Data management and analysis workflow

YSE 550a Fall 2020 Workshop

N.R. Sommer

https://bit.ly/2SQnNnQ

Workshop goals

1. Convince you why you should have good data management and analysis for reproducibility
2. Highlight best practices for data management
3. Highlight best practices for writing data analysis code
4. Give you the tools to get started on your R + GitHub journey

What is reproducibility?

"*Reproducibility* refers to the ability of a researcher to duplicate the results of a prior study using the same materials as were used by the original investigator….

That is, a second researcher might use the same raw data to build the same analysis files and implement the same statistical analysis in an attempt to yield the same results…. **Reproducibility is a minimum necessary condition for a finding to be believable and informative**" – National Science Foundation

What is replicability?

- “Replicability refers to the ability of a researcher to duplicate the results of a prior study if the same procedures are followed but new data are collected.” – National Science Foundation

“The Reproducibility Crisis” refers to….

- 1. Absence of replicated studies in published literature
- 2. Widespread failure to reproduce results of prior published studies
- 3. Lack of transparency and completeness in the reporting of methods, data, and analysis in publication

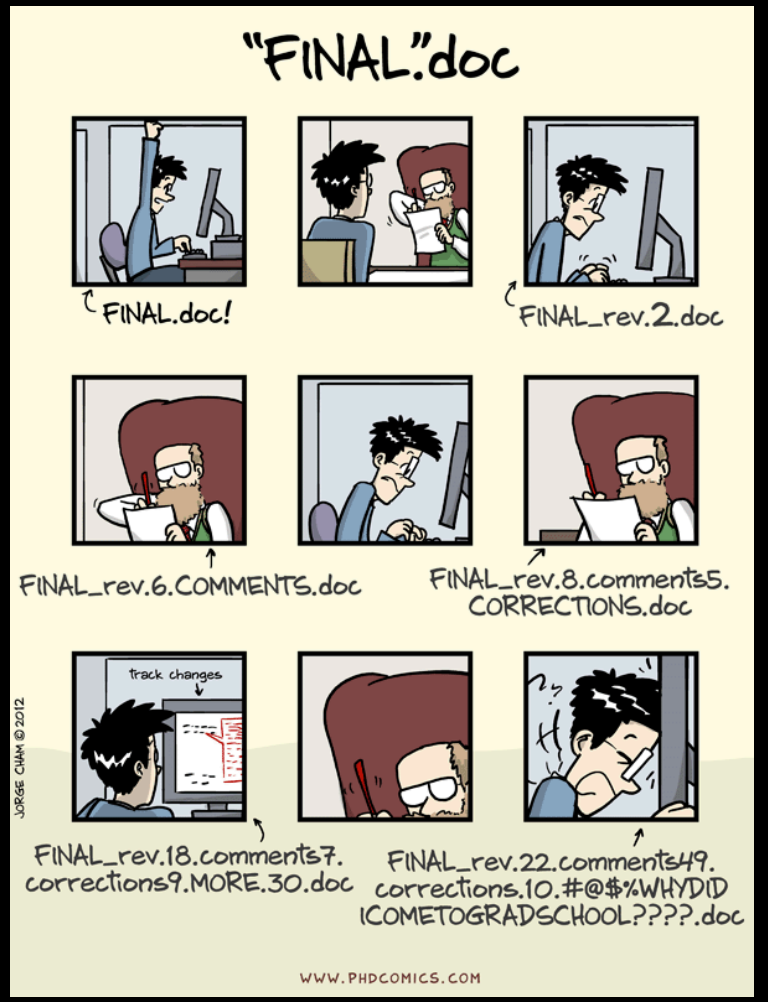
- Something you can’t change – would require a complete overhaul of the academy
- Something you can try to change – publish your null & negative results
- Something you <u>will</u> change – publish your complete methods, data and analysis

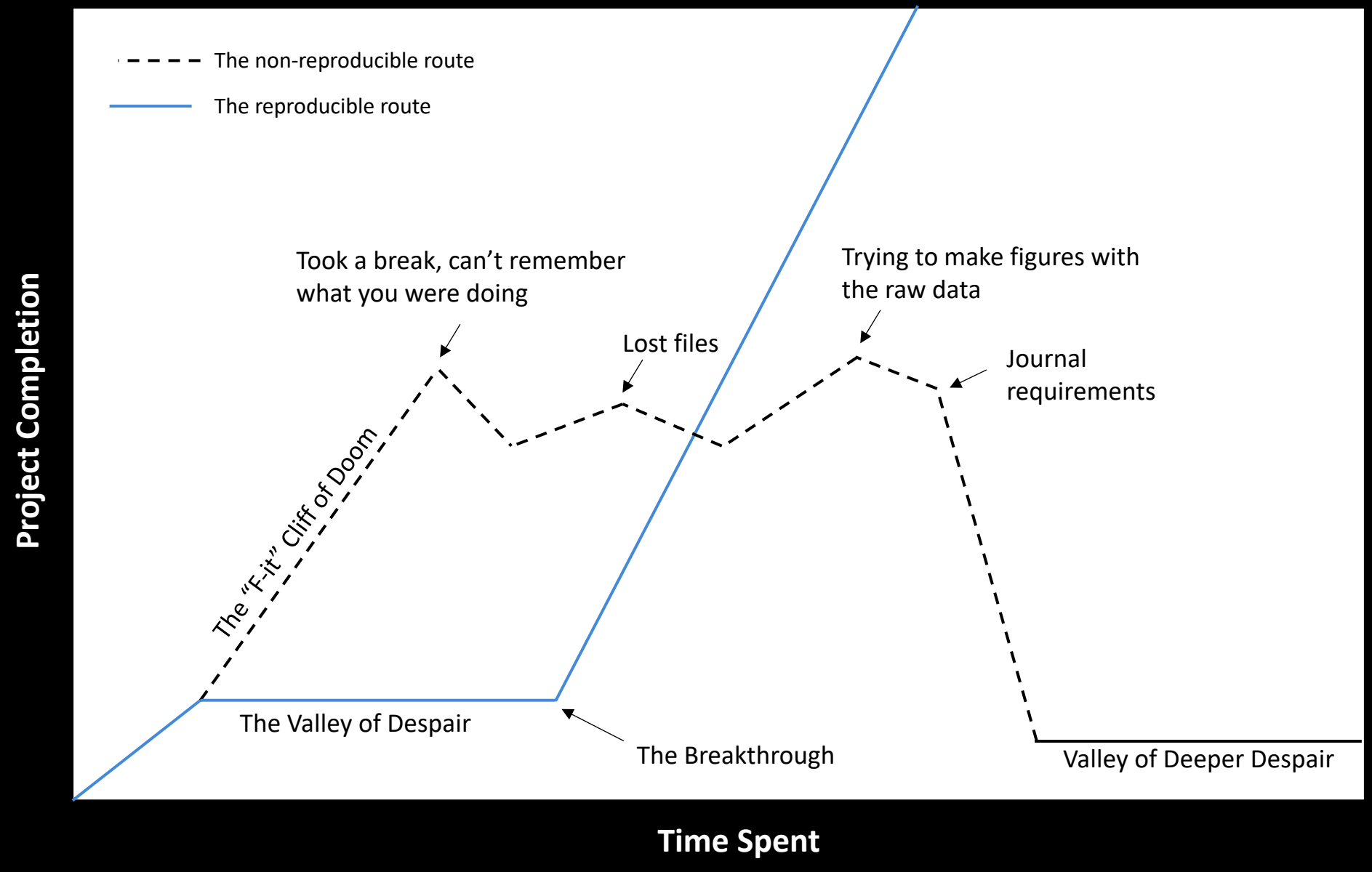
Basic requirements for reproducibility

- A detailed methods section on how the raw data was collected
- The raw data
- Details about the raw data processing
  - In the methods section
  - Within your code
- Details about the analysis
  - In the method section
  - Within your code

Why have a reproducible workflow?

