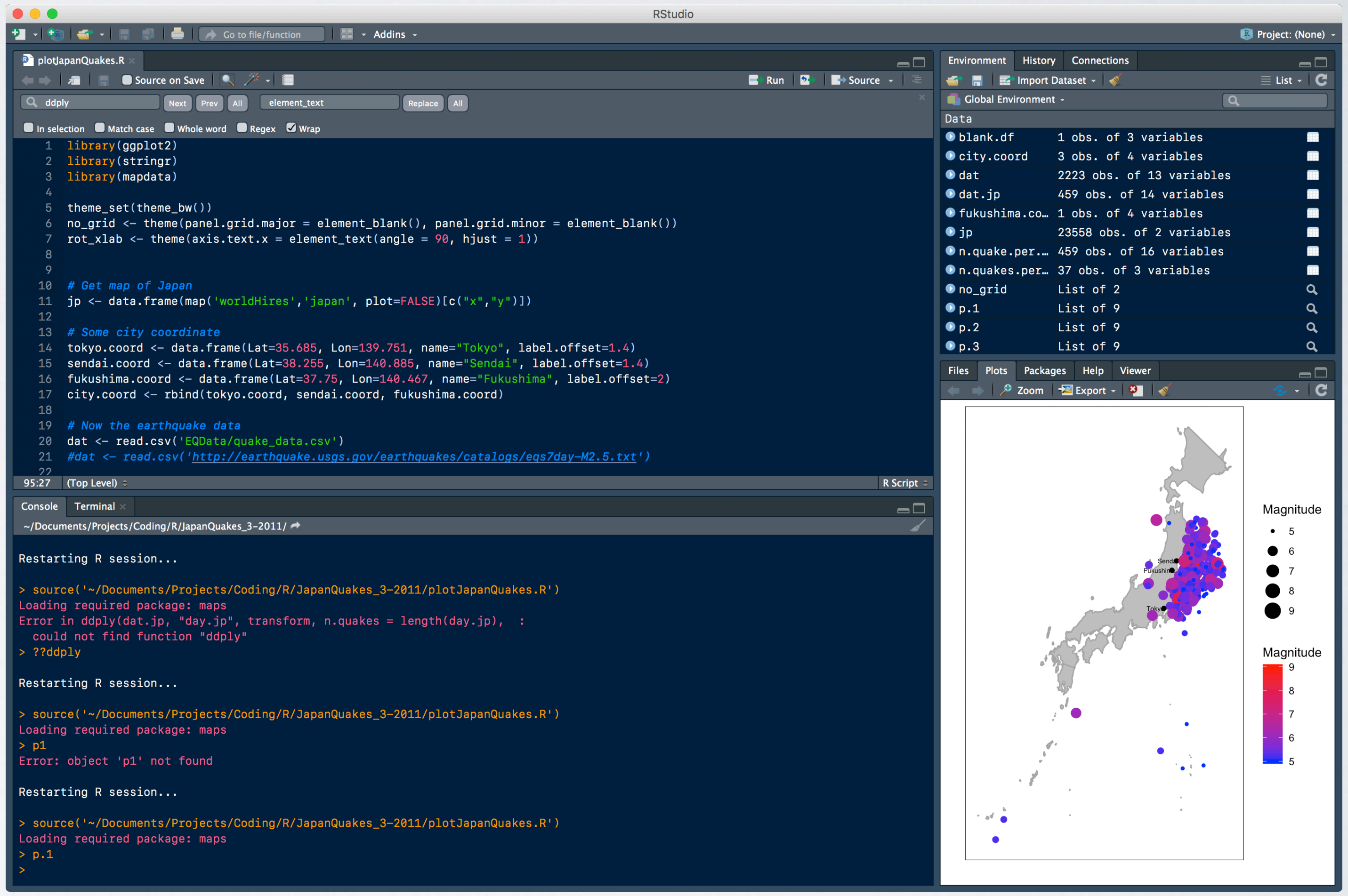
GETTING STARTED WITH RSTUDIO

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

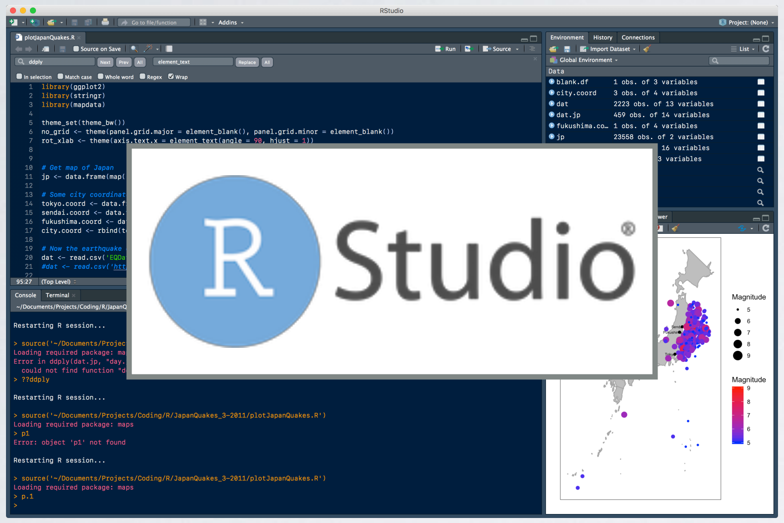


```r
library(ggplot2)
library(stringr)
library(mapdata)

theme_set(theme_bw())
no_grid <- theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
rot_xlab <- theme(axis.text.x = element_text(angle = 90, hjust = 1))


# Get map of Japan
jp <- data.frame(map('worldHires','japan', plot=FALSE)[c("x","y")])

# Some city coordinate
tokyo.coord <- data.frame(Lat=35.685, Lon=139.751, name="Tokyo", label.offset=1.4)
sendai.coord <- data.frame(Lat=38.255, Lon=140.885, name="Sendai", label.offset=1.4)
fukushima.coord <- data.frame(Lat=37.75, Lon=140.467, name="Fukushima", label.offset=2)
city.coord <- rbind(tokyo.coord, sendai.coord, fukushima.coord)

# Now the earthquake data
dat <- read.csv('EQData/quake_data.csv')
#dat <- read.csv('http://earthquake.usgs.gov/earthquakes/catalogs/eqs7day-M2.5.txt')
```

