Welcome to QuACK!

Week 10
11/10/2025
Alyson & Nina

Today’s Agenda

- Warm-up activity
- Open Science
- Github & Git
- Demo: Walkthrough of Github site + how to use basic Git commands + Pre-registration sites
- Practice

Week 10 Warm-up - 10 Minutes

- Go to https://aspredicted.org/
- Create an account
- Click “CREATE” so that you can see the pre-registration template

  (we are not actually going to be submitting anything, so no need to check the self-destruct box, but it’s a helpful feature)
- Once you have skimmed through the template, read this short blog post about how to write a strong pre-reg:

  https://datacolada.org/64

Open Science

UC Berkeley

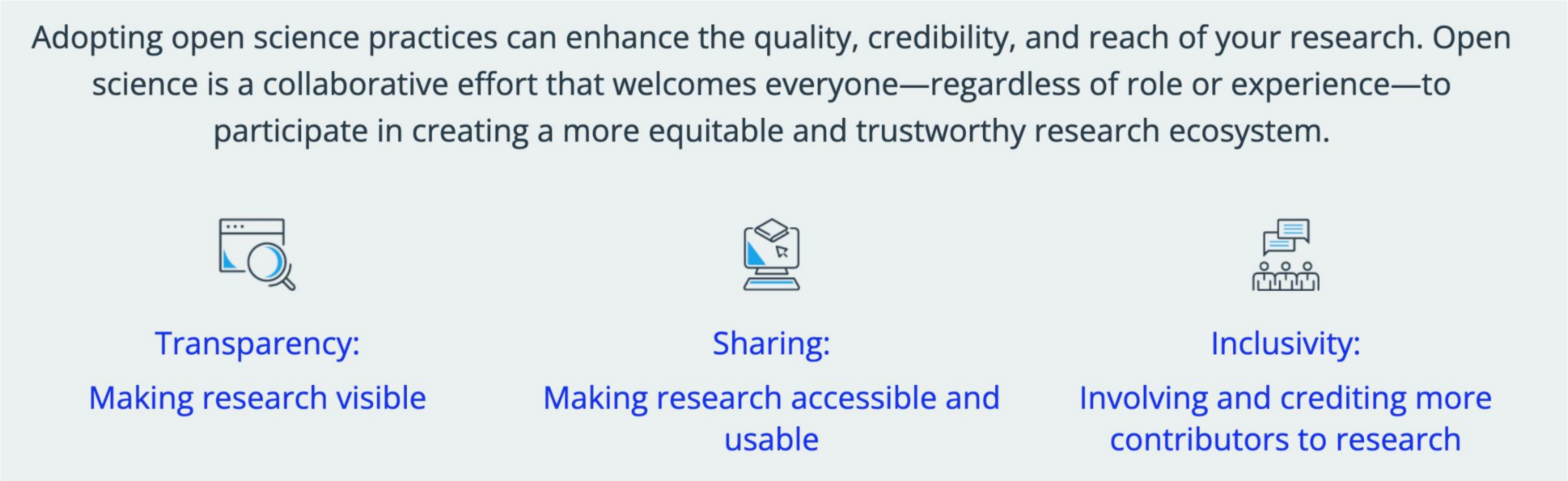
What is Open Science?

“Open science is a global movement that aims to make scientific research and its outcomes freely accessible to everyone. By fostering practices like data sharing and preregistration, open science not only accelerates scientific progress but also strengthens trust in research findings.”

– Center for Open Science

Why is Open Science Important?

Especially given the replication crisis in psychology, open science practices are important for promoting good science practices (e.g., avoiding things like p-hacking), which in turn maintains trust between researchers and also maintains trust from the public.



Adopting open science practices can enhance the quality, credibility, and reach of your research. Open science is a collaborative effort that welcomes everyone—regardless of role or experience—to participate in creating a more equitable and trustworthy research ecosystem.

Transparency:

Making research visible

Sharing:

