Reproducible Workflows

29 September 2021

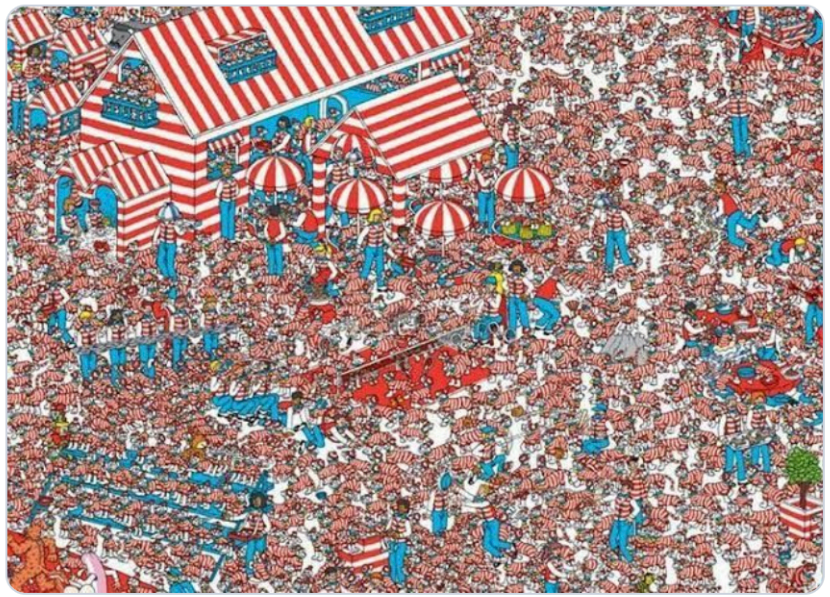
Modern Research Methods

Guy Prochilo 🏳️‍🌈
@GuyProchilo

"Thank you for your interest in our paper. Please find attached the code for reproducing our findings"

