Getting Started with Git and RStudio

William H. Knapp III

April 4, 2016

1 Installing R

First we need to get R installed. To install R go to R's website and select a location in the USA near you for the fastest downloads. Once you've selected a location near you choose to "Download R" for the type of operating system you're using (e.g. Windows or Mac). Use the default values when installing unless you're a power user and know what you're doing. After you've installed R, it's time to install RStudio.

2 Installing RStudio

As I mentioned in the last document, RStudio makes working with R a whole lot easier than using R alone. It will also allow you to create and edit .md and .rmd files. To install RStudio, go to RStudio's website and sellect the installer for your platform. Again, you should select the defaults unless you know what you're doing.

If you prefer to watch a video that will guide you through the installation process and provide you some handy tips for working within RStudio, please watch the following tutorial for either Windows or Mac.

3 Installing Git

If you're using a Mac and installed R and R studio, you might already have git installed. If you find that the git isn't on your Mac later when we try to

use some Git commands, you can download and install git from Git's website and select the appropriate version.

If you're using Windows, download and install Git. Again, use the defaults with your installation, unless you know what you're doing.

If you prefer to see a video tutorial of the process of installing and working with get, please watch the following tutorial for either Windows or Mac.

4 Creating a Github Account

Go to github.com and sign up for an account, if you don't already have one. If you want to protect your privacy, use a username that isn't associated with you. Accept the defaults when setting up your account and make sure you're signed up for a free account. If Github asks you to set up anything for you, decline.

5 Forking a Project

Once you're signed up with Github, make sure you're signed in. Then visit my project for this class. When you're there, look in the upper right hand corner of the project for a button labeled fork and click it. This will create a new version of the project under your username that you can view and edit.

6 Setting up Directories

Directories are another word for folders. You should create one folder for this class. You need to create this folder on your computer, not on your Github account. Inside the folder for this class, you should create another folder for this project. As the project is named capstone-reproducibility, you might want to name the folder for the project with this name. If you don't know how to create a folder for your computer, do a Google search.

7 Downloading the Project

Go to the folder for your project and right click. If there's an option that says something like "Git Bash Here," select that. If not, you'll need to open

up Git Bash. Git Bash is a simple program that you'll type commands into to download, save, and sync your progress. In Windows, you can find Git Bash in the start menu. If you're using Mac, Git Bash will live inside your terminal program. You should be able to perform a search through your applications for terminal. Once you've found the program open it.

Once Git Bash or the Terminal (Mac) is open, you should see the current folder you're in and a dollar sign. The dollar sign is a command prompt, after which you'll enter the commands you need to download, save, and sync your progress.

If you were able to open Git Bash or the Terminal in the folder for this project, you're good to go. If not, you'll need to navigate to the appropriate folder. You do this using the cd command. cd stands for change directory. Take note of the folder location. It will probably be something like C://Users/Username/My Documents/PSY441/capstone-reproducibility.

If you're not already in the project directory, once you know the location of the document type the following commands:

```
cd ``C://Users/Username/My Documents/PSY441/capstone-reproducibility''
```

You should notice that the text before the command prompt has changed to reflect the new directory. *Tip 1*: if you type "cd .." you'll move back up to the containing folder. *Tip 2*: everything in the directory you initialize (more on that shortly) will potentially be visible on your Github account, so make sure you create a directory just for this project.

Once you're in the folder for the project in Terminal or Git Bash, type:

```
git init
```

This will initialize the folder to be tracked by Git and create a hidden folder in the current directory.

Once you have initialized the folder you'll download the project to your computer using the following commands. Make sure you change "username" to your Github username.

```
git pull https://github.com/username/capstone-reproducibility.
```

If everything worked according to plan you should be returned to the command prompt. Now go to the folder for the project and check that the folders and files that were contained in your online fork are now present in your current directory.

8 Editing .md Files

Go to the folder "02 Getting Started with Git and RStudio." In it, you'll find a document called "example.md." Try to open it. If it doesn't open in RStudio, close whatever it opened in. Then right click and try to open it in RStudio or search for RStudio to open it. If it asks you whether you should use RStudio as the default for opening .md files, select yes.

If you're successful, the file will open up in RStudio. If you were unsuccessful let's try one more thing. Open up RStudio. By default RStudio will open a single window with 3 or 4 different panels. In the lower left panel, there's a tab called Files. Select that and navigate to the example.md file and open it by clicking on the file. Alternatively, you can go up to the menu in the upper left corner of RStudio and click "File," then "Open File...," then browse to the file and open it.

Once the file is open, there should be four panels open in RStudio. The panel in the upper left is the editing panel where you'll edit the documents you need to complete the assignments. In the lower left is the console panel in which you can enter R commands directly and see the output for any messages should you make errors. In the upper right hand corner is the Environment/History panel. You can use this to see what variables are currently loaded into R. When you start this should be empty. Finally, in the lower left corner there's a panel that you can use to navigate through files. This panel will also show you any plots that you create and show help if you ask for it in the console panel.

9 Working with .md Files

Now that you have example.md open. Look through the document and see how I've explained the different pieces of Markdown that you'll need to create your own formatted .md files. Let's go through some of the Markdown here.