~/Documents/Projects/Coding/R/JapanQuakes_3-2011/

```
Restarting R session...

> source('~/Documents/Projects/Coding/R/JapanQuakes_3-2011/plotJapanQuakes.R')
Loading required package: maps
Error in ddply(dat.jp, "day.jp", transform, n.quakes = length(day.jp),  :
  could not find function "ddply"
> ??ddply

Restarting R session...

> source('~/Documents/Projects/Coding/R/JapanQuakes_3-2011/plotJapanQuakes.R')
Loading required package: maps
> p1
Error: object 'p1' not found

Restarting R session...

> source('~/Documents/Projects/Coding/R/JapanQuakes_3-2011/plotJapanQuakes.R')
Loading required package: maps
> p.1
>
```

blank.df	1 obs. of 3 variables
city.coord	3 obs. of 4 variables
dat	2223 obs. of 13 variables
dat.jp	459 obs. of 14 variables
fukushima.co…	1 obs. of 4 variables
jp	23558 obs. of 2 variables
n.quake.per.…	459 obs. of 16 variables
n.quakes.per…	37 obs. of 3 variables
no_grid	List of 2
p.1	List of 9
p.2	List of 9
p.3	List of 9



```
library(ggplot2)
library(stringr)
library(mapdata)

theme_set(theme_bw())
no_grid <- theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
rot_xlab <- theme(axis.text.x = element_text(angle = 90, hjust = 1))


# Get map of Japan
jp <- data.frame(map(

# Some city coordinat
tokyo.coord <- data.f
sendai.coord <- data.
fukushima.coord <- da
city.coord <- rbind(t

# Now the earthquake
dat <- read.csv('EQDa
#dat <- read.csv('htt
```

```
~/Documents/Projects/Coding/R/JapanQ

Restarting R session...

> source('~/Documents/Projec
Loading required package: ma
Error in ddply(dat.jp, "day.
  could not find function "d
> ??ddply

Restarting R session...

> source('~/Documents/Projects/Coding/R/JapanQuakes_3-2011/plotJapanQuakes.R')
Loading required package: maps
> p1
Error: object 'p1' not found

Restarting R session...

> source('~/Documents/Projects/Coding/R/JapanQuakes_3-2011/plotJapanQuakes.R')
Loading required package: maps
> p.1
>
```

```
Data
blank.df       1 obs. of 3 variables
city.coord     3 obs. of 4 variables
dat            2223 obs. of 13 variables
dat.jp         459 obs. of 14 variables
fukushima.co…  1 obs. of 4 variables
jp             23558 obs. of 2 variables
                    16 variables
                   3 variables
```

