

# Version Control for Researchers

vASB 2020 Tutorial

Ryan Alcantara, Gary Bruening, & Ross Wilkinson

# Tutorial Outline

- **Introduction to Version Control w/ Git & GitHub** (30 min)
- Breakout & Recap #1: Collaborating via GitHub (30 min)
- Breakout & Recap #2: Undoing Big Changes (20 min)
- Future Considerations (10 min)

# What is version control?

analysis.m      analysis\_v1.m      analysis\_v2.m



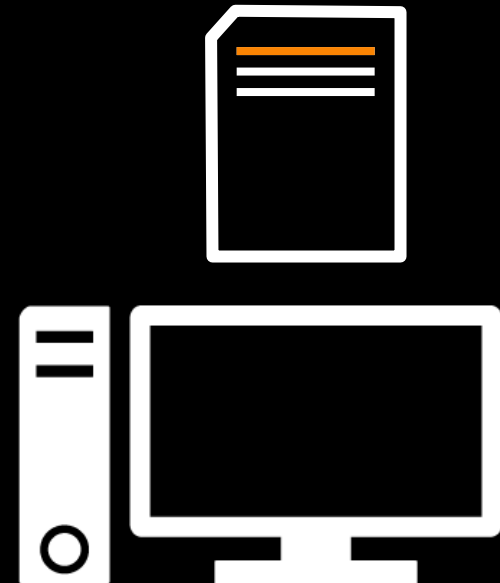
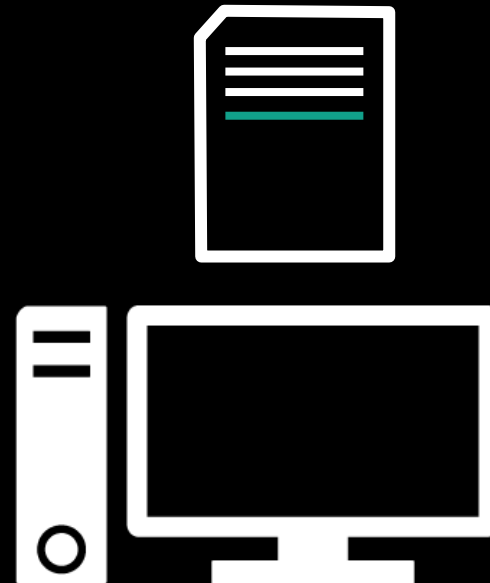
# Why use Git?

