Meet the toolkit:
version control and collaboration

IS 407
JooYoung Seo

Learning goals

By the end of the course, you will be able to...

- gain insight from data
- gain insight from data, **reproducibly**
- gain insight from data, reproducibly, **using modern programming tools and techniques**
- gain insight from data, reproducibly **and collaboratively**, using modern programming tools and techniques
- gain insight from data, reproducibly **(with literate programming and version control)** and collaboratively, using modern programming tools and techniques

Reproducible data analysis

Reproducibility checklist

> What does it mean for a data analysis to be "reproducible"?

Near-term goals:

- Are the tables and figures reproducible from the code and data?
- Does the code actually do what you think it does?
- In addition to what was done, is it clear *why* it was done?

Long-term goals:

- Can the code be used for other data?
- Can you extend the code to do other things?

Toolkit for reproducibility

- Scriptability → R
- Literate programming (code, narrative, output in one place) → R Markdown
- Version control → Git / GitHub

Course toolkit

Doing data science

- Programming:
  - R
  - RStudio
  - tidyverse
  - R Markdown
- Version control and collaboration:
  - Git
  - GitHub

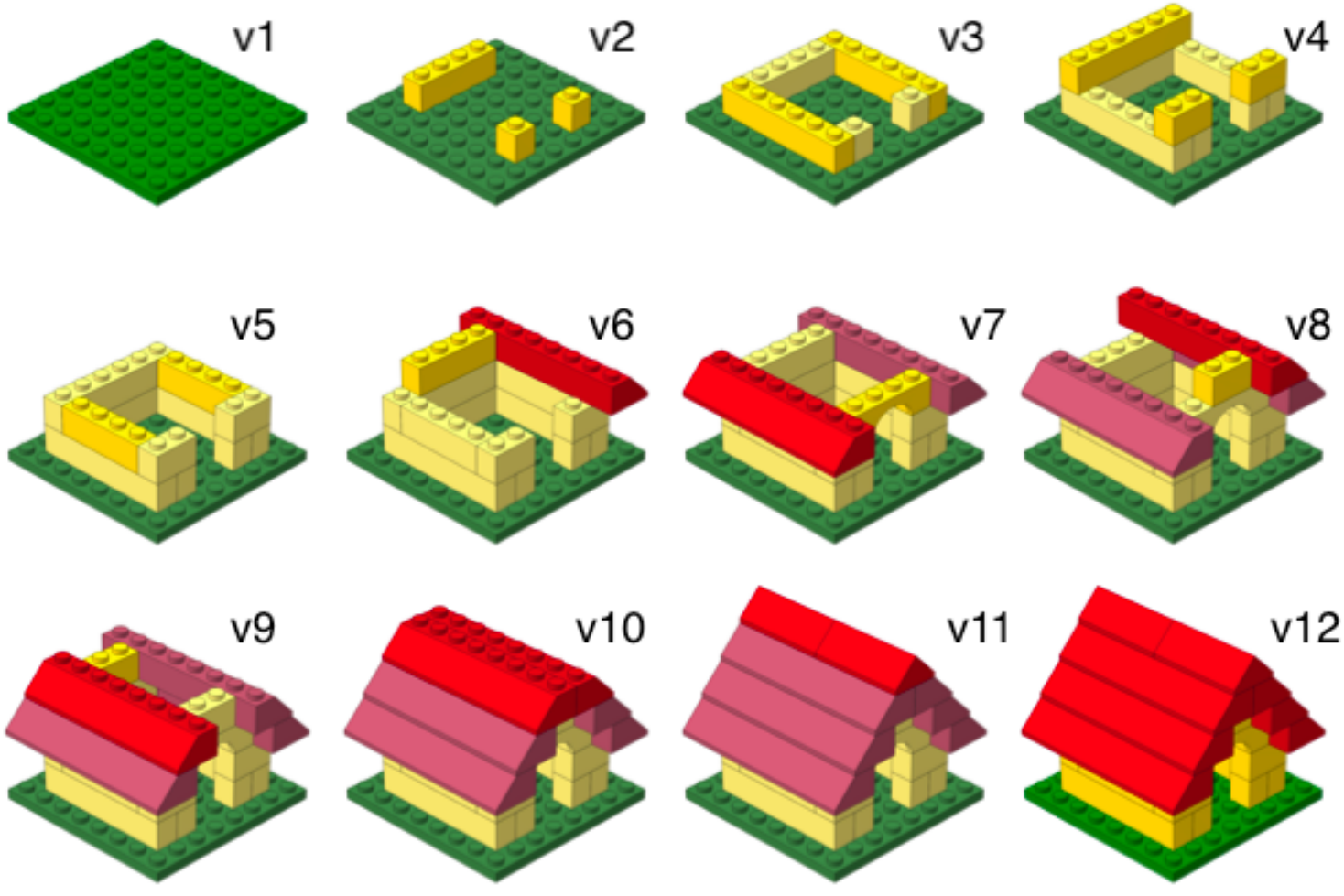
Git and GitHub

Git and GitHub

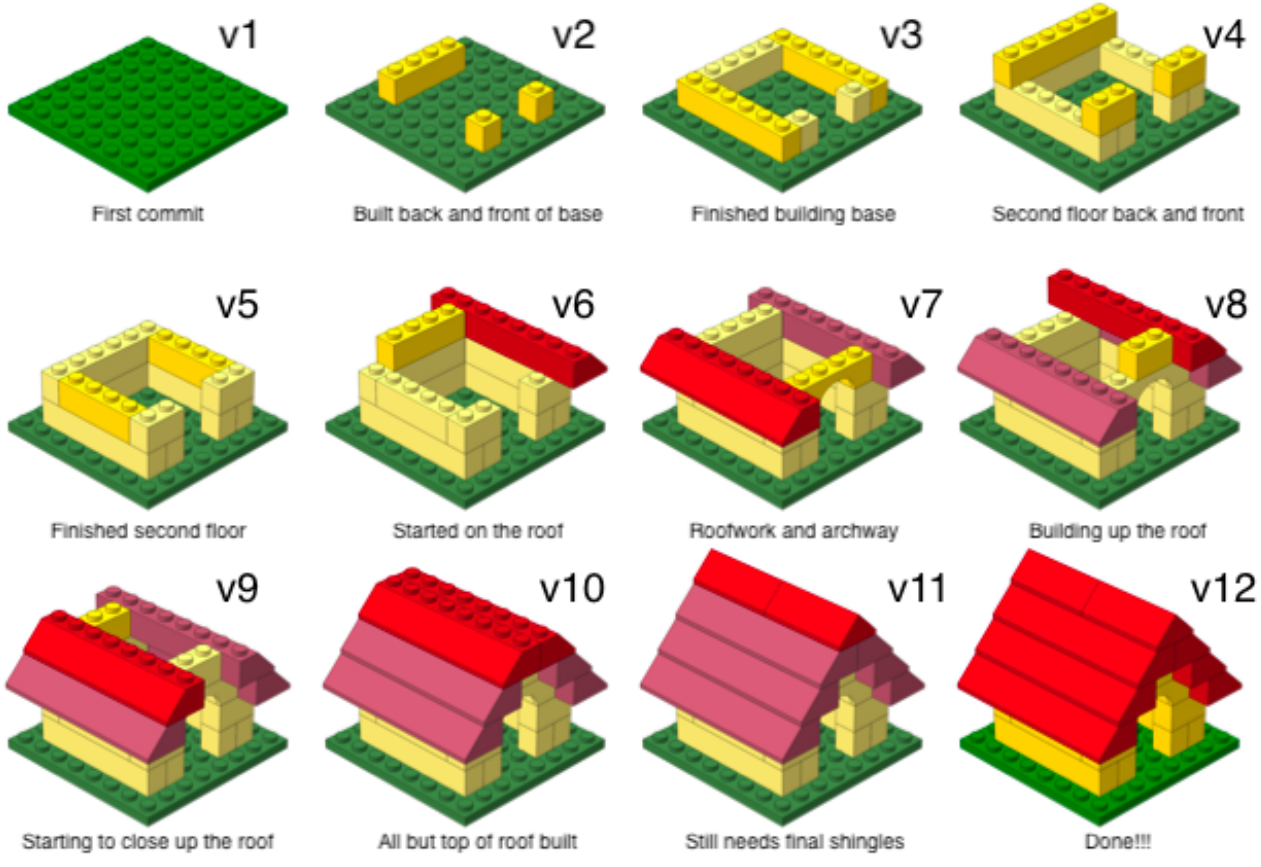
- Git is a version control system -- like “Track Changes” features from Microsoft Word, on steroids
- It's not the only version control system, but it's a very popular one

- GitHub is the home for your Git-based projects on the internet -- like DropBox but much, much better
- We will use GitHub as a platform for web hosting and collaboration (and as our course management system!)

