NR995 Module 9

2017 Fall
Part II: Configuring Git, GitHub, & RStudio
Installing git

Git comes installed with many operating systems. Regardless, it can't hurt to make sure you're working with the latest version. Git is available for free from:

https://git-scm.com/downloads

Once it's installed, you're all set to use git locally to track your files. To reap the full benefits of using git, however, you'll need to create an account with an online hosting service. We'll use GitHub since it's the most common and has some very handy features, though there are others that have their own benefits (e.g., BitBucket).

One of the appeals of GitHub is the online community. There is a very large user base, it is a common hosting site for public repositories, and GitHub automatically makes many code- and text-based files instantly readable. This makes it a simple, easy place to build a presence on the web since so much is done automatically for you.

You can create a free account on GitHub with unlimited public repositories. If you connect your UNH email, you can also receive a number of free private repositories through their educational program. Since your username will be your main identifier, it's best to choose something related to your real name. You can always change it, but it's a bit of a hassle.

Configuring RStudio

You all should have RStudio installed already. If not, it is free for download. Once you have both Git and RStudio installed, open RStudio.

- From the **Tools** menu, open the **Global Options**
- On the pop up window, click the **Git/SVN** tab on the sidebar
- Check the box for *Enable version control interface for RStudio projects*
- If available, check the box for *Use Git Bash as shell for Git projects*
- Restart RStudio for the changes to take effect.

After restarting RStudio, you should see a tab next to *Environment* and *History* that says *Git*, as well as a dropdown *Git* icon at the top.

- Click the *Git* tab
- Click the gear icon that says *More*

- Click *Shell..*

This opens a window with a bash shell. There are two things to do from the command line - only two - and you only need to do it once. In the shell, run the following two lines of code, using your real name (or GitHub username) and the email address you used when creating your GitHub account:

```
git config --global user.name "Neville Longbottom"
git config --global user.email "tentacula@hogwarts.edu"
```

Git on your computer now knows your name and email address for connecting with GitHub. When you make a commit, it will be labelled with the name you entered here.

Creating your first repo

There are several ways to start a new repo and connect it locally through RStudio. In my experience, the simplest is to start from GitHub.

Part I: GitHub

- Open GitHub in your browser
- Click the "+" symbol on the top toolbar and select *New repository*
- Type 'test_repo' as the *Repository Name*
- Initialize with a 'readme', add a .gitignore (select 'R'), and add a license if you'd like
- Click **Create repository**
- Click the green 'Clone or download' button and copy the URL in the popup window

Part II: RStudio

- From the **File** menu, select **New Project...**
- Selet **Version Control** > **Git**
- Paste the URL in the *Repository URL* box
- Name the folder on your computer in the *Project directory name* box
- Choose where to place the folder on your computer in the *Create project as subdirectory of* box
- Click **Create Project**

Stage, commit, push

You now have a local git repository on your computer connected to the remote git repository on GitHub. Test it out:

- Open the README.md file in the RStudio project you just created
- Add a line with 'test commit 1'
- Save the file
- In the *Git* panel, click the check box next to README.md
- Click the **Commit** button
- Enter 'test commit' in the *Commit message* box and click **Commit**
- Click the **Push** button
- Enter your GitHub username and password when asked
- Check on GitHub that the file has been updated

Making it easy: Storing your credentials

In order to push your local commits to GitHub, git needs your login information. The default option is to ask you to type in your username and password each time you push or pull. This is annoying. Instead, there are two options. We will do the first:

1. Store your username and password with a credential helper
    - i. Windows
        - a. Open the Shell again (Git tab > More > Shell..)
        - b. Run `git config --global credential.helper wincred`
        - c. If that doesn't work, download Git Credential Manager and run the .exe file
        - d. Close the shell and proceed to *iii. Testing Stored Credentials* below
    - ii. Mac
        - a. Open the Shell again (Git tab > More > Shell..) or a Terminal window
        - b. Run `git config --global credential.helper osxkeychain`
        - c. Close the shell and proceed to *iii. Testing Stored Credentials* below
    - iii. Testing Stored Credentials
        - a. In RStudio, open the README.md file and add a line with 'test commit 2'
        - b. Save the file
        - c. Stage & commit the changes (don't forget a commit message!)
        - d. Push the commit & enter your username and password
        - e. Push again -- you should not be asked for your username and password
2. Set up SSH keys -- more secure, but more complicated. See this tutorial.