1. Makes your science reproducible
2. Makes your life easier (in the long term)




The non-reproducible route
The reproducible route
Project Completion
Took a break, can't remember what you were doing
Lost files
Trying to make figures with the raw data
Journal requirements
The "F-it" Cliff of Doom
The Valley of Despair
The Breakthrough
Valley of Deeper Despair
Time Spent

Workshop Goals

1. ✓ Convince you why you should have good data management and analysis for reproducibility
   → Good for science
   → Good for you
2. Highlight best practices for data management

What is “real” data?

- Real data = raw data
- Completely unprocessed
- Collected from a primary source

Organization: spreadsheets vs raw data

Spreadsheets

- Functions, formatting, multiple sheets, hidden columns, sorting
- No guarantee that future versions of the software can read older files
- Inconsistent data types within columns

File types:

.xlsm

.xlsx

Organization: spreadsheets vs raw data

Spreadsheets

- Functions, formatting, multiple sheets, hidden columns, sorting
- No guarantee that future versions of the software can read older files
- Inconsistent data types within columns

File types:

.xlsm

.xlsx

Raw files

- “The data” as collected
- If in sheet format, organized into columns of a single data type
- Preserved for future versions of software

File types:

.txt

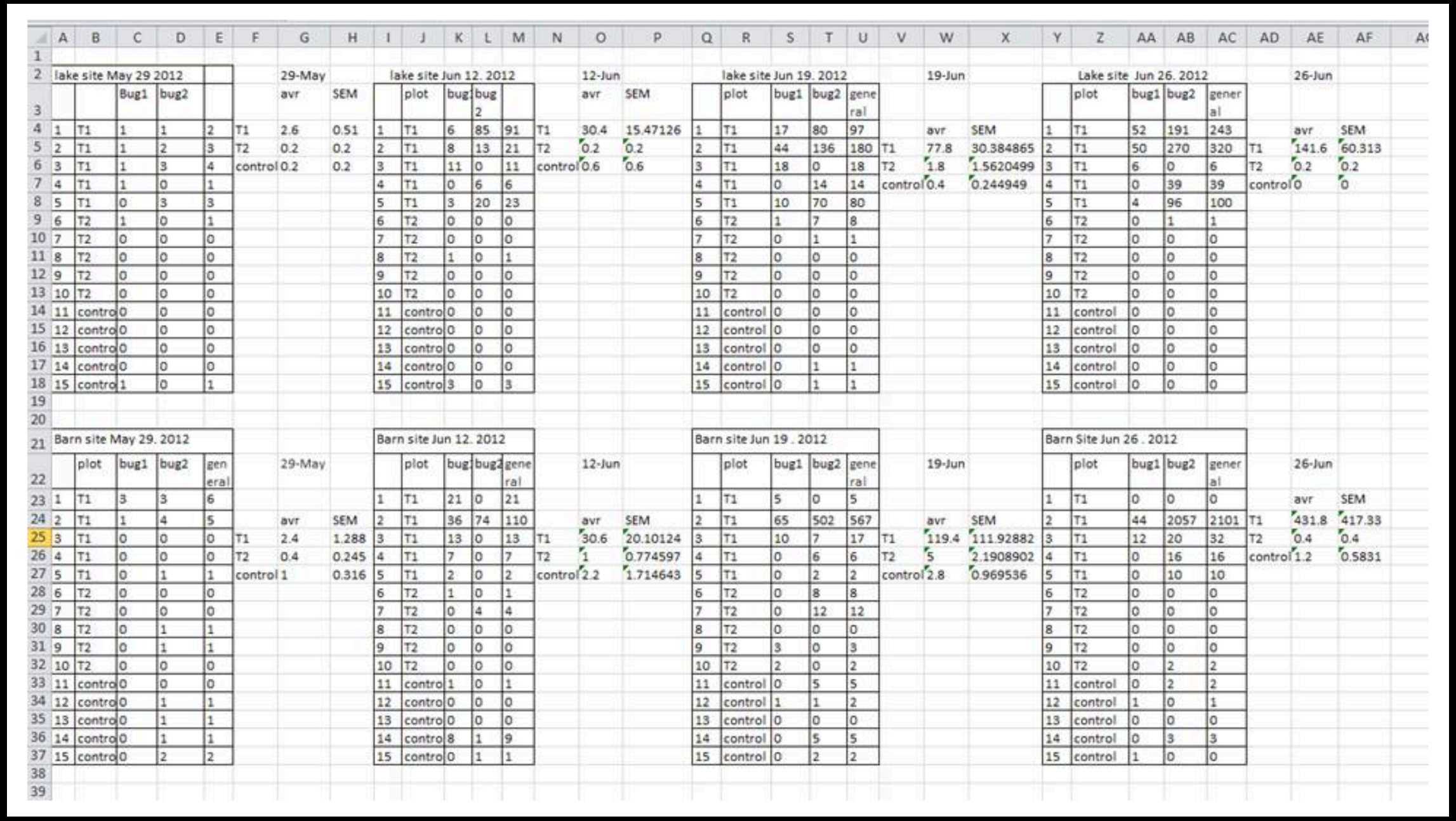
.csv

+ many others for still images, shapefiles, audio and video


EuSpRIG
European Spreadsheet
Risks Interest Group

Best practices for organizing raw data

- Column = the variable
- Row = the observation
- One cell → one item



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF
1																																
2	lake site May 29 2012						29-May		lake site Jun 12. 2012						12-Jun		lake site Jun 19. 2012						19-Jun		Lake site Jun 26. 2012						26-Jun	
3			Bug1	bug2			avr	SEM		plot	bug1	bug2			avr	SEM		plot	bug1	bug2	general					plot	bug1	bug2	general			
4	1	T1	1	1	2	T1	2.6	0.51	1	T1	6	85	91	T1	30.4	15.47126	1	T1	17	80	97		avr	SEM	1	T1	52	191	243		avr	SEM
5	2	T1	1	2	3	T2	0.2	0.2	2	T1	8	13	21	T2	0.2	0.2	2	T1	44	136	180	T1	77.8	30.384865	2	T1	50	270	320	T1	141.6	60.313
6	3	T1	1	3	4	control	0.2	0.2	3	T1	11	0	11	control	0.6	0.6	3	T1	18	0	18	T2	1.8	1.5620499	3	T1	6	0	6	T2	0.2	0.2
7	4	T1	1	0	1				4	T1	0	6	6				4	T1	0	14	14	control	0.4	0.244949	4	T1	0	39	39	control	0	0
8	5	T1	0	3	3				5	T1	3	20	23				5	T1	10	70	80				5	T1	4	96	100			
9	6	T2	1	0	1				6	T2	0	0	0				6	T2	1	7	8				6	T2	0	1	1			
10	7	T2	0	0	0				7	T2	0	0	0				7	T2	0	1	1				7	T2	0	0	0			
11	8	T2	0	0	0				8	T2	1	0	1				8	T2	0	0	0				8	T2	0	0	0			
12	9	T2	0	0	0				9	T2	0	0	0				9	T2	0	0	0				9	T2	0	0	0			
13	10	T2	0	0	0				10	T2	0	0	0				10	T2	0	0	0				10	T2	0	0	0			
14	11	contro	0	0	0				11	contro	0	0	0				11	control	0	0	0				11	control	0	0	0			
15	12	contro	0	0	0				12	contro	0	0	0				12	control	0	0	0				12	control	0	0	0			
16	13	contro	0	0	0				13	contro	0	0	0				13	control	0	0	0				13	control	0	0	0			
17	14	contro	0	0	0				14	contro	0	0	0				14	control	0	1	1				14	control	0	0	0			
18	15	contro	1	0	1				15	contro	3	0	3				15	control	0	1	1				15	control	0	0	0			
19																																
20																																
21	Barn site May 29. 2012								Barn site Jun 12. 2012								Barn site Jun 19 . 2012								Barn Site Jun 26 . 2012							
22		plot	bug1	bug2	general		29-May			plot	bug1	bug2	general		12-Jun			plot	bug1	bug2	general		19-Jun			plot	bug1	bug2	general		26-Jun	
23	1	T1	3	3	6				1	T1	21	0	21				1	T1	5	0	5				1	T1	0	0	0		avr	SEM
24	2	T1	1	4	5		avr	SEM	2	T1	36	74	110		avr	SEM	2	T1	65	502	567		avr	SEM	2	T1	44	2057	2101	T1	431.8	417.33
25	3	T1	0	0	0	T1	2.4	1.288	3	T1	13	0	13	T1	30.6	20.10124	3	T1	10	7	17	T1	119.4	111.92882	3	T1	12	20	32	T2	0.4	0.4
26	4	T1	0	0	0	T2	0.4	0.245	4	T1	7	0	7	T2	1	0.774597	4	T1	0	6	6	T2	5	2.1908902	4	T1	0	16	16	control	1.2	0.5831
27	5	T1	0	1	1	control	1	0.316	5	T1	2	0	2	control	2.2	1.714643	5	T1	0	2	2	control	2.8	0.969536	5	T1	0	10	10			
28	6	T2	0	0	0				6	T2	1	0	1				6	T2	0	8	8				6	T2	0	0	0			
29	7	T2	0	0	0				7	T2	0	4	4				7	T2	0	12	12				7	T2	0	0	0			
30	8	T2	0	1	1				8	T2	0	0	0				8	T2	0	0	0				8	T2	0	0	0			
31	9	T2	0	1	1				9	T2	0	0	0				9	T2	3	0	3				9	T2	0	0	0			
32	10	T2	0	0	0				10	T2	0	0	0				10	T2	2	0	2				10	T2	0	2	2			
33	11	contro	0	0	0				11	contro	1	0	1				11	control	0	5	5				11	control	0	2	2			
34	12	contro	0	1	1				12	contro	0	0	0				12	control	1	1	2				12	control	1	0	1			
35	13	contro	0	1	1				13	contro	0	0	0				13	control	0	0	0				13	control	0	0	0			
36	14	contro	0	1	1				14	contro	8	1	9				14	control	0	5	5				14	control	0	3	3			
37	15	contro	0	2	2				15	contro	0	1	1				15	control	0	2	2				15	control	1	0	0			
38																																
39																																

