R SESSIONS; Stream Ecology lab group

Preface

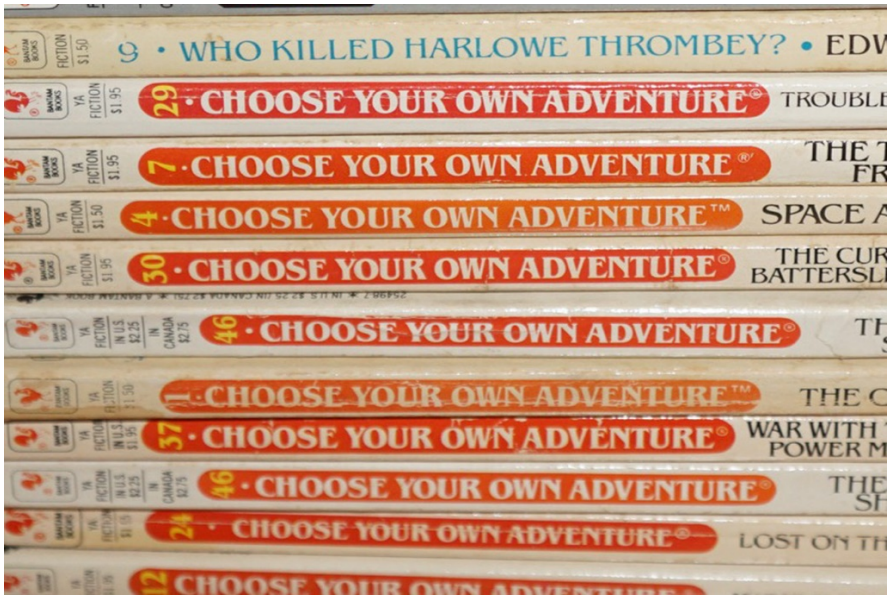
Before we get started on learning and working with R, I wanted to touch on a few things you'll find in this document. I realize there is a wide range of R experience in this lab group, so I will be doing my best to address this issue. More experienced folks – please feel free to browse through this document and glean off what tidbits are useful to you (if any). Less experienced folks – it would probably be a good idea to read through the entire document to familiarize yourself with some of R's basics and capabilities, some of which you might not have realized it possessed.

In order to make this document more accessible, I've split it up into multiple sections that will hopefully be useful guides through R and R Studio's initial setup, where to look if you encounter any issues or have questions about these programs, and the basics of when you open R Studio. I'll also include links to important materials, including the data we will be working with in this first session, to streamline this process and make it as frustration-free as possible.

When we get to using these programs, and we'll primarily be using R Studio because it is more convenient and user-friendly, I'll be using several different text/font types in order to help keep our code more readable.

- *R packages, which are essentially bundles of customized functions that complete specific purposes (e.g., calculating a mean, merging two different datasets together), will be in italics*
- **Specific R functions will be in bold**
- Code annotations will be preceded by a "#". This symbol tells R that this is not actually code, and it is the standard way for R users to annotate their work so it can be duplicated or manipulated by another user. MAKE USE OF THIS!

Lastly, and perhaps most importantly – THERE ARE MULTIPLE WAYS TO ACCOMPLISH AN OBJECTIVE IN R! As we work through these materials, I think you will find that there are multiple avenues you can take that will help you arrive at your goal. R code does not have to be elegant. It doesn't have to be sophisticated and include your own custom-written functions. It is perfectly acceptable if it looks like a Neanderthal sat down and wrote line after repetitive line of code. *The point is – write code that simultaneously works for you and gives you the correct output.*



Overview of R

I've taken the following excerpt directly from the R website. It does a better job of articulating R's purpose and capabilities than I ever could:

R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, …) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity.

One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.

R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.

The R environment

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either on-screen or on hardcopy, and
- a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

The term "environment" is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software.

R, like S, is designed around a true computer language, and it allows users to add additional functionality by defining new functions. Much of the system is itself written in the R dialect of S, which makes it easy for users to follow the algorithmic choices made. For computationally-intensive tasks, C, C++ and Fortran code can be linked and called at run time. Advanced users can write C code to manipulate R objects directly.

Many users think of R as a statistics system. We prefer to think of it as an environment within which statistical techniques are implemented. R can be extended (easily) via *packages*. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics.

R has its own LaTeX-like documentation format, which is used to supply comprehensive documentation, both on-line in a number of formats and in hardcopy.

R support and reference materials

The official page for general R help and frequently asked questions, as well as system-specific resources: https://cran.r-project.org/faqs.html

Use Google – it is your best friend when using R. Use it to look up specific errors or search for what you would like to accomplish with R. Sometimes you get lucky and find the exact snippet of code you need.

