Overview

Data Wrangling, Session 1

Kieran Healy

Code Horizons

July 22, 2024

Housekeeping

10:30am till 12:30pm US EST each day

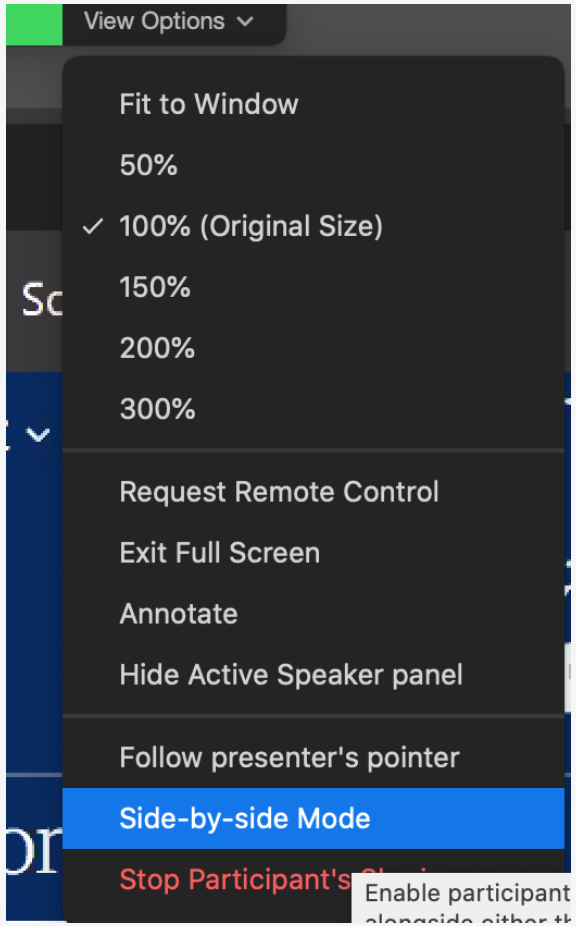
1:30pm to 3:00pm US EST each day

Use the Zoom chat to ask questions, or raise a hand with 👋

In between class sessions

slack

For a better Zoom experience



If you're watching in full-screen view and I'm sharing my screen, then from Zoom's "View options" menu *turn off* "Side-by-Side" mode.

My Setup and Yours

Talking, Slides, and Live-Coding in RStudio

Follow along with RStudio yourself if you can

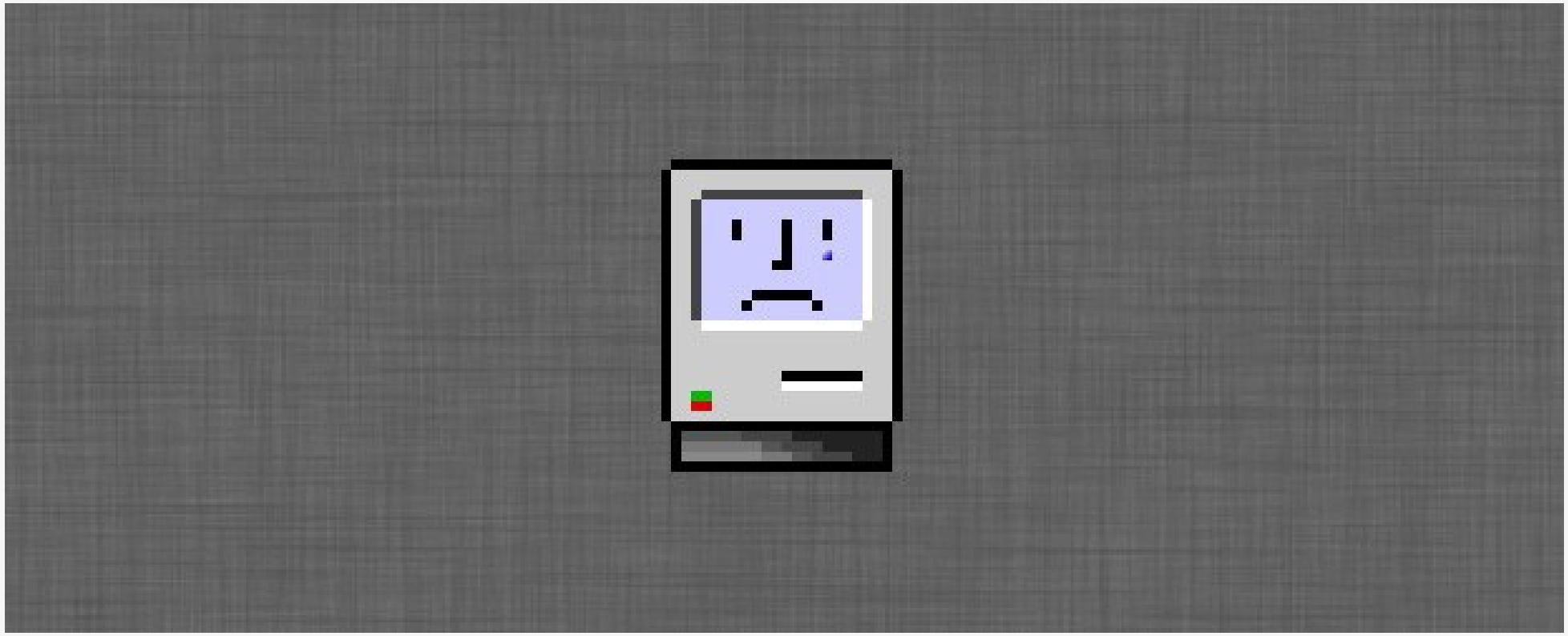
The course packet is also an RStudio project and the place for your notes

Goals for this first session

- Some big-picture motivation & perspective
- Getting familiar with RStudio and its relationship to R
- Getting oriented to R and how it thinks

DATA ANALYSIS is mostly
DATA WRANGLING

Wrangling data is frustrating



Sad Mac

Can we make it fun?

No.

Fun data wrangling

Can we make it fun?

Fun data wrangling

No.

⇦ Not *this* much fun, at any rate

OK but can we eliminate frustration?

Frustration-free data wrangling

Also no.

OK but can we eliminate frustration?

