**R Workshop**   
Geobiology Group  
March 27, 2019

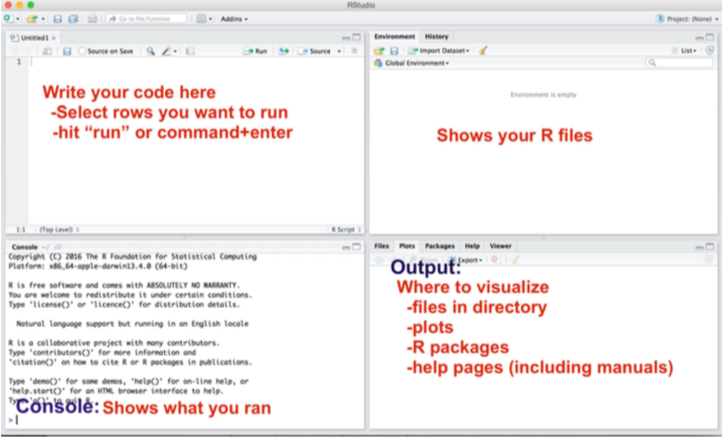
Note: Tania uses R for Windows and Nick uses R for MAC. Direct computer-specific questions to the appropriate user.

* Downloading software

1. Downloading R: <https://www.r-project.org>
2. CRAN mirror: choose the location closest to you. There is a mirror for ISU: <https://mirror.las.iastate.edu/CRAN/>
3. Keep in mind, you will download a version for your specific computer (Windows vs. MAC). **Important:** Make note of the latest release in which you are downloading. Many papers that use R heavily will report the version, especially since updates render packages (you’ll learn what those are later) useless and you always want your data/plots/statistics to be reproducible!
4. Now, we don’t actually use “R” as it is a command line driven program. Unless you are a programmer who likes using the command line, we need something else. We use R Studio! Why? Because it’s a graphic user interface that is user friendly. <https://www.rstudio.com>
   1. You will have the option to run R studio on your computer locally: <https://www.rstudio.com/products/rstudio/download/>
   2. Or online through the server…this if for folks who have large datasets “in the cloud”

<https://www.rstudio.com/products/rstudio/download-server/>

* 1. You will most likely be working locally 99.9% of the time unless you have a dataset that you can’t access from your computer.
* Getting acquainted with the software

1. Open R studio
   1. There are four windows (Image 1):
      1. Top left: Write code; scripts can be opened here; run commands from here with command+return (MAC) or control+enter (Windows)
      2. Top right: Shows which R files you have opened; Import datasets here;
      3. Bottom left: The console shows what you have ran from the top left window; can also type in code to run here
      4. Bottom right: Output is visualized here from code you have ran

(Image from edamamecourse.org)

1. Setting a working directory
   1. A working directory (i.e. some folder you designate) will direct **all output you save to that folder**. Also, once you set a directory the ‘import dataset’ function will default to that folder.
   2. Thirdly, if you are trying to import a file manually (via code), if it is not in your directory you will get an error in the console.
   3. Session > Set working directory > choose directory
2. Importing datasets (top right window)
   1. Knowing your file type is important. For example, even though you may save an excel file as a ‘csv’ file, you **can not** open the file by choosing ‘from excel.’
   2. File types ‘xls’ and ‘xlsx’ can be opened ‘from excel.’ File types ‘txt’ or ‘csv’ can be opened ‘from Text.’ These are most likely the most common file types. I’m sure there are others but we won’t be going into that today.
   3. Click import dataset > choose file you wish to open (txt file, xls file, etc.)
   4. If successful, your dataset will be opened above the top left window.
   5. Now, how do we work with datasets? By opening a new R script!
3. Opening an R script
   1. To do so, go to file > New File > R Script
   2. A new window should pop up in the top left. This will save as a ‘.R’ file
   3. Make sure to save any new scripts you make as these will provide code for future projects you do, as well as appropriate metadata.

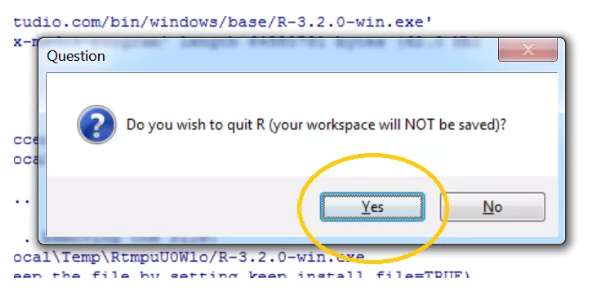
* Coding basics in R

1. Always remember to first set your working directory.
2. R packages: collections of functions and data that allow you to perform a specific function.
   1. To install a package, use this command:   
      install.packages(“PACKAGE”)
   2. You only need to install a package once. To “read in” a package, use this command:   
      library(PACKAGE)
   3. Once a package is loaded, you are now able to use code that has been written into that package. Say you want to learn more about a package. Type:  
      help(package)
      1. This will pop up a help window in the bottom right.
   4. To learn more about “arguments” (i.e. what the code in the package means) type:

??PACKAGE **\*this is very helpful\***

1. A hashtag (#) will void all code out for that line. You can also use it to write down what the purpose of the following line(s) of code are for. **I highly recommend this**. When you come back to the script weeks, months, or years later you will thank your past self for providing a road map to your code.
2. Updating R

Packages are constantly created, therefore, it is beneficial to use the newest version of R. Otherwise, packages created using the newer version will not run on an older version. There are two ways to update R: 1) visit the website or, 2) via installr package (for Windows, as of currently):

* 1. Open R (not RStudio) and type “install.packages(“installr”)”
  2. Turn on the package: library(installr)
  3. Type: updateR()
  4. Wait and follow the pop-up prompts to install the newest version of R
  5. More pop-up prompts will ask if you want to copy packages from older version to the newer version and you can follow through the prompts
  6. Once done, it ***may*** ask you if you want to quit R
  7. 

For Mac: [Update R for Mac](http://www.andreacirillo.com/2018/03/10/updater-package-update-r-version-with-a-function-on-mac-osx/#how-to-install-the-updater-package)

* 1. Open R and install the updateR package by typing:
     1. require(devtools)
     2. Install\_github(‘andreacirilloac/updateR’)
  2. Once the package is installed, run the package to update R by typing this:
     1. updateR(admin\_password = “os\_admin\_user\_password”)

Quick R tutorials and basic beginner stuff:   
<https://www.statmethods.net/index.html>

Cheatsheets (code that is useful in different situations…you’ll be surprised by their repository!): <https://www.rstudio.com/resources/cheatsheets/>

R cookbook (solutions to common tasks and problems in analyzing data):   
<http://www.cookbook-r.com>

List of all R packages (careful! Make sure they can run on your version of R studio):  
<https://cran.r-project.org/web/packages/available_packages_by_name.html>