9.1 Comments

You wish to make a comment put your comment in lines between "<!–" and "–>."

```
<!--
```

```
Comments will go in here.
Here too.
-->
```

9.2 Headings

You can use between 1 and 6 pound signs (#) to create different section headings or titles. The more pound signs you use the smaller the text will be. Follow the last pound sign by a blank and then the text you want to be formatted.

```
# The biggest header/title.
###### The smallest header/title.
```

9.3 Paragraphs

Typing enter at the end of a line won't start a new paragraph. Instead the text below will continue the same paragraph. If you wish to create a new paragraph, you'll need to leave a blank line inbetween separate paragraphs.

```
This
is
one
paragraph.

This is another paragraph.
```

When typing, it's good practice to type enter after you've typed 70-80 characters so the individual lines don't become too long and cumbersome. If you look at the instructions.Rnw file for this assignment, you'll see what I mean.

9.4 Formatting Text

There are several ways you can format text. If you want to use italics, use single asterisks around the text you want italicized. If you want to use bold, use double asterisks around the text. Finally, if you want bold and italics, use triple asterisks around the text.

```
*italicized text*
**bold text**
***bold and italicized text***
```

9.5 Lists

There are two types of lists that you can use. Ordered and unordered. An ordered list will number your list. An unordered list will not. You can combine list types to create sublists. To create an ordered list, type the number of the item followed by a period, then a space, and then the information. For an unordered list use a single asterisk separated from the text by a space.

To create sublists, ordered or unordered, indent the number or asterisk using spaces.

```
1. Ordered list item 1.
  * Unordered list under item 1.
2. Ordered list item 2.
  1. An ordered list under item 1.
```

9.6 Links

You might need to link to different websites that contain materials, videos, etc. To do this you'll need to create a link. To create a link you'll put the text you want to link in brackets. You'll put the actual linking address between parentheses.

```
Here's my favorite [search engine](https://google.com).
```

9.7 Knitting Files

When you knit a file, RStudio will take the markdown document you give it to create output as an html file, a pdf, or a .doc file. We're going to stick with html as you'd need to install a lot of additional software to knit to a pdf.

With my example.md file open in the editing panel, click "Preview HTML." If RStudio asks you to install additional software at this time, please allow it to so it can knit your document properly. You may need to click "Preview HTML" again after the software is installed. If everything works properly,

you should see a website in which my comments are invisible but everything else is formatted as I specified in the .md file. *Tip*: When I'm creating an .md or .rmd file, I knit my file regularly while I edit it. This is good practice as the file won't knit if there are any errors. When I grade your work, I'll download your .md and .rmd files and try to knit them. If they don't knit the first time, you'll lose points, so make sure your document knits before submitting it.

10 Completing the Homework

10.1 Editing homework.md

Open up the homework.md file. It should open in a tab next to the example.md file. I recommend switching back and forth between the files, copying, pasting, and editing what you've pasted to do what you need to do to complete the homework. Instructions for the homework are contained in the comments. You should place your work that addresses the comments immediately after the preceding comment. Remember to knit/preview the document regularly so you can find mistakes more easily. There's little more that's frustrating when completing the homework than trying to find small errors. By knitting often, you'll know that the error is somewhere between where you last knitted and what you entered since then.

10.2 Syncing Changes with Github

Once you've completed the homework, you'll need to save the changes you made using Git and upload those changes to Github. The first time you go through this process, Git might ask you for your name, email, or other information. If it does, it will give you examples, that you can actually use, if you want to maintain your privacy. Otherwise, you can use your real information.

10.2.1 Setting an Origin

Before you can sync any changes with your Github account, you'll need to add your remote (i.e. online) origin. Accomplishing this is simple. You should only need to do this once for each project, which is why I'm separating this

from the rest of the commands that you'll use again and again to sync your files as you progress through the homework. To set up the remote origin go to Git Bash or Terminal in the directory for the homework and enter the following. Remember to replace "username" with your Github username.

```
git remote add origin https://github.com/username/capstone-reproducibility.git
```

Once you've done this it will probably ask you for your username and password. Depending on what system you're working on, when you type your password, you might not see the cursor move. Just trust that it's working and hit enter after you're done.

10.3 Adding, Committing, and Syncing Your Work

Once you've added the remote and have changed the homework.md file it's time to have Git track the additions. We do this with a series of commands, that you'll use anytime you want to push your changes back up to Github.

```
git add -A
git commit -m "A message indicating what you changed"
git push -u origin master
```

The git add -A command adds everything in your project directory to be tracked by git. The git commit command basically takes a picture of everything you've done. Finally, git push sends the changes you made to Github.

When you push, you'll probably be prompted for your username and password. Again, if the cursor doesn't move while entering your password, trust that it is getting entered.

If it's successful, you should see that some files are being processed and you should be returned to the command prompt after all the files have been synced. If it's unsuccessful, double check that you spelled everything correctly. If everything looks right and you can't get it pushed successfully, the best way for me to help you is if you Skype me. My username is knappsych. If you Skype with me, you can share your screen with me so I can see what you're doing and troubleshoot more easily.

11 Quitting RStudio

After you're done working in RStudio for the time being, close it. It might prompt you to save your history. Choose not to. Because if you've made a mistake somewhere, the next time you open R, the mistake will remain. You'll also increase the amount of information in the project directory that will get uploaded to Github when you sync, so say no to saving the history!