Best practices for organizing raw data

- Column = the variable
- Row = the observation
- One cell → one item
- Do not leave cells empty
  - Consider using a code to indicate *why* the cell is empty
    - No data = NA
    - To be collected later = CL

Table 1. Commonly used null values, limitations, compatibility with common software and a recommendation regarding whether or not it is a good option. Null values are indicated as compatible with specific software if they work consistently and correctly with that software. For example, the null value "NULL" works correctly for certain applications in R, but does not work in others, so it is not presented in the table as R compatible.

Null values	Problems	Compatibility	Recommendation
0	Indistinguishable from a true zero		Never use
Blank	Hard to distinguish values that are missing from those overlooked on entry. Hard to distinguish blanks from spaces, which behave differently.	R, Python, SQL	Best option
-999, 999	Not recognized as null by many programs without user input. Can be inadvertently entered into calculations.		Avoid
NA, na	Can also be an abbreviation (e.g., North America), can cause problems with data type (turn a numerical column into a text column). NA is more commonly recognized than na.	R	Good option
N/A	An alternate form of NA, but often not compatible with software		Avoid
NULL	Can cause problems with data type	SQL	Good option
None	Uncommon. Can cause problems with data type	Python	Avoid
No data	Uncommon. Can cause problems with data type, contains a space		Avoid
Missing	Uncommon. Can cause problems with data type		Avoid
-,+,.	Uncommon. Can cause problems with data type		Avoid

White et al 2013

Best practices for organizing raw data

- Column = the variable
- Row = the observation
- One cell → one item
- Do not leave cells empty
  - Have a code to indicate *why* the cell is empty
    - No data = NA
    - To be collected later = CL
- Do not rely on formatting (colors, fonts, size, etc)

Plot: 2			
Date collecte	Species	Sex	Weight
1/8/14	NA		
1/8/14	DM	M	44
1/8/14	DM	M	38
1/8/14	OL		
1/8/14	PE	M	22
1/8/14	DM	M	38
1/8/14	DM	M	48
1/8/14	DM	M	43
1/8/14	DM	F	35
1/8/14	DM	M	43
1/8/14	DM	F	37
1/8/14	PF	F	7
1/8/14	DM	M	45
1/8/14	OT		
1/8/14	DS	M	157
1/8/14	OX		
2/18/14	NA	M	218
2/18/14	PF	F	7
2/18/14	DM	M	52

measurement device not calibrated

Date collecte	Species	Sex	Weight	Calibrated
1/8/14	NA			
1/8/14	DM	M	44	Y
1/8/14	DM	M	38	Y
1/8/14	OL			
1/8/14	PE	M	22	Y
1/8/14	DM	M	38	Y
1/8/14	DM	M	48	Y
1/8/14	DM	M	43	Y
1/8/14	DM	F	35	Y
1/8/14	DM	M	43	Y
1/8/14	DM	F	37	Y
1/8/14	PF	F	7	Y
1/8/14	DM	M	45	Y
1/8/14	OT			
1/8/14	DS	M	157	N
1/8/14	OX			
2/18/14	NA	M	218	N
2/18/14	PF	F	7	Y
2/18/14	DM	M	52	Y

Best practices for organizing raw data

- Column = the variable
- Row = the observation
- One cell → one item
- Do not leave cells empty
  - Have a code to indicate *why* the cell is empty
    - No data = NA
    - To be collected later = CL
- Do not rely on formatting (colors, fonts, size, etc)
- Be consistent
  - Categorical variables
  - Date formatting
  - File name formatting


PUBLIC
OUR DIFF
CAN LEAD
ISO SET
THIS IS
THE FOLLO
02/27/2
201302
27.2.13
MMXIII
((3+3)×(
10/11011/
35
Happy
Birthday
Excel
Coder Loco @coderloco · 11h
Replying to @msexcel
Don't really trust you with dates, so I would double check
1
28
188
Microsoft Excel @msexcel · 9h
We feel you.

Best practices for organizing raw data

- Column = the variable
- Row = the observation
- One cell → one item
- Do not leave cells empty
  - Have a code to indicate *why* the cell is empty
    - No data = NA
    - To be collected later = CL
- Do not rely on formatting (colors, fonts, size, etc)
- Be consistent
  - Categorical variables
  - Date formatting
  - File name formatting
- Choose your column headers carefully

Choose your column headers carefully

Table 1. Examples of good and bad variable names.

good name	good alternative	avoid
Max_temp_C	MaxTemp	Maximum Temp (°C)
Precipitation_mm	Precipitation	precmm
Mean_year_growth	MeanYearGrowth	Mean growth/year
sex	sex	M/F
weight	weight	w.
cell_type	CellType	Cell type
Observation_01	first_observation	1st Obs.

Broman & Woo 2018

Breakout: Organize the poll data

https://bit.ly/3dmkQVz

Best practices for data management

- Save and *lock* the raw data.

Best practices for data management

- Save and *lock* the raw data.
  - No really, don't touch it.

