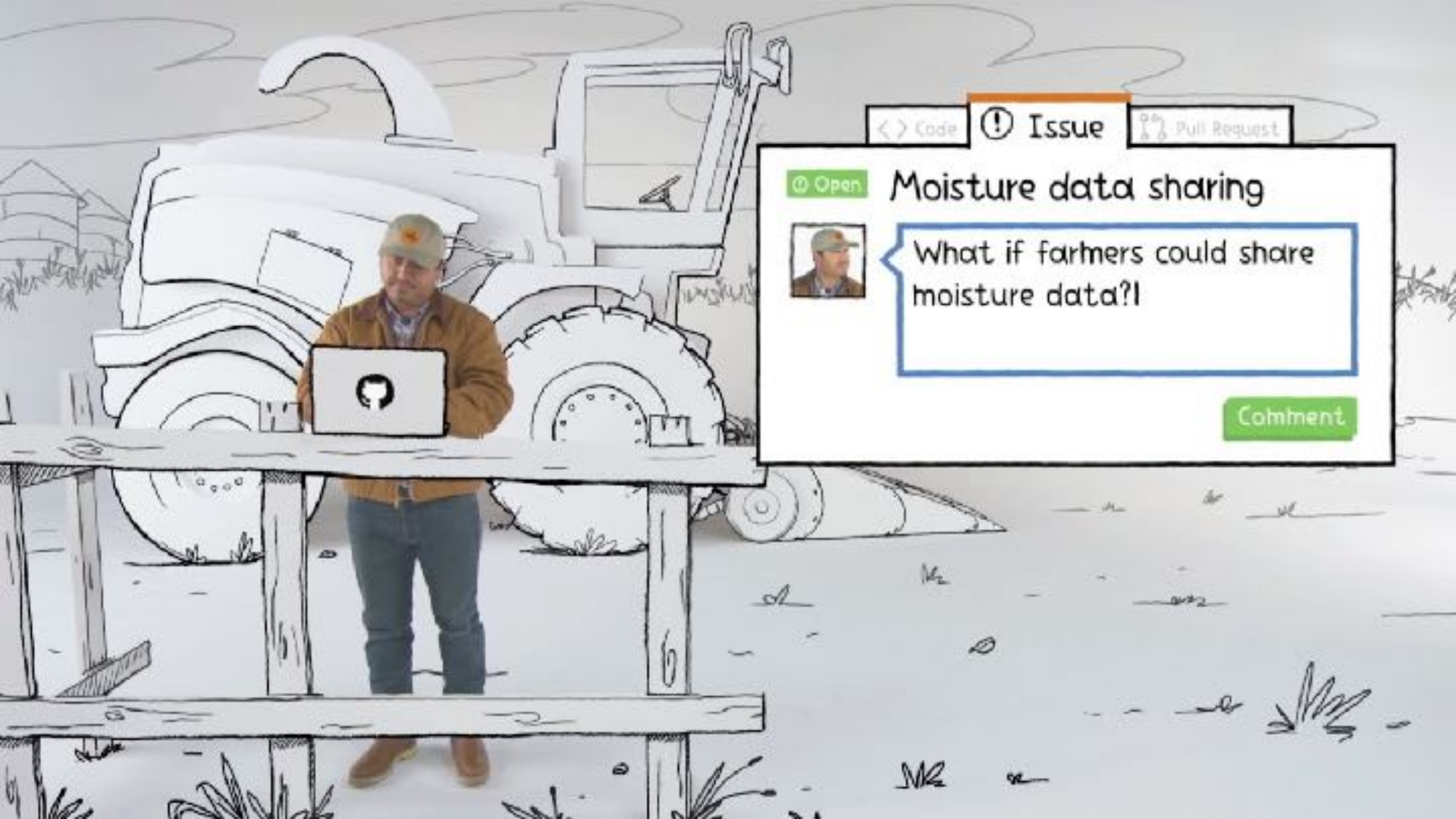
Logistics

- Due date
  - Pre-course survey due Friday (today) at 11:59 PM (extra credit)
  - Practice Assignment due Sunday (was today) at 11:59pm (1%)
  - Due Monday 11:59PM (will be released tonight):
    - Q1 (Canvas quiz)
  - Due *next* Friday 11:59PM (will be released late tonight):
    - D1 (Discussion lab)
    - A1 (Assignment)
    - Group submission (1 Google Form submission per group)

Version Control

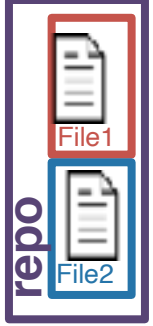
Jason G. Fleischer, Ph.D
UC San Diego

Department of Cognitive Science
jfleischer@ucsd.edu
https://jgfleischer.com
@jasongfleischer


Code
Issue
Pull Request
Open
Moisture data sharing
What if farmers could share moisture data?I
Comment

Each time you create a commit, git tracks the changes made automatically.