Making research accessible and usable

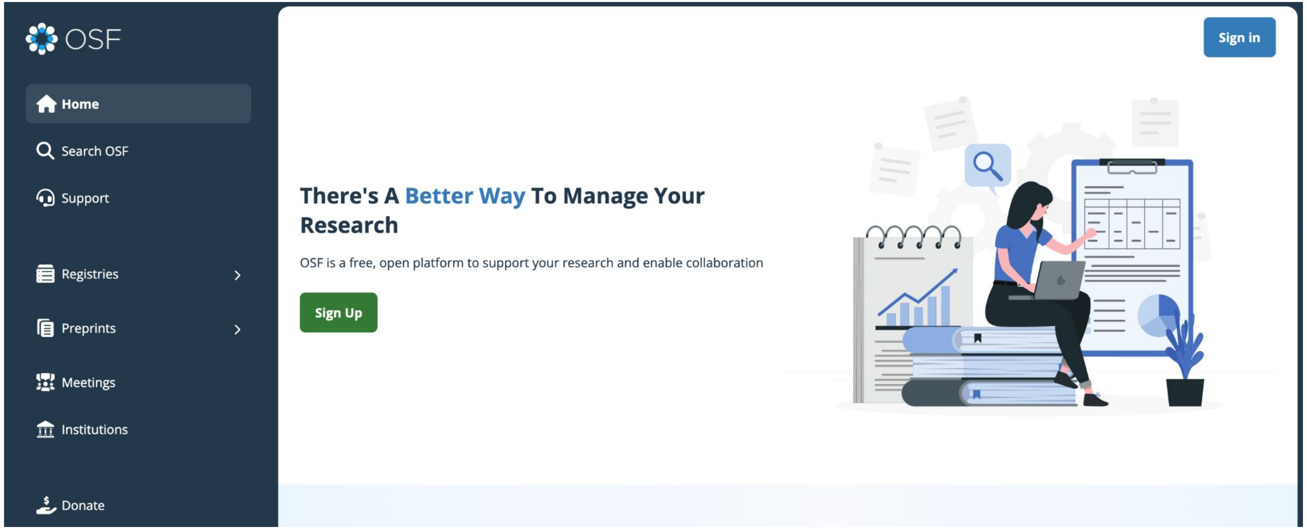
Inclusivity:

Involving and crediting more contributors to research

UC Berkeley

Open-Science Framework (OSF)

Site run by the Center for Open Science where researchers can post their data, data analysis files (e.g., R files), preprints of work, pre-registrations, etc.



There's A Better Way To Manage Your Research

OSF is a free, open platform to support your research and enable collaboration

UC Berkeley

Pre-Registration

Pre-Registration allows researchers to specify and share the details of their research before actually conducting the study.

This is useful for several reasons, such as:

(1) Preventing p-hacking and finding a narrative that fits the data after the fact.
(2) Allows researchers to essentially “call dibs” on research ideas.
(3) Forcing researchers to thoroughly think through their study design and analyses.

Two Main Sites for Pre-Registration

OSF: https://www.cos.io/initiatives/prereg

- OSF Pre-Registration Template: https://docs.google.com/document/d/1gkN0Jp6Gu7GIA4Ne4YCDZ61nCLQRgt32moRdUg9AnVg/edit?usp=sharing

As.Predicted: https://aspredicted.org/

- As.Predicted Pre-Registration Example: https://aspredicted.org/x2nd-byhj.pdf

Registered Reports

Another issue that Open Science aims to address is publication bias – journals only publishing studies that have positive results and not publishing null results.

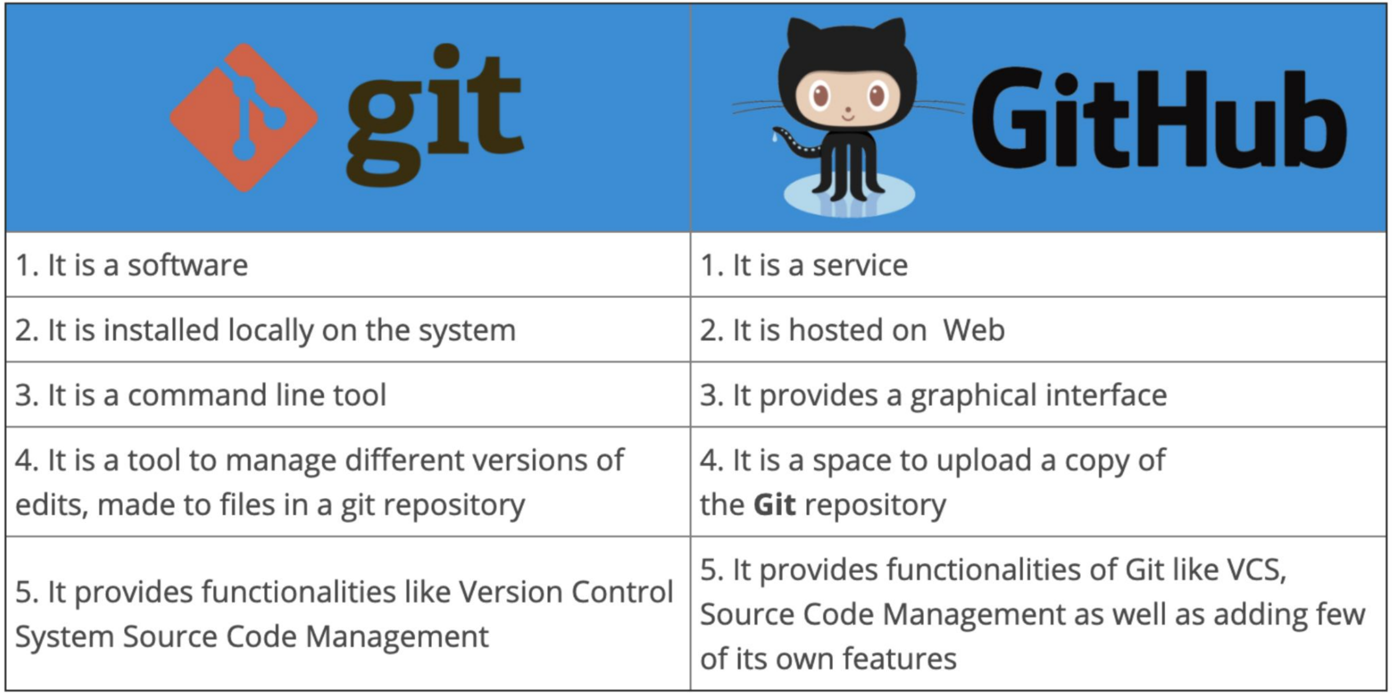
Registered Reports can help reduce this bias:

- These reports are similar to a pre-registration, where researchers can specify their design and analysis plan before running the study.
- The main difference is that these reports are submitted to a journal and undergo peer review.
- If accepted, the researchers will have a specified amount of time to conduct the study and submit the final manuscript with results to the journal, which the journal agrees to publish regardless of the results.

Git & Github

UC Berkeley

What is Git? What is Github?



git	GitHub
1. It is a software	1. It is a service
2. It is installed locally on the system	2. It is hosted on Web
3. It is a command line tool	3. It provides a graphical interface
4. It is a tool to manage different versions of edits, made to files in a git repository	4. It is a space to upload a copy of the **Git** repository
5. It provides functionalities like Version Control System Source Code Management	5. It provides functionalities of Git like VCS, Source Code Management as well as adding few of its own features

Why Use Git and Github?

(1) File Organization: They can help you organize files for projects and help keep track of and different version histories *(think something like how Google Docs keeps a log of your changes and allows you to restore a previous version)*.

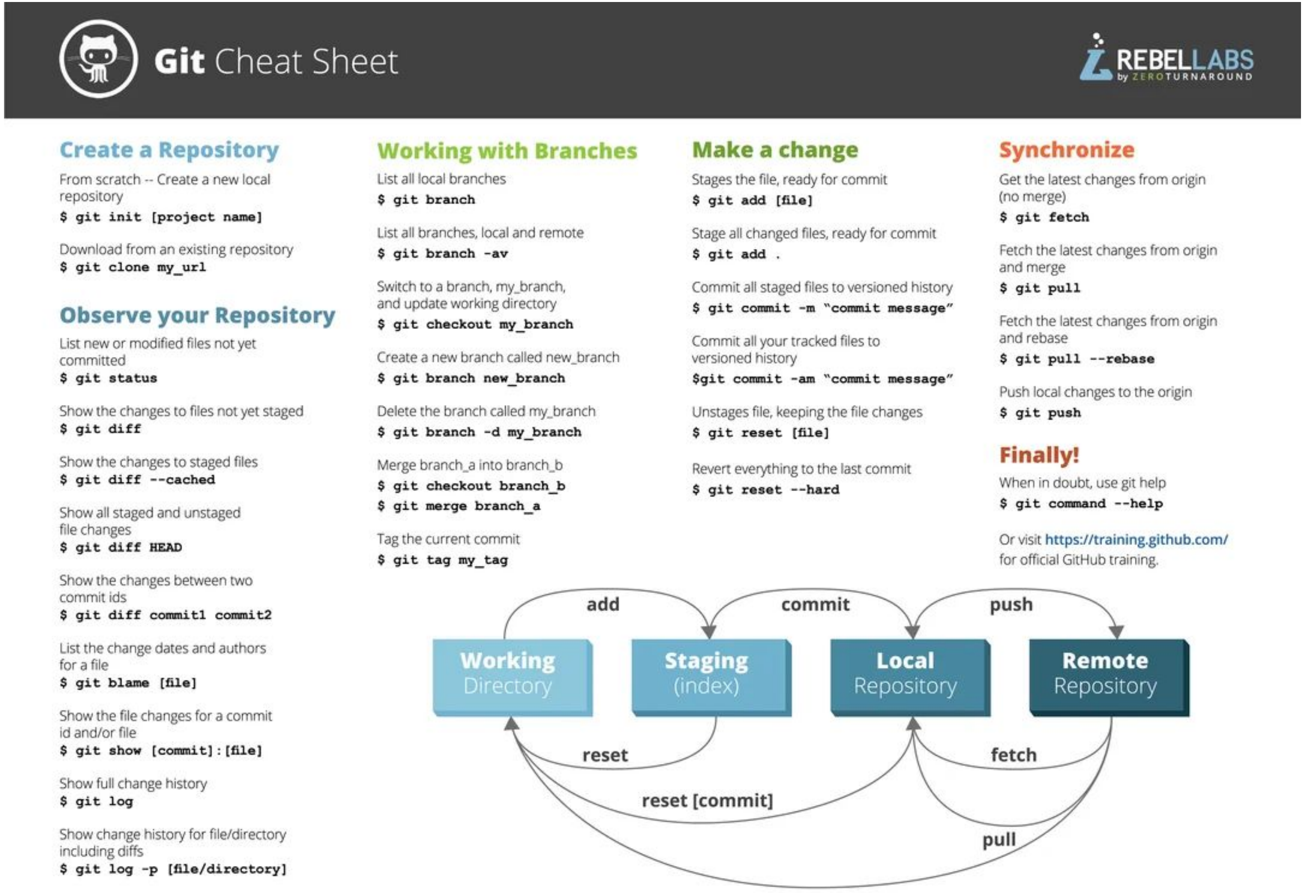
(2) Facilitating Collaboration: Sharing files and working together with others on the same files can be easier with Git and Github, since you can easily upload and modify files (and multiple files all at once) and others can easily download your new and modified files (and all of them at once!).

