

Instructions for

Collaborating on GitHub Tutorial

Breakout 1: Connecting with your partner, create your shared repo and both clone it

- Say hello and introduce yourself
- Connect either on Slack (direct messaging), Zoom chat (make sure to select your partner, not send messages to everyone), phone, whatever you prefer
- Decide who will be Partner 1 and who will be Partner 2
- Exchange GitHub user names

Breakout 1: Connecting with your partner, create your shared repo and both clone it

- Work through sections 9.2 – 9.6 here:
<http://ohi-science.org/data-science-training/collaborating.html>

Working together on a repo

Adapted from Jenny Bryan's HappyGitwithR

<https://happygitwithr.com/>

Dealing with push rejection

You want to push changes, but you are not allowed

```
$ git push
To https://github.com/YOU/REPO.git
 ! [rejected]          master -> master (fetch first)
error: failed to push some refs to 'https://github.com/YOU/REPO.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Your local Git history and that on the GitHub remote are not compatible, i.e. they have diverged.

What is going on?

State of the repo on GitHub:

A – B – C

C represents changes your collaborator had made to B and then pushed while you were also still working on B

State of your local repo:

A – B – D

D represents changes you have made since you pulled B

What is going on?

State of the repo on GitHub:

A – B – C

C represents changes your collaborator had made to B and then pushed while you were also still working on B

State of your local repo:

A – B – D

D represents changes you have made since you pulled B

You can't merge C and D by pushing to GitHub

You have to pull the commit C and somehow integrate it into your D-containing history. Then you will be able to push again

She who pushes first wins

- Push your work to master work (when it's "ready" to share (or at least "ready enough"))
- Committing, pushing, and integrating more often, not less.
This does not eliminate the need to integrate different lines of work, but it makes each integration smaller, less burdensome, and less prone to error
- Stay in touch with your collaborators

Pulling when you have local work

State of the repo on GitHub:

A – B – C

State of your local repo:

A – B – D OR

A – B – (uncommitted changes)

Pulling when you have local work

State of the repo on GitHub:

A – B – C

State of your local repo:

A – B – D OR

A – B – (uncommitted changes)

When working on different files, versions can be merged smoothly

Remote: A--B--C Local before 'git pull':

A--B--(uncommitted changes)

Local after 'git pull':

A--B--C--(uncommitted changes)

Pulling when you have local work

State of the repo on GitHub:

A – B – C

State of your local repo:

A – B – D OR

A – B – (uncommitted changes)

This also works when you have made edits to different parts of the same file

Remote: A--B--C Local before 'git pull':

A--B--(uncommitted changes)

Local after 'git pull':

A--B--C--(uncommitted changes)

Pulling when you have local work

State of the repo on GitHub:

A – B – C

State of your local repo:

A – B – D OR

A – B – (uncommitted changes)

When you have made changes to the same part of a file, you will need to resolve the merge conflict locally before you can push

Remote: A--B--C Local before:

A--B--D

Local after:

A--B--D--(merge commit)
 C/

GitHub Pages

Examples of cool GitHub pages websites

- We're starting simple, but GitHub Pages has functionality to make complex and compelling websites. A few examples
 - <https://twitter.github.io/>
 - <https://justanother-timetraveller.github.io/>