URGS analysis

# April 10

## Set up github

**1. Set Up Your Local Repository**

1. **Navigate to Your Project Folder:**  
   Open your terminal (or command prompt) and navigate to the folder that contains your project files.

cd /path/to/your/project-folder

1. **Initialize a Git Repository:**  
   Run the following command to initialize your project folder as a Git repository:

git init

1. **Create a .gitignore File (Optional but Recommended):**  
   Create a .gitignore file to exclude files you don’t want to track (e.g., virtual environments, cache files). An example for a Python project might be:

gitignore

# Python cache files

\_\_pycache\_\_/

\*.py[cod]

# Virtual environment

venv/

env/

# Jupyter Notebook checkpoints

.ipynb\_checkpoints/

Save this file in your project root.

1. **Add Your Files to the Staging Area:**  
   Add your project files using:

git add .

1. **Make an Initial Commit:**  
   Commit your files with a message:

git commit -m "Initial commit"

**2. Create and Connect to a GitHub Repository**

1. **Create a New Repository on GitHub:**
   * Log in to your GitHub account.
   * Click the "+" icon in the top right and select "New repository".
   * Give your repository a name, choose its visibility (public/private), and click "Create repository".  
     *(For more details, see* [*GitHub’s Create a Repo guide*](https://docs.github.com/en/get-started/quickstart/create-a-repo)*).*

*https://github.com/AndrewDilley/URGS.git*

1. **Link Your Local Repository to GitHub:**  
   In your terminal, add a remote pointing to your new GitHub repository. Replace <YourUsername> and <RepoName> with your details:

git remote add origin https://github.com/AndrewDilley/URGS.git

If your local branch is still called master but GitHub expects main, you can rename your branch:

git branch -M main

1. **Push Your Code to GitHub:**  
   Push your committed changes to the remote repository with:

git push -u origin main

**Verification**

* **Check Your Repository:**  
  Open your GitHub repository page in a web browser to verify that your files have been successfully uploaded.
* **Future Commits:**  
  For future changes, simply repeat the following:
  1. git add .
  2. git commit -m "Your commit message"
  3. git push