How to connect GitHub/Git/RStudio

Created by Gina Nichols (vnichols@iastate.edu) who uses a PC. I can't help you Mac users...
Last updated March 22 2024

Step 1 – Download

1. Download and install the latest versions of R and RStudio.
2. Download and install Git (https://Git-scm.com/downloads). Make sure you keep a note of where it was installed – you will need to find it again later. Accepting all of the defaults is fine.

Step 2 – Sign-up

1. Create a GitHub account online at https://github.com.

NOTE: You can link multiple emails to one account – it doesn't really matter whether you use your personal or school email. Under certain circumstances your school email might give you access to some functionality for free. Remember your username and email, you have to give this information to RStudio, which is hard to do if you forget.

2. Verify your email address.

Step 2 – Prepare a little

1. Create a folder on your desktop titled 'GitHub_repos' or something like that. It will be used to hold things temporarily. I would not suggest putting it in a cloud-based folder. It will annoy you with little updates about files being changed ALL THE TIME. And there is no reason those files should be accessible to anyone. It's literally just a buffer folder.

Step 3 – Connect GitHub to RStudio via
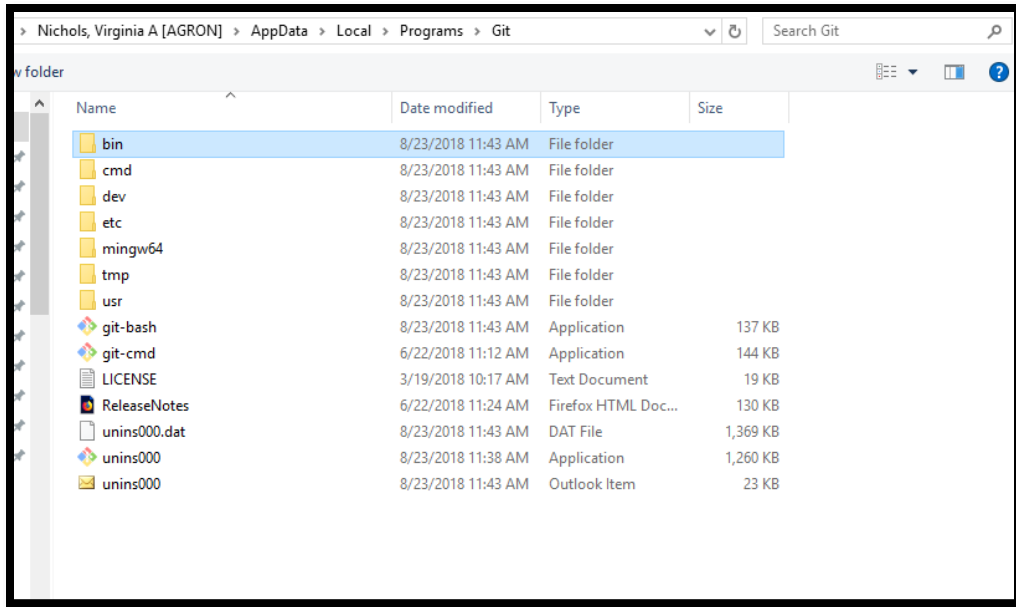
1. In R Studio, go to Tools → Global Options
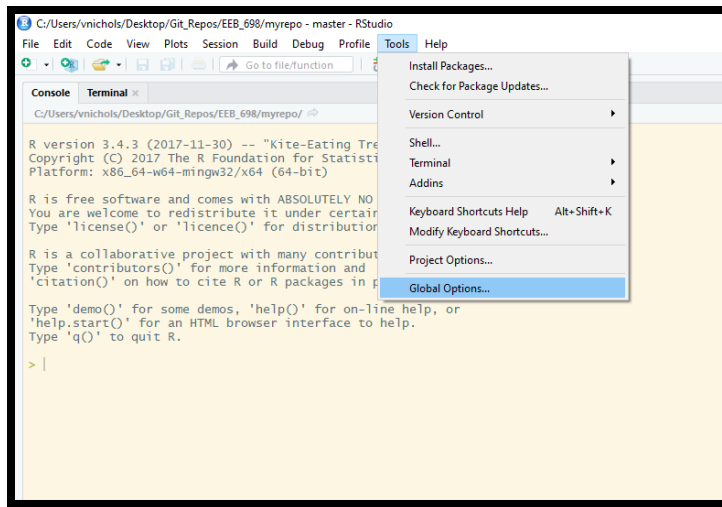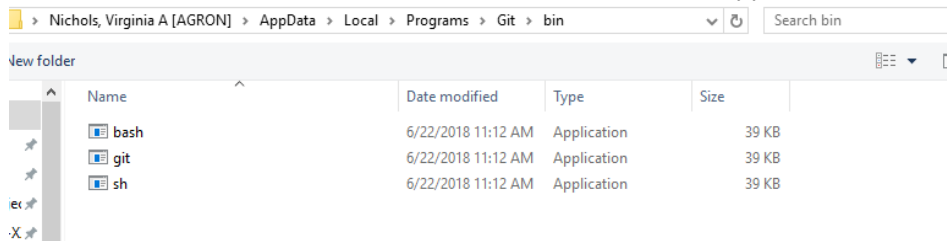
Select Git/SVN on the left-hand menu.
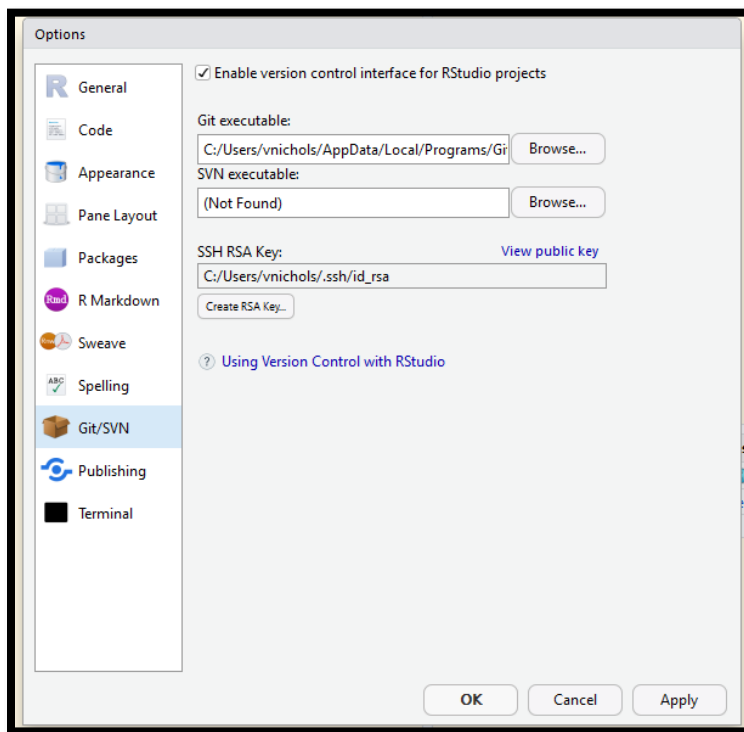
2. Make sure "Enable version control" is checked.
3. In the Git executable, browse to where you saved 'Git'. This can be tricky.