- Improve project organization
- Recall line-by-line changes



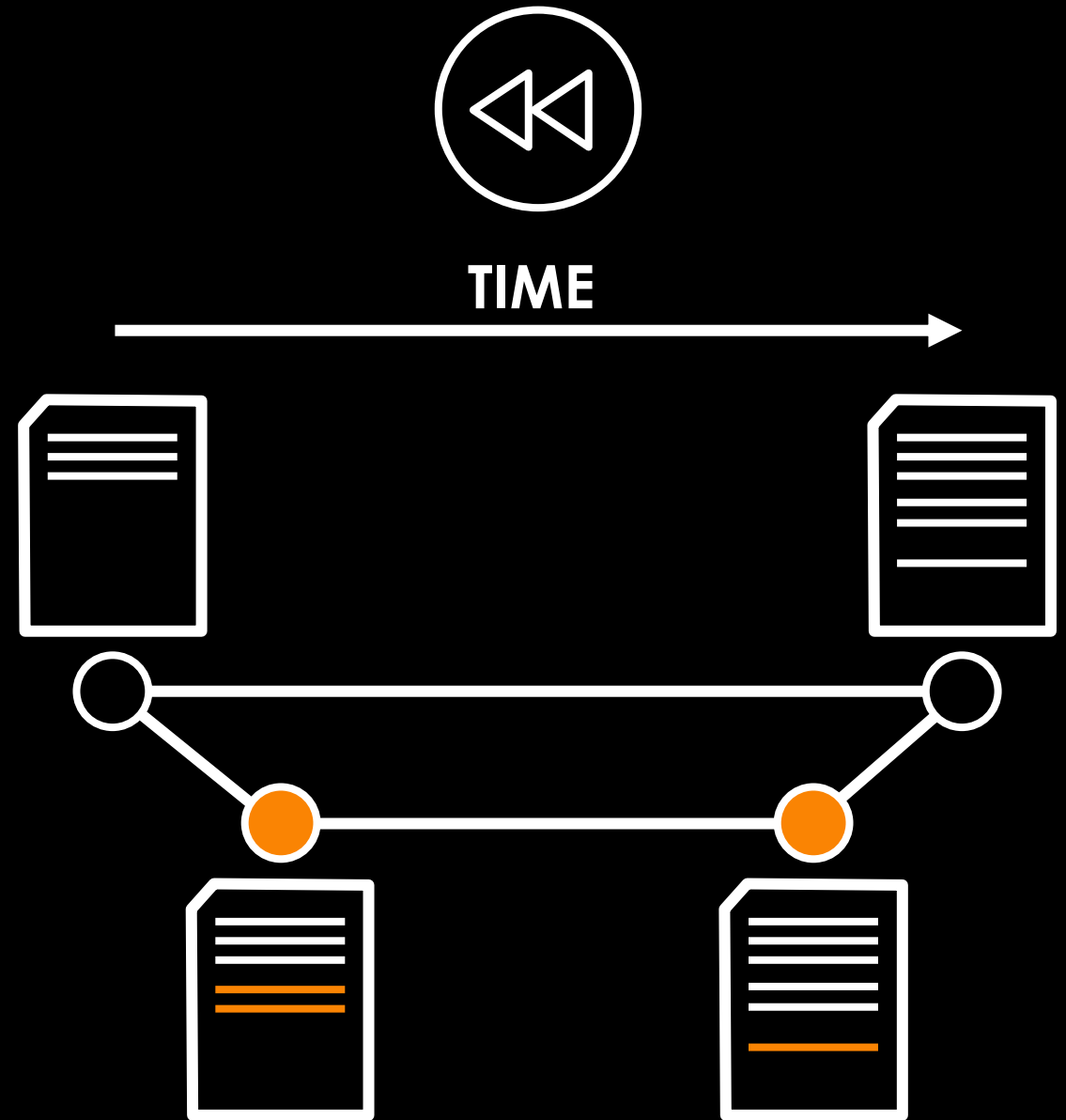
# Why use Git?

- Improve project organization
- Recall line-by-line changes
- Synchronize code between collaborators

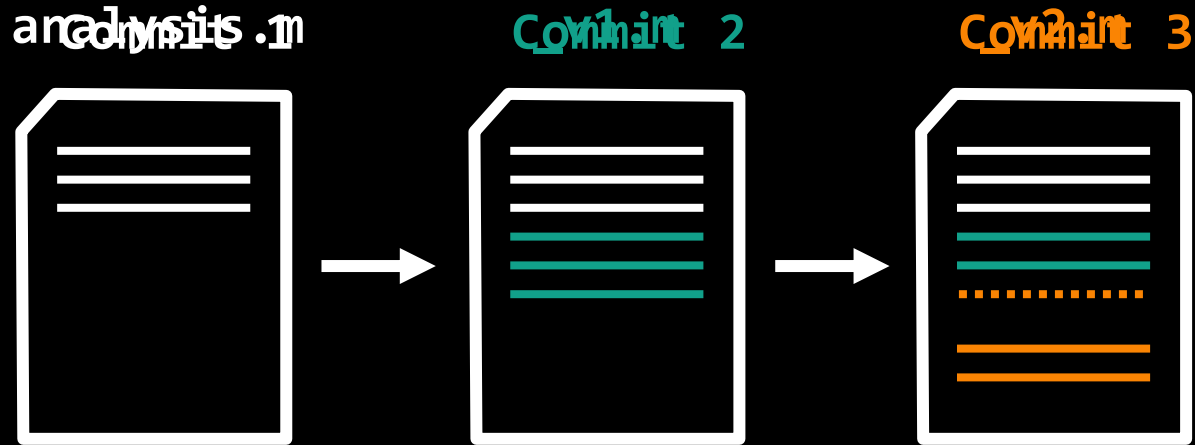


# Why use Git?

- Improve project organization
- Recall line-by-line changes
- Synchronize code between collaborators
- Safely make changes to code
- Revert files to prior state