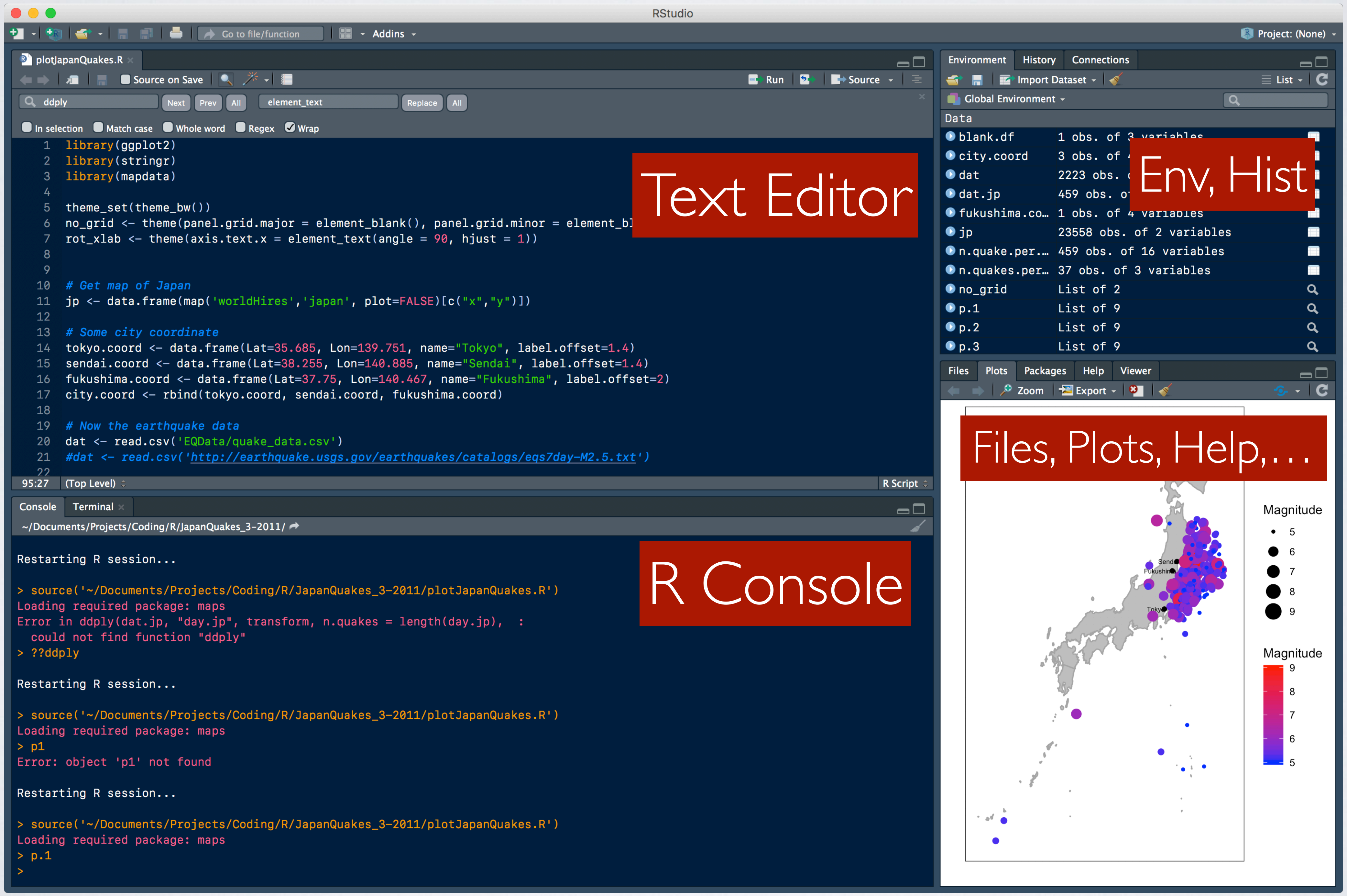
≠

RStudio is an awesome tool for working with R

…but it's not R

You need to install R first, *then* install RStudio

A QUICK TOUR

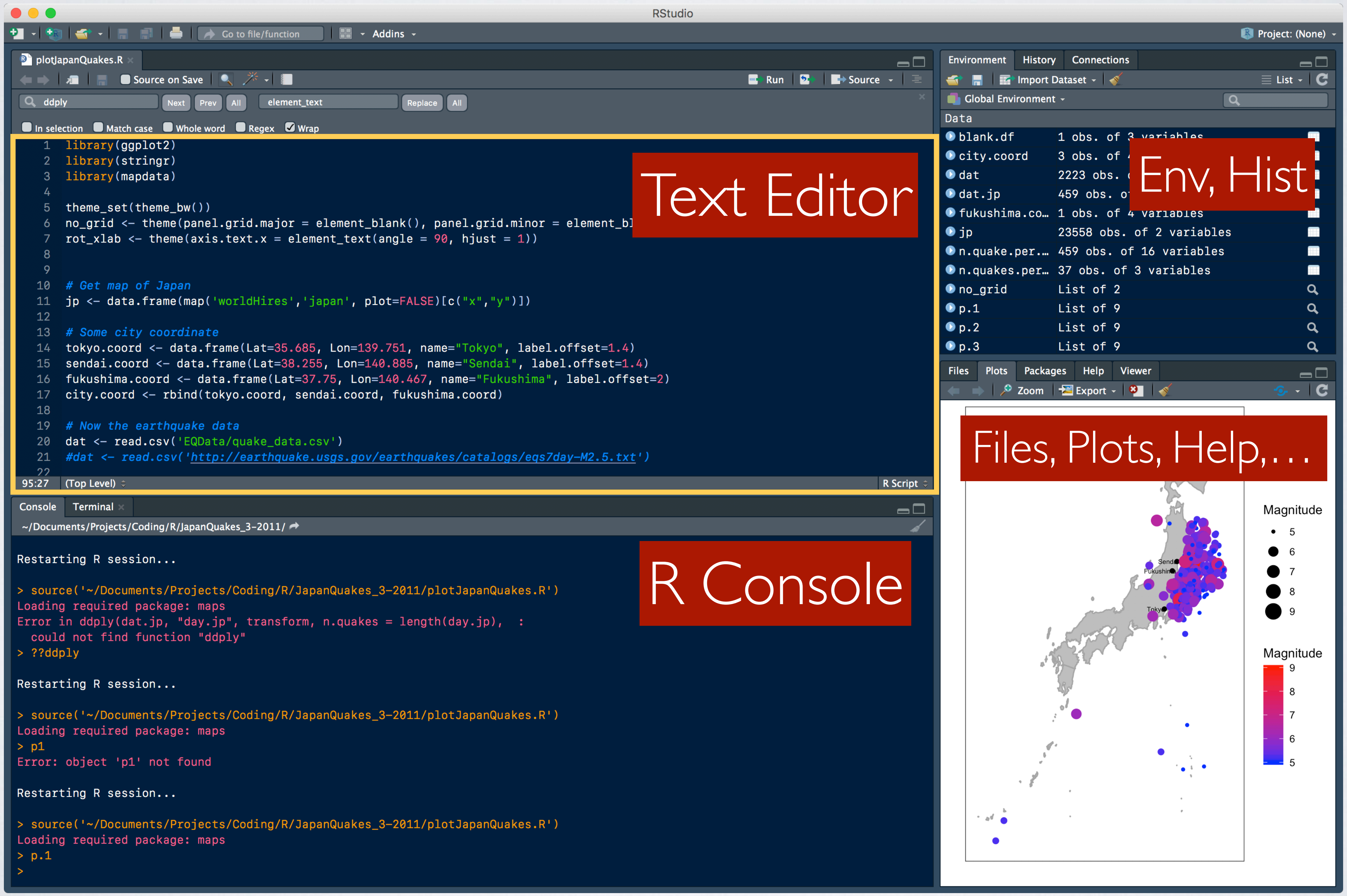


```r
library(ggplot2)
library(stringr)
library(mapdata)

theme_set(theme_bw())
no_grid <- theme(panel.grid.major = element_blank(), panel.grid.minor = element_bl
rot_xlab <- theme(axis.text.x = element_text(angle = 90, hjust = 1))


# Get map of Japan
jp <- data.frame(map('worldHires','japan', plot=FALSE)[c("x","y")])

# Some city coordinate
tokyo.coord <- data.frame(Lat=35.685, Lon=139.751, name="Tokyo", label.offset=1.4)
sendai.coord <- data.frame(Lat=38.255, Lon=140.885, name="Sendai", label.offset=1.4)
fukushima.coord <- data.frame(Lat=37.75, Lon=140.467, name="Fukushima", label.offset=2)
city.coord <- rbind(tokyo.coord, sendai.coord, fukushima.coord)

# Now the earthquake data
dat <- read.csv('EQData/quake_data.csv')
#dat <- read.csv('http://earthquake.usgs.gov/earthquakes/catalogs/eqs7day-M2.5.txt')
```

```
Restarting R session...

> source('~/Documents/Projects/Coding/R/JapanQuakes_3-2011/plotJapanQuakes.R')
Loading required package: maps
Error in ddply(dat.jp, "day.jp", transform, n.quakes = length(day.jp),  :
  could not find function "ddply"
> ??ddply

Restarting R session...

> source('~/Documents/Projects/Coding/R/JapanQuakes_3-2011/plotJapanQuakes.R')
Loading required package: maps
> p1
Error: object 'p1' not found

Restarting R session...

> source('~/Documents/Projects/Coding/R/JapanQuakes_3-2011/plotJapanQuakes.R')
Loading required package: maps
> p.1
>
```

```
blank.df        1 obs. of 3 variables
city.coord      3 obs. of 
dat             2223 obs. 
dat.jp          459 obs. o
fukushima.co…   1 obs. of 4 variables
jp              23558 obs. of 2 variables
n.quake.per.…   459 obs. of 16 variables
n.quakes.per…   37 obs. of 3 variables
no_grid         List of 2
p.1             List of 9
p.2             List of 9
p.3             List of 9
```