Frustration-free data wrangling

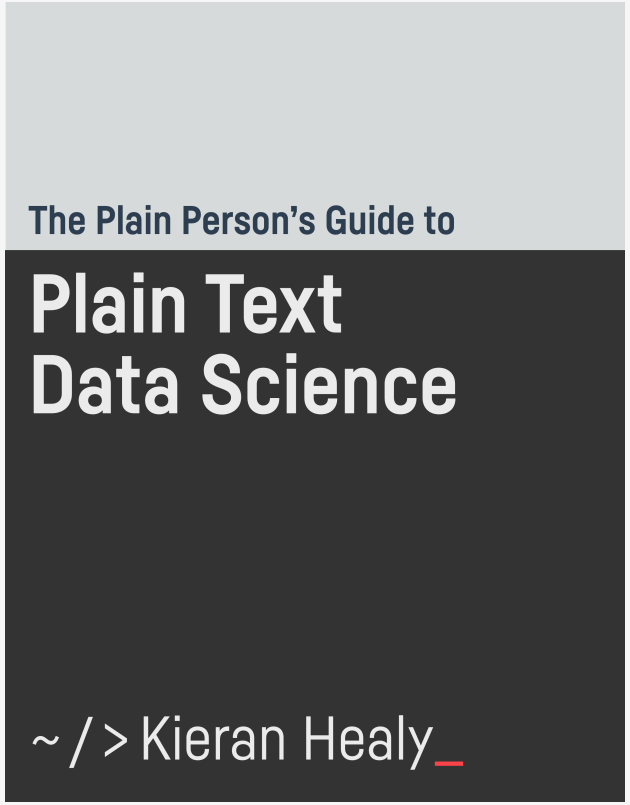
Also no.

(Sorry.)

However, we *can* make it work

Also, it's weirdly satisfying once you get into it.

We take a broadly *Plain Text* approach



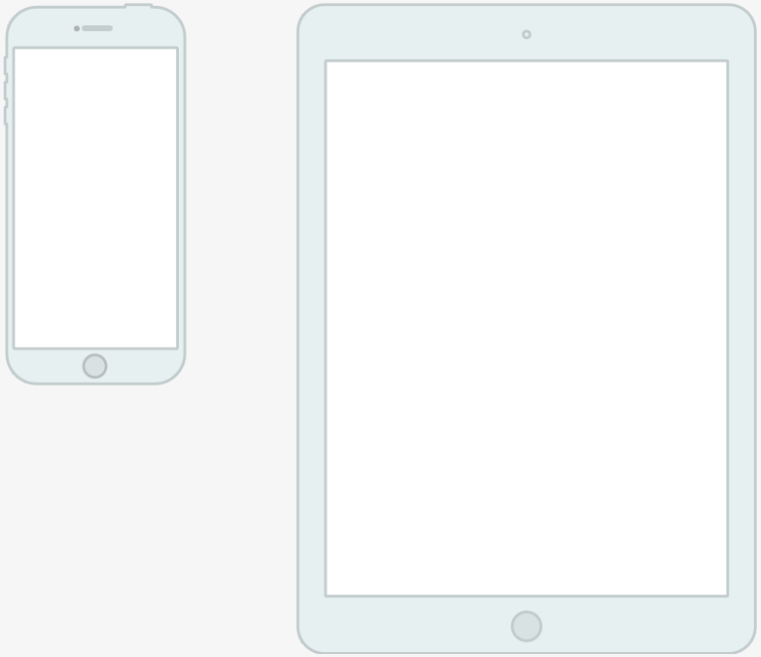
The plain person's guide

Using R and the Tidyverse can be understood within this broader context.

The same principles would apply to, e.g., using Python or similar tools.

Two revolutions in computing

Where the action is



iPhone and iPad

Touch-based user interface

Foregrounds a single application

Dislikes multi-tasking*

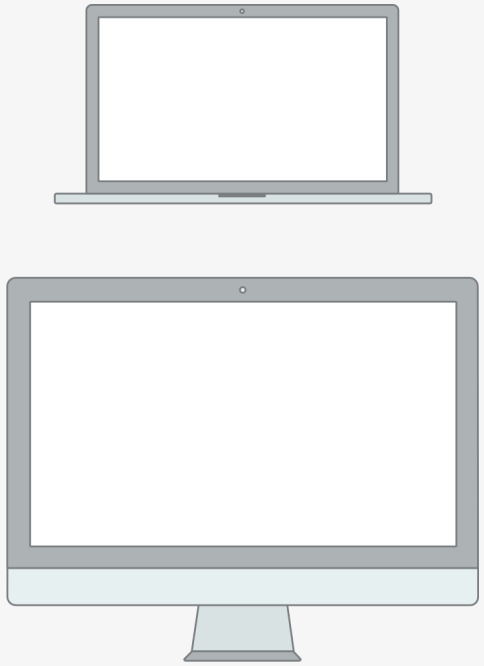
Hides the file system

“Laundry basket” model of where things are

*Multitasking

I mean, “Making different specialized applications and resources work together in the service of a single but multi-dimensional project”, not “Checking Twitter while also listening to a talk and waiting for an update from the school nurse.”

Where statistical computing lives



Desktop and laptop

Windows and pointers.

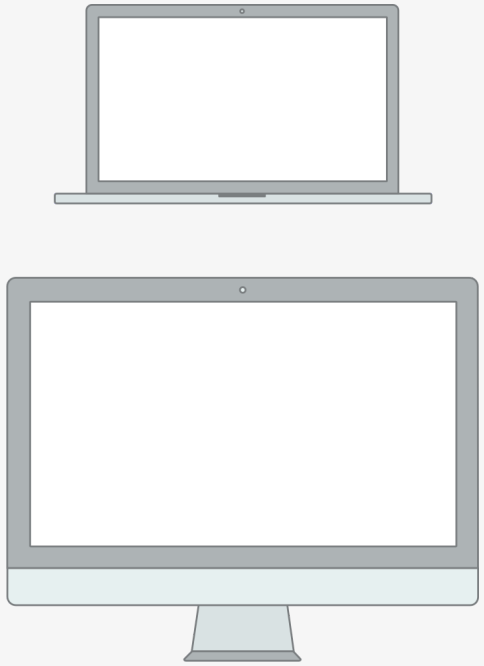
Multi-tasking, multiple windows.

Exposes and leverages the file system.

Many specialized tools in concert.

Underneath, it's the 1970s, UNIX, and the command-line.

Plain-Text Tools for Data Analysis



Desktop and laptop

Better than they've ever been!

Free! Open! Powerful!

Friendly community! Many resources!

But grounded in a UI paradigm that is increasingly far away from the everyday use of computing devices

So why do we use these tools?

Because the research process is *intrinsically messy*

“Office” vs “Engineering” approaches

Questions

What is “real” in your project?

What is the final output?

How is it produced?

How are changes managed?

Different Answers

Office model

Formatted documents are real.

Intermediate outputs are cut and pasted into documents.

Changes are tracked inside files.

Final output is often in the same format you've been working in, e.g. a Word file, or perhaps a PDF.

Different Answers

Office model

Formatted documents are real.

Intermediate outputs are cut and pasted into documents.

Changes are tracked inside files.

Final output is often in the same format you've been working in, e.g. a Word file, or perhaps a PDF.

Engineering model

Plain-text files are real.

Intermediate outputs are produced via code, often inside documents.

Changes are tracked outside files.

Final outputs are assembled programatically and converted to a desired output format.

Different strengths and weaknesses

Office model

Everyone knows Word, Excel, or Google Docs.

“Track changes” is powerful and easy.

Hm, why can’t I remember how I made this figure?.

Where did this table of results come from?