By committing each time you make changes, git allows you to time travel!



By committing each time you make changes, git allows you to time travel!

377dfcd00dd057542b112cf13be6cf1380b292ad

439301fe69e8f875c049ad0718386516b4878e22

456722223e9f9e0ee0a92917ba80163028d89251

There's a unique id, known as a **hash**, associated with each commit.

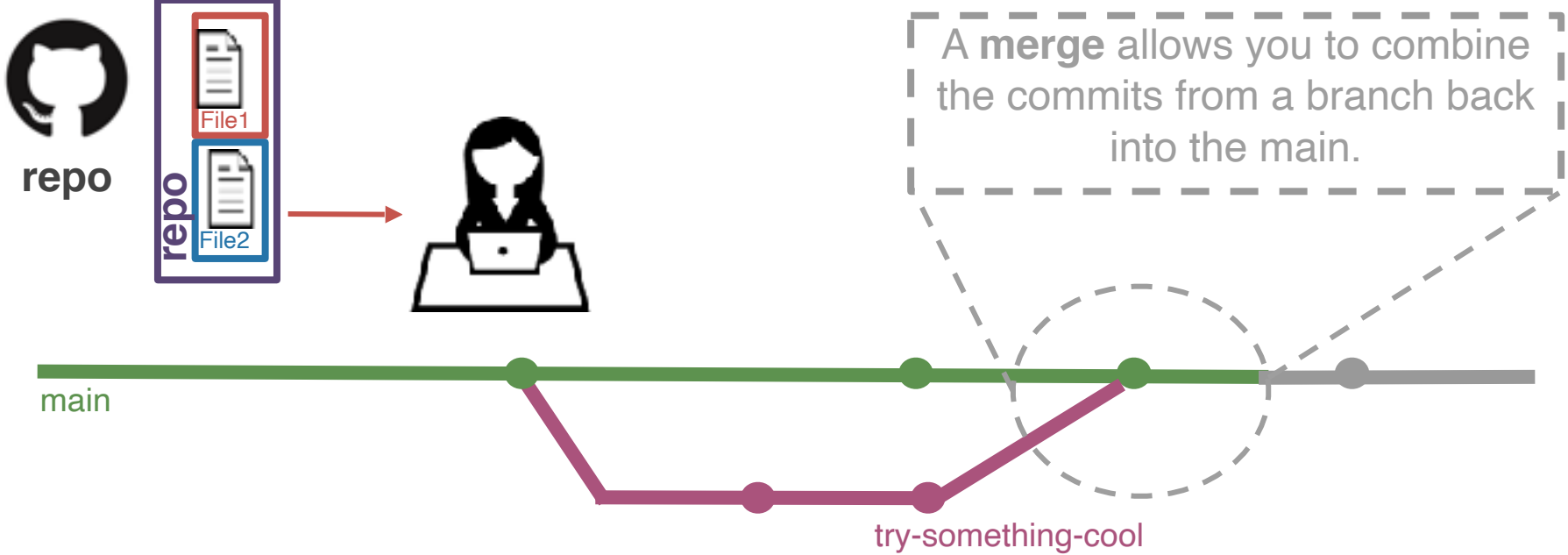
You can return to the state of the repository at any commit. Future commits don't disappear. They just aren't visible when you **check out** an older commit.

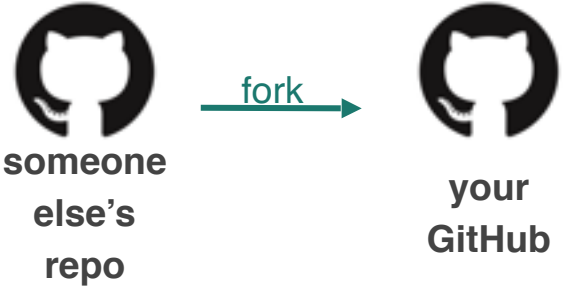
377dfcd00dd057542b112cf13be6cf1380b292ad

But...not everything is always linear.
Sometimes you want to try something out and you're not sure it's going to work.
This is where you'll want to use a **branch**.