The code

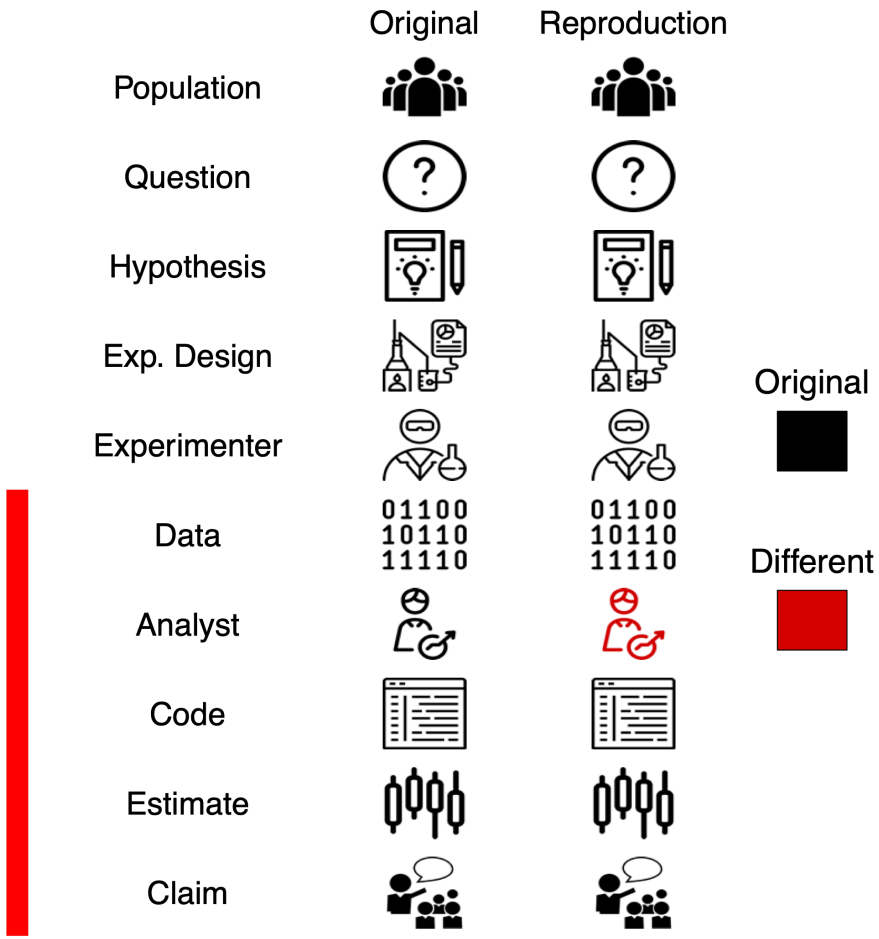
#phdchat #rstats



2:26 PM · Feb 5, 2020 · Buffer

5 Retweets **40** Likes

Last time



REPRODUCE = Get same result from same dataset.

REPRODUCE = "...a second researcher might use the same raw data [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." – NSF Report.

Reproducibility

- For analysis pipeline to be reproducible, both data and analysis code needs to be shared

- Ways to facilitate reproducibility:
  - R and Rmarkdown
  - Good spreadsheet practices
  - Version control for data and code

Spreadsheet horror stories!

EuSpRIG Horror Stories

Spreadsheet mistakes - news stories

Public reports of spreadsheet errors have been sought out on behalf of EuSpRIG by Patrick O'Beirne of Systems Modelling for many years. There are very many reports of spreadsheet related errors and they seem to appear in the global media at a fairly consistent rate.

These stories illustrate common problems that occur with the uncontrolled use of spreadsheets. In many cases, we identify the area of risk involved and then say how we think the problem might have been avoided.

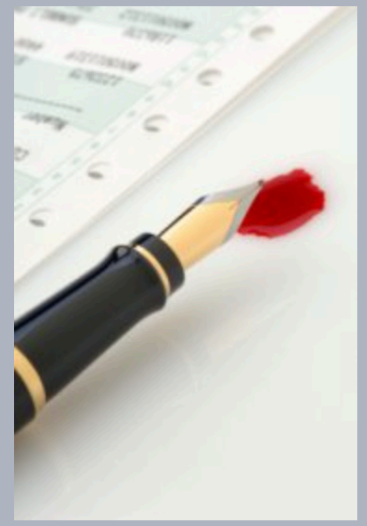
Stories are identified by those who kindly collated and sorted them:

POB: Patrick O'Beirne, Eusprig chair

FH: Felienne Hermans (winner of the 2011 David Chadwick student prize and now an assistant professor at Delft University of Technology).

NS: Tie Cheng, a EuSpRIG committee member.

MPC: Mary Pat Campbell, an actuary, trainer, and a member of the EuSpRIG Discussion group.



LINK:
http://www.eusprig.org/horror-stories.htm

Spreadsheet horror stories

Excel: Why using Microsoft's tool caused Covid-19 results to be lost

By Leo Kelion
Technology desk editor

5 October 2020

Coronavirus pandemic

The badly thought-out use of Microsoft's Excel software was the reason nearly 16,000 coronavirus cases went unreported in England.

Lawmakers examine cause, solutions to $25M education budget error

Shumway said the error was the result of a mathematical formula in an Excel spreadsheet referencing the wrong cell to calculate the state's weighted pupil units. That inaccurate figure was then used for a per-pupil funding estimate that resulted in public education being underfunded by the Utah Legislature.

Good spreadsheet practices

- Use for only data entry and data storage
- Analysis and visualization in R (or some other language)
- Apply good practice principles (Broman & Woo, 2008, Monday's reading)

