How to connect GitHub/Git/RStudio

Created by Gina Nichols (vnichols@iastate.edu) who uses a PC. I can't help you Mac users…

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.

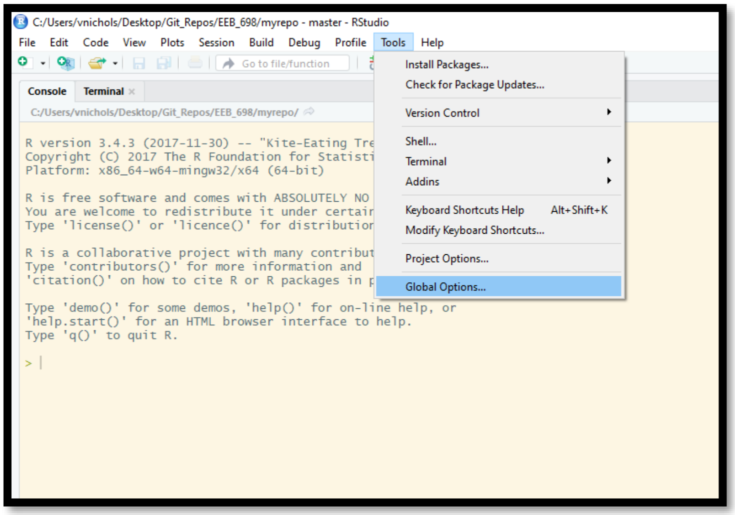
2. Verify your email address.

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.

