HW 1 ~ Setting up software & user accounts

Matt Fitzpatrick

Due Friday, September 2, 2022

R

We will use `R` for all coursework. If you do not currently have `R` installed on your computer, go ahead and download and install the latest version for your operating system: http://cran.r-project.org/. If you have an older version of `R` installed, now might be a good time to update. The most recent version is: `R 4.2.1 "Funny-Looking Kid"` released on 2022/06/23.

RStudio

`RStudio` is a useful IDE (Integrated Development Environment) that integrates `R` coding, testing / executing scripts, plotting, help documentation, and `Git` file versioning software (see next item). Download and install `RStudio` (https://www.rstudio.com/products/rstudio/download/#download). You want the free **Desktop** version for your operating system.

Git & GitHub - Getting setup with `Git` and `GitHub` takes a few steps, as follows:

`Git` is a file version management system and `GitHub` is an online `Git` resource for coding-based projects (you primarily will use `GitHub`, but you will need to install `Git` too - if you haven't already). Note that `GitHub` can be accessed via both a website and an app you install locally. `Git` also interfaces with `RStudio`. We will use `GitHub` to track different versions of our `R` scripts and share code. You will need to install several `Git`-related items, just follow the next steps to setup `Git` and `GitHub`. **To avoid potential problems, close `RStudio` before installing Git.**

1. Download and install `Git` (https://git-scm.com/downloads) for your system. If you are a Mac user, you will have a number of options from which to select. If you do not recognize any of the options listed after you click on `Download for Mac`, I would suggest using the `Binary Installer` (https://sourceforge.net/projects/git-osx-installer/). Windows users should have only one option. Among other things, `Git` interfaces with `RStudio` (see details below) & provides a means of version control for your code. I'll explain how to use `Git/GitHub` later.

2. Once you have `Git` installed, we need to change some settings in `RStudio` to use `Git` for file version tracking. In `RStudio`, under `Tools-Global Options`, click on `Git/SVN` at the bottom-left side of the window (look for the brown box icon). You'll need to tell `RStudio` where `Git` was installed on your system. Where `Git` is installed will differ depending on operating system, so you'll need to look closely. On a Mac it will probably be in `\usr\bin\git`. On Windows try (but check first): `C:/Program Files (x86)/Git/bin/git.exe`.

3. Next, create a free `GitHub` account (https://github.com/join) and then email me (mfitzpatrick@umces.edu) your `GitHub` username.

4. Once I have your username, I will create a repository where you will store code (including for HW assignments and for your final project). Your repository will be named: `mees698C-2022-YOUR-UID-HERE`.

You will receive an email invitation from Github (check your SPAM folder) inviting you to join the repository that I created for you. Open the email from GitHub and click on View Invitation. In the web browser window, click on Accept Invitation. We'll do more with this repository in step #6 below, but for now proceed to step #5.

5. Download and install GitHub Desktop for your operating system: Mac (https://mac.github.com/) or Windows (https://windows.github.com/). GitHub Desktop will serve as a local interface between RStudio and Git / GitHub. You may be asked for your GitHub login credentials during setup of GitHub Desktop. If so, click on Sign in to github.com and then enter your GitHub credentials in the next window. Continue with the setup process, providing the necessary information when requested.

6. The final step is to make a local copy - or *clone* - of the repository I created for you. This will allow you to store your code locally (on multiple computers) and to track versions as you make edits. Click on File-Clone Repository..., check that the Github.com tab is highlighted and then type your UID in the Filter text bar. The repository I created for you should appear in the list. **Before clicking on Clone , check the folder location (see bottom of window where it says 'Local Path') to make sure the repository is being saved where you want it.** Your private repository named mees698C-2022-YOUR-UID-HERE now should be listed in GitHub Desktop.

If you have trouble with any of these steps, let me know.

Spatial libraries in R

Next, within RStudio, run the following R code to install several of the most common R spatial packages we will be using in class. **Note that packages do not always install without error.** If you have any issues installing any of these, let me know ASAP. Of those listed below, rgdal and rJava sometimes cause problems on Macs. Let's hope not!

```
install.packages(c("sp", "sf", "raster", "rasterVis", "maptools", "rgeos", "dismo", "rgdal",
                   "rJava", "gtools"), dependencies=T)
```

Slack

We will use Slack for course communication. You may be asking: why do we yet need another messaging app? Fair question, but if you haven't used Slack before, I think you will soon understand why it is very useful. Slack will allow us to organize all class related messages in one place where we can all see them. In other words, Slack will provide a platform for use to interact with each other on homework assignments, etc. I will send you an invitation to the **Slack channel** for the course. All you should need to do for now is install the Slack app (also available for smartphones). Download and install Slack Desktop for your operating system: Mac (https://slack.com/downloads/mac) or Windows (https://slack.com/downloads/windows).