# Terminology



- **Repository (repo)**
  - Folder containing your project
- **Git**
  - Records changes to local repo
- **GitHub**
  - Place to host repo remotely
- **Commit**
  - “Snapshot” of what tracked files look like at that moment

|          |            |       |  |
|----------|------------|-------|--|
| Commit 1 | 2020-05-01 | Alice | “created analysis.m”                   |
| Commit 2 | 2020-05-25 | Alice | “added feature X”                      |
| Commit 3 | 2020-07-10 | Alice | “added feature Y, fixed feature X bug” |

# 3 States of Files under Version Control

## 1. Modified

- You've changed a file, but not committed it yet

## 2. Staged

- You've marked a modified file to be included in next commit (snapshot)

## 3. Committed

- Changes safely recorded locally (in a hidden folder called `.git`)



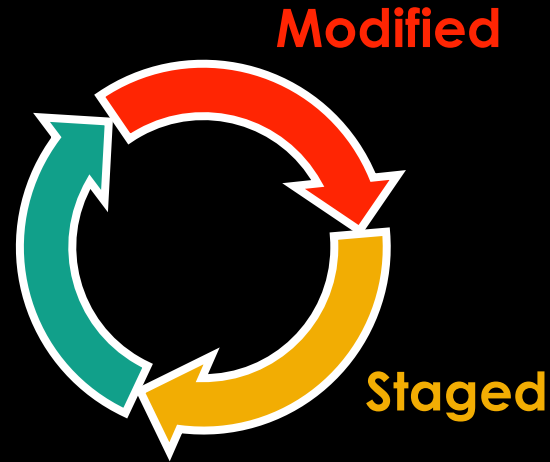


# Git + GitHub



# Basic Workflow

Committed



URL

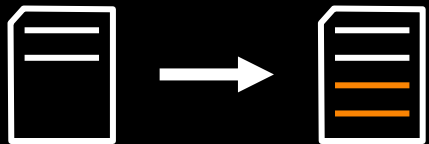
git pull

git push

Working Directory

Staging Area

Local Repository



git add FILE

git commit

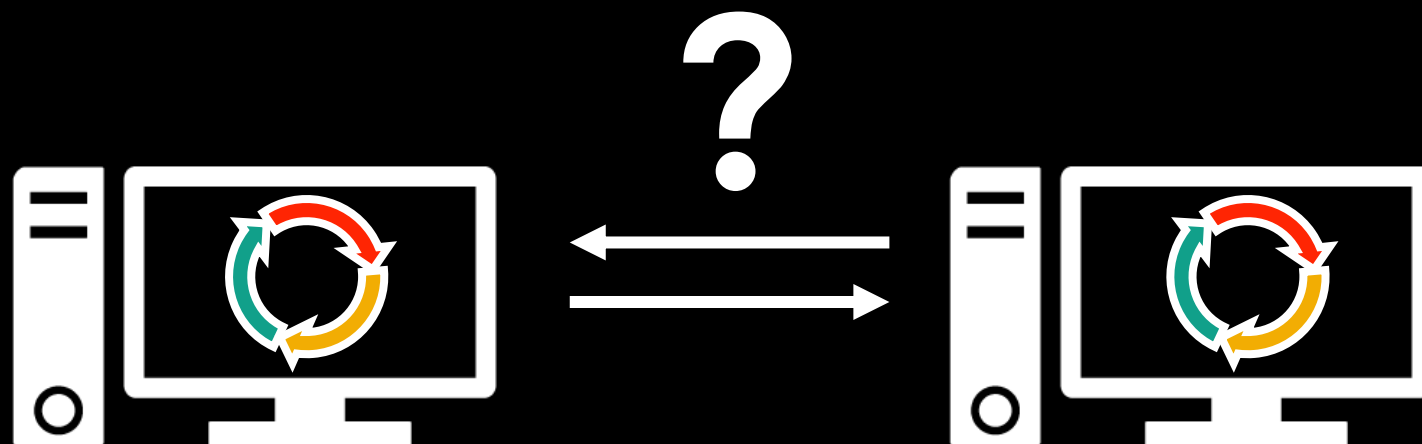
|          |            |
|----------|------------|
| Commit 1 | 2020-05-01 |
| Commit 2 | 2020-05-25 |
| Commit 3 | 2020-07-10 |



