Session 4

Intro to Pandas

Loading Data

Cleaning Data

9.24.19

Link to Jupyter Notebooks:
https://mybinder.org/v2/gh/data-voyage-solutions/oag-session-mats/master

01

Review/Wrap Up

1 hour

02

Loading data with Pandas

1 hour

03

Common data cleaning tasks

1 hour

04

Git / GitHub

Remaining time

Schedule/ Topics

Meeting Date	Module	Sub-topic
~~8/20/19~~	~~1: Python Fundamentals~~	~~Control Flow Part 1 and Dictionaries~~
~~8/27/19~~		~~Control Flow Part 2~~
~~9/3/19~~		~~Versioning Control (Git)~~
9/24/19	2: Data Wrangling/ Preparation	Loading data/Intro to Pandas
10/1/19		Common data cleaning tasks
10/8/19		Common errors encountered & solutions
10/15/19	3: EDA & Intro to Visualizations	Basic summary/descriptive statistics
10/22/19		How to choose the right/best chart
10/29/19		How to create different visuals in Python
11/5/2019	4: Visualizations (e.g., Bokeh)	Design principles/Formatting
11/12/2019		Interactive visuals
11/19/2019		Creating dashboards

Review

Practice Set 1

Common Data Cleaning Tasks

- *Load Data*
- *Inspect data*
- Rename columns
- Drop columns
- Data types
- Drop duplicates
- ...
- ...

Git
Version Control

Resources for *The Basics*

- https://try.github.io/
  - https://github.com/jlord/git-it-electron#what-to-install
  - https://learngitbranching.js.org/

What's the point?

Git is a program for keeping track of changes over time, known in programming as ***version control***.

If you've used a track changes feature in a text editing software then you're already familiar with the concept!

Lingo: Repository

- Collection of related files for a project.
- Think of it as a **project folder** that is tracked by Git.
- Called "repo" for short.

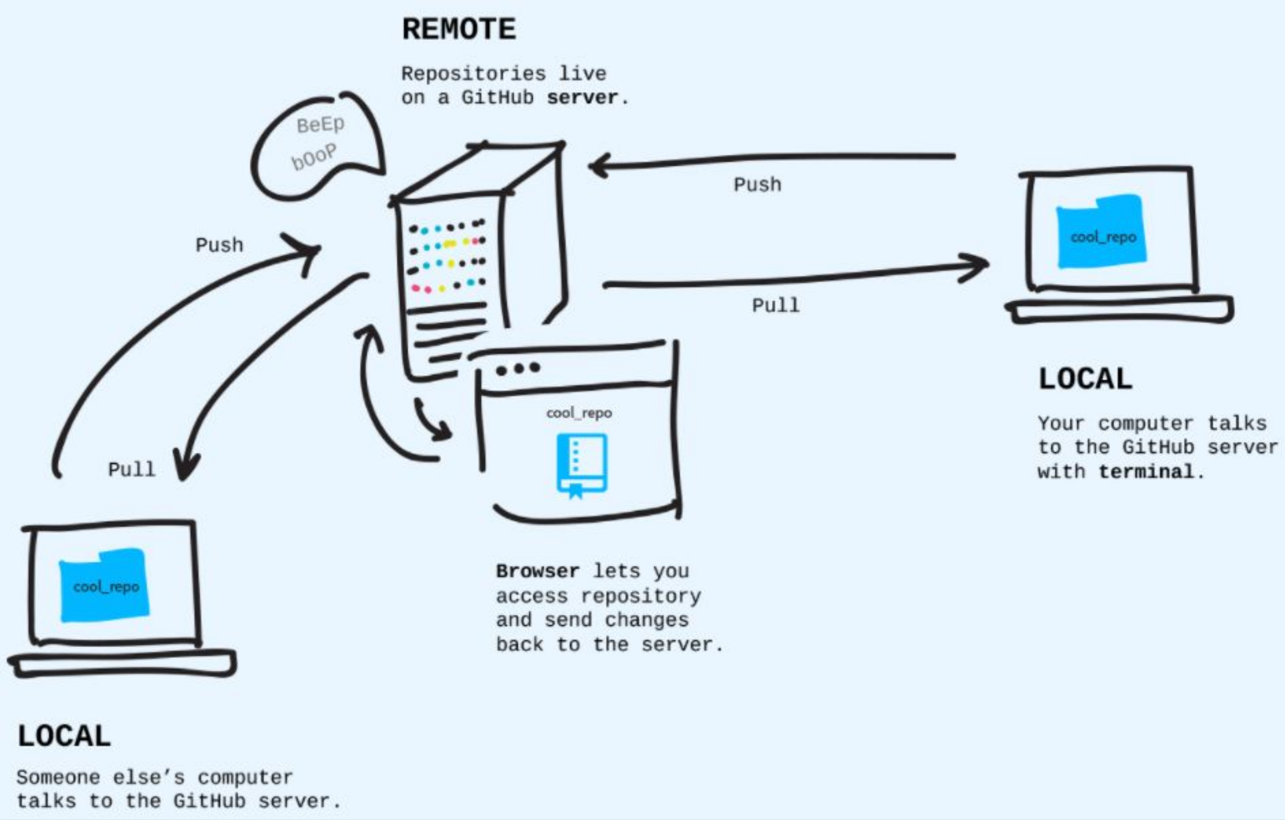
In order for you to be able to share and collaborate with others (without giving them access to your computer), you use GitHub.

- GitHub acts as a central repository for you and everyone else to share.
- Push changes to it and pull down changes from others.