SPEND MOST OF YOUR TIME IN THE EDITOR



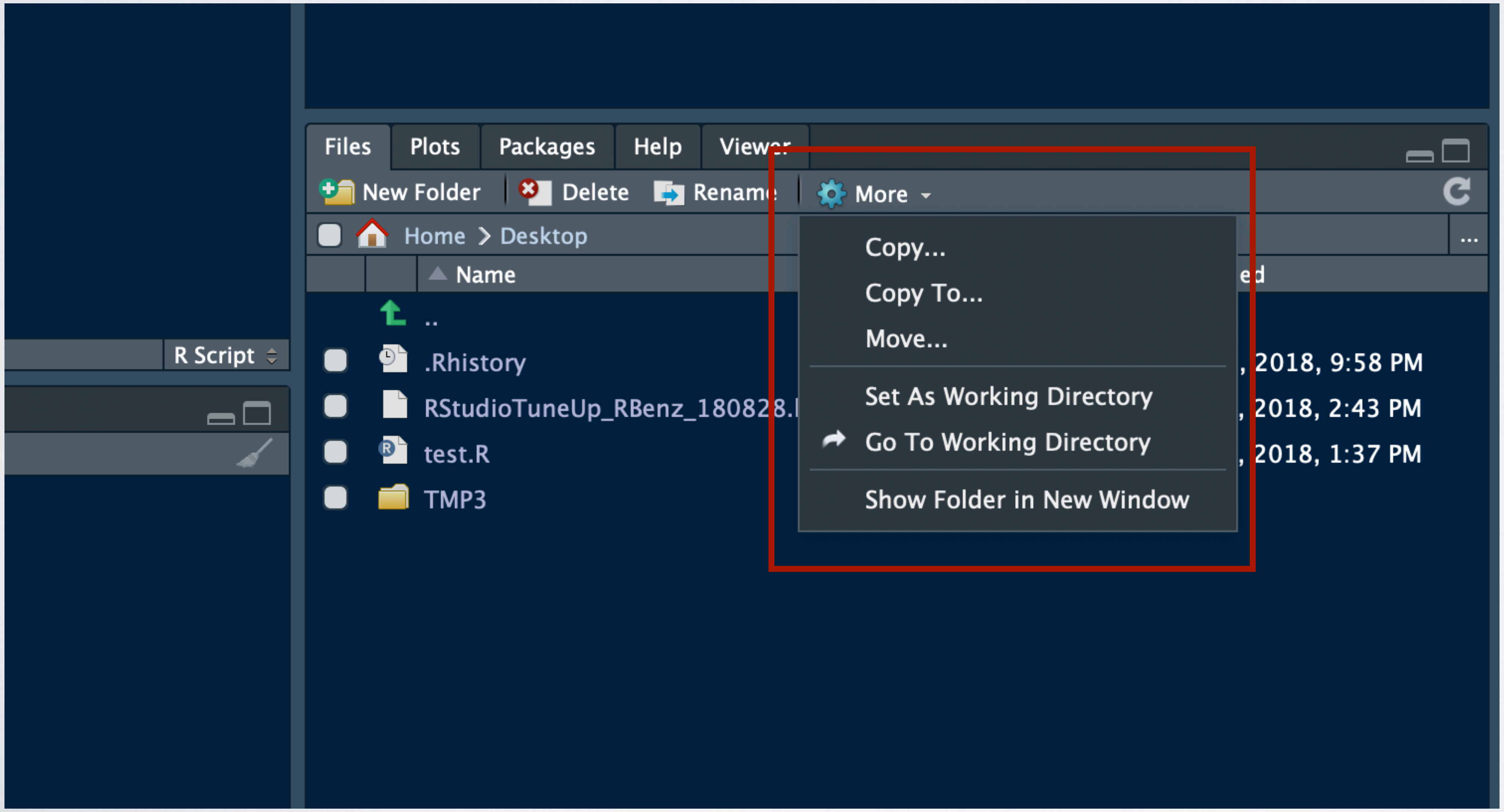
```r
library(ggplot2)
library(stringr)
library(mapdata)

theme_set(theme_bw())
no_grid <- theme(panel.grid.major = element_blank(), panel.grid.minor = element_bl
rot_xlab <- theme(axis.text.x = element_text(angle = 90, hjust = 1))


# Get map of Japan
jp <- data.frame(map('worldHires','japan', plot=FALSE)[c("x","y")])

# Some city coordinate
tokyo.coord <- data.frame(Lat=35.685, Lon=139.751, name="Tokyo", label.offset=1.4)
sendai.coord <- data.frame(Lat=38.255, Lon=140.885, name="Sendai", label.offset=1.4)
fukushima.coord <- data.frame(Lat=37.75, Lon=140.467, name="Fukushima", label.offset=2)
city.coord <- rbind(tokyo.coord, sendai.coord, fukushima.coord)

# Now the earthquake data
dat <- read.csv('EQData/quake_data.csv')
#dat <- read.csv('http://earthquake.usgs.gov/earthquakes/catalogs/eqs7day-M2.5.txt')
```

```
Data
blank.df          1 obs. of 3 variables
city.coord        3 obs. of 
dat               2223 obs. 
dat.jp            459 obs. o
fukushima.co…     1 obs. of 4 variables
jp                23558 obs. of 2 variables
n.quake.per.…     459 obs. of 16 variables
n.quakes.per…     37 obs. of 3 variables
no_grid           List of 2
p.1               List of 9
p.2               List of 9
p.3               List of 9
```

```
~/Documents/Projects/Coding/R/JapanQuakes_3-2011/

Restarting R session...

> source('~/Documents/Projects/Coding/R/JapanQuakes_3-2011/plotJapanQuakes.R')
Loading required package: maps
Error in ddply(dat.jp, "day.jp", transform, n.quakes = length(day.jp),  :
  could not find function "ddply"
> ??ddply

Restarting R session...

> source('~/Documents/Projects/Coding/R/JapanQuakes_3-2011/plotJapanQuakes.R')
Loading required package: maps
> p1
Error: object 'p1' not found

Restarting R session...

> source('~/Documents/Projects/Coding/R/JapanQuakes_3-2011/plotJapanQuakes.R')
Loading required package: maps
> p.1
>
```