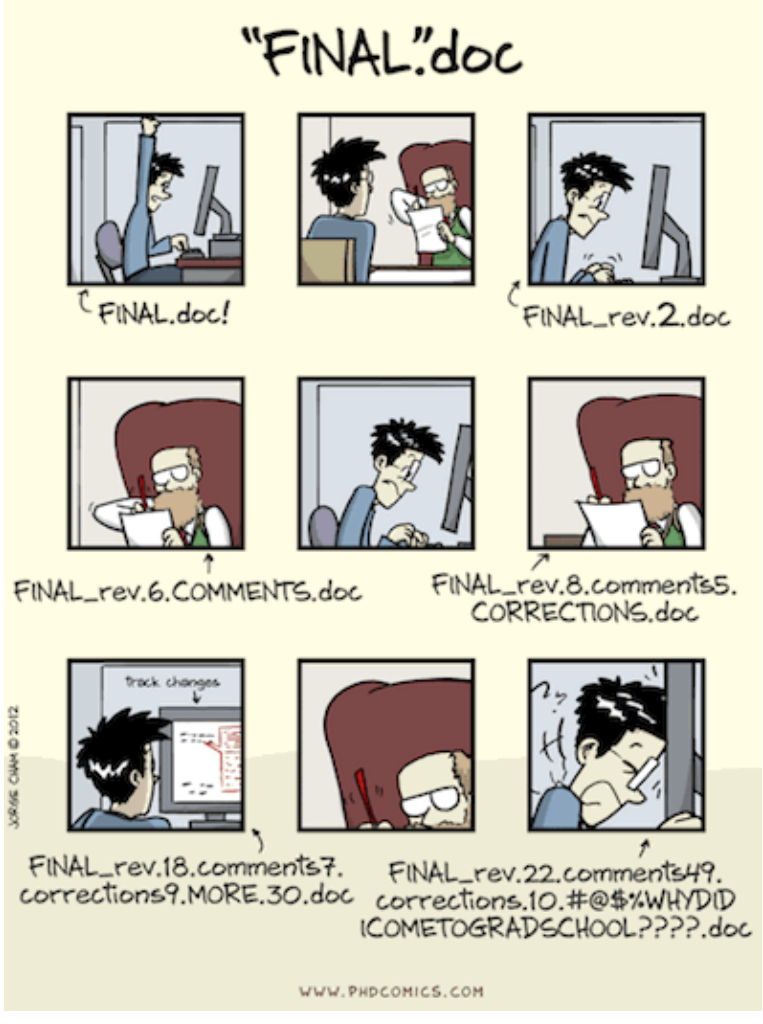
Versioning



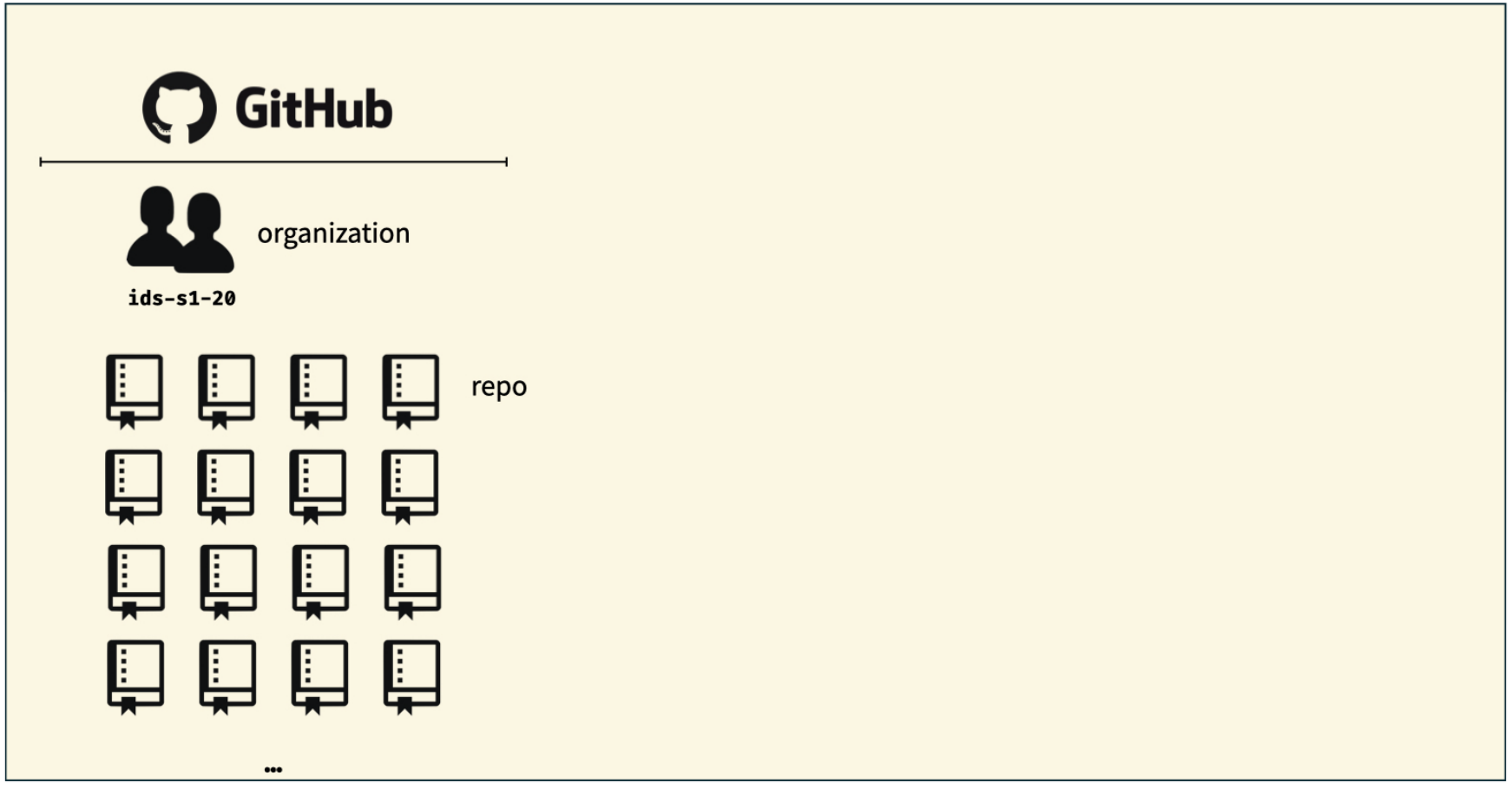