`Paper_Final_edits_FINAL_kh-1a.docx`

Different strengths and weaknesses

Office model

Everyone knows Word, Excel, or Google Docs.

“Track changes” is powerful and easy.

Hm, why can’t I remember how I made this figure?.

Where did this table of results come from?

`Paper_Final_edits_FINAL_kh-1a.docx`

Engineering model

Plain text is universally portable.

Push button, recreate analysis.

Why can’t I make R do this simple thing?

This version control stuff is a pain.

```
Object of type 'closure'
      is not subsettable
```

Each approach generates solutions to its own problems

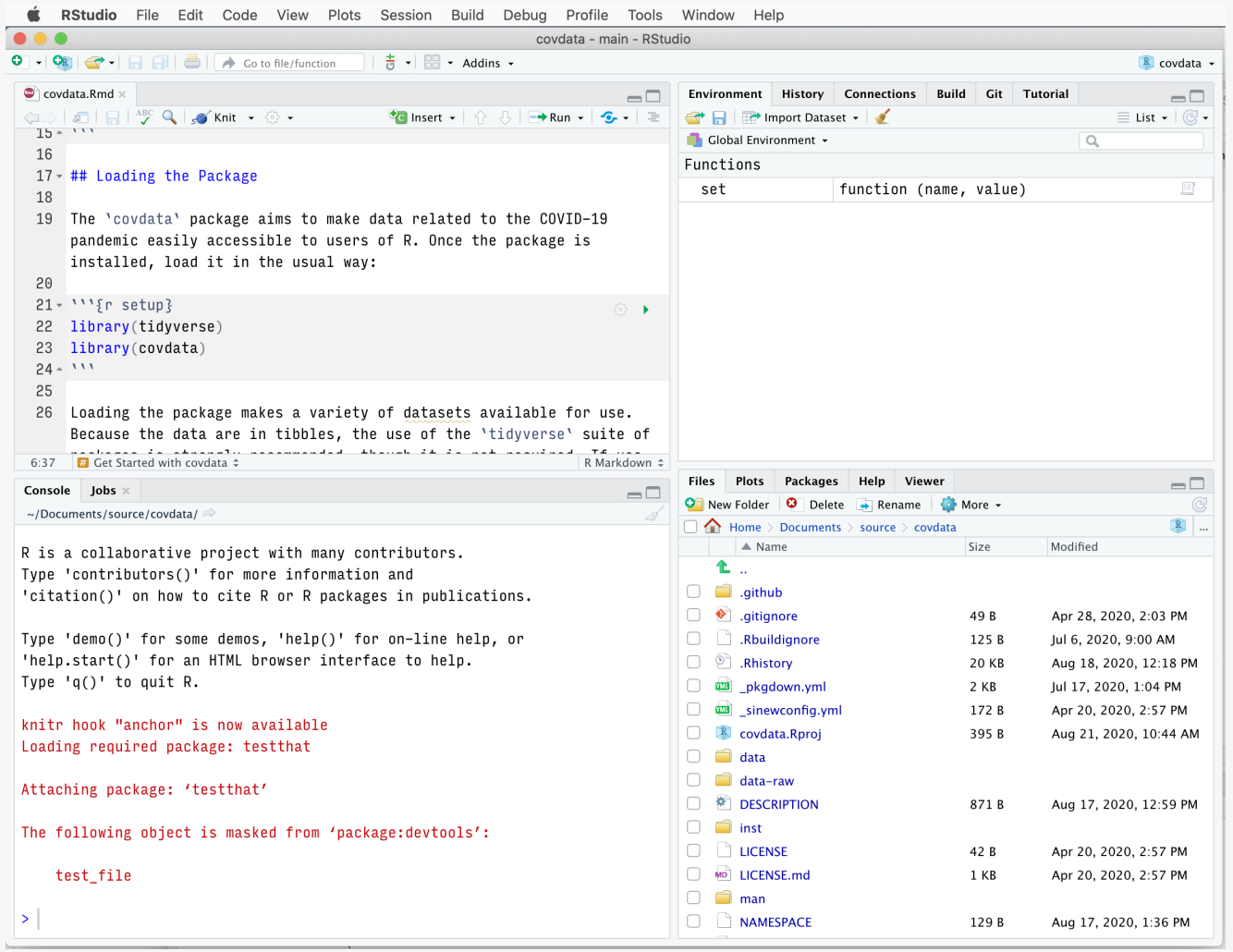
Into the Kitchen


R
Studio®

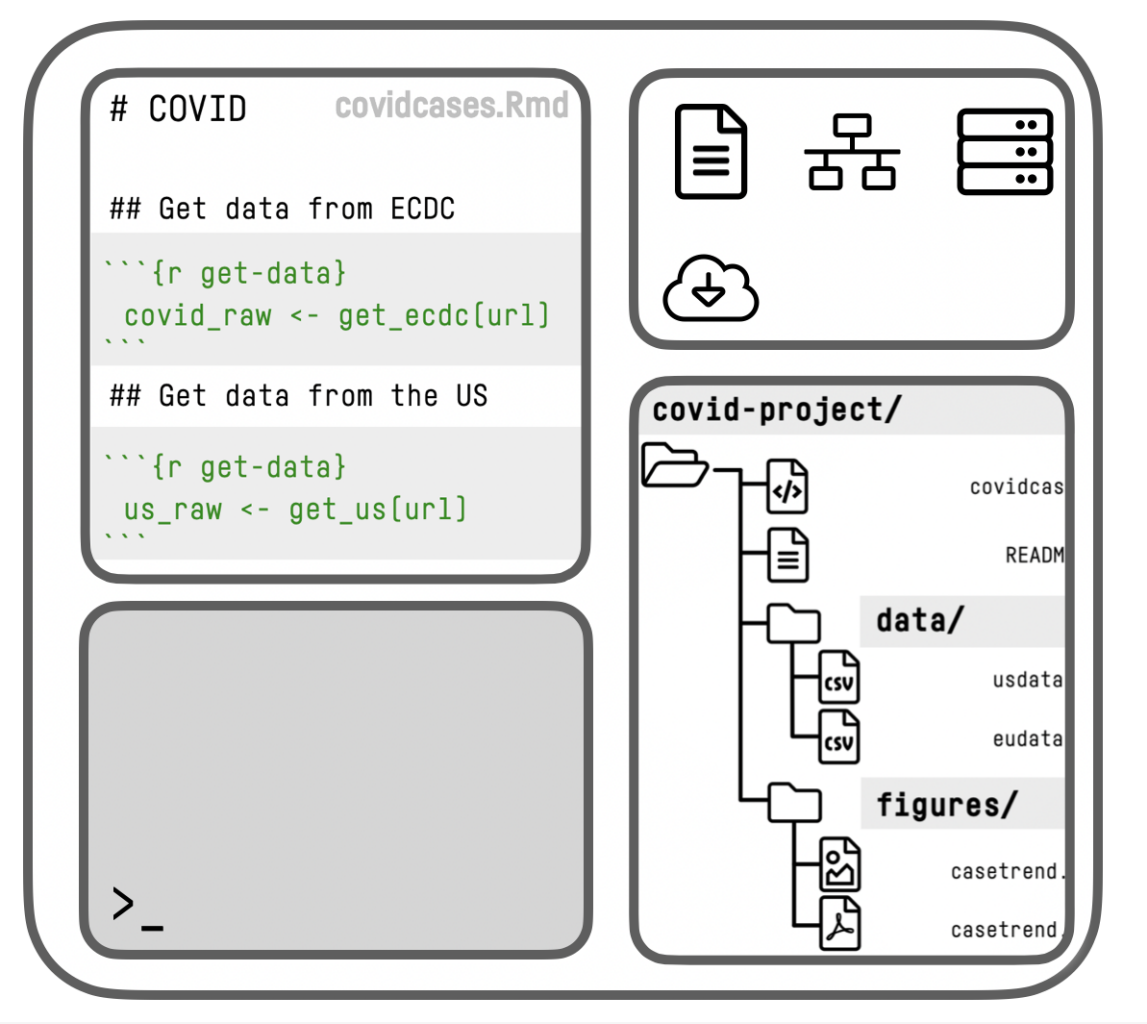
