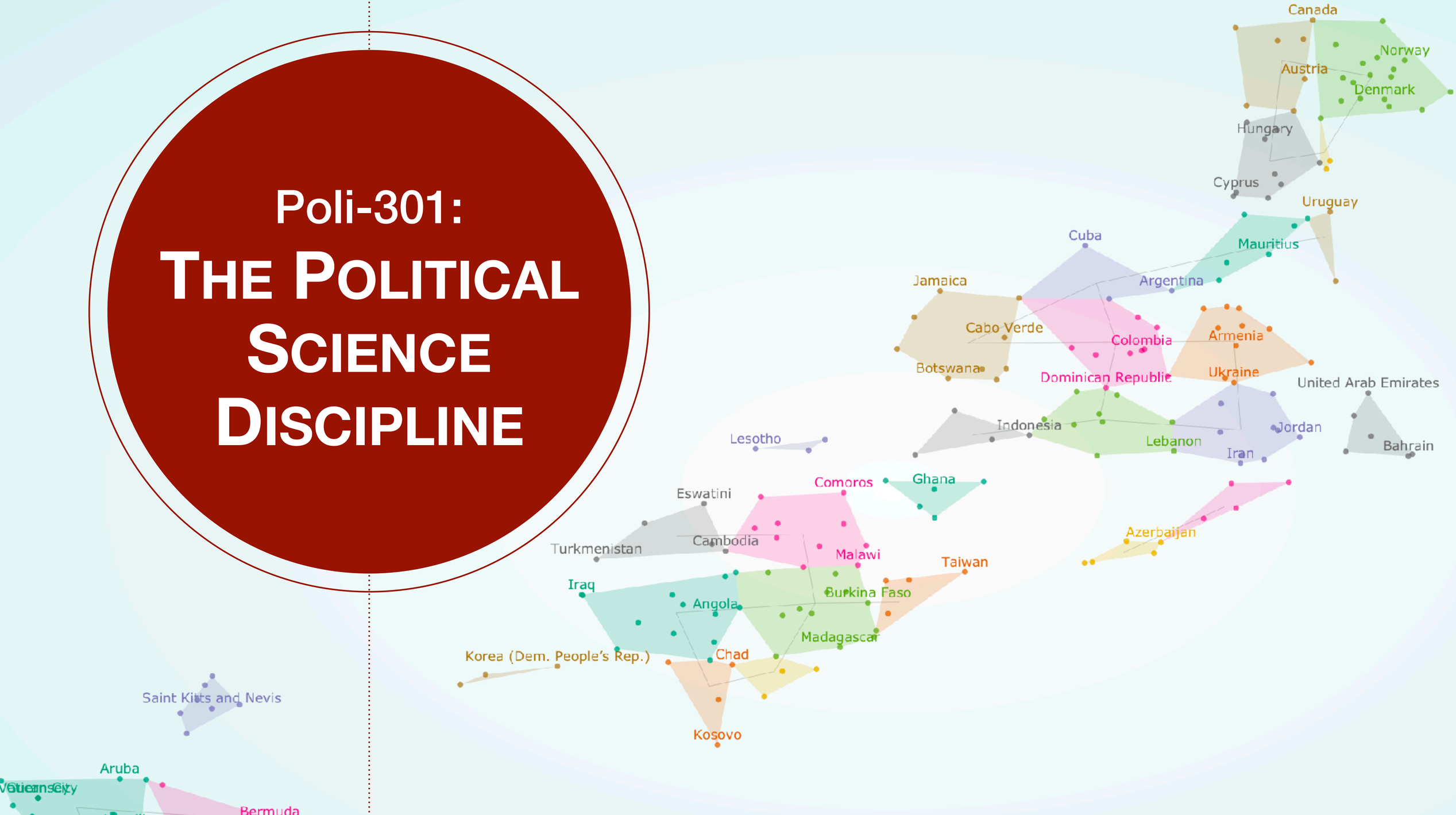
Poli-301:
The Political Science Discipline



TODAY'S AGENDA

1. Class software: Slack, R, RStudio
2. Class Demo: RStudio
3. Basics
4. Class Demo: Markdown + first dataset

Slack

Slack is a messaging platform used by teams to communicate

Can send *files, code, images*, etc.