Versioning
with human readable messages



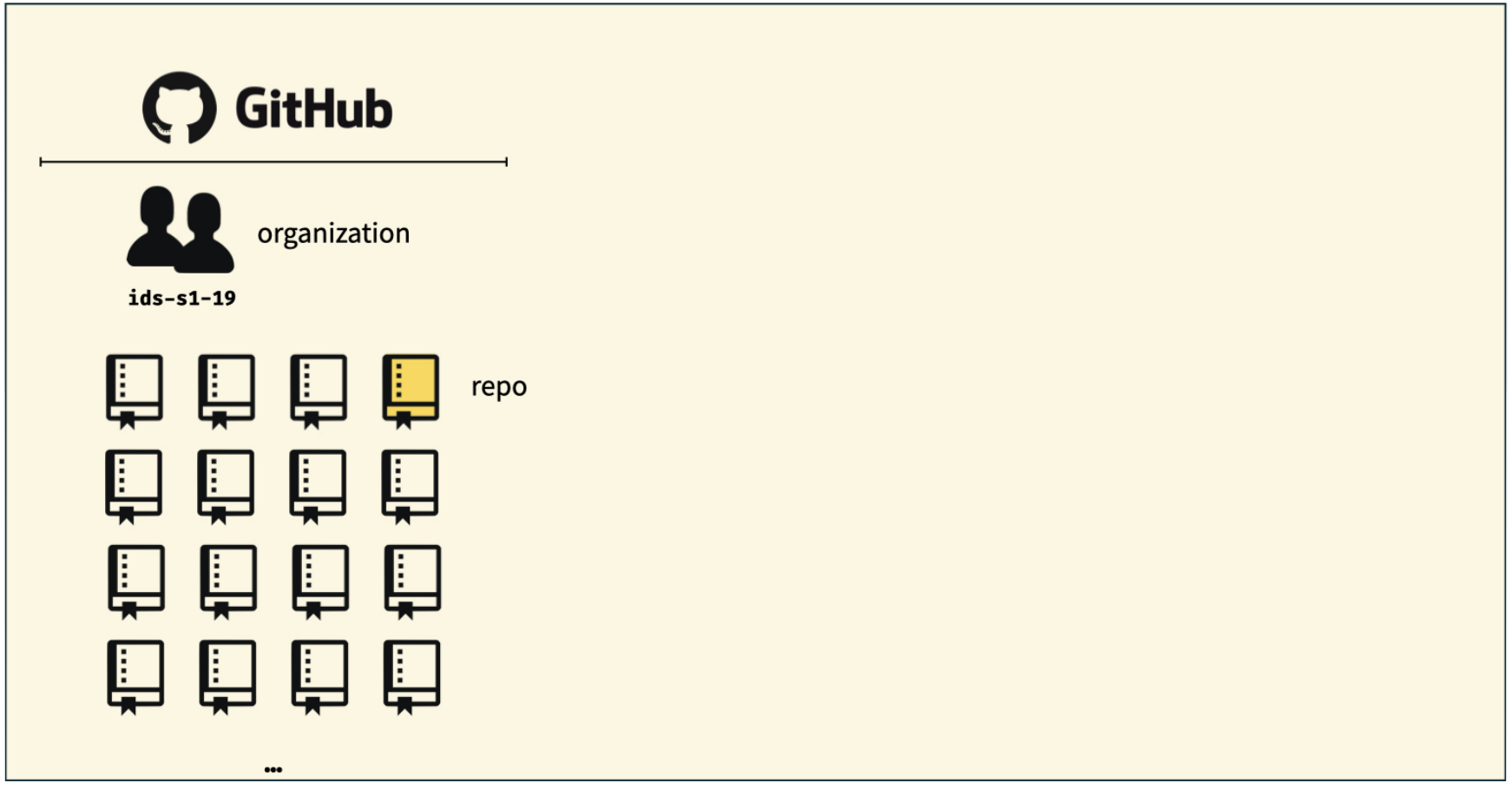