Data needs to be easily read by **both** humans and computer

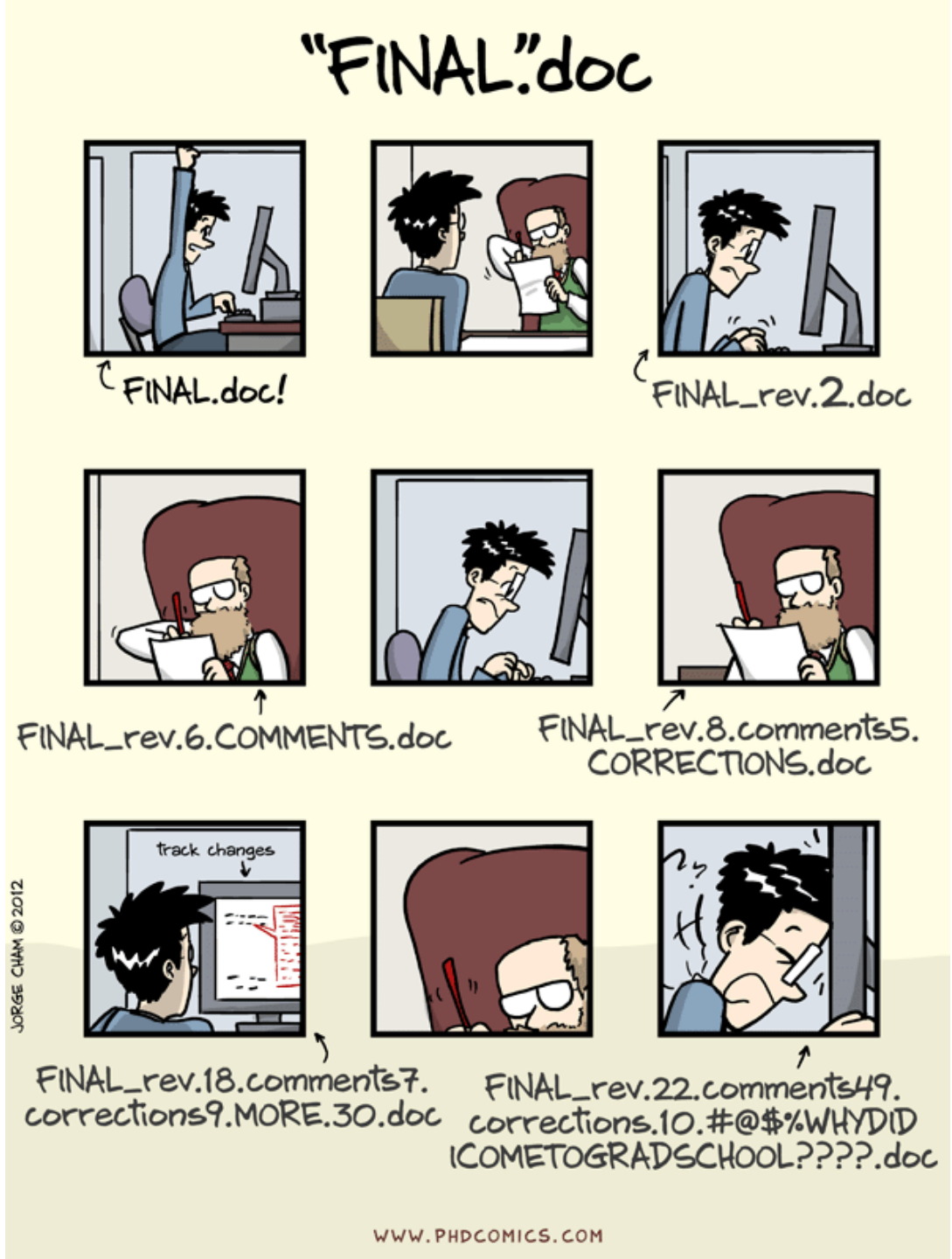
Image source: natalka_dmitrova - www.freepik.com

Good spreadsheet practices

1. Be consistent
   - Whatever you do, do it consistently
   - Use consistent codes for categorical variables.
   - Use a consistent fixed code for any missing values
   - Use consistent variable names across files
   - Use consistent subject identifiers.
   - Use consistent file names.
   - Be careful about extra spaces within cells

Good spreadsheet practices

2. Choose good names for things
   - Avoid spaces!!
     - You'll have to use double quotes for spaced variable names, "glucose 6 weeks", rather than just writing glucose_6_weeks.
     - Use underscores instead
   - Avoid special characters, except for underscores and hyphens
     - Other symbols ($, @, %, #, &, *, (,), !, /, etc.) have special meaning to programming languages
   - Make labels meaningful
   - Never name something "final"!



