Module 1 – lesson 06

Script – part 1

So what is Github? It’s a cloud repository, which hosts things like code, files and documents. It’s very similar to Dropbox, Google drive and Microsoft’s One Drive.

However, Github also includes version control and tracking using Git, which we'll get to shortly. Github has a web-based interface that includes support for desktop and mobile integration.

Github provides access control and collaboration features such as bug tracking, feature requests, task management, and wikis.

It also has native support and interpretation of markdown that's much easier to use and write than HTML. We’re going to learn more about markdown at the end of this module.

So let’s set up your Github account. Go to <https://github.com>

1. Choose a Good Username for Your Github Account
   1. Pick something professional that represents you.
   2. This will be your identity on Github and will be viewable by everyone.
   3. NOTE: For this course, I assume that you are creating a PUBLIC Github account, which is FREE. You can create a PRIVATE Github account for a fee.
2. You can register one Github account per email.
3. Once you get logged into your Github Account, go to your account settings to customize your photo, bio, email, website URL, and more...
4. When you first get started you won't have any repositories, but we will be creating repositories for each project.

[BEGIN computer demo]

[NOTES TO MYSELF - COMPUTER DEMO]

* Show create account and log in screen
* Once you are signed in, click on the icon on the top right – click the pull down arrow to see selection options…such as accessing your profile and settings
* Click on your settings – check your name and email. These are IMPORTANT – you need to know these to set up Git for version control and connectivity from your cloud account to your local computer
* Add your bio summary, a URL, your photo, and any other information you want to share with everyone
* View your profile page – this is your “home” on Github – your page will look different from mine. When you first create your account you won’t have any repositories. However, we will be creating new repositories for this course shortly.

[END computer demo]

Now that you have your Github account created and you are logged in, we’re going to install Git. GIT is a source code management system for software development. It was designed and developed in 2005 by the Linux developers.

GIT is a distributed version control system with complete history & version-tracking capabilities. You may have heard of other version control systems, like Subversion, CVS, Perforce, and ClearCase.

Unlike some of these, GIT is FREE (cost) and freely distributed under the terms of the GNU General Public License.

[BEGIN computer demo]

[NOTES TO MYSELF - COMPUTER DEMO]

* Download and install Git from <https://git-scm.com/> - click "Downloads" – at the lower left side of the web page
* This opens another web page. This page has the links for downloading the installer files for Mac, Linux and Windows operating systems. Choose the download link for your operating system – NOTE: Clicking these links starts the file download.
* Run the installer you just downloaded to install Git on your computer. Follow the instructions and accept the defaults.

For example on my windows computer, I can go to the start programs and see that Git was installed and has 3 options for running GIT:

* Git Bash – for this course we will use the Git Bash option
* Git CMD
* And Git GUI

[computer demo continued]

Now that you have Git installed on your computer and you’ve created your Github account, let's test your setup.

1. Open your browser and log back into your Github account
2. Click on your Profile, and then Click on Repositories – now we’re going to create a new repository
3. Click NEW to create a new repository.
   1. type in a name for your repository such as "MyFirstRepo"
   2. put in a short description like "My First Github Repository"
   3. this will be a PUBLIC repository, but as you can see if you have paid for a PRIVATE Github account you do have the option to create Private repositories
   4. Go ahead and click the box to select "Initialize this repository with a README"
   5. keep everything else the same (use the defaults)
   6. click "Create Repository"

It takes a moment for the repository to be created, but you’ll notice that your repository now has 1 file in it. README.md, which is your readme for the repository.

Now we’re going to connect everything back to your local drive using Git.

We need to create a place on your local drive where you want to save your work for this course. We’re going to end up creating multiple repositories for this course, so I create a central folder on your computer like “C:\RepTemplates” where you'll keep everything organized.

You can see this folder created on my computer. It is this folder where I will store and link all of my Github repositories for this course.

Let’s go ahead and run GIT. As I mentioned, we will use the Git Bash command window for running and executing GIT commands.

Once the GIT Bash window opens, you’ll see some information and details in the window about what directory/folder it's currently in. On my system, GIT Bash defaults to my “users” directory.

However, we want to change out of this directory. Keep typing

cd ..

until you get to the main “C” drive. Then we’re going to change to the RepTemplates folder we just created. Type

cd RepTemplates

You should see the directory folder change at the GIT Bash command line, but you can also type the command

pwd

To get the “path with directory” to verify that you ended in your C:\RepTemplates folder as intended.