Best practices for data management

- Save and *lock* the raw data.
  - No really, don’t touch it.
- Back up your raw data, frequently
  - DropBox, GitHub, GoogleDrive

Best practices for data management

- Save and *lock* the raw data.
  - No really, don't touch it.
- Back up your raw data, frequently
  - DropBox, GitHub, GoogleDrive
- Create a "Data Dictionary" or Metadata (data about your data)
  - ReadMe
  - Cornell ReadMe Style Guide for Metadata
  - This should be a text file

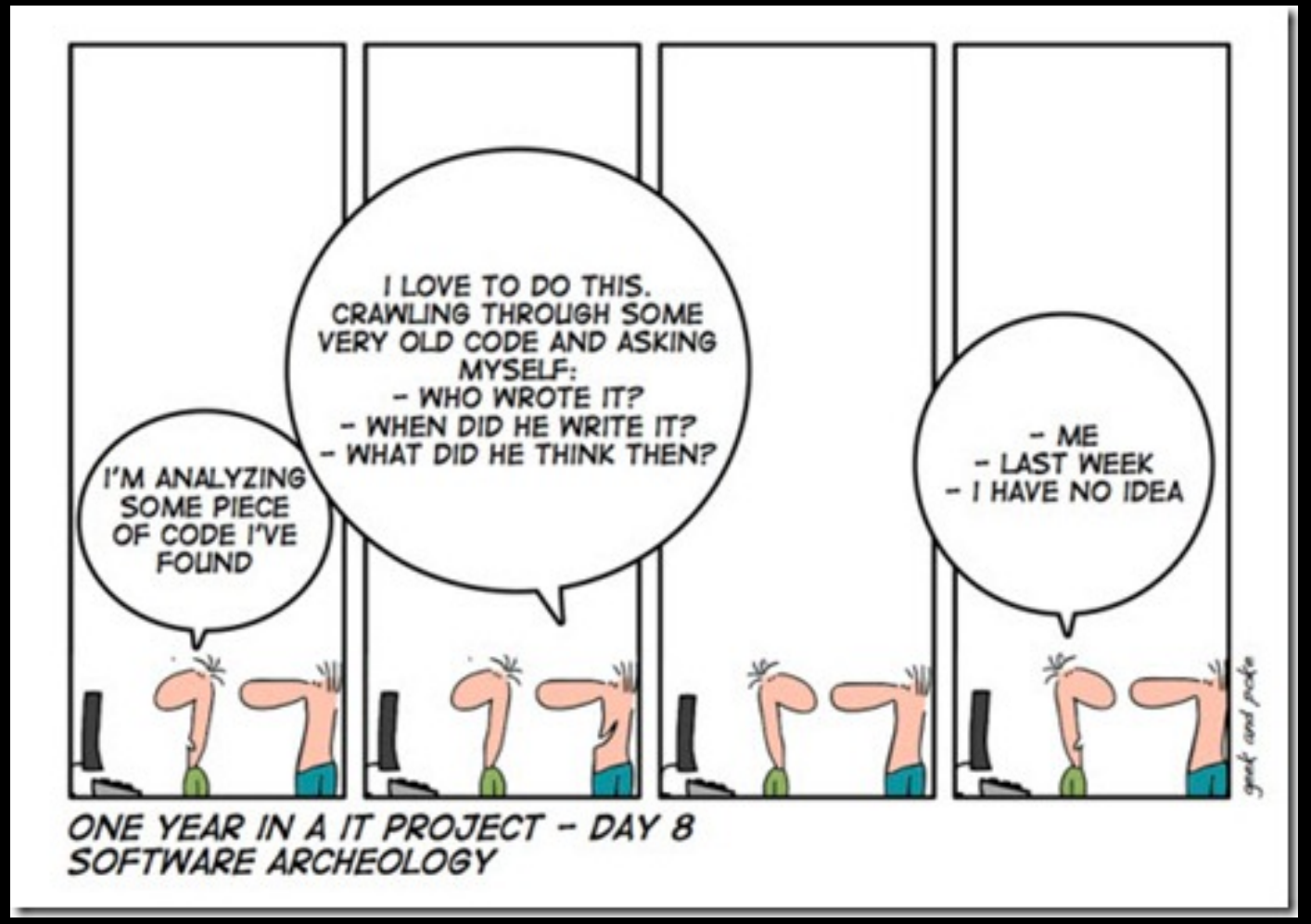
Goals

1. ✓ Convince you why you should have good data management and analysis for reproducibility
2. ✓ Highlight best practices for data management
3. Highlight best practices for writing data analysis code

Best practices for writing data analysis code

ANNOTATE EVERYTHING

Best practices for writing data analysis code



Best practices for writing data analysis code

ANNOTATE EVERYTHING

BE CONSISTENT

- At the start of your code, include
  - The project name
  - The author of the code
  - The dependencies (i.e. required packages and their respective versions)
  - [Usually, a working directory but pls don’t, there is a better way, stay tuned]

Best practices for writing data analysis code

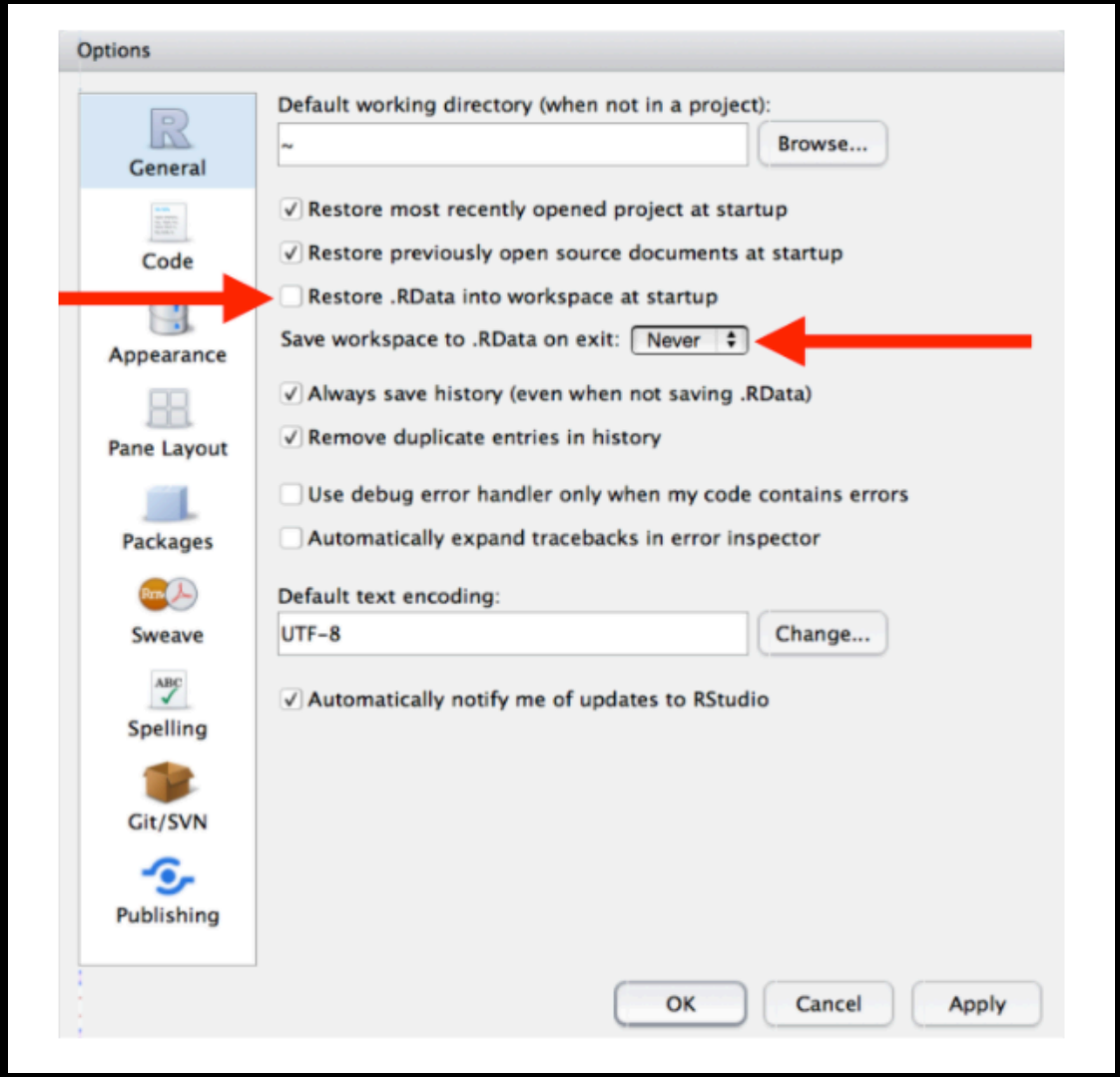
- Create discrete sections using #comment syntax
- Give objects and functions meaningful names
- Always start with a clean environment

PSA: Your R environment isn't real

- Your data is real.
- Your code creates objects from your data.
- Your code manipulates objects.
- Your code analyzes objects.

