. Create R scripts, add source data, save files & plots
within this project directory (or sub-directories)

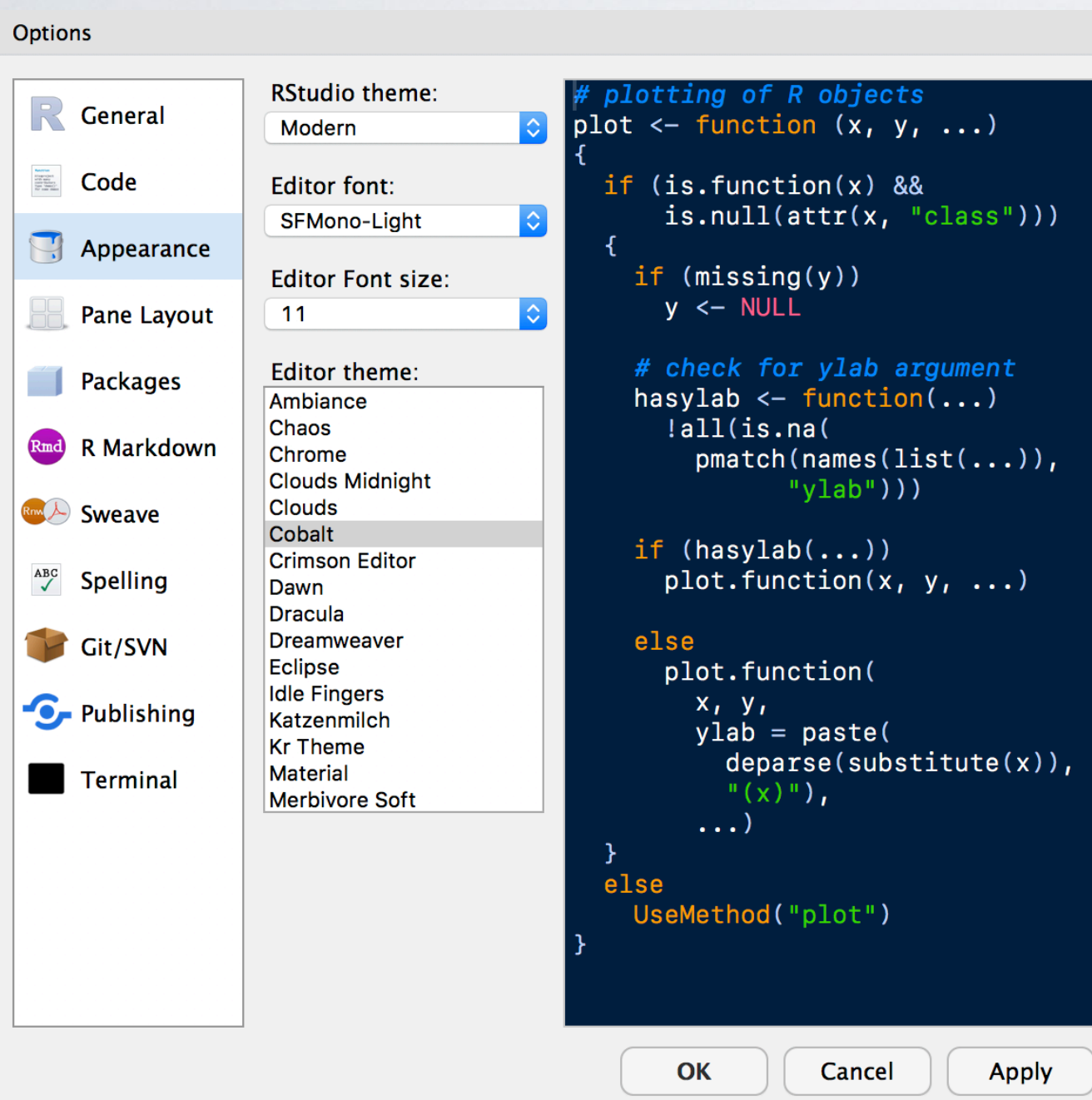
4. Return to this project at a later time by launching the
project's .Rproj file

EXAMPLE WORKFLOW #2

1. Open a terminal and `mkdir` your project directory
2. Open RStudio in the terminal *from this directory*:
`open -a RStudio .`
3. Create R scripts, add source data, save files & plots
within this project directory (or sub-directories)
4. Return to this project at a later time by `cd`'ing to this directory and launching RStudio (2. above)