Talk to everyone in class, specific people, or small groups

R and RStudio

R is a programming language developed for **statistics**
RStudio is an interface that makes using R easier

R: Engine	RStudio: Dashboard
	

FIGURE 2.1: Analogy of difference between R and RStudio.

R and RStudio

We will do everything in this class in RStudio

R: Do not open this	RStudio: Open this
	

What else can we do with R?

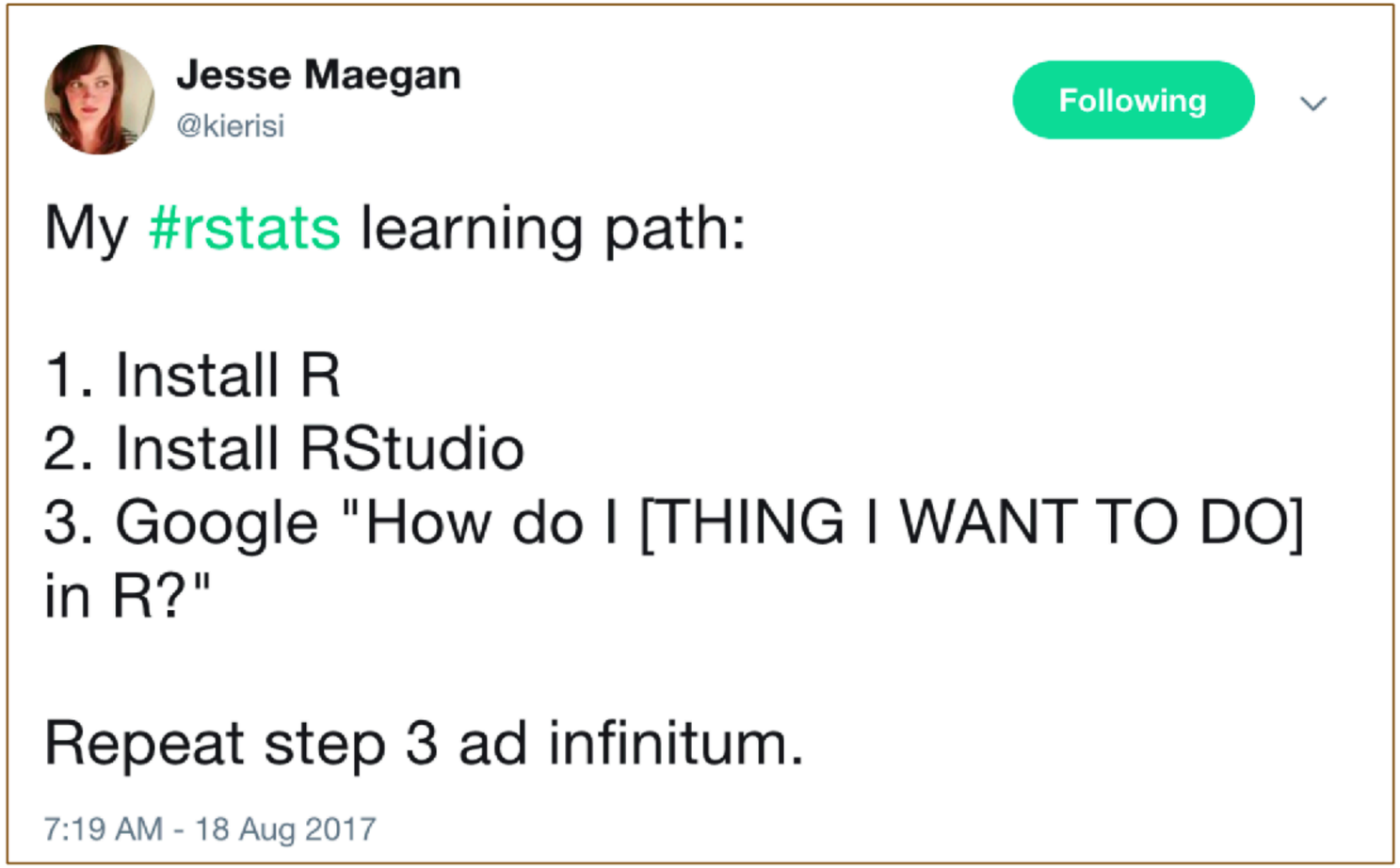
We can make **apps**

We can make **websites** (like mine, for free)