http://www.phdcomics.com/comics/archive.php?comicid=1531

Good spreadsheet practices

3. Write dates as YYYY-MM-DD
   - "ISO 8601" standard
   - In gene lists contained within papers from 18 journals from 2005–2015, ~20% had errors in the gene names, due to the conversion of gene symbols to dates or floating-point numbers (Ziemann, Eren, and El-Osta, 2016)

Good spreadsheet practices

4. No empty cells
   - Fill in all cells and use some common code for missing data

A

	A	B	C
1	id	date	glucose
2	101	2015–06–14	149.3
3	102		95.3
4	103	2015–06–18	97.5
5	104		117.0
6	105		108.0
7	106	2015–06–20	149.0
8	107		169.4

B

	A	B	C	D	E	F	G	H	I
1		1 min				5 min			
2	strain	normal		mutant		normal		mutant	
3	A	147	139	166	179	334	354	451	474
4	B	246	240	178	172	514	611	412	447

Good spreadsheet practices

5. Put just one thing in a cell
   - “sam-experimental”, “ling-control” → no! Make two columns
   - Don’t include units in a cell “60 g” → no! Better to include unit in column name (”weight_grams”)
   - No notes in a cell
   - Don’t merge cells!

Good spreadsheet practices

6. Make data rectangular (preferably tidy!)

	A	B	C	D	E	F	G	H	I	J	K
1			week 4			week 6			week 8		
2	Mouse ID	SEX	date	weight	glucose	date	weight	glucose	date	weight	glucose
3	3005	M	3/30/2007	19.3	635	4/11/2007	31	460.7	4/27/2007	39.6	530.2
4	3017	M	10/6/2006	25.9	202.4	10/19/2006	45.1	384.7	11/3/2006	57.2	458.7
5	3434	F	11/22/2006	26.6	238.9	12/6/2006	45.9	378	12/22/2006	56.2	409.8
6	3449	M	1/5/2007	27.5	121	1/19/2007	42.9	191.3	2/2/2007	56.7	182.5
7	3499	F	1/5/2007	19.8	220.2	1/19/2007	36.6	556.9	2/2/2007	43.6	446

	A	B	C	D	E	F
1		GTT date	GTT weight	time	glucose mg/dl	insulin ng/ml
2	321	2/9/15	24.5	0	99.2	lo off curve
3				5	349.3	0.205
4				15	286.1	0.129
5				30	312	0.175
6				60	99.9	0.122
7				120	217.9	lo off curve
8	322	2/9/15	18.9	0	185.8	0.251
9				5	297.4	2.228
10				15	439	2.078
11				30	362.3	0.775
12				60	232.7	0.5
13				120	260.7	0.523
14	323	2/9/15	24.7	0	198.5	0.151
15				5	530.6	off curve lo

A

	A	B	C
1	id	GTT date	GTT weight
2	321	2/9/15	24.5
3	322	2/9/15	18.9
4	323	2/9/15	24.7

B

	A	B	C	D	E
1	id	GTT time	glucose mg/dl	insulin ng/ml	note
2	321	0	99.2	NA	insulin below curve
3	321	5	349.3	0.205	
4	321	15	286.1	0.129	
5	321	30	312	0.175	
6	321	60	99.9	0.122	
7	321	120	217.9	NA	insulin below curve
8	322	0	185.8	0.251	
9	322	5	297.4	2.228	
10	322	15	439	2.078	
11	322	30	362.3	0.775	
12	322	60	232.7	0.5	
13	322	120	260.7	0.523	
14	323	0	198.5	0.151	
15	323	5	530.6	NA	insulin below curve

Good spreadsheet practices

7. Create a data dictionary
   - A separate file that explains what all of the variables are
   - Should contain (minimally):
     - The exact variable name as in the data file
     - A longer explanation of what the variable means
     - The measurement units
     - Expected minimum and maximum values