- Don't rely on your environment.
- Make it work in your code.

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



Best practices for writing data analysis code

- Create discrete sections using #comment syntax
- Give objects and functions meaningful names
- Always start with a clean environment
- Put functions and their definitions at the top of the script, or within the appropriate section
- Use version control (hang tight…)

Best practices for data *analysis*

- Depends on your subfield….
- In general:
  - Clean your data
  - Explore your data (outliers, normality, collinearity, etc)
  - Analyze your data
  - ^ Keep all of these steps in your final code

- Choose your packages carefully
  - For an open-sourced language like R, anyone can write a package
  - Come to Rrodeo and we'll teach you how to write your own

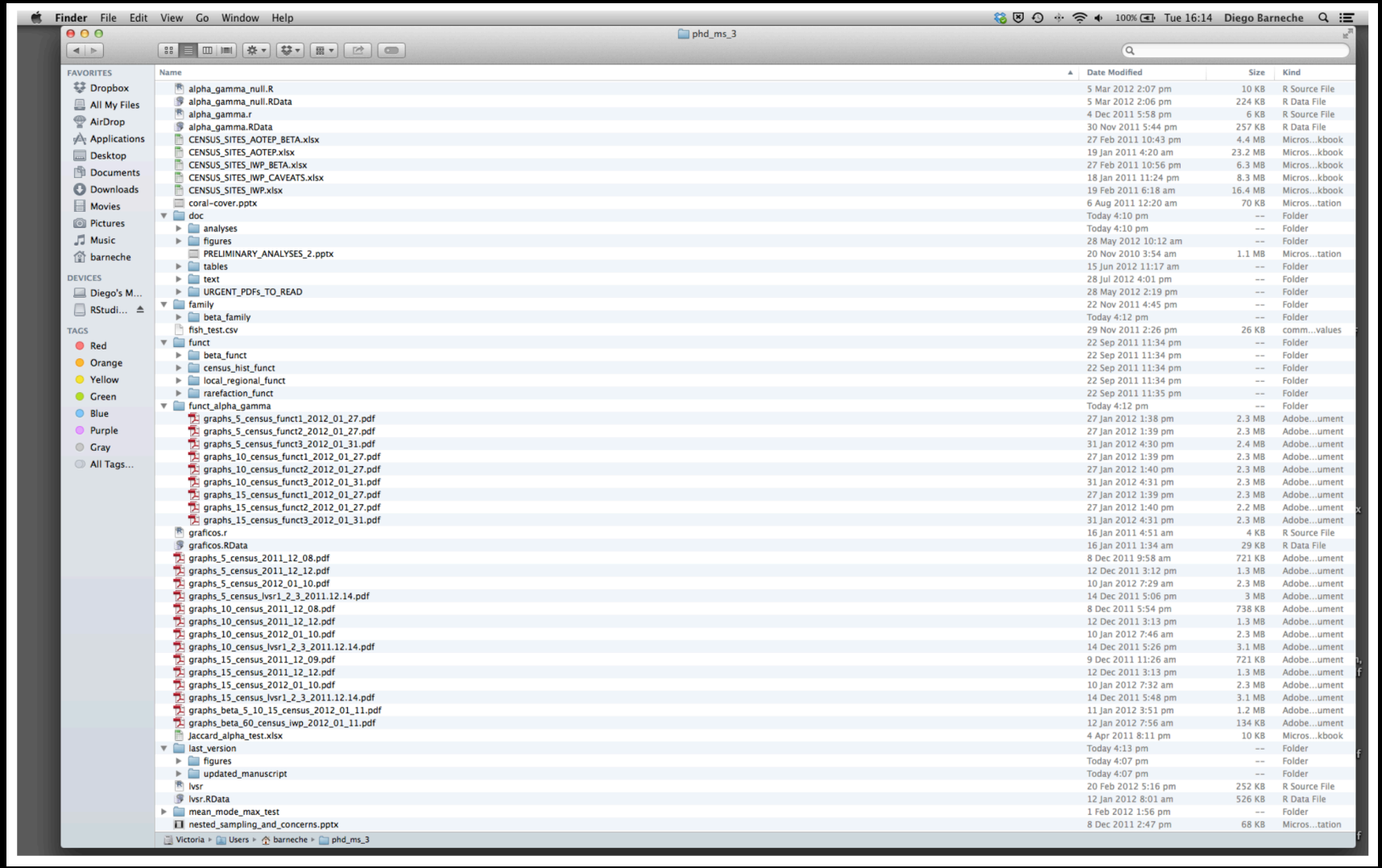
- Here's a great paper on data exploration for ecologists.

Your R & GitHub Journey :-)

These commands are no longer in your coding vocabulary

- `rm(list = ls())`
- `setwd()`

Your R & GitHub Journey :-)



RProjects

.Rproj

- Stores all the settings specific to that project
  - The ‘working directory’
  - All files within the folder
  - Connection to version control
- Opens with a ‘clean’ environment

Even if you do not use GitHub, you should use Rprojects

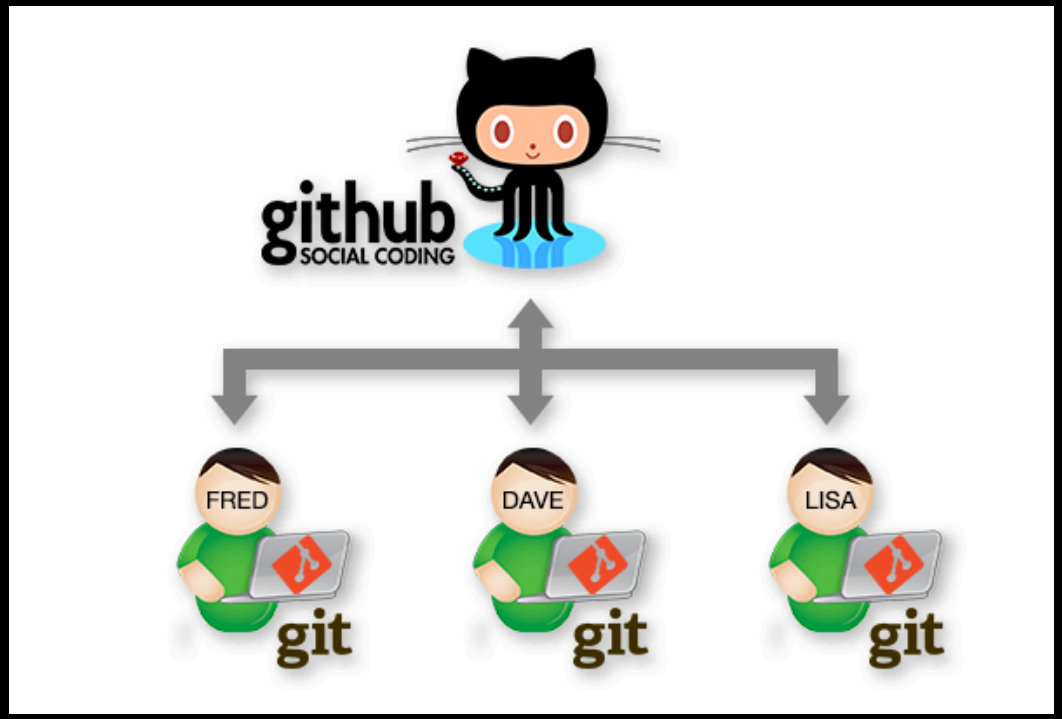
Git

- Language of version control
- Open-sourced
- Used across a massive number of other languages
- Contains the full history of changes for any given file, locally