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*



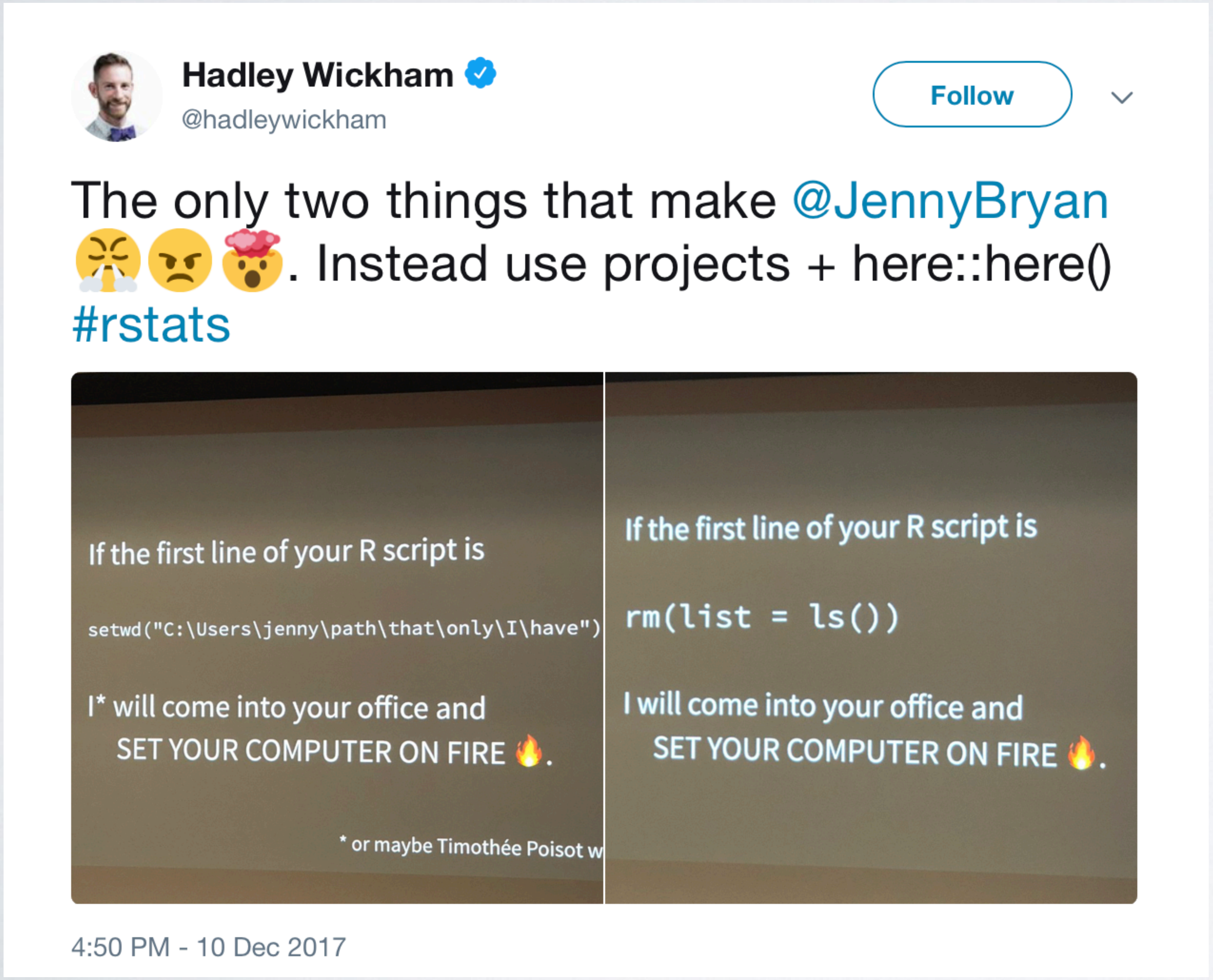
Manually set a path

<u>Commands</u>

getwd()
setwd()

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

THINK ABOUT YOUR WORK IN TERMS OF PROJECTS


Hadley Wickham
@hadleywickham

The only two things that make @JennyBryan 😤😠🤯. Instead use projects + here::here() #rstats

If the first line of your R script is

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

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

* or maybe Timothée Poisot w

If the first line of your R script is

```
rm(list = ls())
```

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

4:50 PM - 10 Dec 2017

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

AVOID STARTING RSTUDIO FROM FROM *APPLICATIONS*

- Create a new RStudio project

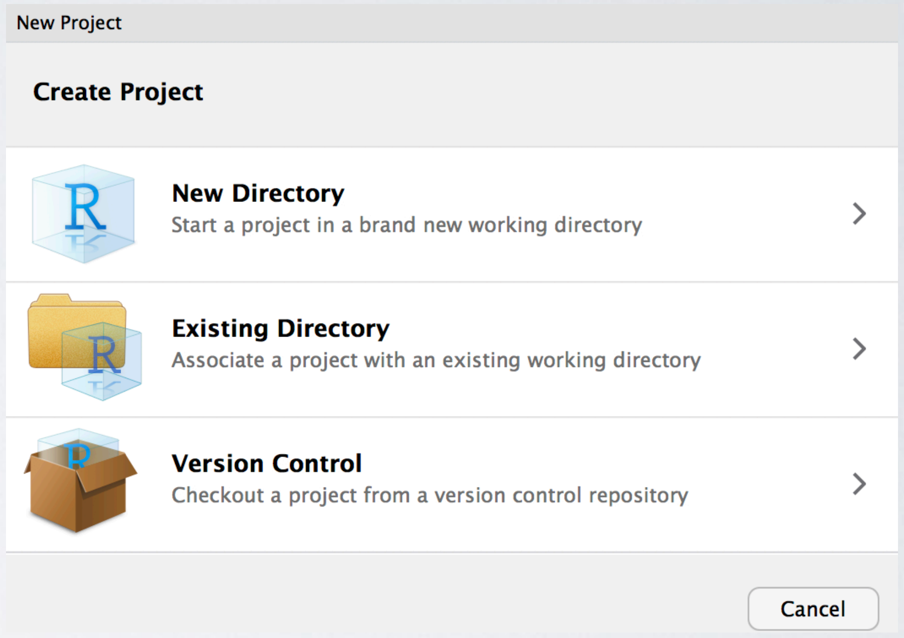
  `File → New Project…`

- 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


New Project

Create Project

New Directory
Start a project in a brand new working directory

Existing Directory
Associate a project with an existing working directory

Version Control
Checkout a project from a version control repository

Cancel

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*:

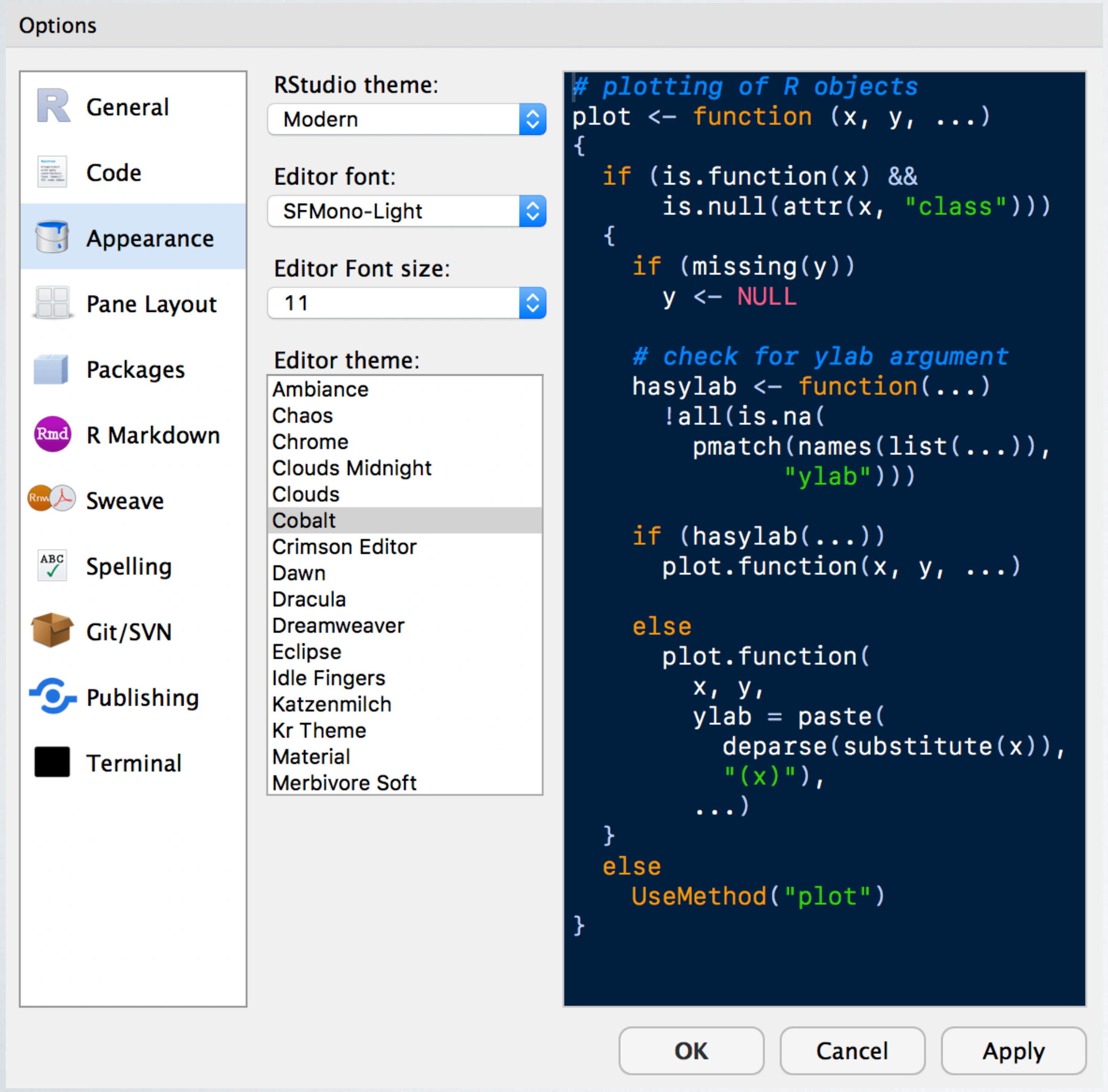
```
open -a RStudio .
```

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



RStudio theme: Modern

Editor font: SFMono-Light

Editor Font size: 11

Editor theme:

Ambiance
Chaos
Chrome
Clouds Midnight
Clouds
Cobalt
Crimson Editor
Dawn
Dracula
Dreamweaver
Eclipse
Idle Fingers
Katzenmilch
Kr Theme
Material
Merbivore Soft

```
# plotting of R objects
plot <- function (x, y, ...)
{
  if (is.function(x) &&
      is.null(attr(x, "class")))
  {
    if (missing(y))
      y <- NULL

    # check for ylab argument
    hasylab <- function(...)
      !all(is.na(
        pmatch(names(list(...)),
            "ylab")))

    if (hasylab(...))
      plot.function(x, y, ...)

    else
      plot.function(
        x, y,
        ylab = paste(
          deparse(substitute(x)),
          "(x)"),
        ...)
  }
  else
    UseMethod("plot")
}
```