We can also view the contents of this folder, by typing either

ls

To “list” the files in this directory or you can also type

dir

To get a “directory” listing of the contents. You’ll notice at the moment there is nothing in this folder. That's fine. That's correct. In a minute we’re going to link back up to our newly create Github repository “MyFirstRepo”.

[END computer demo]

As we go through this course, I will refer many times to the book by Jenny Bryan entitled: **Happy Git and Github for the useR**

You can access this book for FREE online at <http://happygitwithr.com/> There's a lot of good information on setting up R and RStudio and for getting setup using Git and Github.

[BEGIN computer demo]

*…Go to* [*http://happygitwithr.com/*](http://happygitwithr.com/) *and walk students through book sections…*

Now to get started using GIT, you need to “introduce yourself.” At this point, you should already be logged into your Github account. But we need to make sure that GIT understands how to talk to your Github account. So, we’re going to type in 3 GIT commands in your GIT Bash window. Open your GIT Bash window.

This first command tells GIT your name – be sure to type in the same name you used when you set up your Github account. Your name goes between the 2 single quote marks.

git config --global user.name 'Jennifer Bryan'

Next we also have to tell GIT the email account you used when you set up your Github account. Again put your email in between the 2 single quote marks.

git config --global user.email 'jenny@stat.ubc.ca'

Finally, to check to make sure everything went in correctly, type in the following GIT command to list your global settings and you should see the user.name and user.email you just typed in.

git config --global –list

If you see these, CONGRATULATIONS you have successfully introduced yourself to GIT!!

KEEP your GIT Bash window open.

[END computer demo]

**"pushmi-pullyu" SLIDE with PUSH / PULL GRAPHIC – insert here**

We'll be using the terms PUSH and PULL to talk about moving files back and forth from our local computer to the Github cloud repository and from the cloud back to our local computer.

The “pushmi-pullyu” was a fictional animal in the Doctor Dolittle series of children's books by Hugh Lofting with two heads on opposite ends of its body, so you never knew if the animal was coming or going.

Hopefully, we won’t have that confusion in this course, but we will be PUSH'ing and PULL'ing content in and out of your project repository between your local computer and your Github account using Git version control.

A PULL moves content from the cloud to your local computer.

A PUSH moves content from your local computer to the cloud.

[BEGIN computer demo]

Now let's CLONE your Github repository to copy the repository contents from your Github cloud repository down to your local computer.

Open your browser, and go to your “MyFirstRepo” repository. At the top right, there's a green button to “Clone or Download” your repository. Just below that green button, there's a little icon to the right to “copy to the clipboard” the long URL address you will need when we use GIT to clone your repository.

[END computer demo]

**First PULL to Clone your repository SLIDE – insert graphic illustrating a PULL from the cloud**

When you CLONE your repository, this is your first PULL. You will be PULLing the content down from your Github account to your local computer.

[BEGIN computer demo]

To execute a clone using GIT, open your GIT Bash window. Check to make sure you are in your C:\RepTemplates directory.

Go back to the “MyFirstRepo” repository and click “copy to clipboard” to get the Github repo URL. Make sure you have the option for “Clone with HTTPS” shown to get the correct URL.

Back in GIT Bash, Type git clone followed by the URL. Since the URL is now COPYied into your “clipboard”, you can PASTE it into the GIT Bash window

git clone https://github.com/melindahiggins2001/MyFirstRepo.git

This will take a minute to run, but it should say that it is cloning your repository and you should not get any errors.

Now type in a ls or dir command to view the contents of your directory. You should now see a new folder created called “MyFirstRepo” in your C:\RepTemplates directory.

Then type in

cd MyFirstRepo

to change into this new directory and type ls or dir to view the contents. VIOLA!! You should now see the README.md file in this directory.

You can also see this file by viewing the directory contents in your file explorer. You may also be able to see a hidden folder called /.git which was created when you did the clone. If you can’t see this folder, that’s OK- it’s usually hidden by default. I changed the settings on my computer so I can view these hidden folders.

[END computer demo]

TADA!! You have now successfully cloned your Github repository and have it linked from your local computer to Github using version control and tracking with GIT!!

We’re going to do this again in the next part using the RStudio interface.