For a PC:
The Git folder might look something like this.

You'll want to select the 'bin' folder, then select the Git application in this folder.


For now, ignore the SVB executable. It's optional, and can be set up later. It's related to file security.

2. Hit "*Create RSA Key…*"
This creates the key in a popup window. It looks like gibberish, that's good. Hit Close.

3. Click "*View public key*" (this will only be available after you create a key). Copy the output.

4. Go to your GitHub online account. Click on your little picture and select *Settings.*
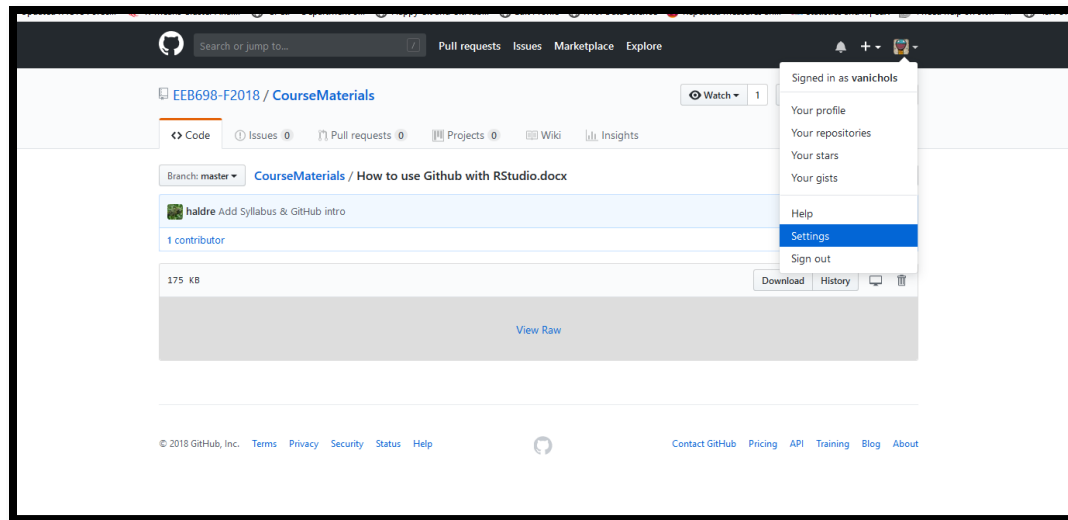
One the lefthand menu, select *'SSH and GPG keys'*. Select *'New SSH key'*, and paste the copied link from RStudio here.

You have now linked GitHub to RStudio. They will communicate via Git software.

Step 4 – Connect RStudio to GitHub

1. Type 'shell' in the windows search bar, and click on the 'Command Prompt' icon. This will open up the terminal. It will look scary, but don't worry.
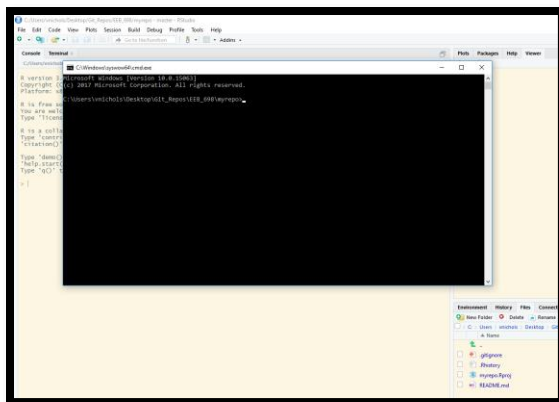
2. Tell RStudio and Git your GitHub information so it can label changes. Type the following into the terminal, it's code so type it EXACTLY. Spaces, characters, etc. All exactly.

Git config --global user.email "youremailaddress"
(replace youremailaddress with your email that you used for GitHub)
For example I will type:
Git config --global user.email "virginia.nichols@gmail.com"

Git config --global user.name "yourGitname"
*(replace yourGitname with your GitHub username)*

*For example I will type:*
Git config --global user.name "vanichols"

You have now given RStudio all the information it needs to communicate with GitHub and vice versa. Nothing will look different, it's anticlimactic.