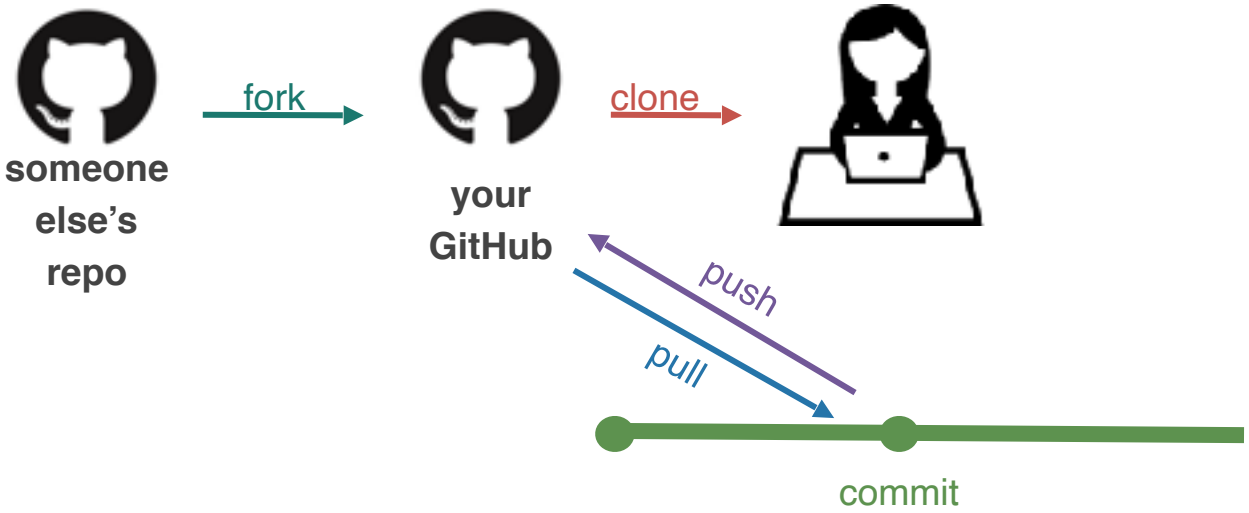
It's a good way to experiment. It's pretty easy to get rid of a branch later on should you not want to include the commits on that branch.

But...what if you DO want to include the changes you've made on your try-something-cool branch into the main branch?

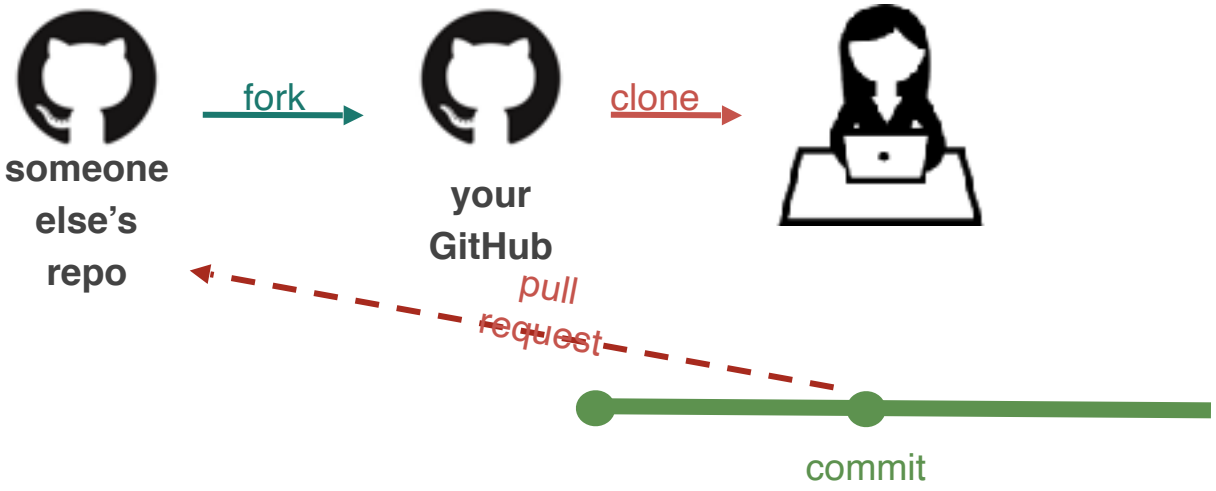

repo
repo
File1
File2
A **merge** allows you to combine the commits from a branch back into the main.
main
try-something-cool