RStudio is an IDE for R

A kitchen is an IDE for Meals

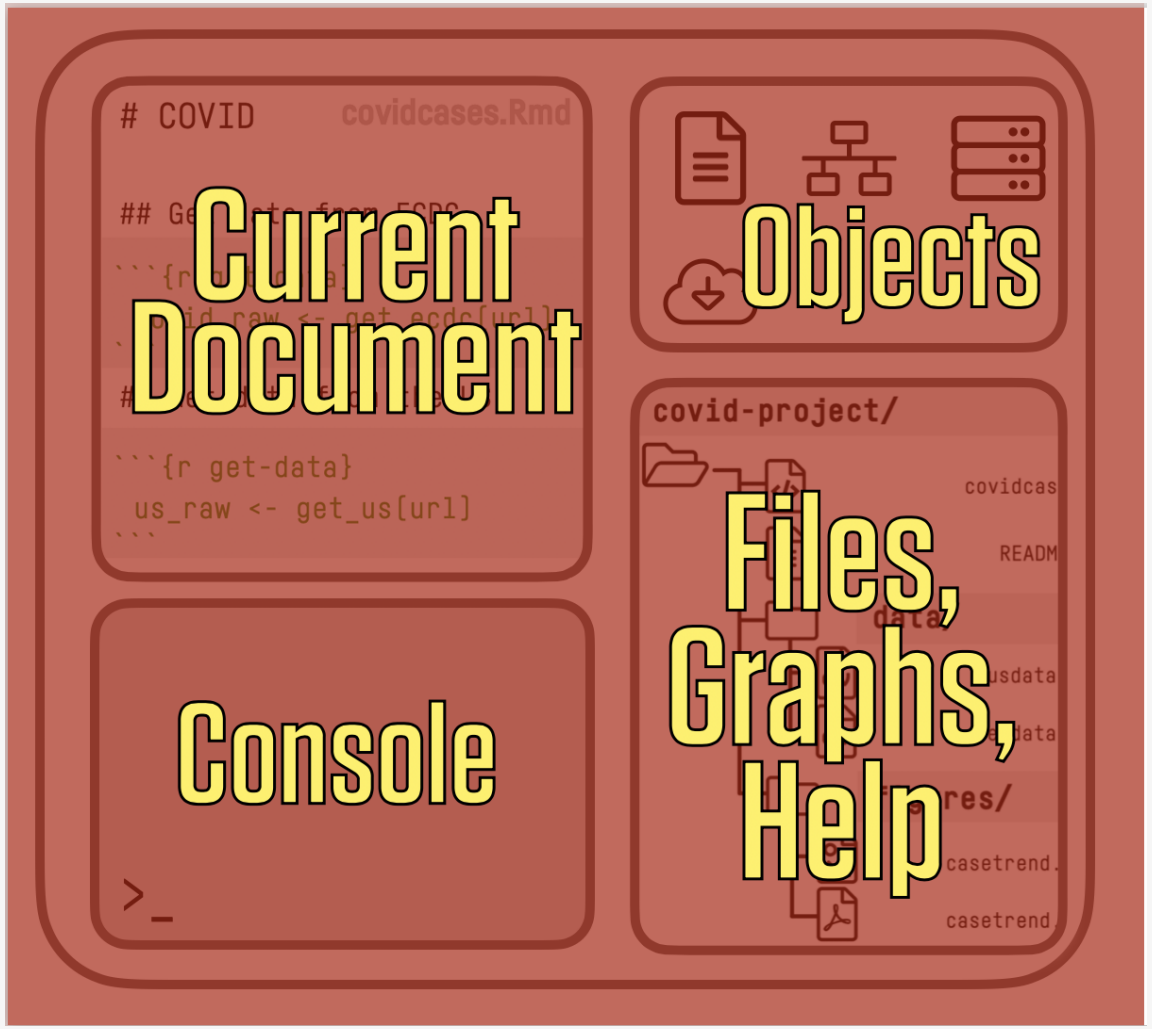
R and RStudio



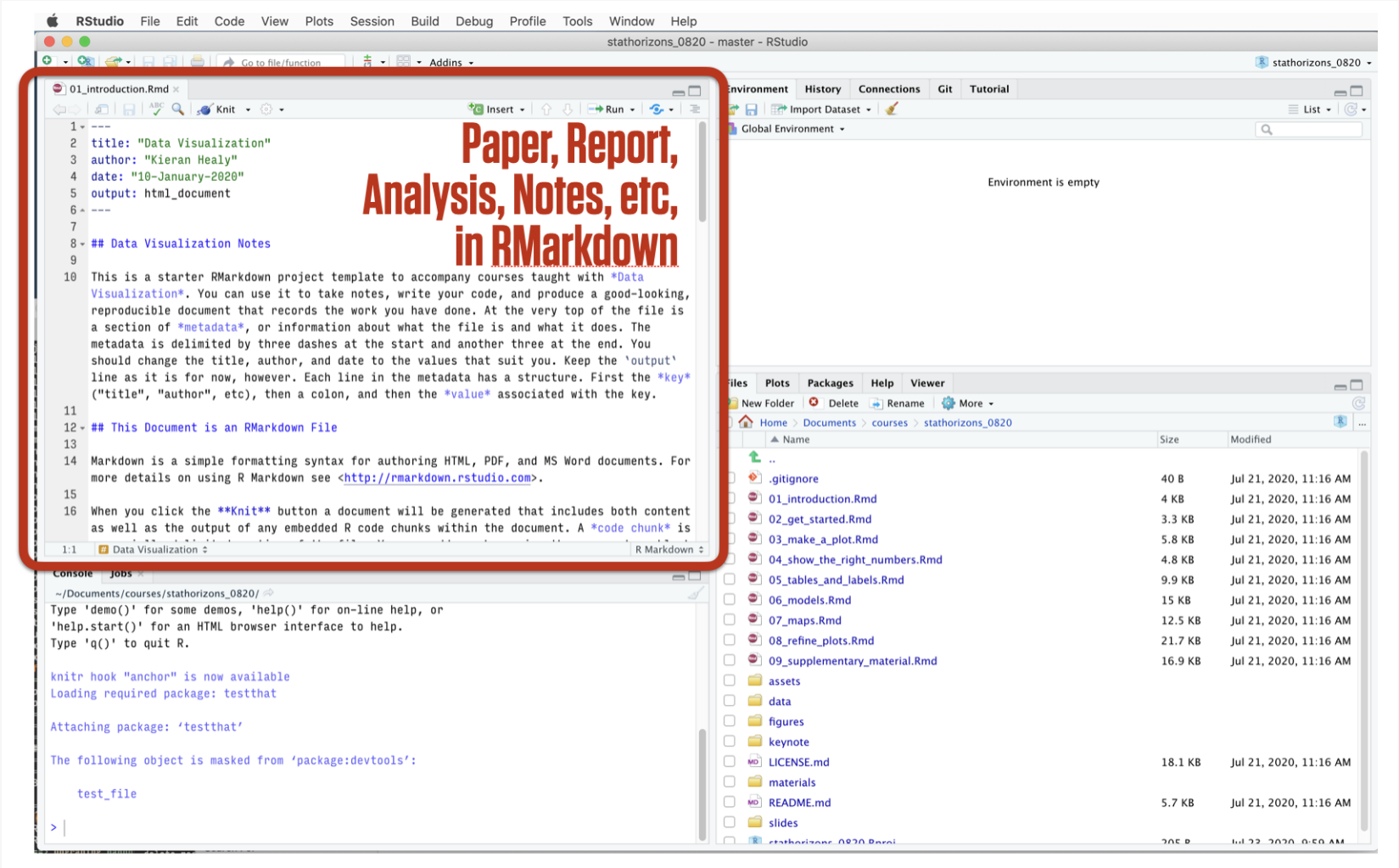
R and RStudio



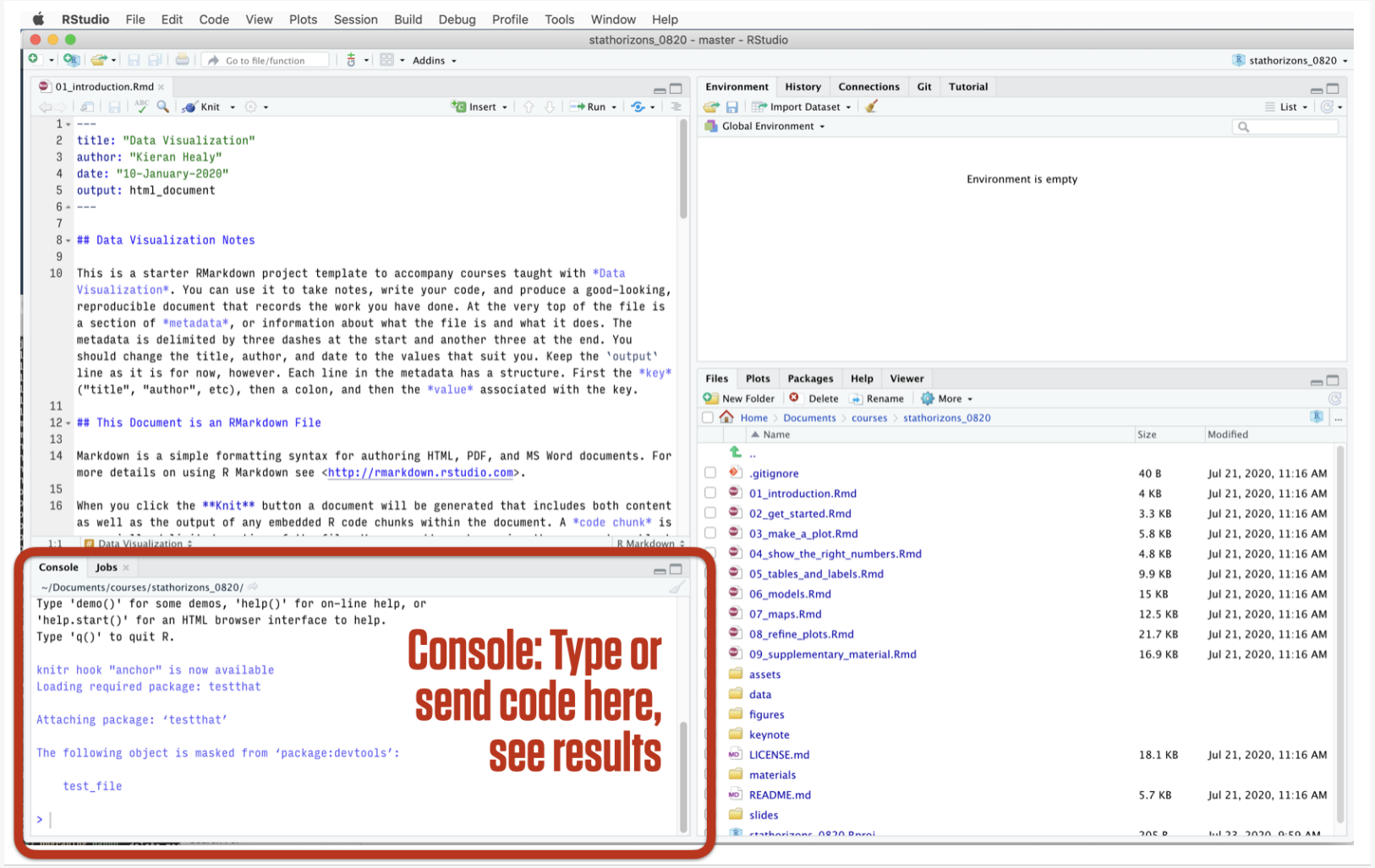
R & RStudio


Current Document
Objects
Files, Graphs, Help
Console
COVID
covidcases.Rmd
```{r get-data}
us_raw <- get_us[url]
```
covid-project/
>_