Why do we need version control?



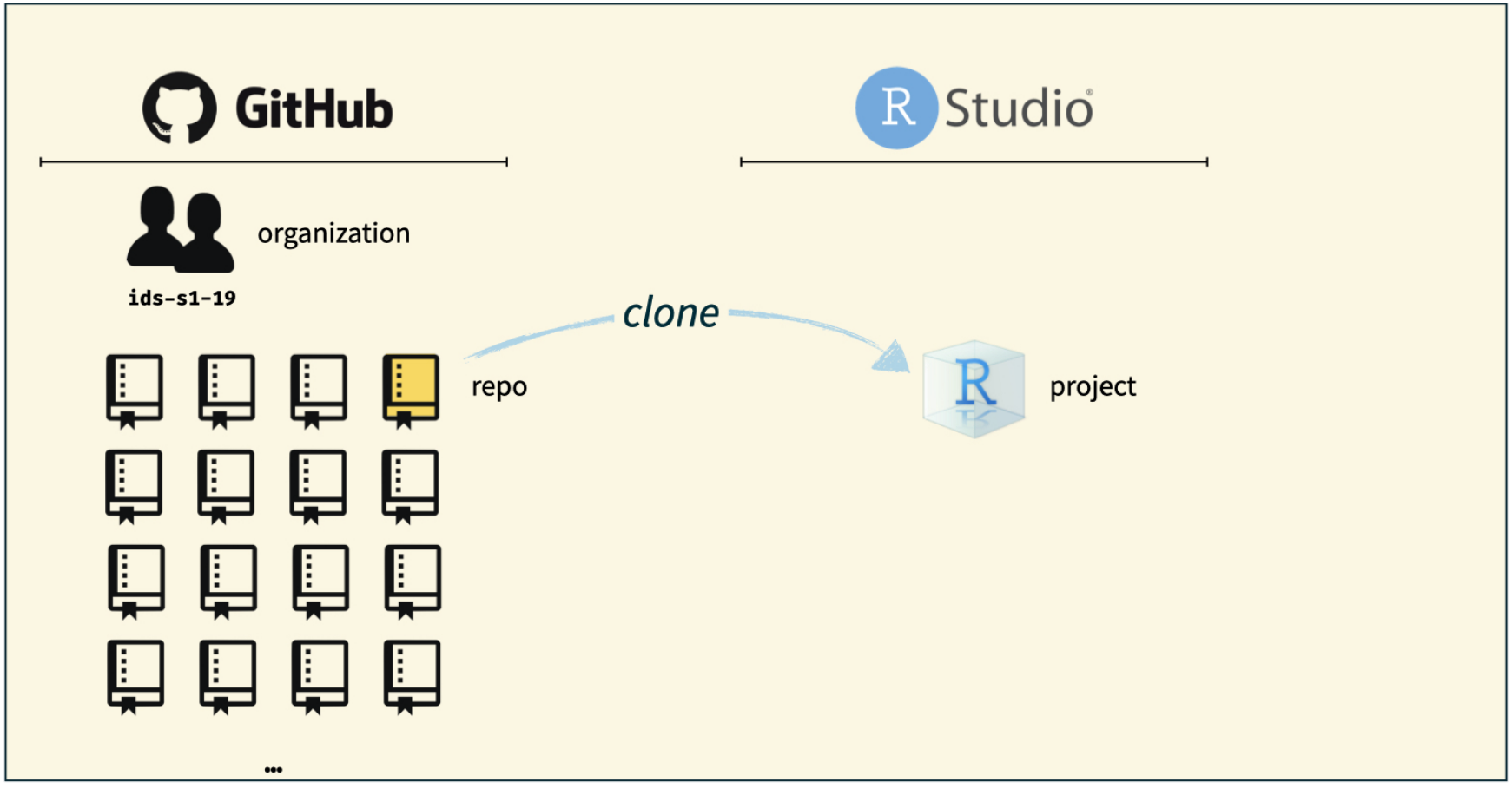
How will we use Git and GitHub?