We can **scrape** websites and Twitter

Write full papers and books (ModernDive)

Learning R

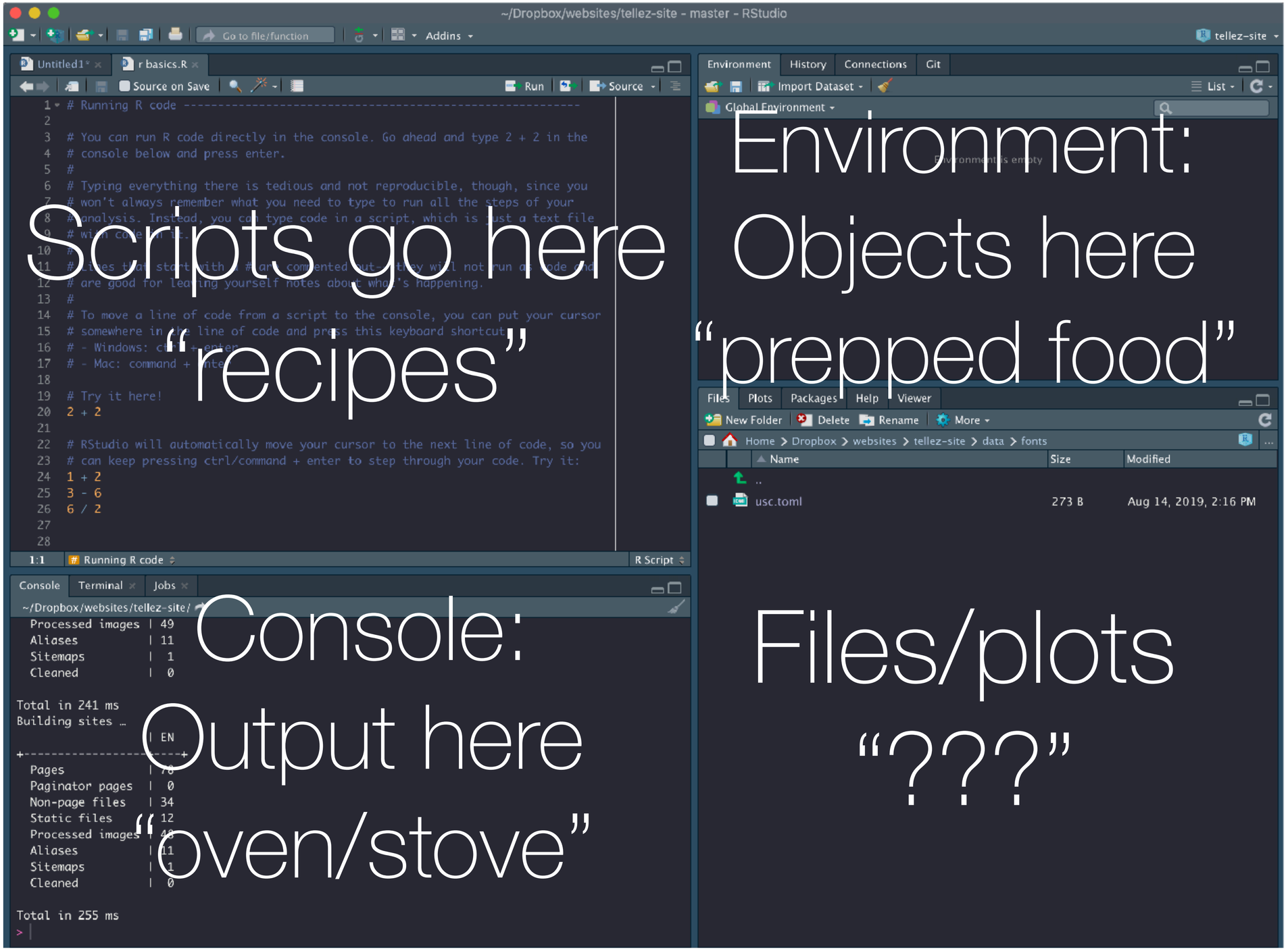


Consult Stackoverflow!