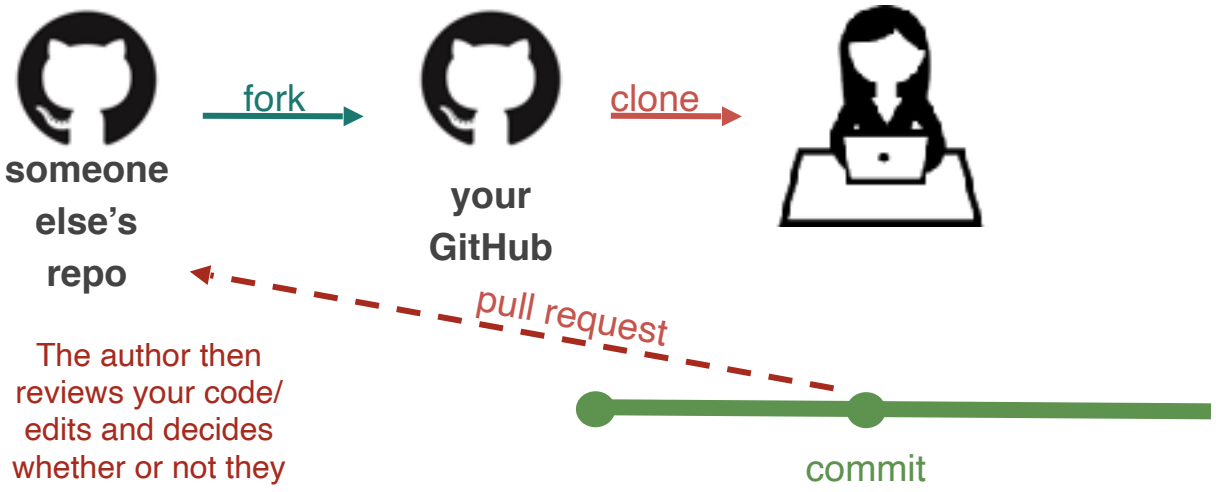
What if someone else is working on something cool and you want to play around with it? You'll have to **fork** their repo.



After you fork their repo, you can play around with it however you want, using the workflow we've already discussed.



But what if you think you've found a bug in their code, a typo, or want to add a new feature to their software? For this, you'll submit a **pull request** (aka **PR**).



The author then reviews your code/ edits and decides whether or not they want to **merge your pull request**.

But what if you think you've found a bug in their code, a typo, or want to add a new feature to their software? For this, you'll submit a **pull request** (aka **PR**).

someone else's repo

Last but not least...what if you find a bug in someone else's code OR you want to make a suggestion but aren't going to submit a suggestion with a PR. For this, you can file an **issue** on GitHub.

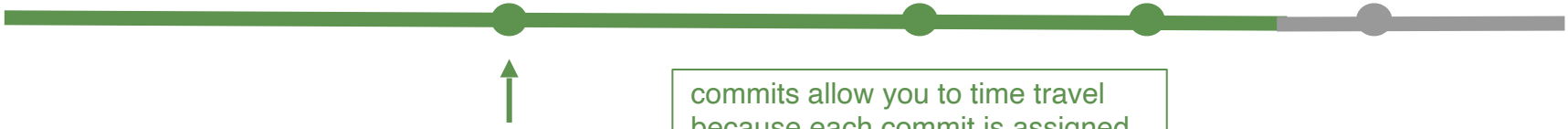
someone else's repo

Last but not least...what if you find a bug in someone else's code OR you want to make a suggestion but aren't going to submit a suggestion with a PR. For this, you can file an **issue** on GitHub.

Issues are *bug trackers*. While, they can include bugs, they can also include feature requests, to-dos, whatever you want, really!

They can be assigned to people.