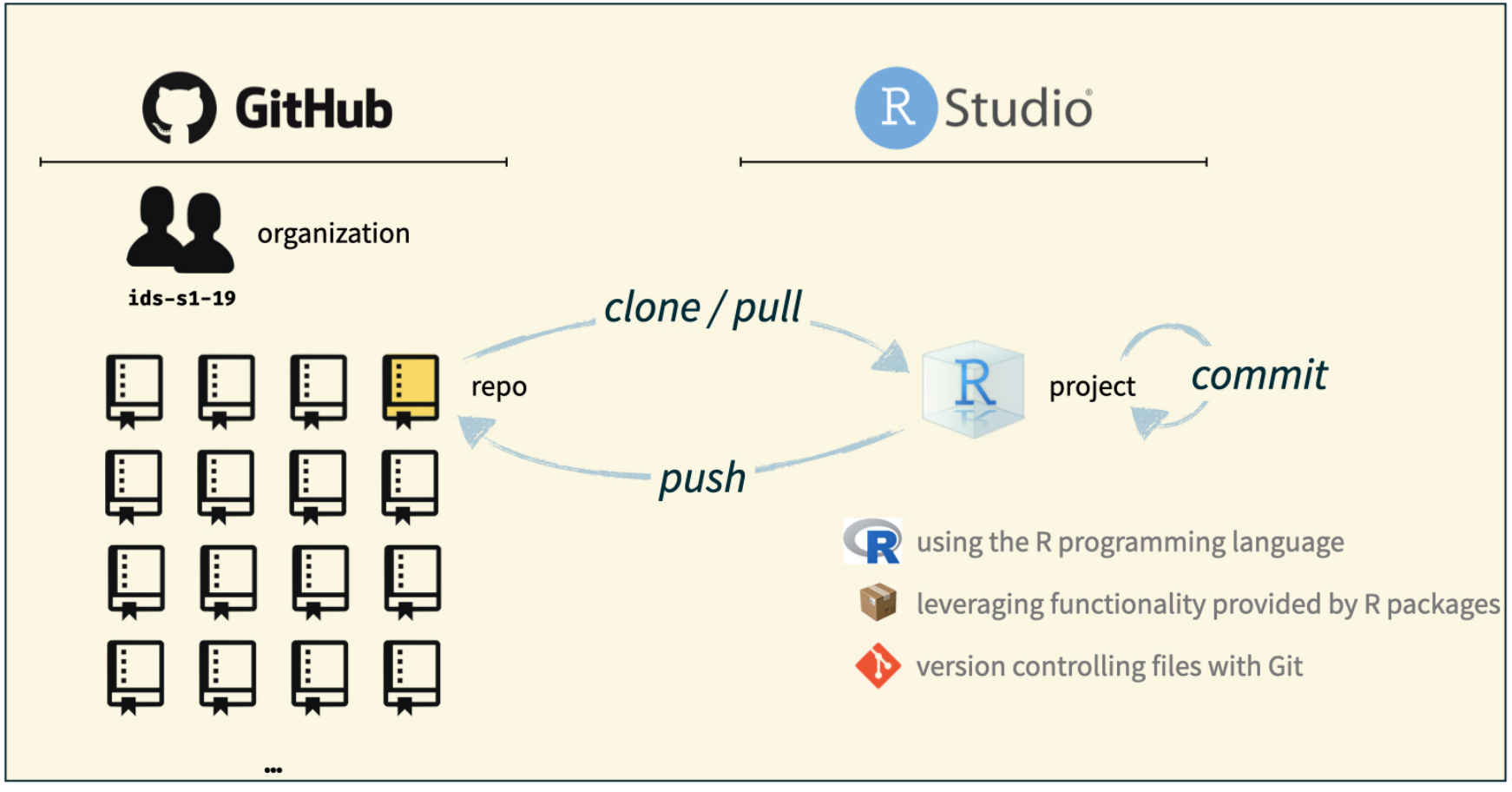
How will we use Git and GitHub?