You do not need to know Git to use GitHub

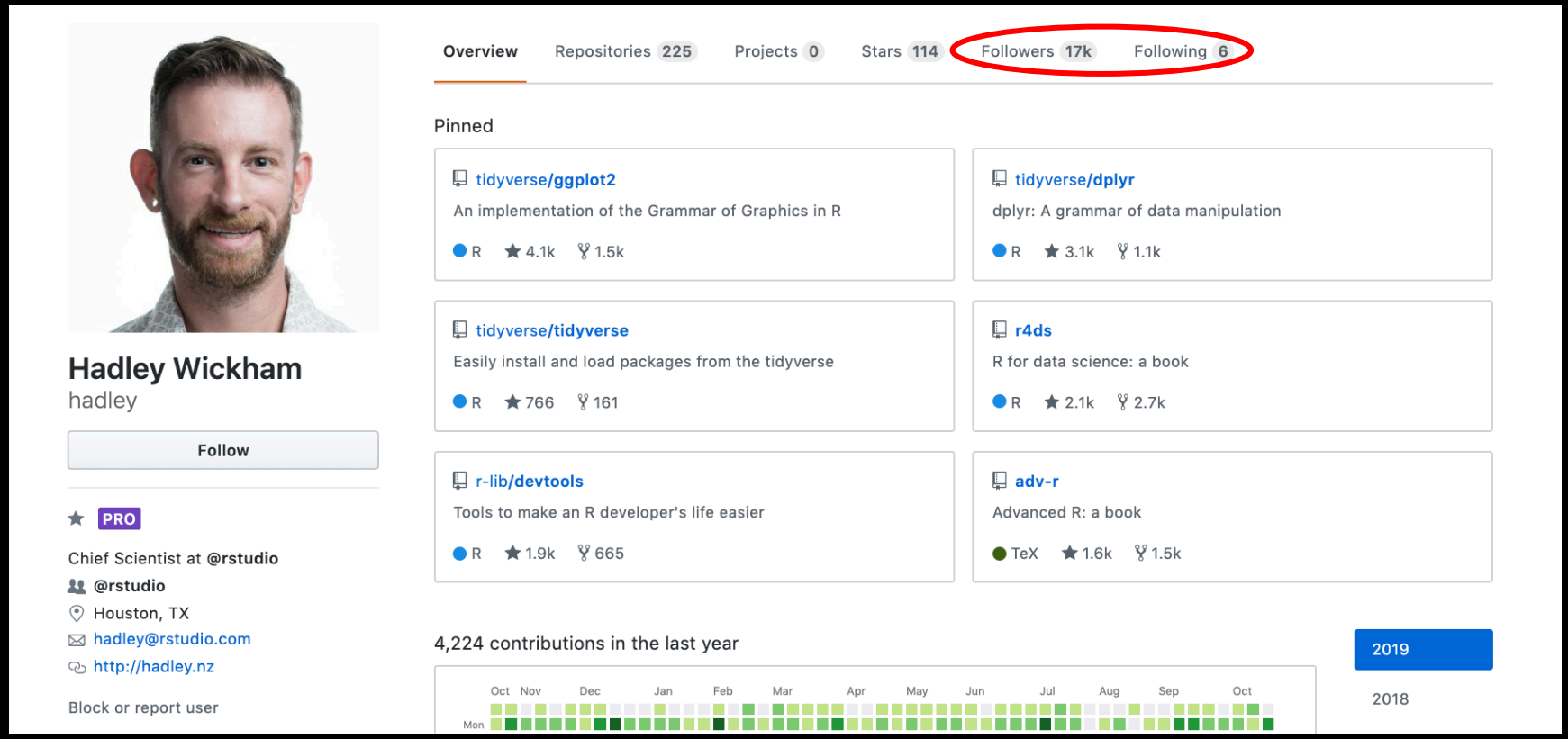
GitHub

- The “cloud” for version control & collaboration
  - Edit history of a google doc within a google drive