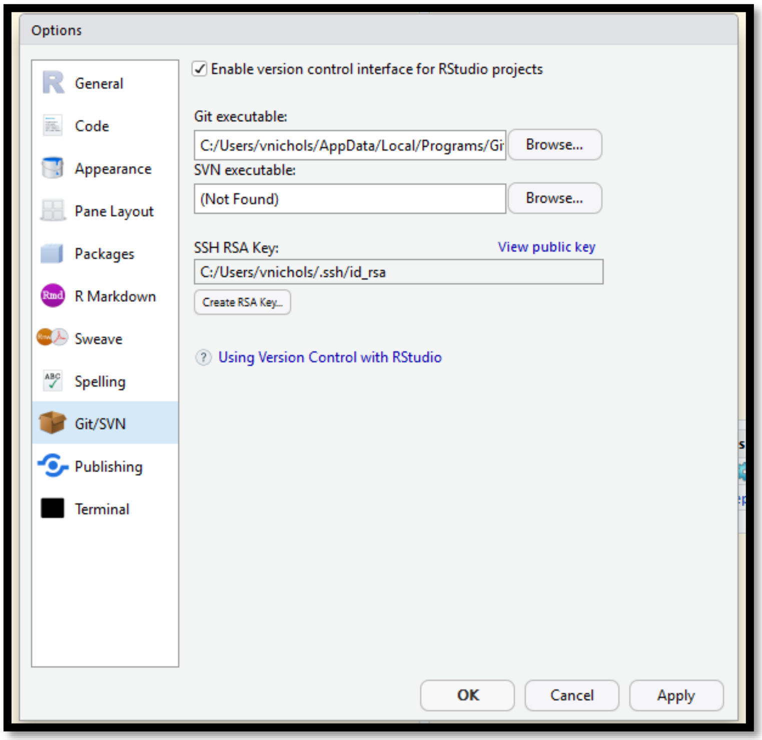
Step 3 – Connect GitHub to RStudio via

1. In R Studio, go to Tools → Global Options


C:/Users/vnichols/Desktop/Git_Repos/EEB_698/myrepo - master - RStudio
File
Edit
Code
View
Plots
Session
Build
Debug
Profile
Tools
Help
Go to file/function
Console
Terminal
C:/Users/vnichols/Desktop/Git_Repos/EEB_698/myrepo/
Install Packages...
Check for Package Updates...
Version Control
Shell...
Terminal
Addins
Keyboard Shortcuts Help Alt+Shift+K
Modify Keyboard Shortcuts...
Project Options...
Global Options...

```
R version 3.4.3 (2017-11-30) -- "Kite-Eating Tre
Copyright (C) 2017 The R Foundation for Statisti
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO
You are welcome to redistribute it under certair
Type 'license()' or 'licence()' for distributior

R is a collaborative project with many contribut
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in p

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

Select Git/SVN on the left-hand menu.



☑ Enable version control interface for RStudio projects

Git executable: C:/Users/vnichols/AppData/Local/Programs/Gi

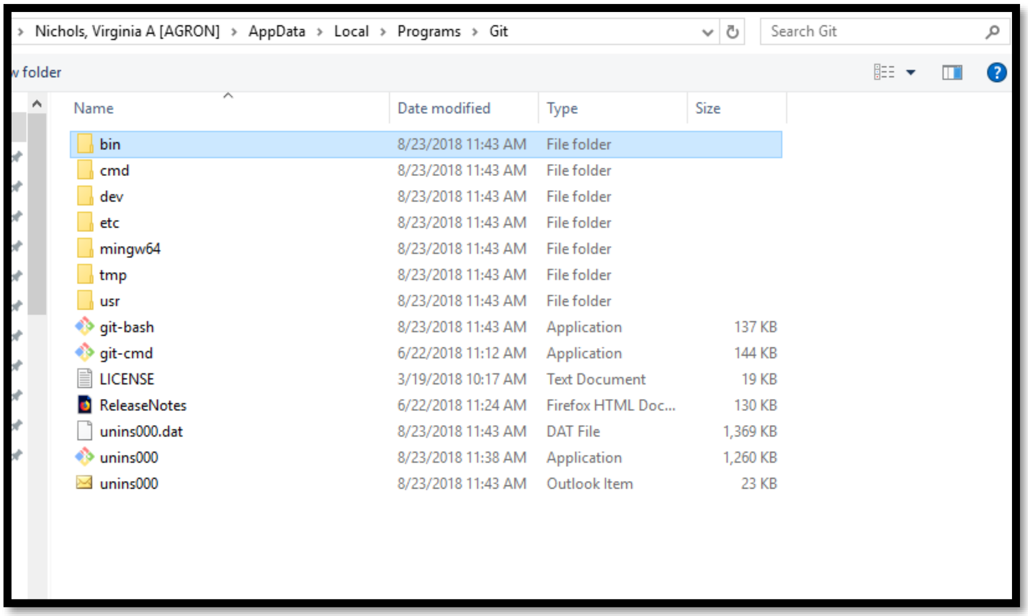
SVN executable: (Not Found)

SSH RSA Key: C:/Users/vnichols/.ssh/id_rsa

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:

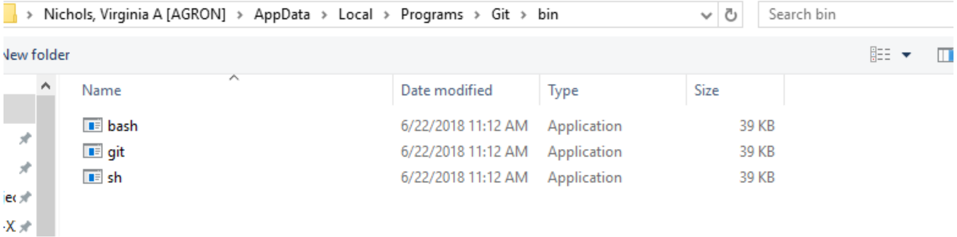
The Git folder might look something like this.



› Nichols, Virginia A [AGRON] › AppData › Local › Programs › Git

