git init .

git config --global user.email "your@email.com"

git config --global user.name "your name"

git status

git add .

git commit -m "first commit"

In Github, create a new public repository called “chunks”.

Do NOT check “Initialize with README”. You want a blank repository.

Create a folder somewhere (Remember good workflow. Maybe move it to your Documents folder or a new folder for Projects?) and also call it “chunks”.

Copy and paste the files, including 01\_chunk.html and 01\_chunk.rmd files into that “chunks” folder.

Then in terminal:

navigate to your folder, e.g.

cd /Users/james/Desktop/Rprogramming/PRACTICE/Seamounts

git init

git add .

git commit -m "first commit"

git remote add origin git@github.com:Sahelanth/Seamounts.git

git push -u origin master

Refresh your Github repo and all your files should be there

github.io is the address for hosted files.

What if you get some new data and wanted to update this?

Change the data imported in the local project

Generate a new HTML from the .Rmd file

git add the files in the terminal/cmd

git commit -m with the message “new data” or whatever

git push -u origin master to move the new file changes to Github

Refresh the URL address and the changes should be reflected!

Let’s say you’ve created a repo on Github. You’ve set up your local project folder as a repo with “git init” in your console. You’ve typed in “git remote add origin git@github.com:username/yourrepo.git”

What two commands are next?

1. git commit -m “first commit for this project”
2. 2. git push -u origin master

Provide summarized data

While it’s excellent to include the scripts that detail the cleaning and wrangling process it took to turn raw data into the polished set you’ve published, there is a large audience of people who just want to download and play with the finalized data.

Include a folder in the repo that you can point them to so they don’t have to dig through your methodology to reproduce the summarized data.

.gitignore

Use .gitignore to exclude certain files from being uploaded to GitHub. Such as:

Files larger than 100 mb

GitHub will refuse to upload the data

Files with your keys or passwords

Extraneous files like your R history

You can borrow this .gitignore file for inspiration.

Include readmes and data dictionaries

Let people know what they’re dealing with.

Be as specific as possible, including where you got the data from.

Buzzfeed is a good model for how they index their story links and repos as a table

Licensing

Be sure to include a license in each repo.

This lets others know that it’s open source, and sets the limits on how people can use, change, or distribute your work.

For example, The Washington Post usually publishes their work in GitHub under an Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license.

This means users can share, copy, and redistribute our data in any medium or format and can remix, transform, and build upon our work. However, they must give appropriate credit and indicate if any changes were made. And they must not use it for commercial purposes and must also share their work under the same license.

There’s also the MIT license, which is very similar.

Have a discussion with your folks and remain consistent.