Module 1 – lesson 06

Script – part 2

[BEGIN computer demo]

Let’s take a moment and look at some of the other content in the **Happy Git and Github for the user** book by Jenny Bryan. <http://happygitwithr.com/>

There are many chapters in this book you may want to read and take a look at. For example, chapter 5 has information on setting up a Github account and chapter 6 has information on installing or upgrading both R and RStudio which you’ve already done. And chapter 7 covers installing Git which you’ve also already done.

Then in Chapter 8 there is information on introducing yourself to GIT which you’ve just completed.

If you would like to move beyond using just the Git Bash window and command line interface for using Git for version control, I recommend reading Chapter 9 on installing a more full-featured Git client. Jenny Bryan recommends either SourceTree or GitKracken.

Chapter 10 covers getting connected to Github which you just completed.

We’re going to spend some time in this next part of the lesson, learning about setting up credentials on your computer using either HTTPS or SSH to securely connect to your Github account. These details are covered in chapters 11 and 12.

There is additional information in chapters 13, 14 and 15 on using RStudio with Git to connect to Github and manage your projects. I will be showing you how to use RStudio to connect to Github using Git shortly.

The later chapters 16, 17 and 18 provide examples of linking up projects with Github depending on whether the project is new or existing and whether you setup the project on Github first or last. For the projects we will be doing in this course, we will be creating new projects by setting up Github first.

The next section of the book provides some workflow examples. I point out Chapter 22 which covers Git commands some of which you’ve already learned. I also mention Chapter 26 entitled Burn it all down which is helpful to read when you have problems and Git stops communicating between Github and RStudio.

Now we’re going to connect to your Github account using Git but from the RStudio interface instead of from the Git Bash window. Go ahead and start RStudio.

When you open RStudio you should see a screen similar to this but it won’t look exactly like this and that is OK. Your layout should be similar. There are a few options we need to review and setup to make sure that RStudio known that you want to use Git.

In the tools menu, click on Global options. Click on the button for GIT/SVN. In this window we want to make sure that the box is checked for “enable version control interface for RStudio projects”. Next we need to find where the GIT executable file is located on your computer. On my computer it is located on my program files folder for Git/bin. For example, if I click browse it shows where this is on my computer’s hard drive. You’ll notice that the file is named “git.exe” and is located in the “/bin” folder. You may also see an icon like the one shown here next to the filename. There is also a similar file under the “/cmd” folder, but this is NOT the one we want. We also DO NOT want the file for “git-bash.exe” NOR the one names “git-cmd.exe”

Also make sure you have the box checked for “Use Git Bash as shell for Git projects”. This why I showed you earlier how to use the Git Bash shell window with your projects.

Since we’re not using SVN you can ignore the line for SVN executable

[END COMPUTER DEMO]

[BACK TO SLIDES]

Now that we’ve got some of the options setup in RStudio for using Git, we next need to setup your Github account credentials on your computer so that each time you run a GIT command to connect and sync to your Github account you won’t have to keep typing in your login name and password. You can setup your credentials by using either HTTPS (hyper text transfer protocol secure) or SSH (Secure Shell). These are two different approaches for setting up your credentials. I’m going to show you how to setup SSH from RStudio.

[BEGIN COMPUTER DEMO]

Back in RStudio in the Global options window for Git/SVN options, were going to setup your SSH RSA Key. This is for setting up a public key/private key cryptosystem. Click on the button to “Create RSA Key” and use the defaults. This is where you create the key. You can add a pass phrase or password, but this is optional. Note where on your hard drive it tells you where the security key will be created. Then click “create” to create your key. If you’d like to view your public key, click on the link to the right. Once you’re done, click OK

Let’s double check that GIT also now sees your SSH Key. Open your Git Bash window and type in this command

ls –al ~/.ssh

When you do this, you should see two files id\_rsa (which is your private key) and id\_rsa.pub (which is your public key). This is explained in more detail in the Happy Git book in chapter 12.2. You can also click the [?] Using Version Control with RStudio to get to the help webpages at RStudio.

Make sure you are in the local directory for your new repository C:/RepTemplates/MyFirstRepo. You should see this listed in your Git Bash window prompt or you can also type pwd to get the “path with directory”

You can double check your settings in the git bash window by typing

git config –global --list

You should be pretty much setup and ready to go at this point. If you are still getting errors, you might have a credentialing conflict. For example, if you have multiple Github accounts with different emails, you might have to remove one credential and add the other one instead. Search Stack Overflow <https://stackoverflow.com/> or the Github help documentation <https://help.github.com/> for help.

Let’s go back to RStudio and create a New Project.