Name	Date modified	Type	Size
bin	8/23/2018 11:43 AM	File folder	
cmd	8/23/2018 11:43 AM	File folder	
dev	8/23/2018 11:43 AM	File folder	
etc	8/23/2018 11:43 AM	File folder	
mingw64	8/23/2018 11:43 AM	File folder	
tmp	8/23/2018 11:43 AM	File folder	
usr	8/23/2018 11:43 AM	File folder	
git-bash	8/23/2018 11:43 AM	Application	137 KB
git-cmd	6/22/2018 11:12 AM	Application	144 KB
LICENSE	3/19/2018 10:17 AM	Text Document	19 KB
ReleaseNotes	6/22/2018 11:24 AM	Firefox HTML Doc...	130 KB
unins000.dat	8/23/2018 11:43 AM	DAT File	1,369 KB
unins000	8/23/2018 11:38 AM	Application	1,260 KB
unins000	8/23/2018 11:43 AM	Outlook Item	23 KB

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



› Nichols, Virginia A [AGRON] › AppData › Local › Programs › Git › bin

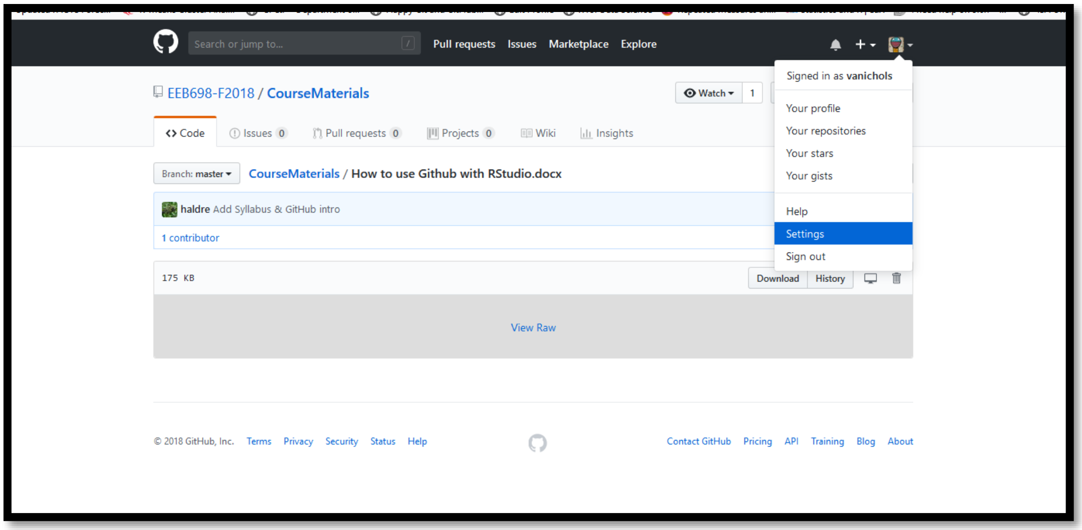
Name	Date modified	Type	Size
bash	6/22/2018 11:12 AM	Application	39 KB
git	6/22/2018 11:12 AM	Application	39 KB
sh	6/22/2018 11:12 AM	Application	39 KB

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.*

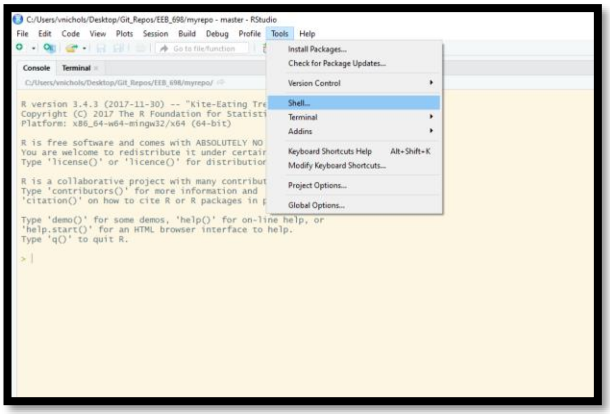


One the lefthand menu, select *'SSH and GPG keys'*. Select *'New SSH key'*, and paste the copied link from RStudio here.

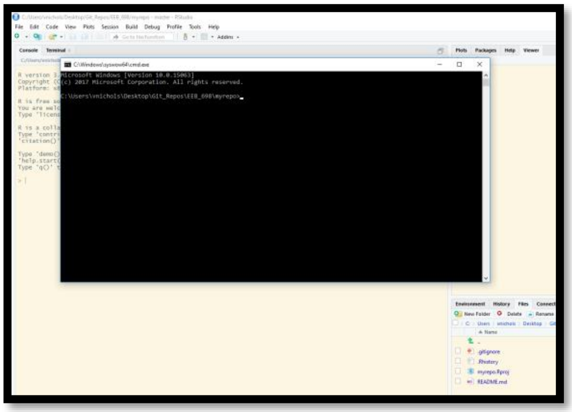
You have now linked GitHub to RStudio. They will communicate via Git software.

Step 4 – Connect RStudio to GitHub

1. Return to RStudio. Select Tools → Shell. This will open up the terminal. It will look scary, but don't worry.