# Collaboration

*“Email the script”*

*“USBs handed back and forth (pre-COVID)”*

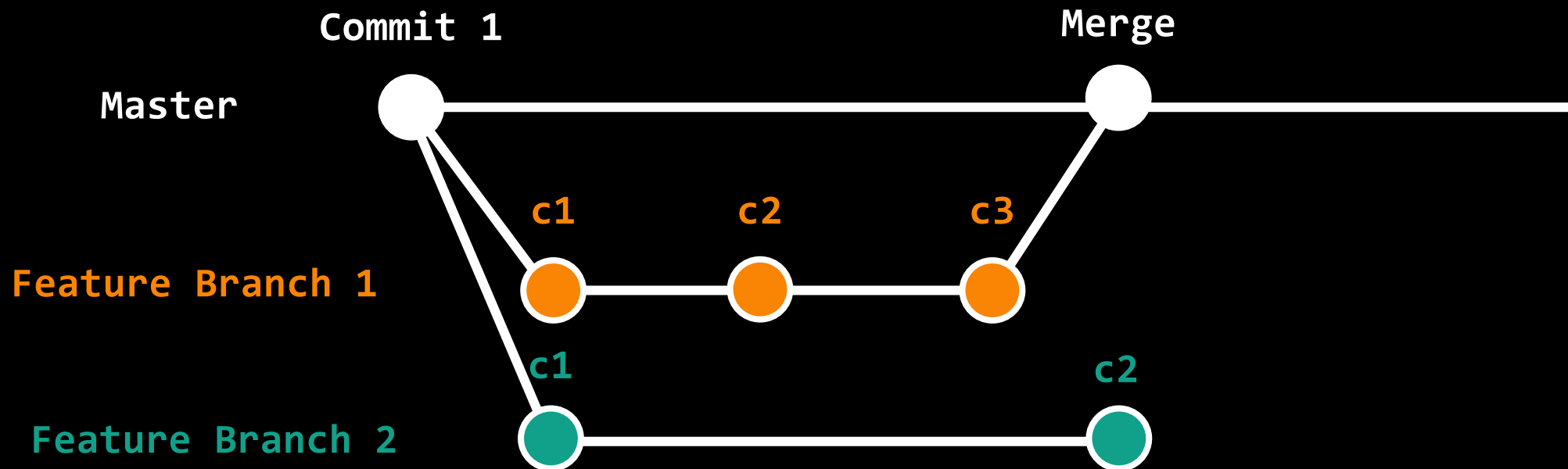


*“Change name of script to Name\_v1”*

*“Shared codes stored on lab computer”*

# Branches

Keep Master  
Branch Functional

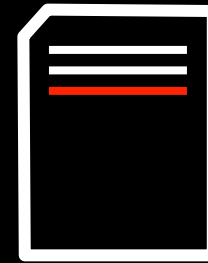


# Merge Conflicts

```
# script you're working on  
plot(x, y)  
<<<<<<< HEAD  
xlim([0,10])  
=====  
xlim([-10, 10])  
>>>>>>> COMMIT IDENTIFIER  
ylim([5, 25])
```

Code on your local version

Code on GitHub version

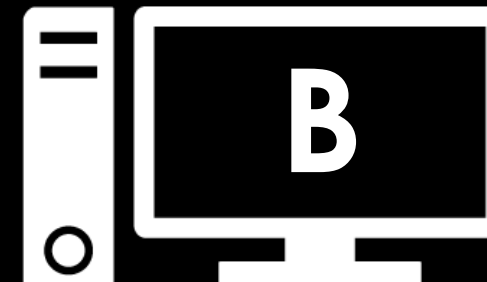
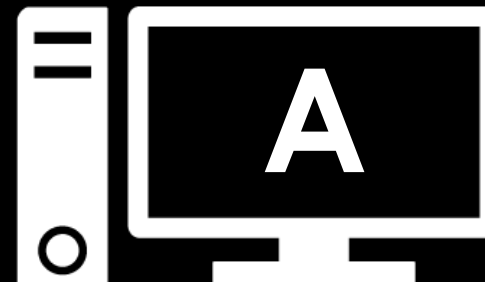
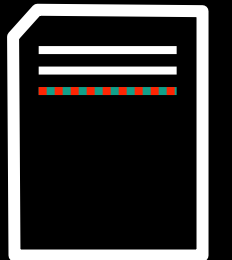


