GIT & GIT HUB

In Git and GitHub, several commands are commonly used for version control, collaboration, and managing repositories in data science projects. Here’s a list of essential Git commands and how they are used:

1. Repository Management

* **Clone a Repository**:

Command: git clone <repository-url>

* + Example: git clone https://github.com/user/repository.git
  + This command creates a local copy of a GitHub repository on your computer.
* **Create a New Repository**:

Command : git init

* + This initializes a new Git repository in your current directory.

2. Working with Files

* **Check Status**:

Command : git status

* + Shows the status of your working directory and staging area (e.g., which files are modified, staged for commit, etc.).
* **Add Files to Staging Area**:

Command : git add <file-name>

* + Example: git add data.csv
  + Adds a specific file to the staging area.
  + To add all changes: git add .
* **Remove Files**:

Command : git rm <file-name>

* + Removes the file from the working directory and stages the removal for the next commit.

3. Committing Changes

* **Commit Changes**:

Command : git commit -m "Commit message"

* + This records changes in the repository with a descriptive message.
* **Amend Last Commit**:

Command : git commit --amend

* + Allows you to edit the previous commit message or add additional changes.

4. Branching and Merging

* **Create a New Branch**:

Command : git branch <branch-name>

* + Example: git branch feature-xyz
  + Creates a new branch.
* **Switch to a Branch**:

Command : git checkout <branch-name>

* + Example: git checkout feature-xyz
  + Switches to the specified branch.
* **Create and Switch to a New Branch**:

Command : git checkout -b <branch-name>

* + Example: git checkout -b feature-xyz
  + Creates a new branch and switches to it.
* **Merge a Branch**:

Command : git merge <branch-name>

* + Merges the specified branch into the current branch.
* **Delete a Branch**:

Command : git branch -d <branch-name>

* + Deletes the specified branch.

5. Collaboration

* **Pull Changes from Remote Repository**:

Command : git pull origin <branch-name>

* + Fetches and merges changes from the remote repository.
* **Push Changes to Remote Repository**:

Command : git push origin <branch-name>

* + Pushes committed changes to the remote repository.
* **Fork a Repository**:
  + On GitHub, you can fork a repository via the GitHub website, which creates a copy of the repository under your account.
* **Create a Pull Request**:
  + After pushing your changes to your GitHub fork, you can create a pull request to propose your changes to the original repository.

6. Viewing History

* **View Commit History**:

Command : git log

* + Shows the history of commits in the current branch.
* **View a Specific Commit**:

Command : git show <commit-hash>

* + Displays detailed information about a specific commit.

**7. Undoing Changes**

* **Revert Changes**:

Command : git revert <commit-hash>

* + Reverts the changes made by a specific commit.
* **Reset to a Previous Commit**:

Command : git reset --hard <commit-hash>

* + Resets the current branch to a specific commit, discarding all subsequent changes.

These commands form the foundation of using Git and GitHub for version control, especially in collaborative data science projects. They allow you to manage your project’s history, collaborate with others, and ensure your work is versioned and backed up.