```
R version 3.4.3 (2017-11-30) -- "Kite-Eating Tre
Copyright (C) 2017 The R Foundation for Statisti
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO
You are welcome to redistribute it under certair
Type 'license()' or 'licence()' for distributior

R is a collaborative project with many contribut
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in p

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```



```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\vnichols\Desktop\Git_Repos\EEB_698\myrepo>_
```

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)

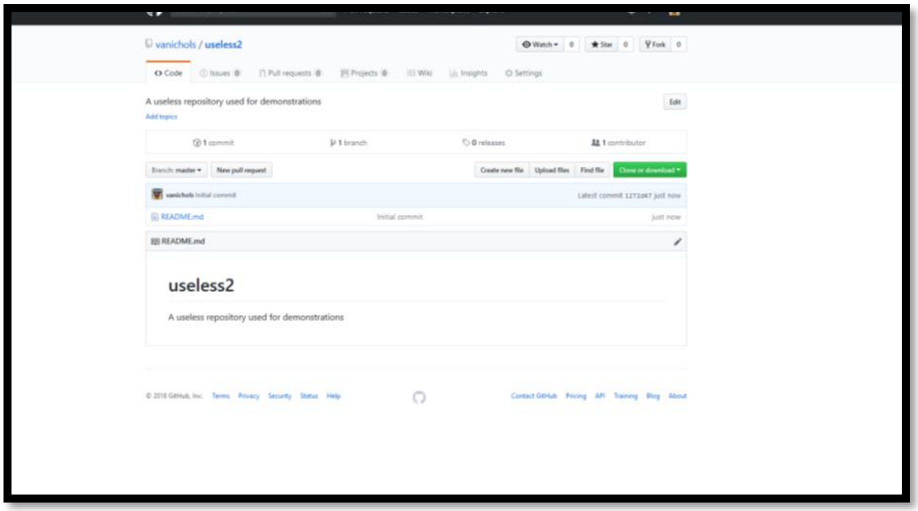
```
Git config --global user.name "yourGitname"
```

(replace "yourGitname" with your GitHub username)

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.



vanichols / useless2

A useless repository used for demonstrations

vanichols Initial commit — Latest commit 1272d47 just now

README.md — Initial commit — just now

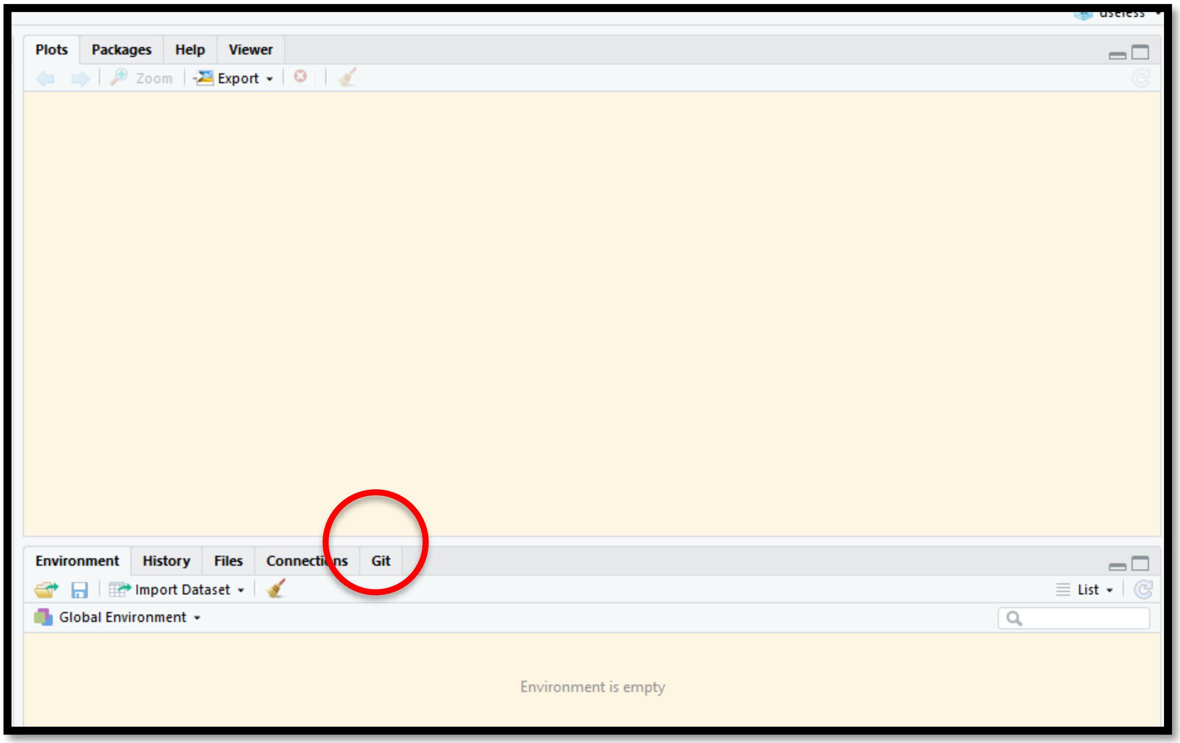
useless2

A useless repository used for demonstrations

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.

Resources

Jenny Bryan, a software engineer and adjunct faculty at University of British Columbia, has a great resource that gives a much more in-depth exploration of this process and of Git, GitHub, RStudio, and R.

http://happyGitwithr.com/