CHOOSE A GREAT FONT AND THEME

Preferences → Appearance



These are some fonts I like

Inconsolata
SF Mono
IBM Plex Mono

These are some themes I like

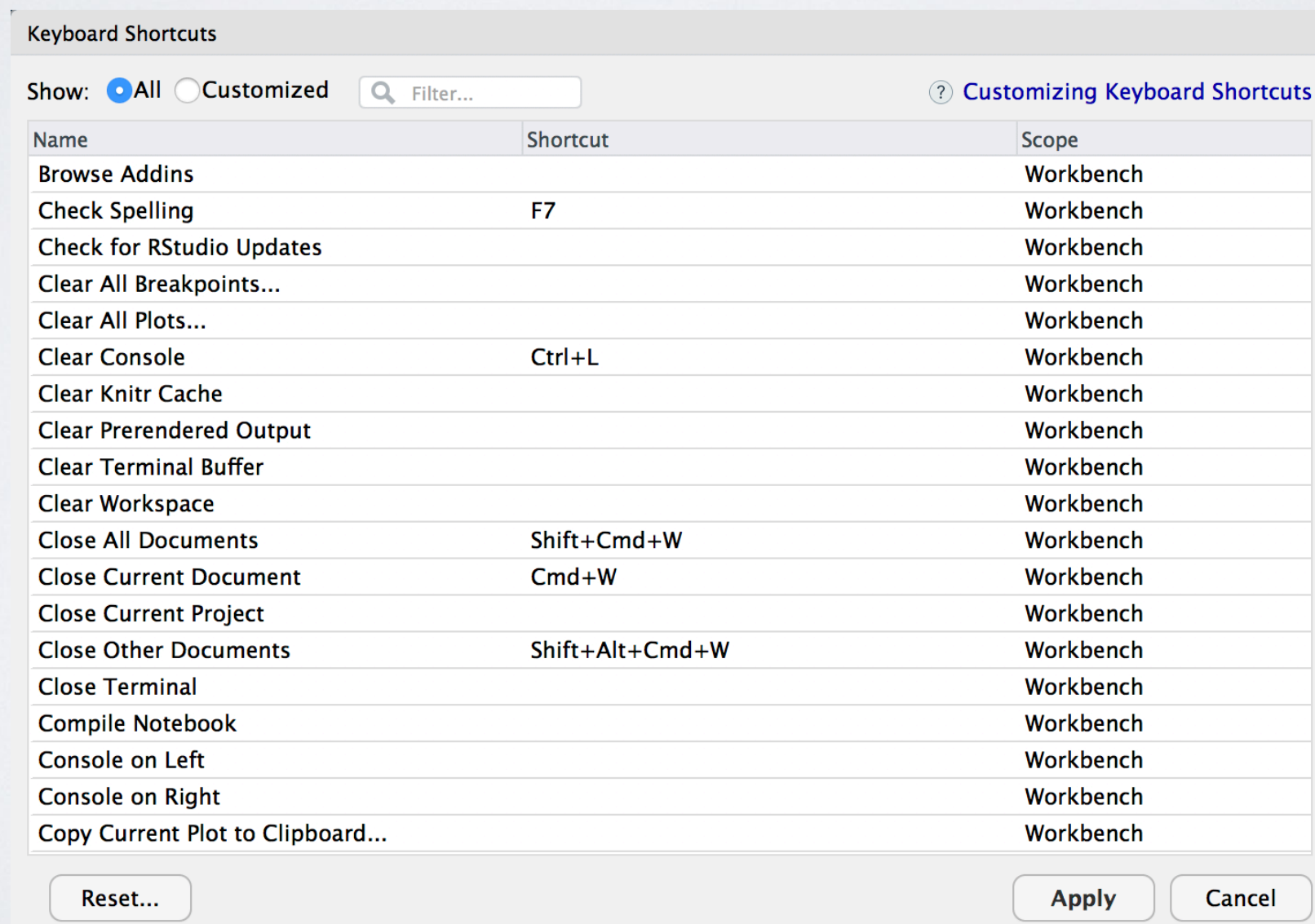
Cobalt
Idle Fingers
Material

LEARN KEYBOARD SHORTCUTS

(AT LEAST A FEW...)

Tools → Keyboard Shortcuts Help

Tools → Modify Keyboard Shortcuts...



IF YOU ONLY LEARN ONE...

(WELL ACTUALLY TWO)

THE Drill

Step 1: Restart your R session

SHIFT + CMD + F10

might also need the fn key on a laptop

Step 2: (re) Run your script

SHIFT + CMD + S

Do this frequently as you write your code

- Reminds you to capture your code in the editor, not the console
- Prevents coding left-overs from messing up your work
- Helps you find coding problems more quickly
- Essential component of a reproducible workflow

OTHER KEYBOARD SHORTCUTS I LIKE

- Execute the current line of code
CMD + Return
- Clean-up your code
CMD+A, CMD+I
- Navigating your code
CMD+arrow keys (use with SHIFT to select)

OTHER KEYBOARD SHORTCUTS I LIKE

- Comment / uncomment
CMD+SHIFT+C (but I remapped to CMD+ /)
- Switch to Editor & Console
CTRL+1 (Editor), CTRL+2 (Console)

CHANGE THESE SETTING NOW!

Preferences → General

Options

General

- Code
- Appearance
- Pane Layout
- Packages
- R Markdown
- Sweave
- Spelling
- Git/SVN
- Publishing
- Terminal

Default working directory (when not in a project):
~ Browse...

☒ Re-use idle sessions for project links
☒ 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
☐ Show .Last.value in environment listing
☒ Use debug error handler only when my code contains errors
☐ Automatically expand tracebacks in error inspector
☐ Wrap around when navigating to previous/next tab
☒ Automatically notify me of updates to RStudio

OK Cancel Apply

*The defaults are
reproducibility
nightmares*



**Uncheck
Never**

CHANGE THESE SETTING NOW!

Preferences → General

Options

General

Code

Appearance

Pane Layout

Packages

R Markdown

Sweave

Spelling

Git/SVN

Publishing

Terminal

Default working directory (when not in a project):
~ Browse...

☒ Re-use idle sessions for project links

☒ 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

☐ Show .Last.value in environment listing

☒ Use debug error handler only when my code contains errors

☐ Automatically expand tracebacks in error inspector

☐ Wrap around when navigating to previous/next tab

☒ Automatically notify me of updates to RStudio

OK Cancel Apply

I uncheck these too

Personal View

Starting to work on
a project should be
a deliberate action

FINAL THOUGHTS

- **Basic RStudio**

use it as an editor to write code and a console to execute code

- **Advanced RStudio**

learn keyboard shortcuts, tune it's settings, utilize it's power for writing packages, building Shiny apps, writing RMarkdown documents, using version control, ...

- Think about your workflow — how can you streamline your work process?
- The little bit of time you spend tuning your set-up can save you lots of time in the long run