

Git Workflows - Do's and Don'ts

Do's:

- Separate branches
- Separate notebooks
- Save off “cleaned” data
- Add only the files you ‘changed’
- Group sync
 - Beginning and end of day
- Clear notebook outputs
- Ask for help!!!

Dont's:

- git add .
- Work in the same notebook
 - At the same time
- Push to main/master
- Ignore conflicts (force merge)
- Leave notebooks/servers up
- Pull request without communicating with group



Common Git Errors & How to Resolve

- Accidental typo in commit message
 - **git commit --amend**
- Forgot file in last commit
 - **git add** <missing file name>
 - **git commit --amend**
- Trying to push and receiving “upstream” error
 - **git push -u origin** < branch>
 - Sets upstream for all future push/pull actions

Common Git Errors & How to Resolve

- Added unwanted file into repo
 - **git reset** <filename>
- Added unwanted file AND committed
 - **git reset --soft HEAD~1**  Undos commit
 - **git reset** <filename>  Unstages file
 - **git commit -m** “message”  Re commit
- Escape unwanted merge
 - **git merge --abort**

Common Git Errors & How to Resolve

- Accidentally committed all changes to master/main
 - **git branch** <new branch name>  New branch
 - **git reset HEAD~ --hard**  Rolls back master
 - **git checkout** <new branch name>
- Misspelled branch name
 - **git branch -m** <old-name> <new-name>
- Misspelled branch name AND already pushed
 - **git push origin --delete** <old-branch-name>
 - **git push origin** <new-name>

Last Resorts

If you have exhausted all other options and possibilities, and are still stuck with a git error and unable to move forward, you can essentially rewind your actions as a last resort.

Save all your changes in a duplicate location outside your repo before doing this!!!

- **git log** or **git reflog** to see all actions you have taken
- Find index of where you want to return to (unique ID)
- **git reset HEAD @ {index}**

Nuke from orbit option. AGAIN save all changes in a duplicate location outside the repository!!!

- **Delete local repository** - terminal or finder
- **Re-clone** locally from github

What Worked - What To Do Better

Worked:

- Splitting investigation/analysis
 - Independent but share b/t members
- Creating 'outline' or vision to project
- Communicate who is working on what and where
 - Syncing repos
- Open to group member ideas/exploration
- Troubleshoot code together
 - PP, sharing screen
- Assign responsibility for deliverables
 - Ownership
- Divide and conquer
 - Regular check-ins
- Visualizations

Do Better:

- MVP, final numbers before writeups
- Outline deliverables early on
- Markdown and comment as you go
- Spend time as group setting up repository
- Setting up group checkpoints
- *Working off same dataset

6 BENEFITS OF GROUP WORK

1. You're exposed to a Variety of Perspectives

2. You Improve your Vocabulary

3. You Learn to Teach

4. You learn to Manage Personalities

5. You can Leverage the Talents of your Peers

6. You Learn Negotiating Skills

Questions?