	A	B	C	D
1	name	plot_name	group	description
2	mouse	Mouse	demographic	Animal identifier
3	sex	Sex	demographic	Male (M) or Female (F)
4	sac_date	Date of sac	demographic	Date mouse was sacrificed
5	partial_inflation	Partial inflation	clinical	Indicates if mouse showed partial pancreatic inflation
6	coat_color	Coat color	demographic	Coat color, by visual inspection
7	crumblers	Crumblers	clinical	Indicates if mouse stored food in their bedding
8	diet_days	Days on diet	clinical	Number of days on high–fat diet

Good spreadsheet practices

8. No calculations in the raw data file – just the data!
9. No font color or highlighting!
10. Save the data as plain text (e.g., csv)
11. Make backups
    - Should save all versions of file so if something gets corrupted you can go back and fix

A

	A	B	C
1	id	date	glucose
2	101	2015–06–14	149.3
3	102	2015–06–14	95.3
4	103	2015–06–18	97.5
5	104	2015–06–18	1.1
6	105	2015–06–18	108.0
7	106	2015–06–20	149.0
8	107	2015–06–20	169.4

B

	A	B	C	D
1	id	date	glucose	outlier
2	101	2015–06–14	149.3	FALSE
3	102	2015–06–14	95.3	FALSE
4	103	2015–06–18	97.5	FALSE
5	104	2015–06–18	1.1	TRUE
6	105	2015–06–18	108.0	FALSE
7	106	2015–06–20	149.0	FALSE
8	107	2015–06–20	169.4	FALSE