How will we use Git and GitHub?



How will we use Git and GitHub?



Git and GitHub tips

- There are millions of git commands -- ok, that's an exaggeration, but there are a lot of them -- and very few people know them all. 99% of the time you will use git to add, commit, push, and pull.
- We will be doing Git things and interfacing with GitHub through RStudio, but if you google for help you might come across methods for doing these things in the command line -- skip that and move on to the next resource unless you feel comfortable trying it out.
- There is a great resource for working with git and R: happygitwithr.com. Some of the content in there is beyond the scope of this course, but it's a good place to look for help.

Tour: Git and GitHub

- Create a GitHub account
- Verify your GitHub email
- Adjust your GitHub settings for a more pleasant GitHub experience

Next week...

Work with R, RStudio, Git, and GitHub together![+]

[+]Just like a real data scientist!

Course toolkit

Doing data science

- Programming:
  - R
  - RStudio
  - tidyverse
  - R Markdown
- Version control and collaboration:
  - Git
  - GitHub

R and RStudio

R and RStudio

- R is an open-source statistical **programming language**
- R is also an environment for statistical computing and graphics
- It's easily extensible with *packages*

- RStudio is a convenient interface for R called an **IDE** (integrated development environment), e.g. *"I write R code in the RStudio IDE"*
- RStudio is not a requirement for programming with R, but it's very commonly used by R programmers and data scientists

R packages

- **Packages** are the fundamental units of reproducible R code. They include reusable R functions, the documentation that describes how to use them, and sample data[1]

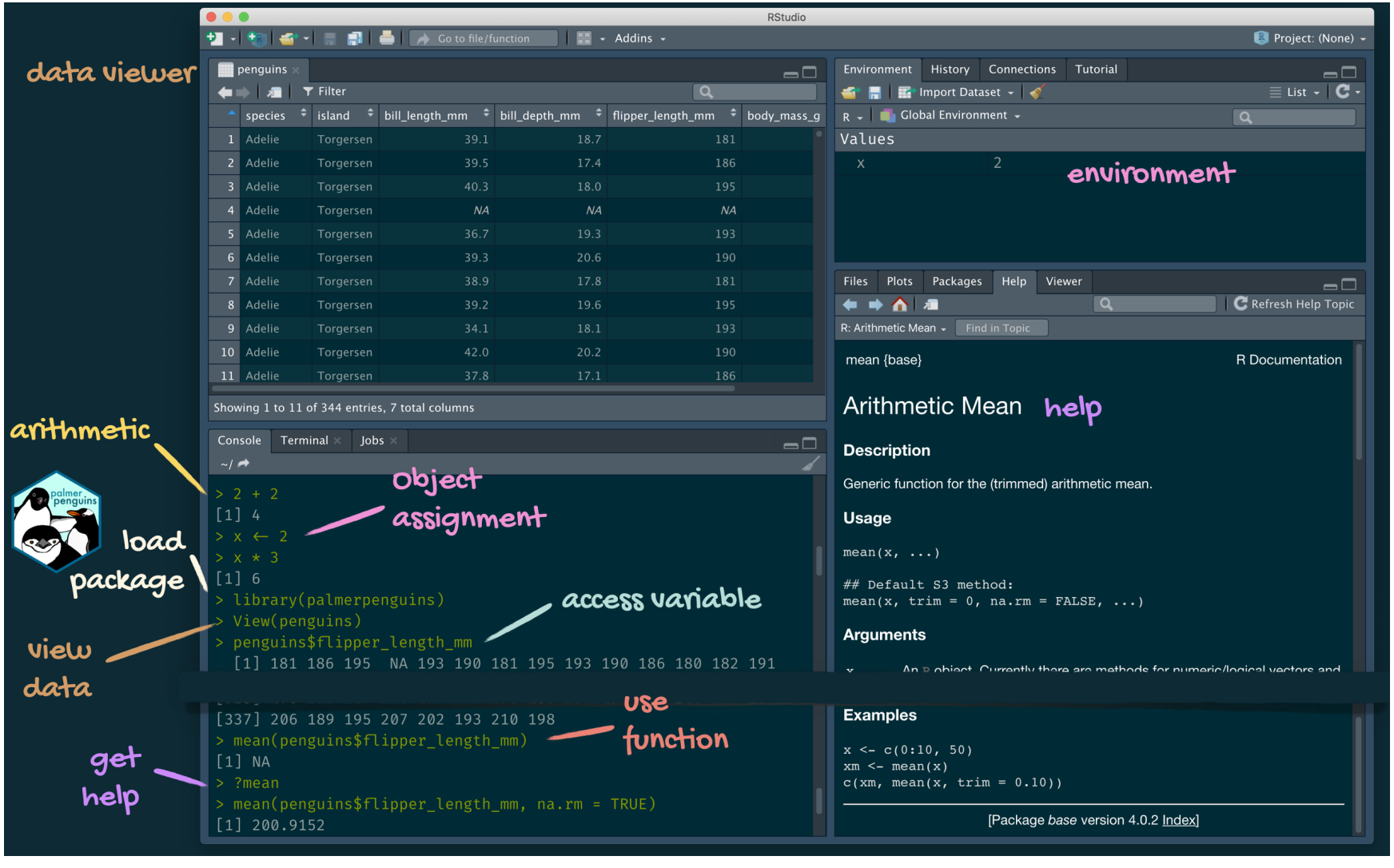
There are over 16,000 R packages available on **CRAN** (the Comprehensive R Archive Network)[2]

- We're going to work with a small (but important) subset of these!