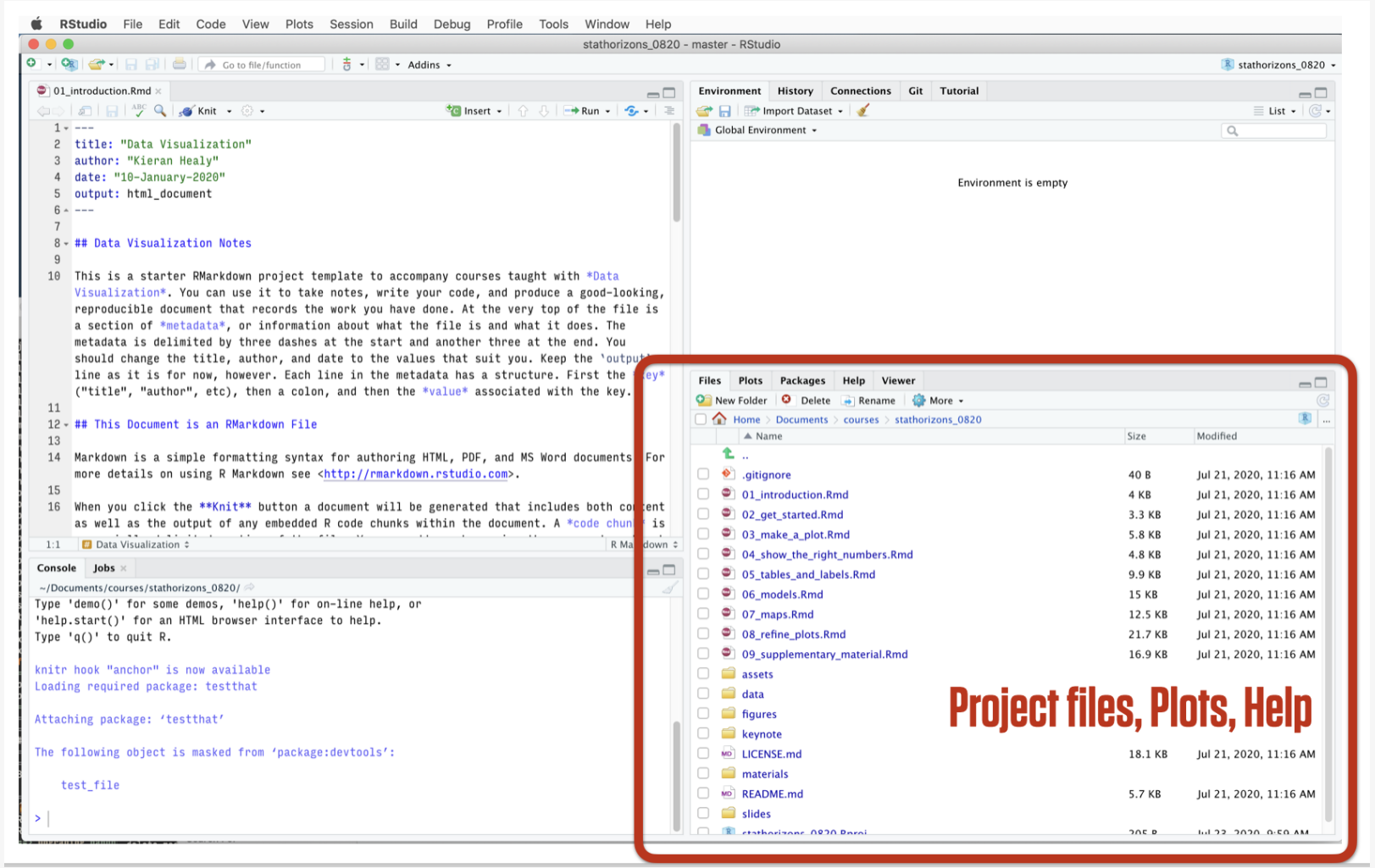
R & RStudio


RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
stathorizons_0820 - master - RStudio
01_introduction.Rmd
Paper, Report, Analysis, Notes, etc, in RMarkdown
Environment History Connections Git Tutorial
Import Dataset
Global Environment
Environment is empty
Files Plots Packages Help Viewer
New Folder Delete Rename More
Home Documents courses stathorizons_0820
Name Size Modified
.gitignore 40 B Jul 21, 2020, 11:16 AM
01_introduction.Rmd 4 KB Jul 21, 2020, 11:16 AM
02_get_started.Rmd 3.3 KB Jul 21, 2020, 11:16 AM
03_make_a_plot.Rmd 5.8 KB Jul 21, 2020, 11:16 AM
04_show_the_right_numbers.Rmd 4.8 KB Jul 21, 2020, 11:16 AM
05_tables_and_labels.Rmd 9.9 KB Jul 21, 2020, 11:16 AM
06_models.Rmd 15 KB Jul 21, 2020, 11:16 AM
07_maps.Rmd 12.5 KB Jul 21, 2020, 11:16 AM
08_refine_plots.Rmd 21.7 KB Jul 21, 2020, 11:16 AM
09_supplementary_material.Rmd 16.9 KB Jul 21, 2020, 11:16 AM
assets
data
figures
keynote
LICENSE.md 18.1 KB Jul 21, 2020, 11:16 AM
materials
README.md 5.7 KB Jul 21, 2020, 11:16 AM
slides
Console Jobs
~/Documents/courses/stathorizons_0820/
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
knitr hook "anchor" is now available
Loading required package: testthat
Attaching package: 'testthat'
The following object is masked from 'package:devtools':
test_file
Data Visualization
R Markdown