They can be closed once addressed ….or if the software maintainer doesn't like the suggestion

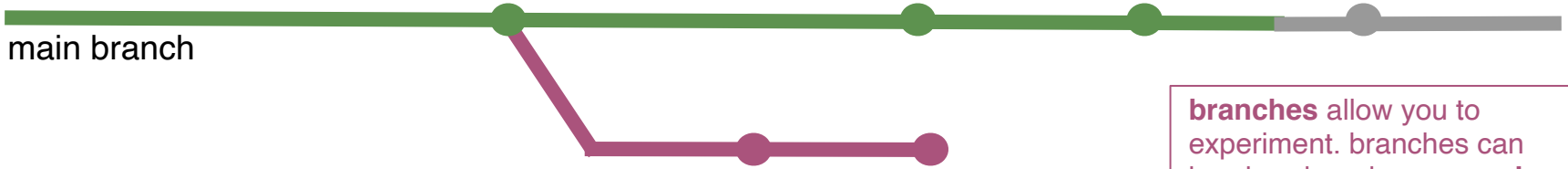


One more git recap...

One more git recap...

377dfcd00dd057542b112cf13be6cf1380b292ad

commits allow you to time travel because each commit is assigned a unique **hash**



branches allow you to experiment. branches can be abandoned or **merged**

someone else's repo

fork

your GitHub

You can work on others' repos by first **forking** their repository onto your GitHub

One more git recap...

You can work on others' repos by first **forking** their repository onto your GitHub

Pull requests allow you to make specific edits to others' repos

Issues allow you to make general suggestions to your/others' repos

One more git recap...

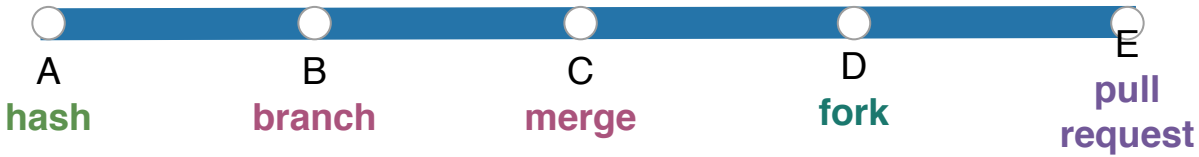
Review & Question Time

Version Controller III

https://forms.gle/eyxgHB3wvqmy17uR9

To experiment within your own repo (test out a new feature, make some changes you're not sure will work)...

what should you do?

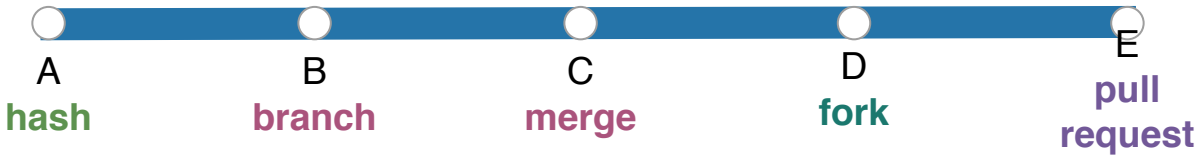


Version Controller IV

https://forms.gle/eyxgHB3wvqmy17uR9

If you've made edits to someone else's repo that you're not a collaborator on…

what would *they* have to do to incorporate your changes?

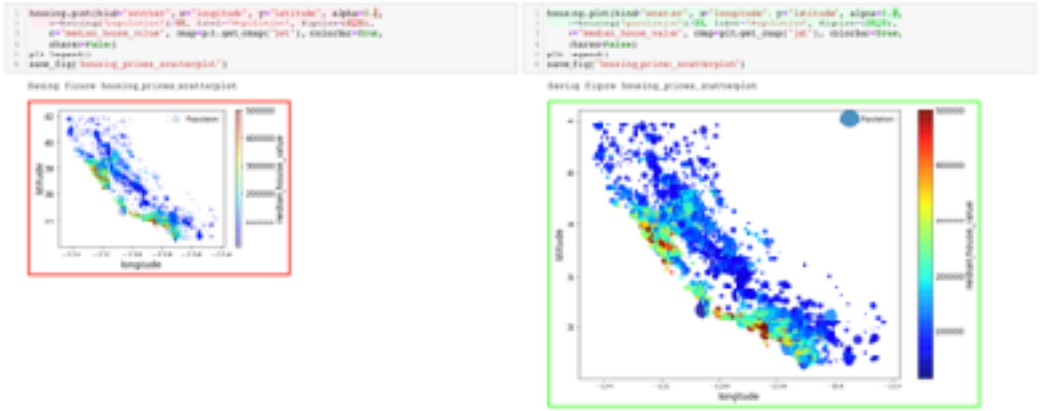


Jupyter notebooks suck to version control

https://nextjournal.com/schmudde/how-to-version-control-jupyter

ReviewNB

ReviewNB is a GitHub app that also offers visual diffing with an interface that looks similar to the traditional Jupyter IDE. Because the outputs are visualized, problems associated with committing binary blobs disappear.