push

pull

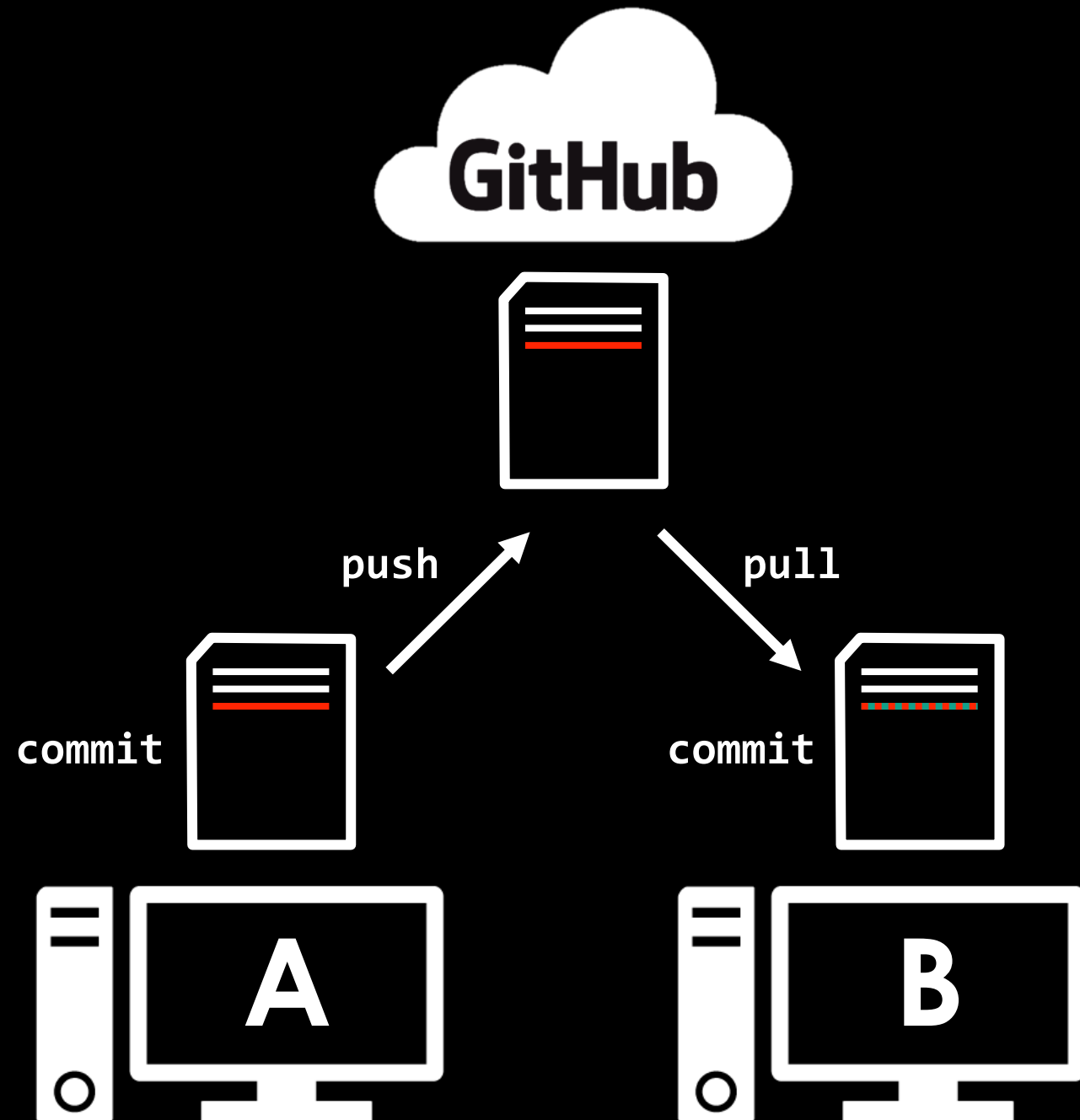
commit

commit



# Merge Conflicts

```
# script you're working on  
plot(x, y)  
xlim([-10, 10])  
ylim([5, 25])
```



# Breakout Room Instructions

- Groups of 5 where you will practice using some of these commands while collaborating via GitHub.
- Introduce yourselves!
- Helpers are available for questions or troubleshooting!