<u>These are some fonts I like</u>

Inconsolata

SF Mono

IBM Plex Mono

<u>These are some themes I like</u>

Cobalt

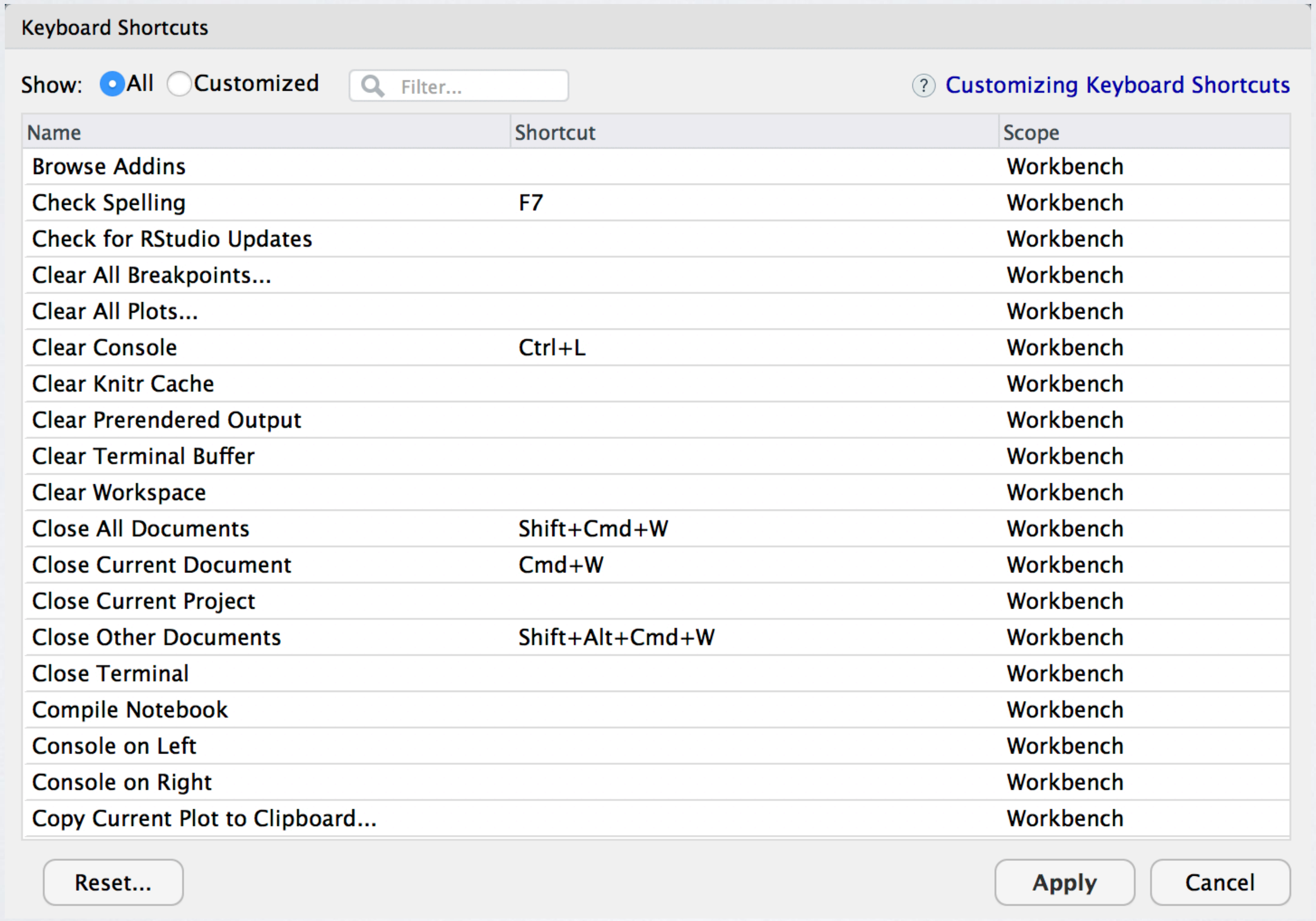
Idle Fingers

Material

LEARN KEYBOARD SHORTCUTS (AT LEAST A FEW…)

Tools → Keyboard Shortcuts Help

Tools → Modify Keyboard Shortcuts…


Keyboard Shortcuts

Show: [✓ All] Customized   Filter...   Customizing Keyboard Shortcuts

Name	Shortcut	Scope
Browse Addins		Workbench
Check Spelling	F7	Workbench
Check for RStudio Updates		Workbench
Clear All Breakpoints...		Workbench
Clear All Plots...		Workbench
Clear Console	Ctrl+L	Workbench
Clear Knitr Cache		Workbench
Clear Prerendered Output		Workbench
Clear Terminal Buffer		Workbench
Clear Workspace		Workbench
Close All Documents	Shift+Cmd+W	Workbench
Close Current Document	Cmd+W	Workbench
Close Current Project		Workbench
Close Other Documents	Shift+Alt+Cmd+W	Workbench
Close Terminal		Workbench
Compile Notebook		Workbench
Console on Left		Workbench
Console on Right		Workbench
Copy Current Plot to Clipboard...		Workbench

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

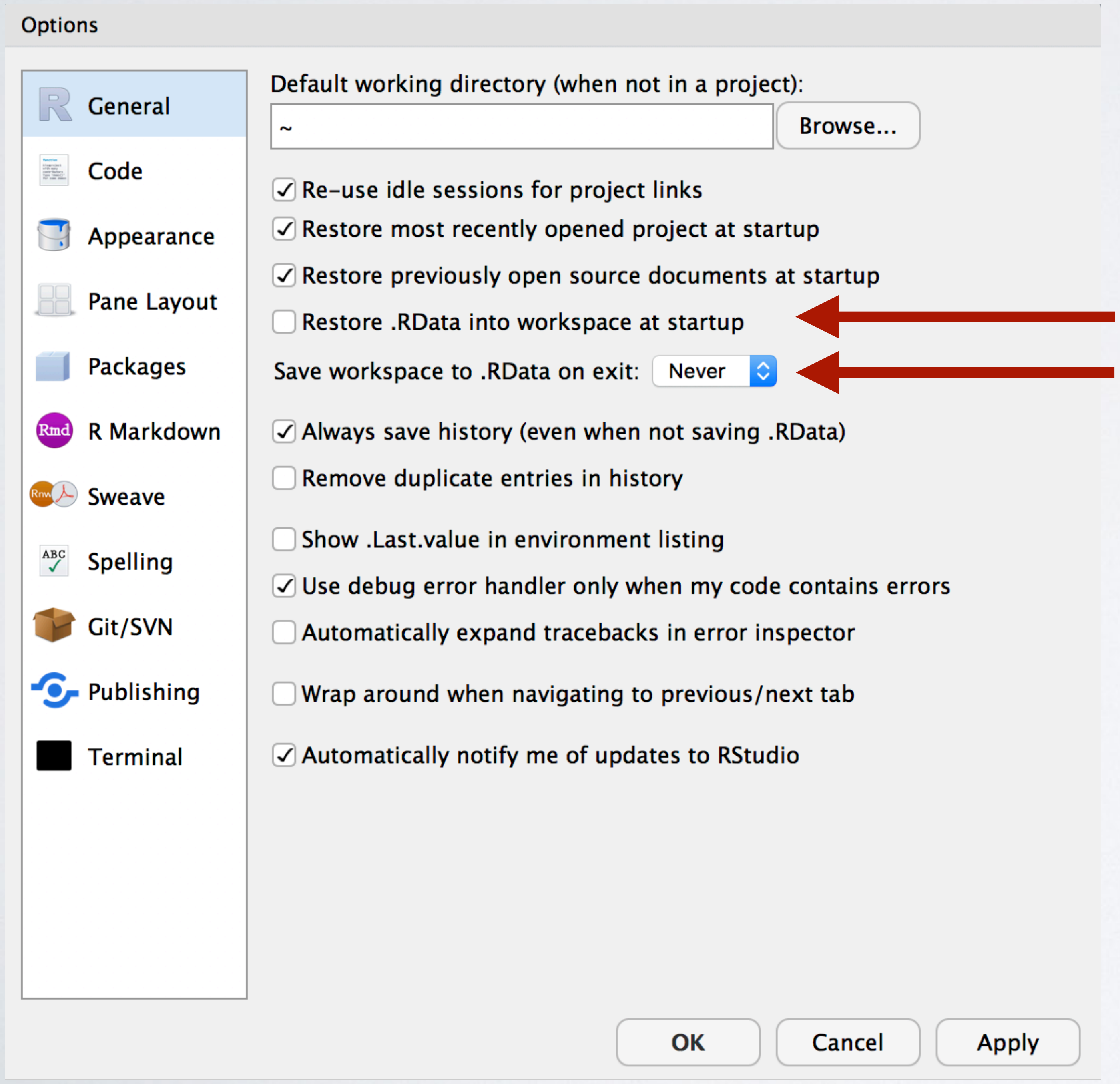
  `CMD+SHIFT+C` (but I remapped to `CMD+/`)