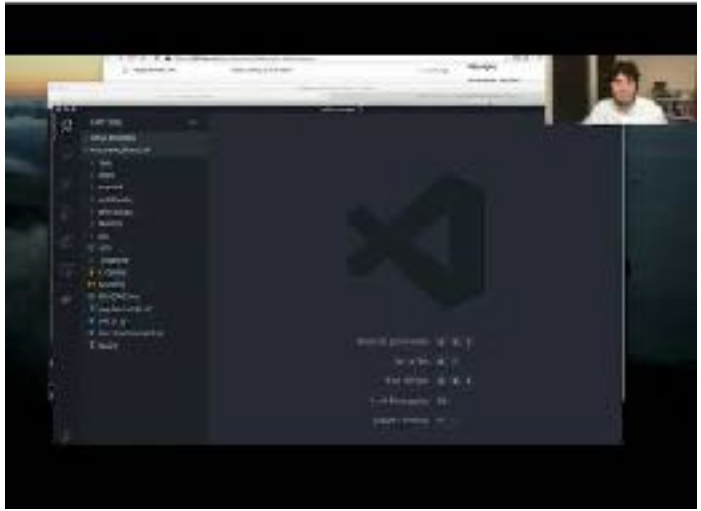
ReviewNB example courtesy of the ReviewNB website

More options

nbautoexport

docs stable | pypi v0.5.0 | conda-forge v0.5.0 | tests passing | codecov 99%

Making it easier to code review Jupyter notebooks, one script at a time.



-jupy

+text

Version Control: Practice

Please do not put assignments on GitHub.

- Discussion Lab 1: Part 3
- Assignment 1: Part 1
  - This will get you practice with git & GitHub
  - Understand what you're doing in the assignment!
  - You may have to google, ask others, spend some time with this!
  - Part II is a Python review; each part of this assignment is self-contained
  - Do this part of the assignment ASAP
- git & Github == How to get the course lectures/materials
  - Assignment 1 will have you fork the Lectures and Project repos
  - You can keep the lectures up-to-date throughout the quarter
- you'll be using GitHub for your final projects

Git and GitHub Crash Course

https://youtu.be/RGOj5yH7evk

COGS 108 Final Projects

The COGS 108 Final Project will give you the chance to explore a topic of your choice and to expand your analytical skills. By working with real data of your choosing you can examine questions of particular interest to you.

- You are encouraged to work on a topic that matters to the world (your family, your neighborhood, a state/province, country, etc).
- Taboo Topics: Movie Predictions/Recommendation System; YouTube Data Analysis, Kickstarter success prediction/analysis,prediction of what makes a song popular on Spotify Whatever is MOST popular EVER and whatever is HOTTEST RN on Kaggle

Final Project: Objectives

- Identify the problems and goals of a *real* situation and dataset.
- Choose an appropriate approach for formalizing and testing the problems and goals, and be able to articulate the reasoning for that selection.
- Implement your analysis choices on the dataset(s).
- Interpret the results of the analyses.
- Contextualize those results within a greater scientific and social context, acknowledging and addressing any potential issues related to privacy and ethics.
- Work effectively to manage a project as part of a team.

Upcoming Project Components

Project Group Signup - 1 submission per group (due Fri Week 2)

Project Review (5%) - Before Mon of week 3, your group will be assigned a previous COGS 108 project to review; A google Form will be released to guide your thinking/discussion about and review of what a previous COGS 108 group did for their project. (due Fri Week 3)

Project Proposal (8%) - a GitHub repo will be created for your group; ‘submit’ on GitHub (due Fri Week 4)

Project Proposal (8%)

Full project guidelines are here:
https://github.com/COGS108/Projects/blob/master/FinalProject_Guidelines.md