It's easy when you start out programming to get really frustrated and think, **"Oh it's me, I'm really stupid," or, "I'm not made out to program."** But, that is absolutely not the case. Everyone gets frustrated. I still get frustrated occasionally when writing R code. It's just a natural part of programming. So, it happens to everyone and gets less and less over time. Don't blame yourself. Just take a break, do something fun, and then come back and try again later.

Hadley Wickham,
Chief Data Scientist at RStudio

Demo: RStudio and customization


~/Dropbox/websites/tellez-site - master - RStudio
Go to file/function
Addins
tellez-site
Untitled1*
r basics.R
Source on Save
Run
Source
Running R code ----------
You can run R code directly in the console. Go ahead and type 2 + 2 in the
console below and press enter.
Typing everything there is tedious and not reproducible, though, since you
won't always remember what you need to type to run all the steps of your
analysis. Instead, you can type code in a script, which is just a text file
with code in it.
Lines that start with # are commented out--they will not run as code and
are good for leaving yourself notes about what's happening.
To move a line of code from a script to the console, you can put your cursor
somewhere in the line of code and press this keyboard shortcut:
- Windows: ctrl + enter
- Mac: command + enter
Try it here!
2 + 2
RStudio will automatically move your cursor to the next line of code, so you
can keep pressing ctrl/command + enter to step through your code. Try it:
1 + 2
3 - 6
6 / 2
1:1
Running R code
R Script
Scripts go here
"recipes"
Environment
History
Connections
Git
Import Dataset
List
Global Environment
Environment is empty
Environment:
Objects here
"prepped food"
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
Home > Dropbox > websites > tellez-site > data > fonts
Name
Size
Modified
..
usc.toml
273 B
Aug 14, 2019, 2:16 PM
Files/plots
"???"
Console
Terminal
Jobs
~/Dropbox/websites/tellez-site/
Processed images | 49
Aliases | 11
Sitemaps | 1
Cleaned | 0
Total in 241 ms
Building sites …
| EN
Pages | 78
Paginator pages | 0
Non-page files | 34
Static files | 12
Processed images | 49
Aliases | 11
Sitemaps | 1
Cleaned | 0
Total in 255 ms
>
Console:
Output here
"oven/stove"

Demo: first script

Packages/libraries

R comes with lots of built-in functions

But it's real power comes from **packages** built by users

R: A new phone	R Packages: Apps you can download
	

FIGURE 2.4: Analogy of R versus R packages.

The Tidyverse

The “Stuff” in R so far

objects: our data, tibbles, data frames

functions: do stuff to objects

comments: preceded by #, used to make notes

packages: extra functions not in base R

Installing packages

Install fivethirtyeight:

In the Files pane of RStudio:

a. Click on the “Packages” tab.
b. Click on “Install” next to Update.
c. Type the name of the package under “Packages (separate multiple with space or comma):” In this case, type `fivethirtyeight`.
d. Click “Install”.

RMarkdown

This might feel like **a lot** for the first day, that's OK.

The last thing we will look at is **RMarkdown**. RMarkdown is a built-in tool for RStudio that turns your scripts into reports.

It allows you to write both **normal text** and **code**. This way you can write *about* your code, your plots, and your results in the same space.

Once you're all done, you can nicely output to PDF, Word, or HTML.

"YAML":
important info about your document

"chunks" of code

Normal text

Section headers
or ## or ###…

Demo: RMarkdown

Pipes: $ > $

```
plot(select(mutate(filter(data, country = "India"), gdp_capita = gpd/pop), gdp_capita))
```

What's more readable?

```
data %>%
  filter(country == "India") %>%
  mutate(gdp_capita = gdp/pop) %>%
  select(gdp_capita) %>%
  plot()
```

The “Stuff” in R so far

objects: our data, tibbles, data frames

functions: do stuff to objects

comments: preceded by #, used to make notes

packages: extra functions not in base R

pipes: move stuff between functions

chunks: bits of R script in markdown files