Connecting to GitHub

Andrew Ba Tran

Contents

This is from the seventh chapter of learn.r-journalism.com.

Github is one of web-based hosting services for sharing and collaborating with code and data.

It all depends on your needs and preferences. Each are pretty similar.

Options

• Github
• BitBucket
• GitLab

We’re going to publish a folder to Github because it offers free hosting to open-source software projects.

It’s already where most R packages are hosted when they’re being developed.

**Warning:** If you’re using two-factor authentication in Github, you’ll need to go through some more steps for generating an SSH key to connect. This shouldn’t concern you if you’re using Github after just creating a new account.

Creating a Github repository

Go to Github.com and create a new account if you haven’t yet. Use the same username and email address that you set up with your git global option.

For this example, we’re going to upload our folder of chunks that we created in the Publishing chapter.

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

Do NOT check “Initialize with README”. You want a blank repository.
This will create a new repository page, which needs to be synchronized with the local repository we’re about to create.

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.
Open up your CMD/terminal and navigate to the folder.

For me, that would look like:

```
[Andrews-MacBook:images andrewtran$ cd /Users/andrewtran/Projects/chunks
Andrews-MacBook:chunks andrewtran$]
```

Type in the git commands we learned before to initialize, add, and commit.

```
git init
git add .
git commit -m "first commit"
```
Now we need to connect this repository on the local computer to the repository on Github.

The lines were provided when you created the repository on Github:
Type the equivalent of that into your terminal/cmd:

```sh
git remote add origin git@github.com:YOURUSERNAME/chunks.git
git push -u origin master
```
Refresh your Github repo and all your files should be there:

```bash
create mode 100644 03_chunk.Rmd
create mode 100644 03_chunk.html
create mode 100644 04_chunk.Rmd
create mode 100644 04_chunk.html
create mode 100644 05_chunk.Rmd
create mode 100644 05_chunk.html
create mode 100644 06_chunk.Rmd
create mode 100644 06_chunk.html
create mode 100644 07_chunk.Rmd
create mode 100644 07_chunk.html

Andreas-MacBook:chunks andrewtran$ git remote add origin git@github.com:andrewtran/chunks.git
Andreas-MacBook:chunks andrewtran$ git push -u origin master

Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 4 threads.
Compressing objects: 100% (16/16), done.
Writing objects: 100% (16/16), 723.58 KiB | 2.28 MiB/s, done.
Total 16 (delta 10), reused 0 (delta 0)
remote: Resolving deltas: 100% (10/10), done.
To github.com:andrewtran/chunks.git
  * [new branch]  master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
Andreas-MacBook:chunks andrewtran$
```
Now you can get those HTML files hosted.

**Hosting with Github Pages**

Turn http://github.com/username/reponame/index.html ->
Into http://username.github.io/reponame/index.html

github.io is the address for hosted files. Can you spot what to swap?

First we need to set up Github to host the files.

**Setting up GH Pages**

After uploading your repo, click on **Settings** on the top right.
Scroll down to the GitHub Pages section of Settings.

Change the Source from None to master branch and click save.

For future projects, you can organize your files so that all output HTML are stored in a docs folder. Then the option master branch\docs will host those files and ignore everything in the root folder.

Now go to https://YOURUSERNAME.github.io/chunks/06_chunk.html and you should have your hosted file!
That’s it!

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

1. Change the data imported in the local project
2. Generate a new HTML from the .Rmd file
3. git add the files in the terminal/cmd
4. git commit -m with the message “new data” or whatever
5. git push -u origin master to move the new file changes to Github
6. Refresh the URL address and the changes should be reflected!

And that is that!

With this published repo, you’re now part of the reproducible journalism community.

Last thing we’ll go over are some tips and best practices.