[1] Wickham and Bryan, R Packages.

[2] CRAN contributed packages.

Tour: R and RStudio



A short list (for now) of R essentials

- Functions are (most often) verbs, followed by what they will be applied to in parentheses:

```
verb(object)
eat(food)
do_that(to_this, to_that, with_those)
```

- Packages are installed with the install.packages function and loaded with the library function, once per session:

```
install.packages("package_name")
library(package_name)
```

R essentials (continued)

- Columns (variables) in data frames are accessed with $:

```
dataframe$var_name
```

- Object documentation can be accessed with ?

```
?mean
```

tidyverse

tidyverse.org

- The **tidyverse** is an opinionated collection of R packages designed for data science
- All packages share an underlying philosophy and a common grammar

rmarkdown.rstudio.com

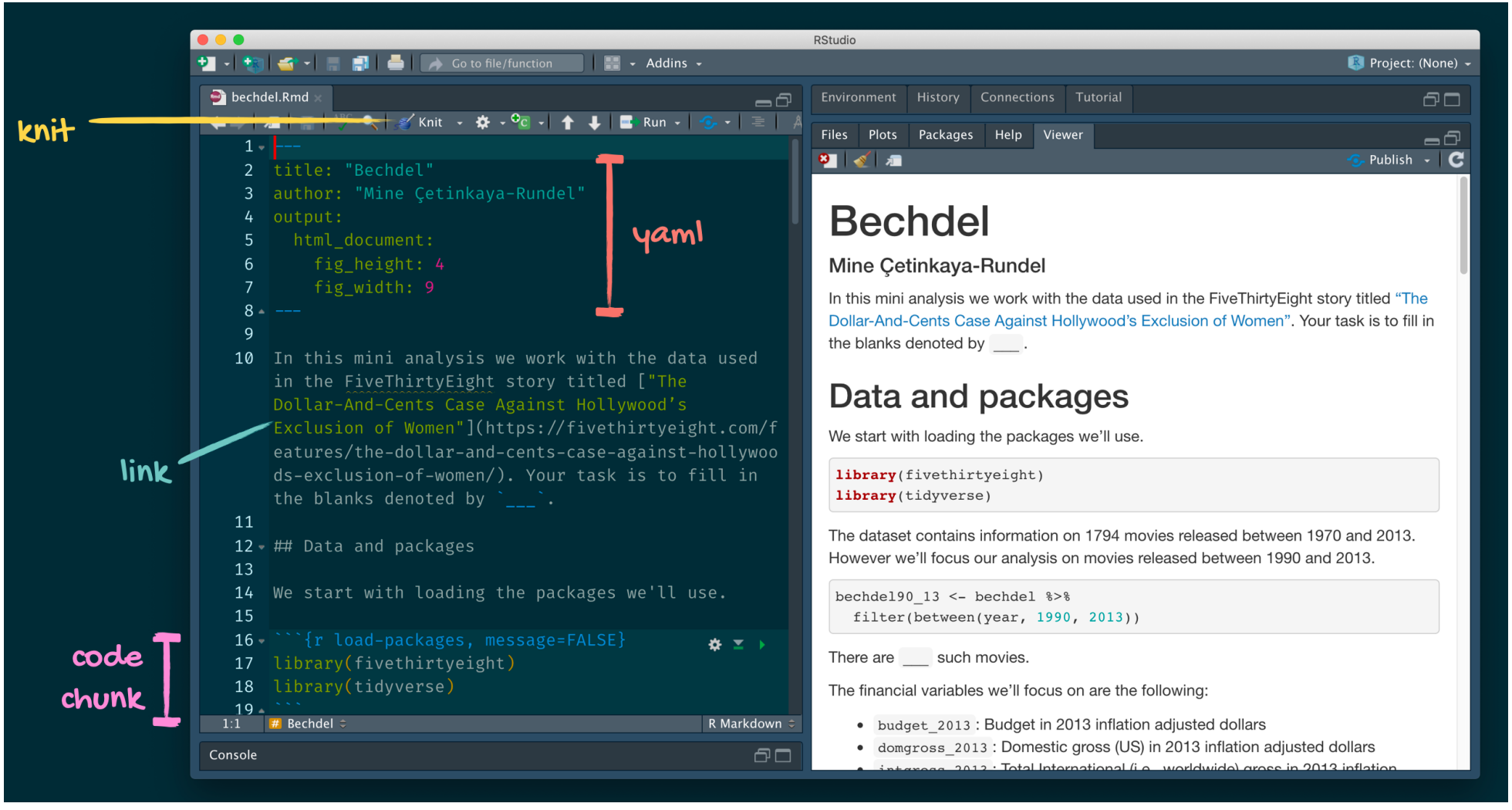
- **rmarkdown** and the various packages that support it enable R users to write their code and prose in reproducible computational documents
- We will generally refer to R Markdown documents (with .Rmd extension), e.g. *"Do this in your R Markdown document"* and rarely discuss loading the rmarkdown package