Commonly Used Git Commands

Command	Function	Example Use Case
`git clone [url]`	Copy a repo from a remote location onto your computer	*Nina doesn't have the Quack repo from Github on her computer, so she clones it to add new files for class*
`git pull`	Retrieve any changes that have been made to a repo since you cloned or last pulled	*Alyson cloned the Quack repo in 2024, but many changes have been made since then → She uses the "pull" command to update her local version*
`git rm`	Remove file from repo	*Nina wants to delete 2024 files from Quack 2025 repo*
`git mv`	Move file location within repo	*She also moves Week 10 files into their own folder within the Quack repo*
`git commit -m "[description]"`	Save changes made to a local repo and keeps history of changes made	*Nina saves her changes with the description "deleted 2024 files and moved Week 10 files" so she has a log of what changes she made*
`git push`	Apply local changes to remote repo	*Nina pushes her local changes to the remote Github repo so Alyson can pull them*

UC Berkeley

Git Command Cheat Sheet



Git Cheat Sheet

REBELLABS by ZEROTURNAROUND

Create a Repository

From scratch -- Create a new local repository

```
$ git init [project name]
```

Download from an existing repository

```
$ git clone my_url
```

Observe your Repository

List new or modified files not yet committed

```
$ git status
```

Show the changes to files not yet staged

```
$ git diff
```

Show the changes to staged files

```
$ git diff --cached
```

Show all staged and unstaged file changes

```
$ git diff HEAD
```

Show the changes between two commit ids

```
$ git diff commit1 commit2
```

List the change dates and authors for a file

```
$ git blame [file]
```

Show the file changes for a commit id and/or file

```
$ git show [commit]:[file]
```

Show full change history

```
$ git log
```

Show change history for file/directory including diffs

```
$ git log -p [file/directory]
```

Working with Branches

List all local branches

```
$ git branch
```

List all branches, local and remote

```
$ git branch -av
```

Switch to a branch, my_branch, and update working directory

```
$ git checkout my_branch
```

Create a new branch called new_branch

```
$ git branch new_branch
```

Delete the branch called my_branch

```
$ git branch -d my_branch
```

Merge branch_a into branch_b

```
$ git checkout branch_b
$ git merge branch_a
```

Tag the current commit

```
$ git tag my_tag
```

Make a change

Stages the file, ready for commit

```
$ git add [file]
```

Stage all changed files, ready for commit

```
$ git add .
```

Commit all staged files to versioned history

```
$ git commit -m "commit message"
```

Commit all your tracked files to versioned history

```
$git commit -am "commit message"
```

Unstages file, keeping the file changes

```
$ git reset [file]
```

Revert everything to the last commit

```
$ git reset --hard
```

Synchronize

Get the latest changes from origin (no merge)

```
$ git fetch
```

Fetch the latest changes from origin and merge

```
$ git pull
```

Fetch the latest changes from origin and rebase

```
$ git pull --rebase
```

Push local changes to the origin

```
$ git push
```

Finally!

When in doubt, use git help

```
$ git command --help
```

Or visit https://training.github.com/ for official GitHub training.

UC Berkeley

More Git Resources

Another Git command cheat sheet:
https://education.github.com/git-cheat-sheet-education.pdf

Longer + more thorough Git guide: https://git-scm.com/book/en/v2

Exit Ticket

Link: https://tinyurl.com/quack-fall2025-exit-survey