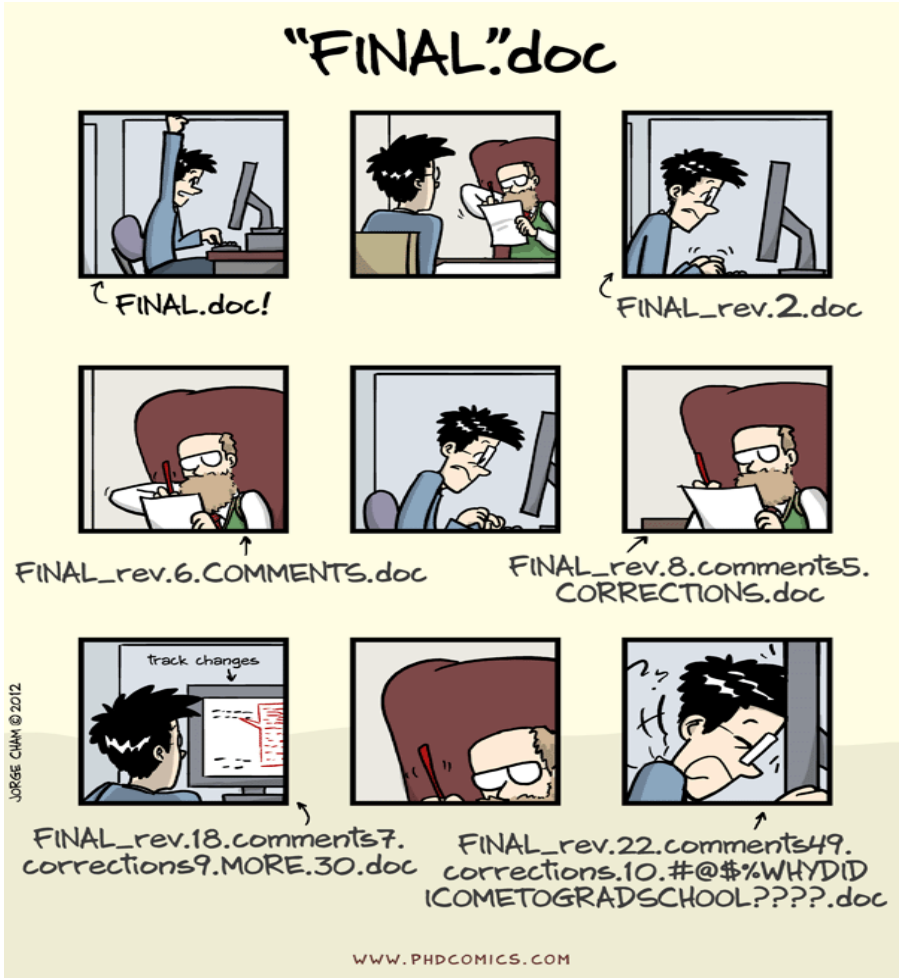
A

	A	B	C	D	E
1	id	sex	glucose	insulin	triglyc
2	101	Male	134.1	0.60	273.4
3	102	Female	120.0	1.18	243.6
4	103	Male	124.8	1.23	297.6
5	104	Male	83.1	1.16	142.4
6	105	Male	105.2	0.73	215.7

B

```
id,sex,glucose,insulin,triglyc
101,Male,134.1,0.60,273.4
102,Female,120.0,1.18,243.6
103,Male,124.8,1.23,297.6
104,Male,83.1,1.16,142.4
105,Male,105.2,0.73,215.7
```

"Version control"



Version control allows you to:

- See history of changes
- Go back to old version

http://www.phdcomics.com/comics/archive.php?comicid=1531

Sharing data (ideally, with versions)

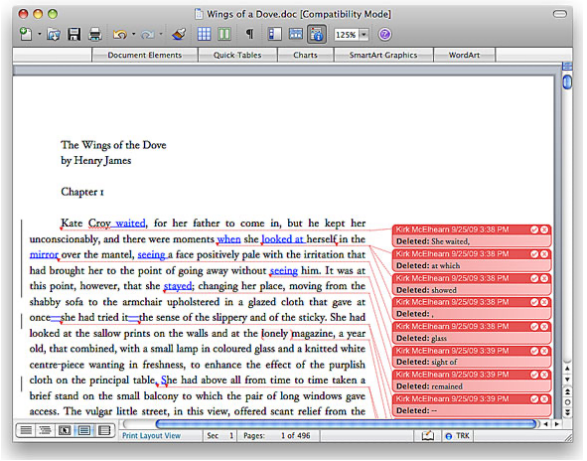
- Possible solutions:
  - Email
  - Texting
  - whatsapp
  - Google docs
  - Google drive
  - Usb drives
  - Paper
  - Verbally??

https://pandelisz.github.io/

One Solution:

- https://osf.io/
- Free and open source project management tool that supports researchers throughout their entire project lifecycle.
- This solves the sharing problem and sort of solves the version control problem
- Easy to use
- Interface clunky, not great version control

A more sophisticated solution: Git



- Git is a free and open source software
- Version control system
- Original purpose was to help groups of developers work collaboratively on big software projects.
- Like track changes in Microsoft Word/Google Docs
- Repurposed by data science/science community

https://pandelisz.github.io/

GitHub != Git (but they're similar)

- GitHub is a web-based **hosting service** for version control using Git (like Dropbox or Google Drive).

- Has many extra features designed to improve how people collaborate.

https://pandelisz.github.io/

Repository as the unit of storage

- "repo"
- In our context, one repository = one project
- Repo contains **all** the files associated with a project.