R Markdown

R Markdown

- Fully reproducible reports -- each time you knit the analysis is ran from the beginning
- Simple markdown syntax for text
- Code goes in chunks, defined by three backticks, narrative goes outside of chunks

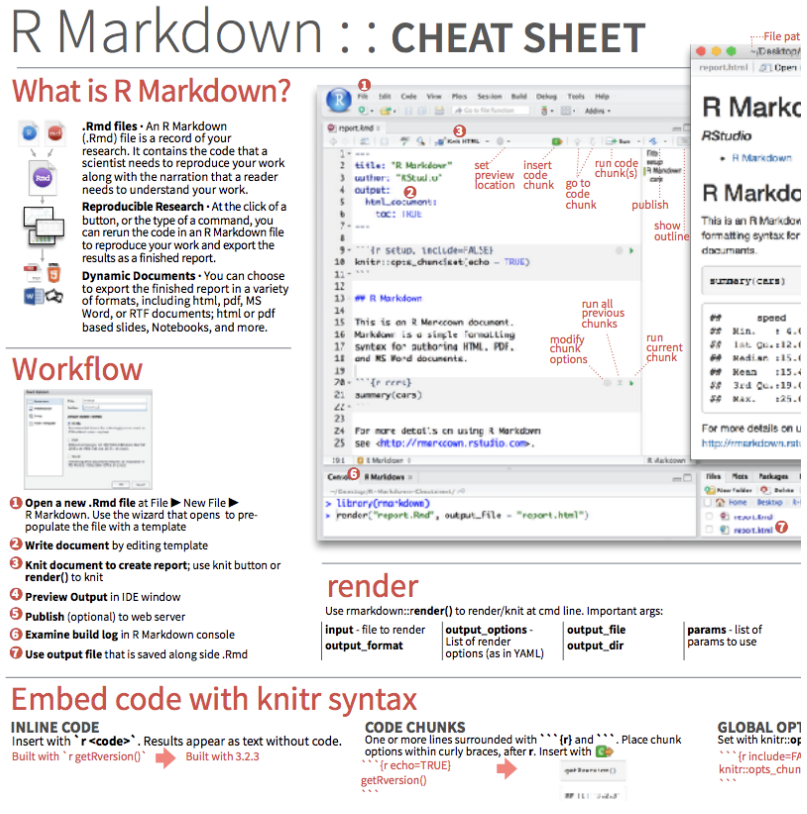
Tour: R Markdown



R Markdown help

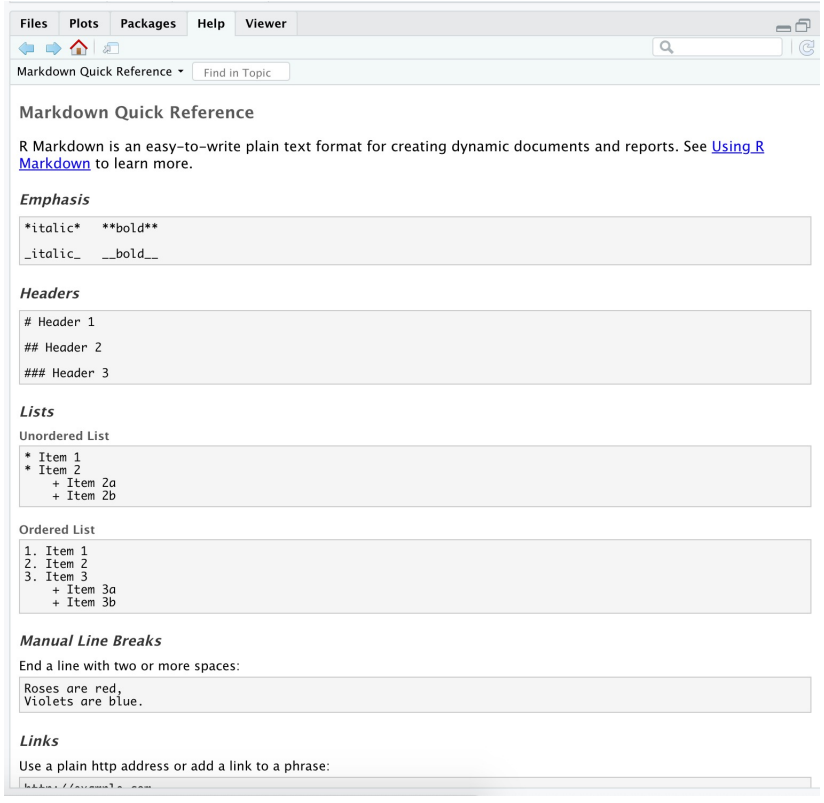
R Markdown Cheat Sheet

Help -> Cheatsheets



Markdown Quick Reference

Help -> Markdown Quick Reference



How will we use R Markdown?

- Every assignment / report / project / etc. is an R Markdown document
- You'll always have a template R Markdown document to start with
- The amount of scaffolding in the template will decrease over the semester

Your turn: Cumulative deaths from COVID-19

covid.Rmd