Lingo: Remote Repository

- A repo that lives on one of GitHub’s servers.
- By **pushing your local changes** to a remote, you are updating the remote repo.
- By **pulling** your updated changes **down from the remote** repo, collaborators can get the latest from your work.

Diagram about Repos



REMOTE

Repositories live on a GitHub **server**.

LOCAL

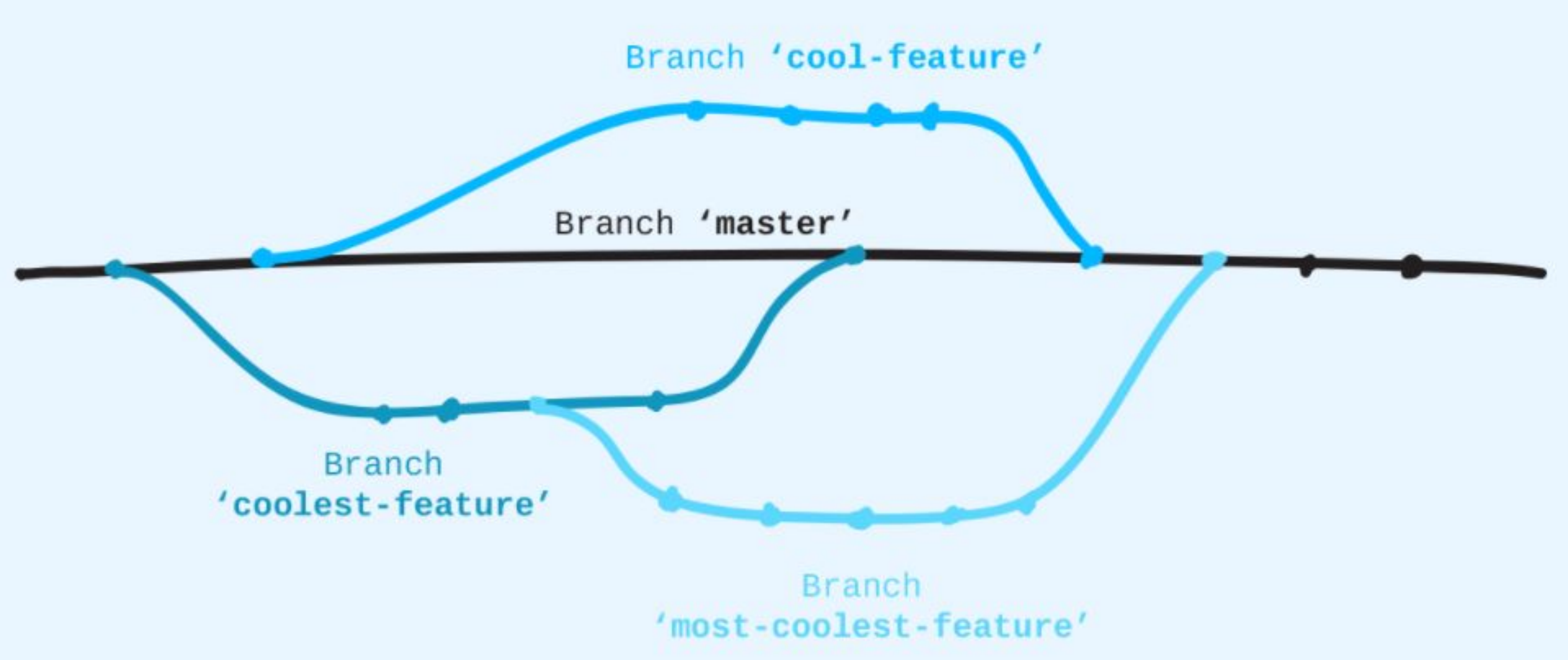
Your computer talks to the GitHub server with **terminal**.

Browser lets you access repository and send changes back to the server.

LOCAL

Someone else's computer talks to the GitHub server.

Feature Branch Workflow



Our Git Workflows

- Functions folder/Project Templates
- Team Member A's projects
- Team Member B's projects

- Master and "DA Stage" Branch
  - Project A
    - Branch: EDA
    - Branch: Cleaning/Preprocessing
    - Branch: Analysis 1
  - Project B
  - Project C

NOTE:

We will use **`git rebase`** for our merges.

Let's have a practice run!

Assuming Git is installed and already configured on your local computer:

- ❑ Open terminal or shell
- ❑ Navigate to a desired ***parent*** directory
- ❑ One way to set-up: Clone a remote repo on GitHub
- ❑ Navigate to cloned repo on your local computer
- ❑ Make some changes to the local repo
- ❑ Push changes to the remote repo:
    - ❑ git status
    - ❑ git diff
    - ❑ git add <filename> or .
    - ❑ git commit -m "ur commit msg" (aka save history...with a short message)

“Round Robin” Game

1. Starting spot: https://github.com/orgs/data-voyage-solutions/dashboard
2. Create a new remote repo
3. Add collaborators
4. Kelly starts the round:
   a. **git clone a remote repo to local**
   b. **make some changes and save**
   c. **save history of changes**
   d. **push changes to remote repo from local (update remote repo)**
5. Next person up! Complete #4 steps, one person at a time.

“Round Robin” Game -- Round 2

Once Round 1 has been completed:

1. Kelly starts the round:
   a. **Check status of local repo**
   b. **Do a `git pull`! It’s like an update…**
   c. **Check the logs….vs a diff**
2. Everyone else, at the same time (except the last person that pushed changes)! Complete #1 steps.