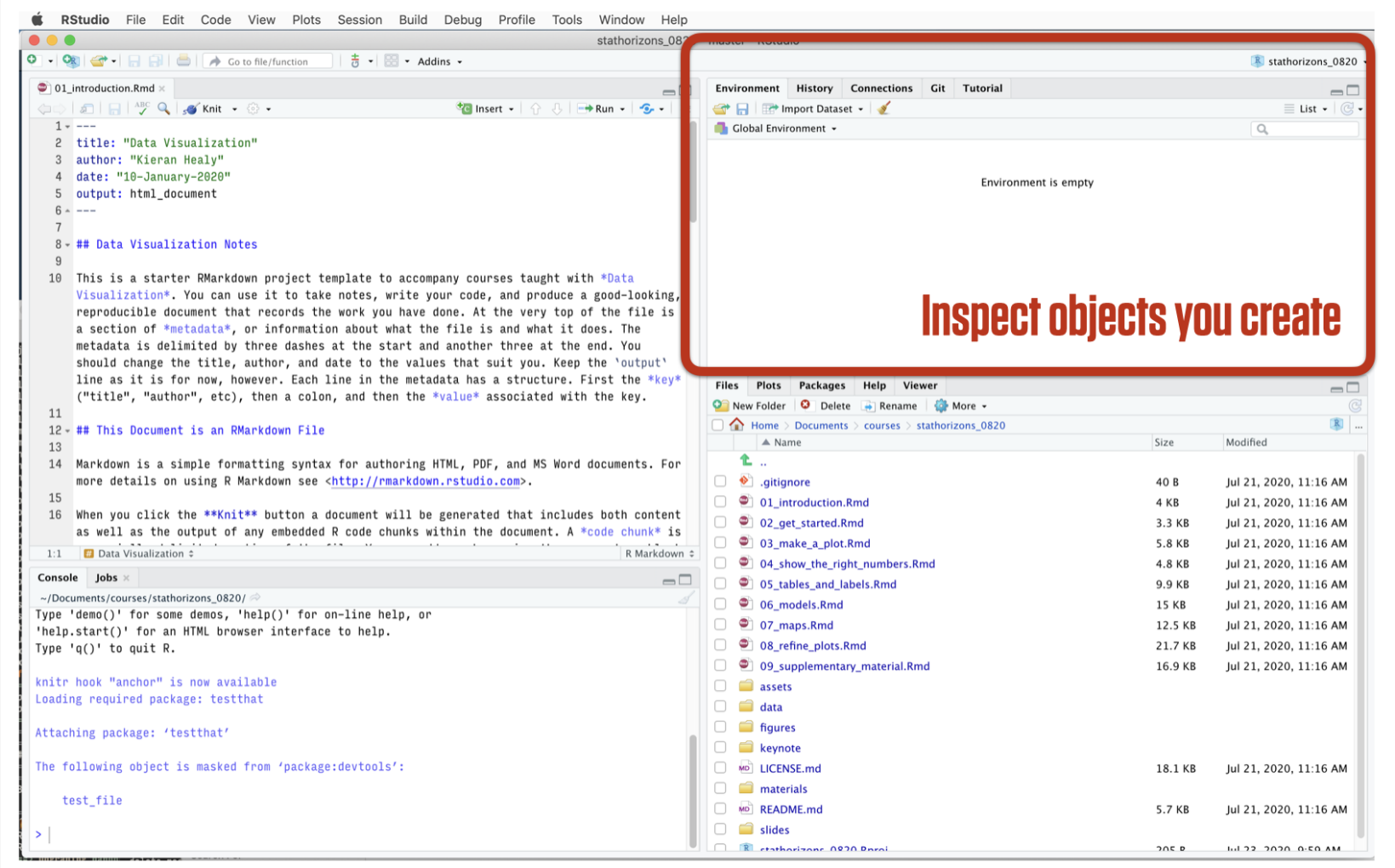
R & RStudio


RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
stathorizons_0820 - master - RStudio
01_introduction.Rmd
Environment History Connections Git Tutorial
Environment is empty
Files Plots Packages Help Viewer
Console Jobs
Console: Type or send code here, see results

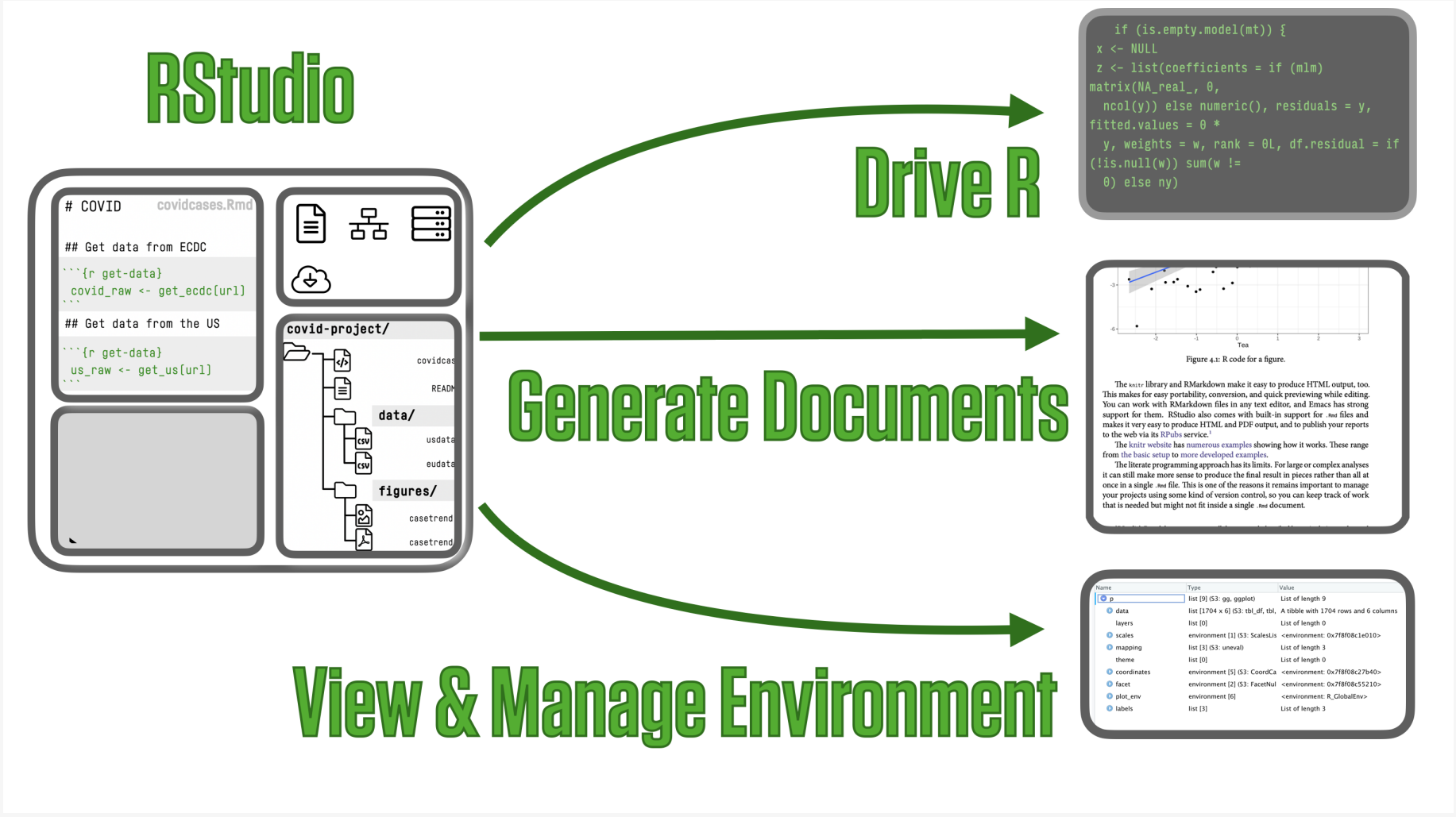
R & RStudio


RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
stathorizons_0820 - master - RStudio
01_introduction.Rmd
Environment History Connections Git Tutorial
Environment is empty
Files Plots Packages Help Viewer
Console Jobs
Project files, Plots, Help