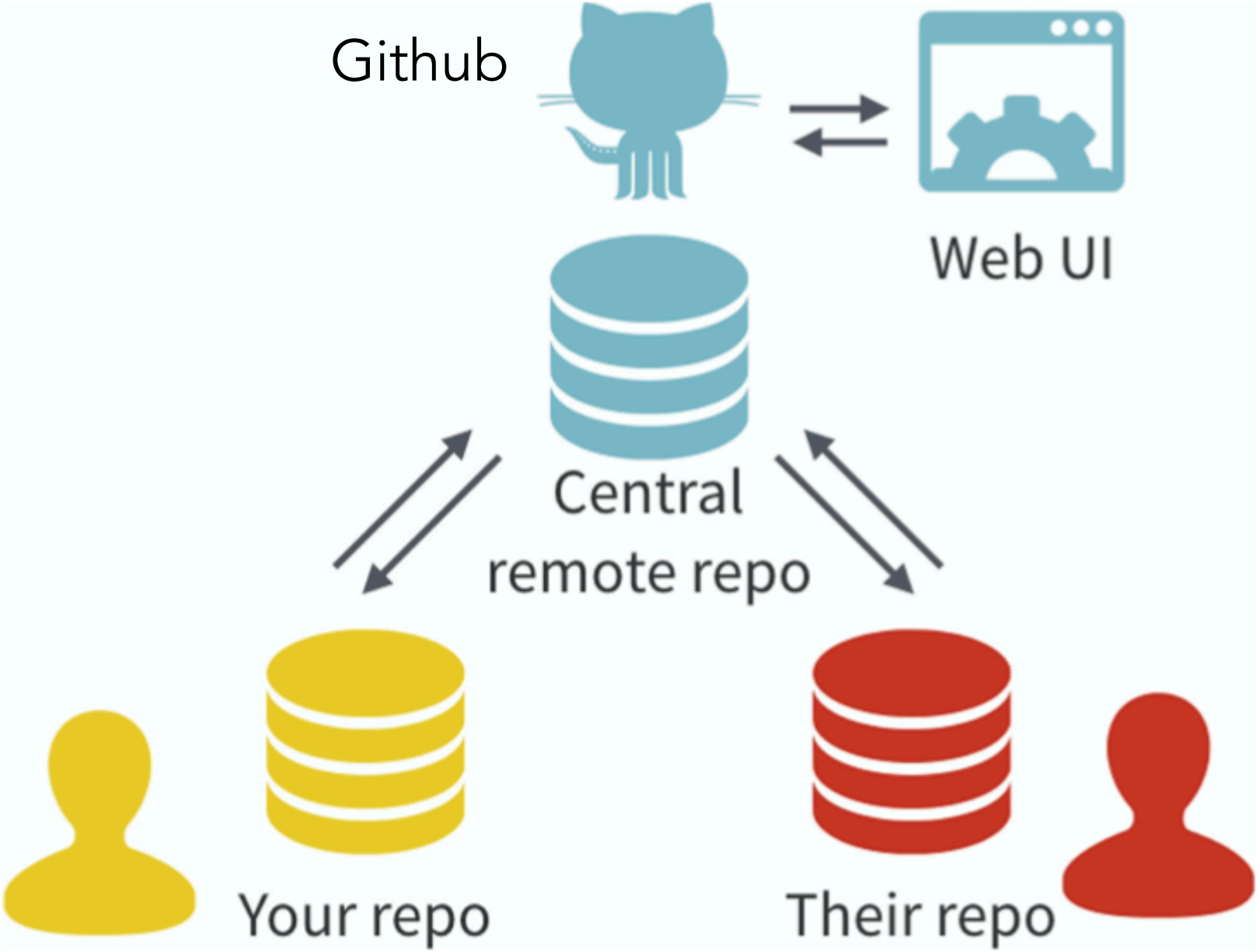
= class project

= software package

= paper

= code for a website

e.g., https://github.com/mllewis/cumulative-science



(Bryan, 2018)