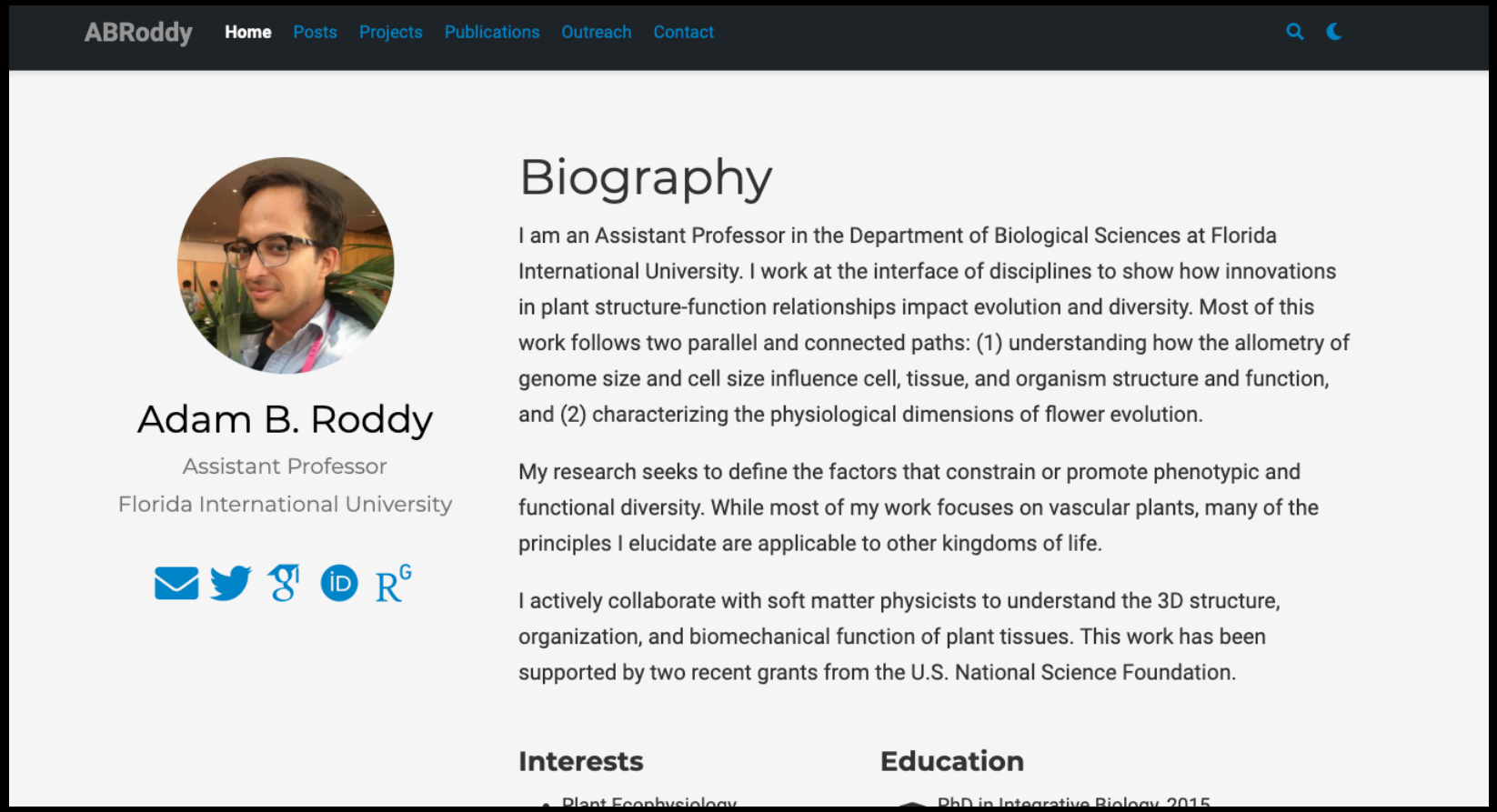


GitHub

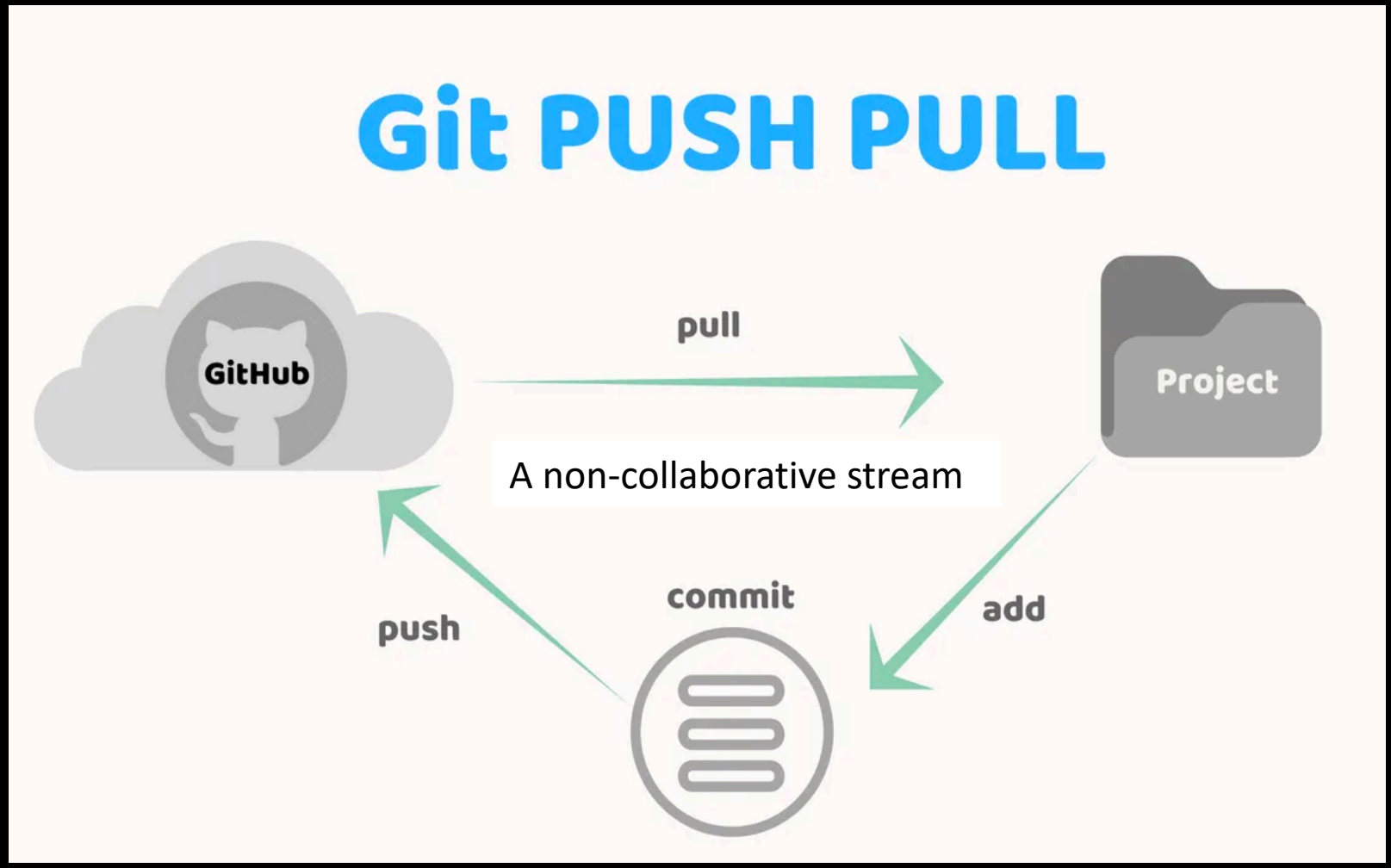
- Another social media



GitHubPages

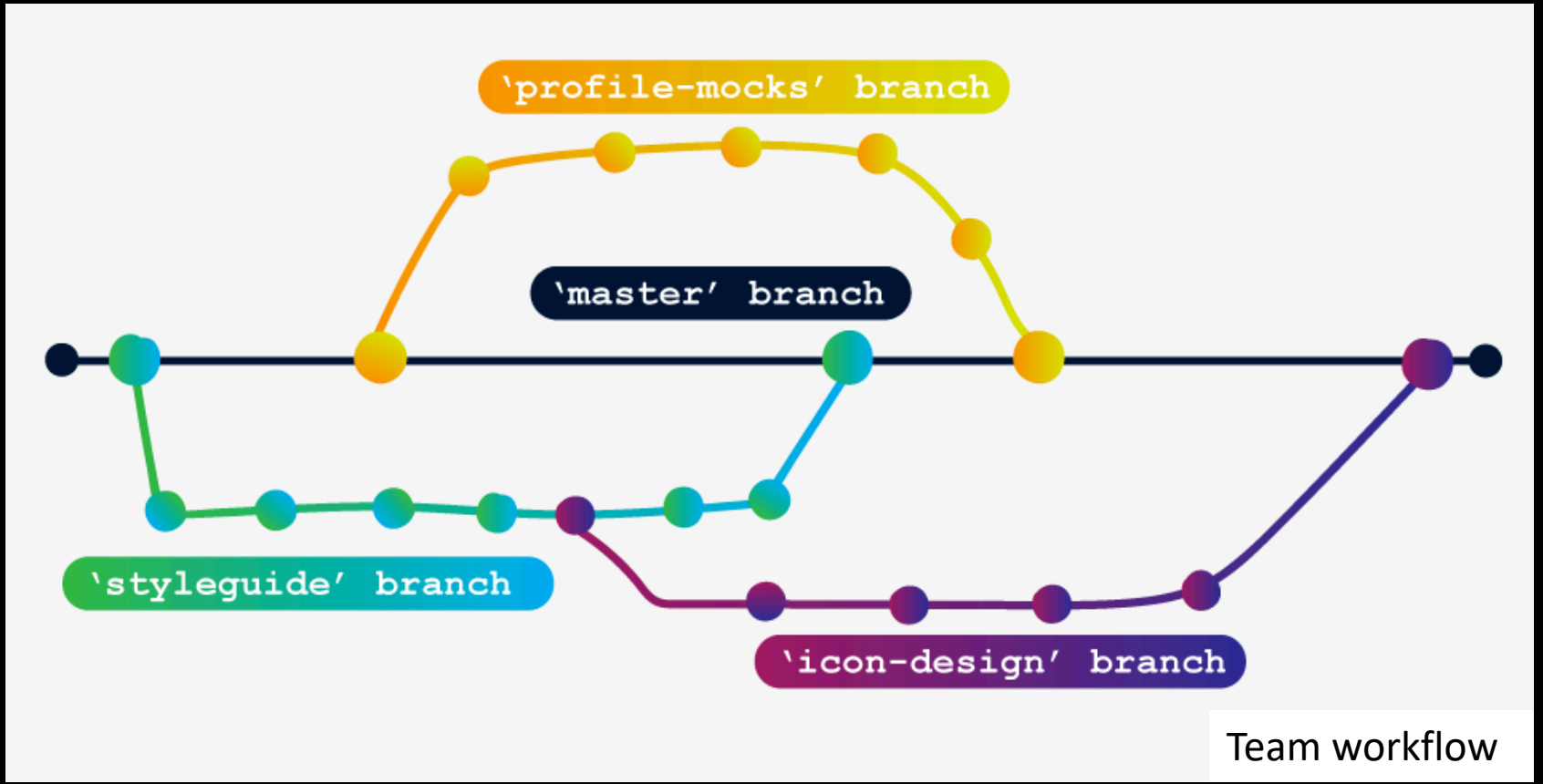


GitHub 101



- Pull
- Commit
- Push

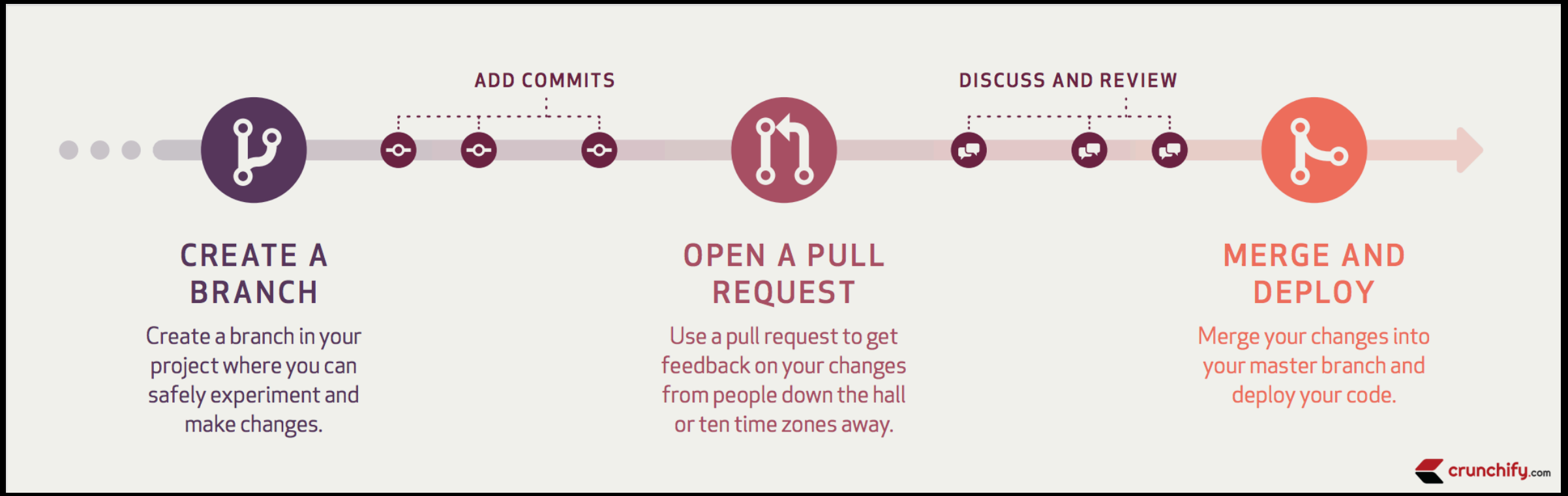
GitHub 101



Team workflow

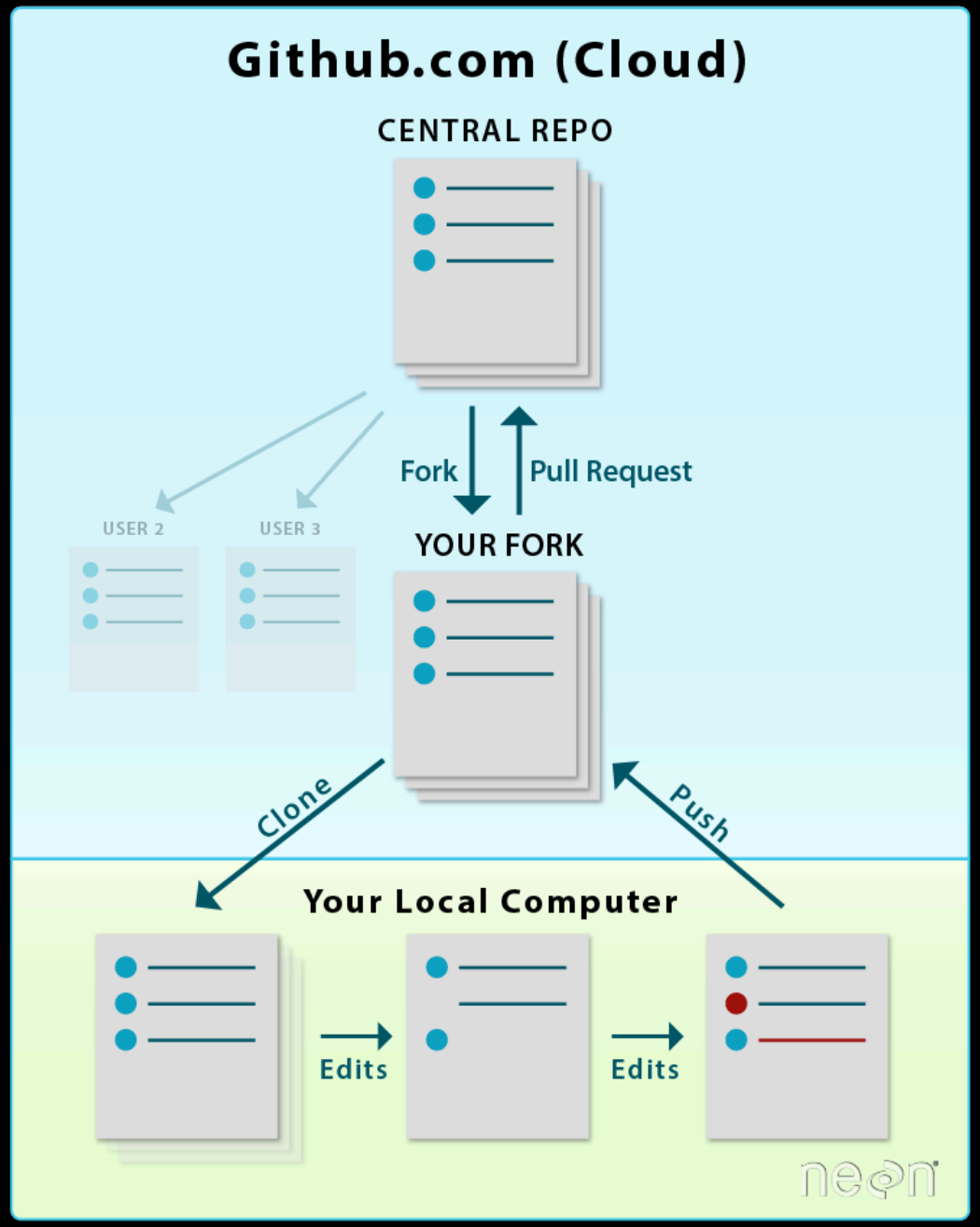
- Branches
  - Master
  - Fork
- Pull request

GitHub 101


ADD COMMITS
DISCUSS AND REVIEW
CREATE A BRANCH
Create a branch in your project where you can safely experiment and make changes.
OPEN A PULL REQUEST
Use a pull request to get feedback on your changes from people down the hall or ten time zones away.
MERGE AND DEPLOY
Merge your changes into your master branch and deploy your code.
crunchify.com

GitHub 101

- “Pull” is to pull from upstream
  - Catches your local device up on any changes that may have been made
- To “fork” and clone is to “branch” from the master and continue working on something, even as other commits are being made to the master
- A “pull request” initiates the re-merge of your fork to the master.
- Changes are three step: Commit, Stage, Push



Let’s try it out!

GitHub and your thesis

- For each thesis question* (read: publishable unit)
  - Establish a Project Folder
  - Establish a GitHub repo
- Make commits early and often
- Use effective commit messages
- Practice!
  - Use dummy repos and datasets until you get more comfortable

- * If you are working with human subjects or sensitive data, you'll need to have a data management plan that may not include GitHub.

Some parting comments….

- There are many different venues for DOI-issuing repositories
- If a journal says “you must submit to Dryad”, it means that Dryad is the *minimum*…
  - You can share your GitHub instead and archive to Zenodo
- The convention in ecology is to do a clean commit of the entire repo prior to review
  - Either:
    - Make an entirely new repo with only the initial commit
    - Or create an orphan branch
- Keep your repo private until it’s going out for review.

Some parting comments....

Additional Tutorials

- https://ourcodingclub.github.io/tutorials/git/
- https://happygitwithr.com/existing-github-last.html