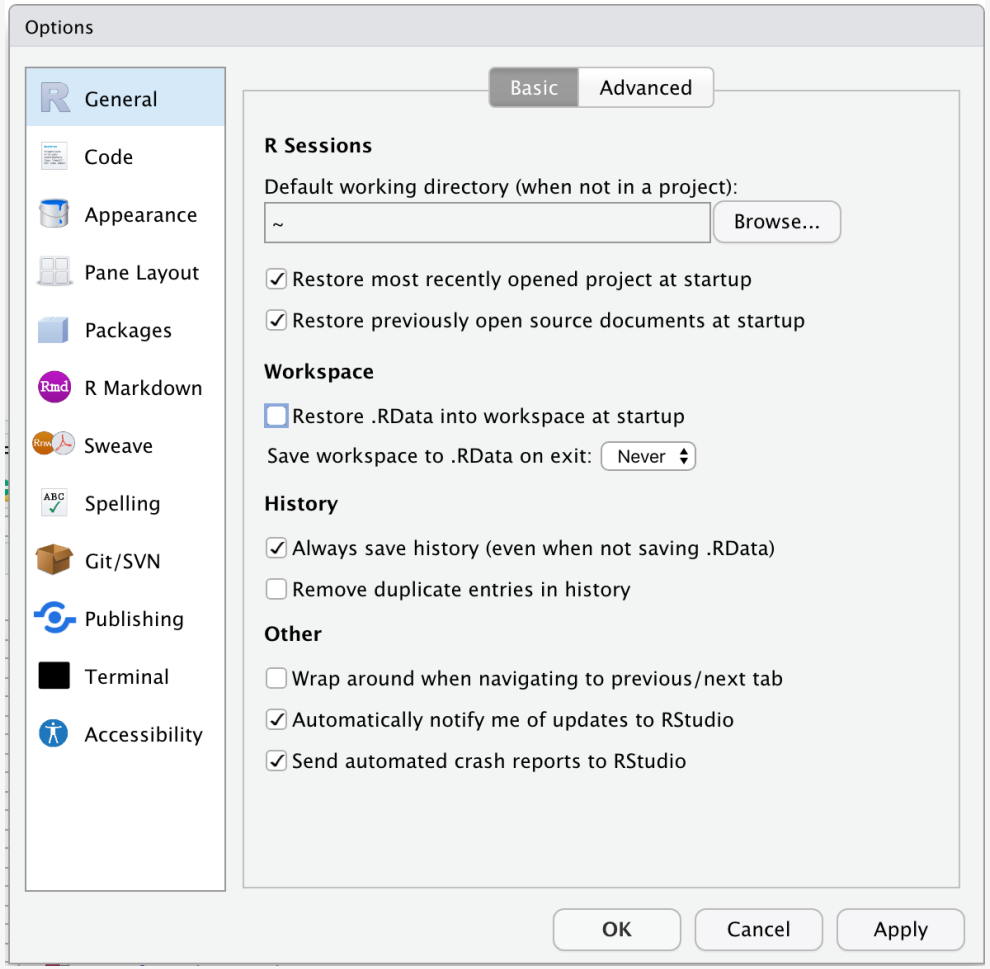
R & RStudio


RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
01_introduction.Rmd
Knit
Insert
Run
1 ---
2 title: "Data Visualization"
3 author: "Kieran Healy"
4 date: "10-January-2020"
5 output: html_document
6 ---
8 ## Data Visualization Notes
10 This is a starter RMarkdown project template to accompany courses taught with *Data Visualization*. You can use it to take notes, write your code, and produce a good-looking, reproducible document that records the work you have done. At the very top of the file is a section of *metadata*, or information about what the file is and what it does. The metadata is delimited by three dashes at the start and another three at the end. You should change the title, author, and date to the values that suit you. Keep the 'output' line as it is for now, however. Each line in the metadata has a structure. First the *key* ("title", "author", etc), then a colon, and then the *value* associated with the key.
12 ## This Document is an RMarkdown File
14 Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.
16 When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. A *code chunk* is
1:1 Data Visualization R Markdown
Console Jobs
~/Documents/courses/stathorizons_0820/
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
knitr hook "anchor" is now available
Loading required package: testthat
Attaching package: 'testthat'
The following object is masked from 'package:devtools':
test_file
>
stathorizons_0820
Environment History Connections Git Tutorial
Import Dataset
List
Global Environment
Environment is empty
Inspect objects you create
Files Plots Packages Help Viewer
New Folder Delete Rename More
Home Documents courses stathorizons_0820
Name Size Modified
.. .gitignore 40 B Jul 21, 2020, 11:16 AM
01_introduction.Rmd 4 KB Jul 21, 2020, 11:16 AM
02_get_started.Rmd 3.3 KB Jul 21, 2020, 11:16 AM
03_make_a_plot.Rmd 5.8 KB Jul 21, 2020, 11:16 AM
04_show_the_right_numbers.Rmd 4.8 KB Jul 21, 2020, 11:16 AM
05_tables_and_labels.Rmd 9.9 KB Jul 21, 2020, 11:16 AM
06_models.Rmd 15 KB Jul 21, 2020, 11:16 AM
07_maps.Rmd 12.5 KB Jul 21, 2020, 11:16 AM
08_refine_plots.Rmd 21.7 KB Jul 21, 2020, 11:16 AM
09_supplementary_material.Rmd 16.9 KB Jul 21, 2020, 11:16 AM
assets
data
figures
keynote
LICENSE.md 18.1 KB Jul 21, 2020, 11:16 AM
materials
README.md 5.7 KB Jul 21, 2020, 11:16 AM
slides

R & RStudio


RStudio
COVID
covidcases.Rmd
Get data from ECDC
```{r get-data}
covid_raw <- get_ecdc[url]
```
Get data from the US
```{r get-data}
us_raw <- get_us[url]
```
covid-project/
covidcas
READM
data/
usdata
eudata
figures/
casetrend
casetrend
Drive R
if (is.empty.model(mt)) {
x <- NULL
z <- list(coefficients = if (mlm)
matrix(NA_real_, 0,
ncol(y)) else numeric(), residuals = y,
fitted.values = 0 *
y, weights = w, rank = 0L, df.residual = if
(!is.null(w)) sum(w !=
0) else ny)
Generate Documents
Figure 4.1: R code for a figure.
The knitr library and RMarkdown make it easy to produce HTML output, too. This makes for easy portability, conversion, and quick previewing while editing. You can work with RMarkdown files in any text editor, and Emacs has strong support for them. RStudio also comes with built-in support for .Rmd files and makes it very easy to produce HTML and PDF output, and to publish your reports to the web via its RPubs service.
The knitr website has numerous examples showing how it works. These range from the basic setup to more developed examples.
The literate programming approach has its limits. For large or complex analyses it can still make more sense to produce the final result in pieces rather than all at once in a single .Rmd file. This is one of the reasons it remains important to manage your projects using some kind of version control, so you can keep track of work that is needed but might not fit inside a single .Rmd document.
View & Manage Environment
Name
Type
Value
p
list [9] (S3: gg, ggplot)
List of length 9
data
list [1704 x 6] (S3: tbl_df, tbl,
A tibble with 1704 rows and 6 columns
layers
list [0]
List of length 0
scales
environment [1] (S3: ScalesLis
<environment: 0x7f8f08c1e010>
mapping
list [3] (S3: uneval)
List of length 3
theme
list [0]
List of length 0
coordinates
environment [5] (S3: CoordCa
<environment: 0x7f8f08c27b40>
facet
environment [2] (S3: FacetNul
<environment: 0x7f8f08c55210>
plot_env
environment [6]
<environment: R_GlobalEnv>
labels
list [3]
List of length 3

Your code is what's real in your project



Consider not showing output inline