Step 4 – Test it

1. Go to your home page on GitHub. Click on 'Repositories', then the 'New' button (it's green).
   a) Name it whatever you want, it will be a useless repository. You could even name it 'useless'. Leave it public. Click the box '*Initialize this repository with a README'.* You'll see what that does in a second.
   b) Click the green '*Create repository*' button (green).
   c) Click on the '*Clone or download*' button (again, green). You are cloning with HTTPS. Just copy the link, either using ctl+C or the little clipboard-looking-button.
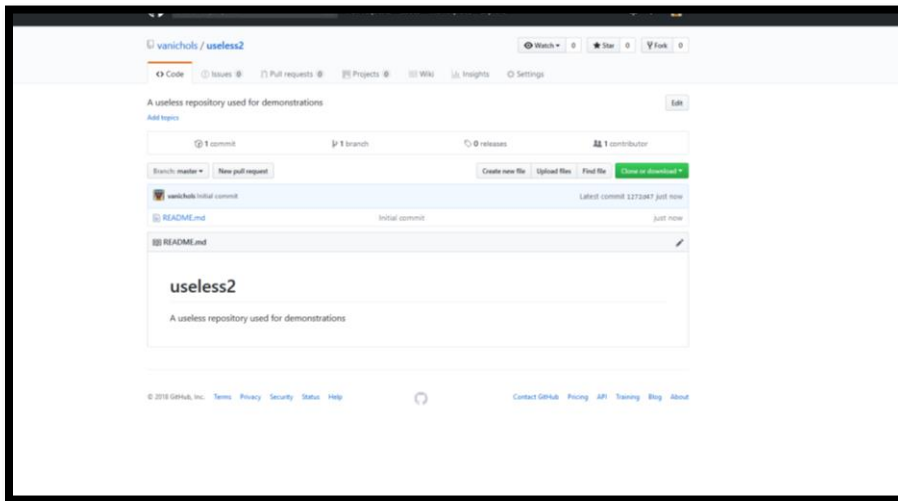
2. Go back to RStudio. If you don't normally use R Projects, just do it for the sake of this tutorial. It won't matter.
   a) Click File → New Project
   b) Select 'Version Control'
   c) Select 'Git'
   d) Paste the URL you copied from GitHub in the '*Repository URL'* box.
   e) Name the directory whatever you want.
   f) Create the project as a subdirectory of that temporary GitHub folder you made in **Step 3.**
   g) Now you should see a Git tab in your RStudio. I have my windows arranged differently than most, but here you can see it.
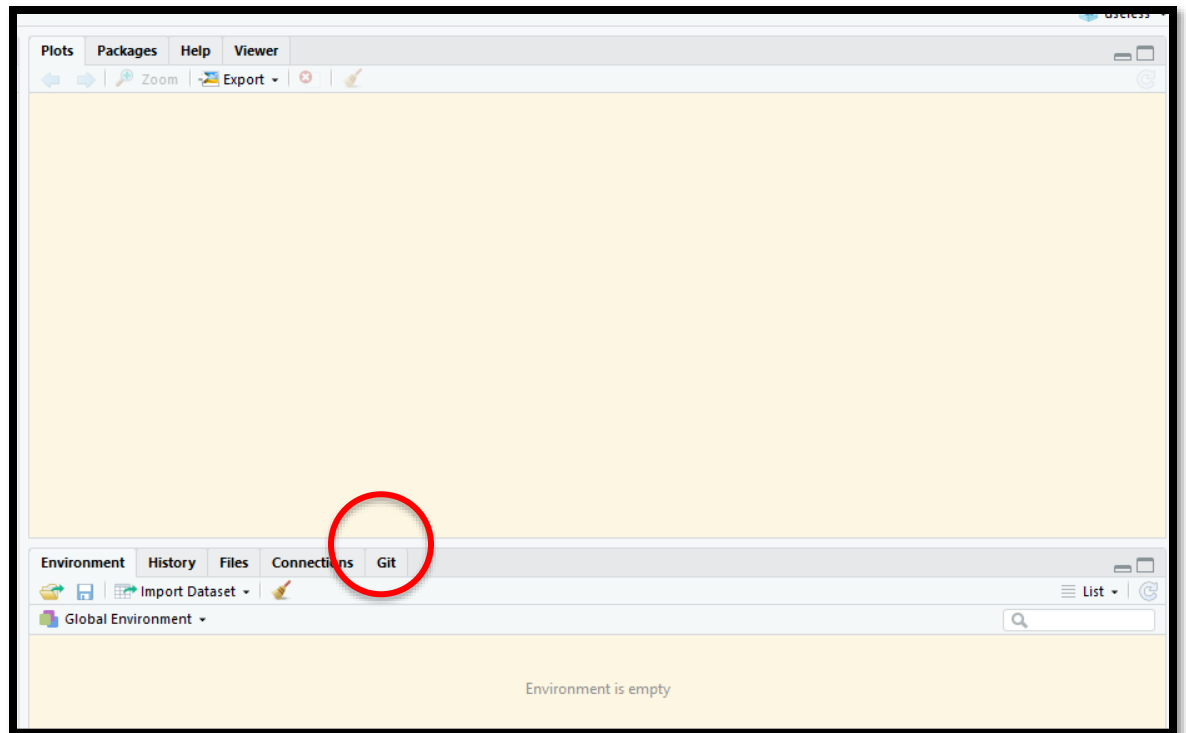
These are the first steps in getting everything connected. There are lots of ways to interact with files once they are connected – those ways are for another time.

Cheat sheet for common git commands to use in terminal

- git add . *(this stages everything that has changed, the . means 'everything')*
- git commit -m "updated documentation"
- git push