- Switch to Editor & Console

  `CTRL+1` (Editor), `CTRL+2` (Console)

CHANGE THESE SETTING NOW!

Preferences → General



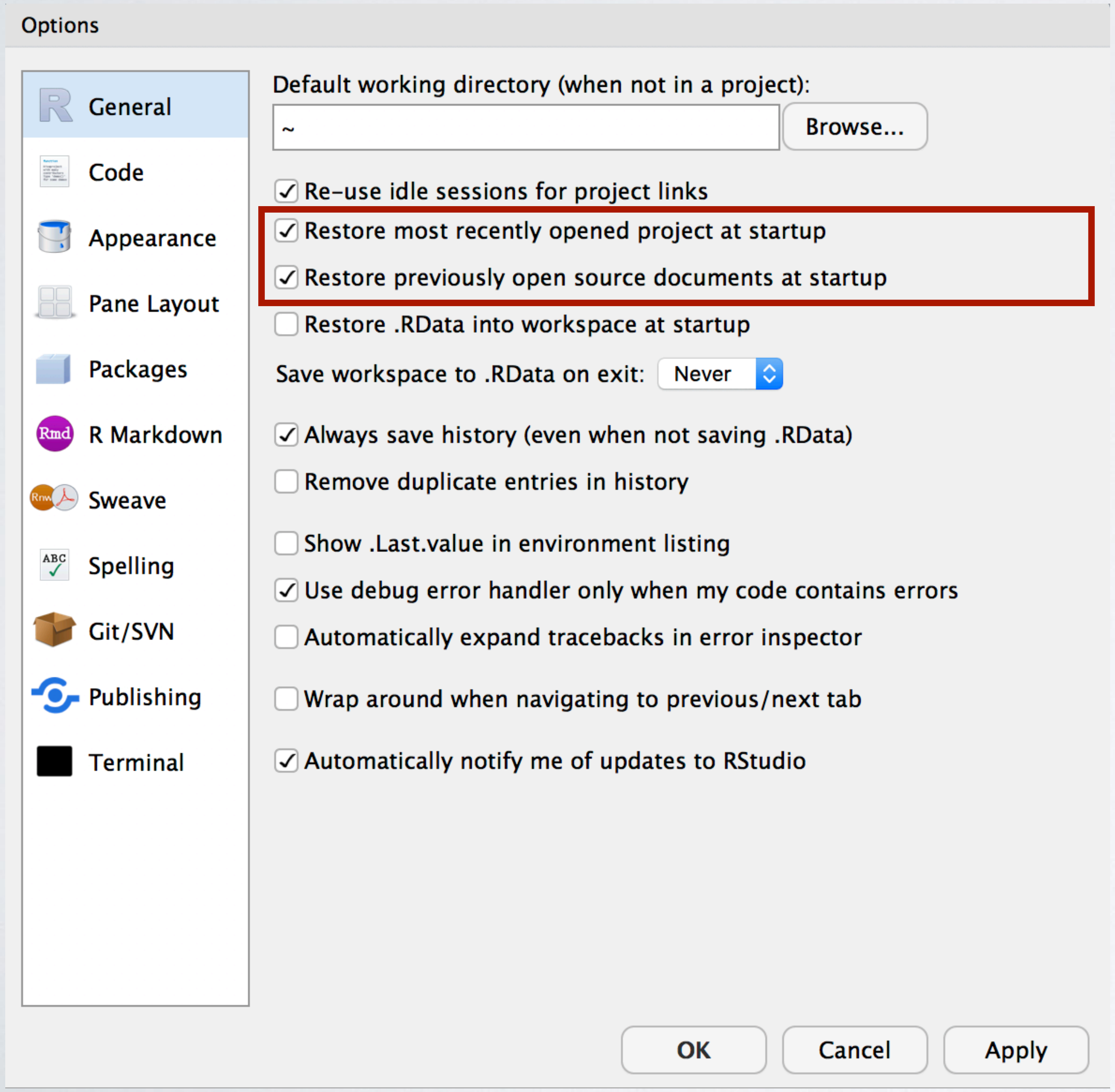
Default working directory (when not in a project): ~

- ✓ 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 ← **Uncheck**
- Save workspace to .RData on exit: Never ← **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

The defaults are reproducibility nightmares

CHANGE THESE SETTING NOW!

Preferences → General



Options

Default working directory (when not in a project):

~

- [✓] 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

I uncheck these too

<u>Personal View</u>

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

- 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