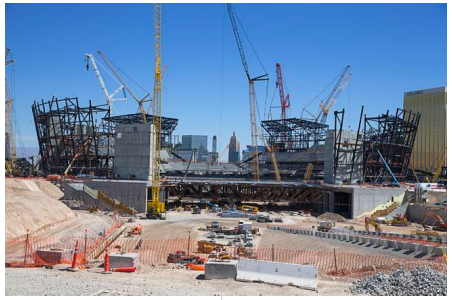
The workflow

Your Working Directory

Staging Area

Local Repo

Central Repo on Github



add

rm

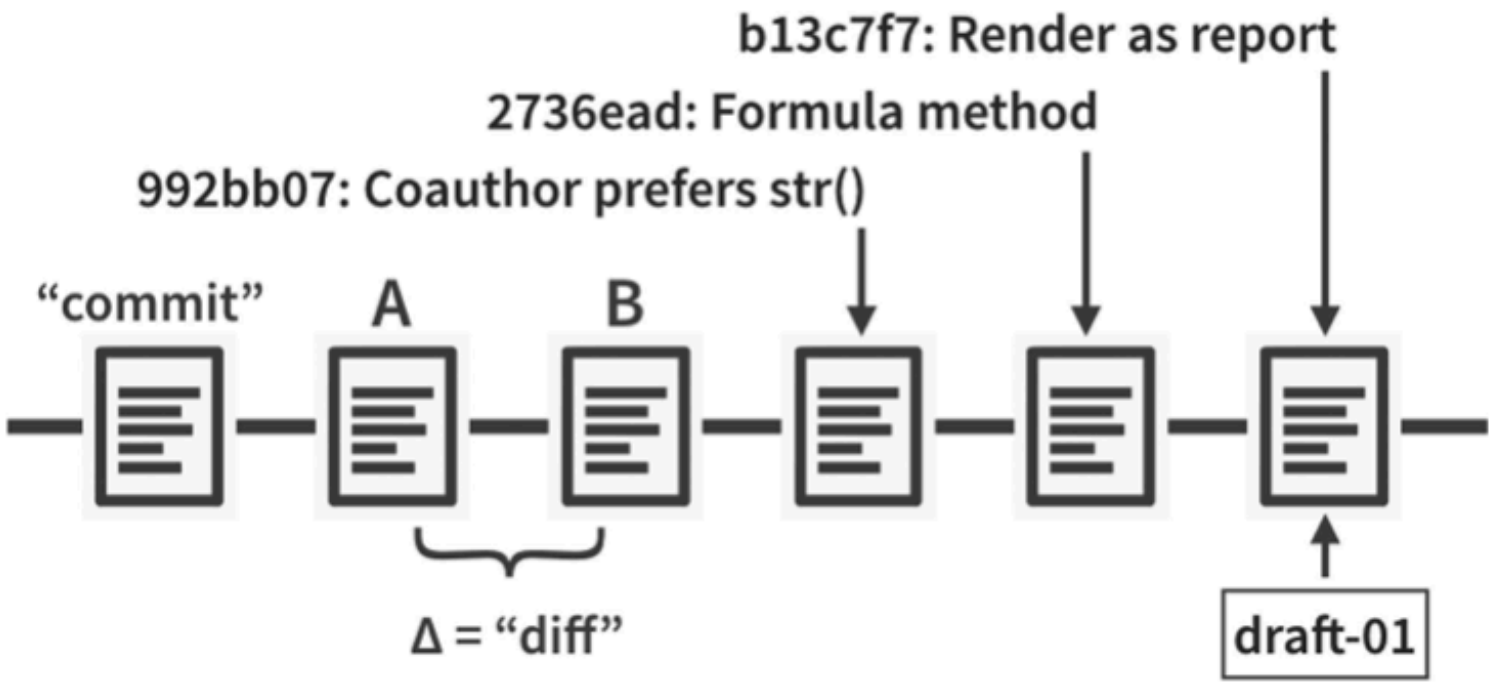
commit

push

pull

clone (first time)

Commit is like a “snapshot” of your file at a point in time



You should include an informative message when you make a commit.

You can always go back and see what your project looked like a different points in your commit history.

(Bryan, 2018)

How do you interact with git?

- At the command line in Terminal:

```
git <VERB> <ARGS>
```

- Various GUIs

Explore the complexity project repo on Github

Contents lists available at ScienceDirect

Cognition

journal homepage: www.elsevier.com/locate/COGNIT

Original Articles

The length of words reflects their conceptual complexity

Molly L. Lewis *, Michael C. Frank

Department of Psychology, Stanford University, United States

https://github.com/mllewis/RC

https://github.com/mllewis/RC

In groups of 2-3, answer the following questions:

1. What is the directory structure of the repo?
2. How many commits are there?
3. What was the first commit message on Aug 21, 2015
4. Can you find the object to the right? (obj_15.jpg)
5. How many collaborators were there on this project?