Other sites, like StackOverflow (https://stackoverflow.com/questions/tagged/r) are incredibly useful and often have your exact question. Answers are provided by other R-proficient users, and there are often multiple answers to each question.

Lastly, R has its own help pages and documentation that you can use to look up the purpose of specific functions or packages, as well as their respective arguments. We'll touch on this during our first session.

Materials you'll need

You will first need to download R and R Studio. It is preferable that you download the newest versions available (i.e., R is version 3.6.3 whereas R Studio is version 1.2.5033). R versions are always given "interesting" names – version 3.6.3 is called "Holding the Windsock". Please check the following sites to download these programs, as well as see when newer versions will be made widely available.

- To download R, please visit https://www.r-project.org/. The downloadable material will consist of a tar.gz file – these are essentially compiled and zipped files.
- To download R Studio, please visit https://rstudio.com/products/rstudio/download/. There are a variety of options to choose from depending on your computer's operating system.

The data we will be using for our R sessions can be found at:
https://uofi.box.com/s/rlwmuw3m4wc0g9kvdah0f1o1ln2mwhm3
These data include raw data, of multiple types (e.g., Excel spreadsheets, GIS files). Please download these data into whatever hard drive directory you wish.

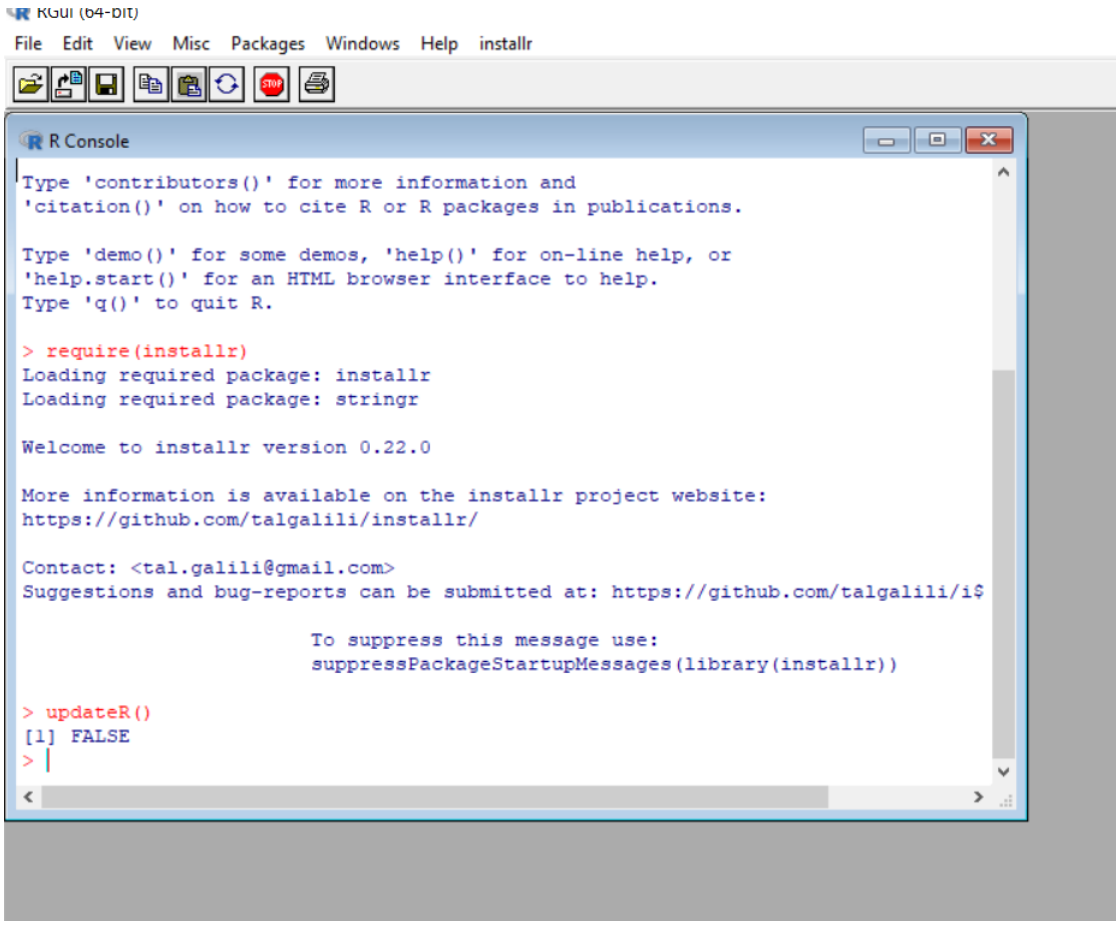
Installing and/or updating R and R Studio

If you do not have R and R Studio installed on your computer:

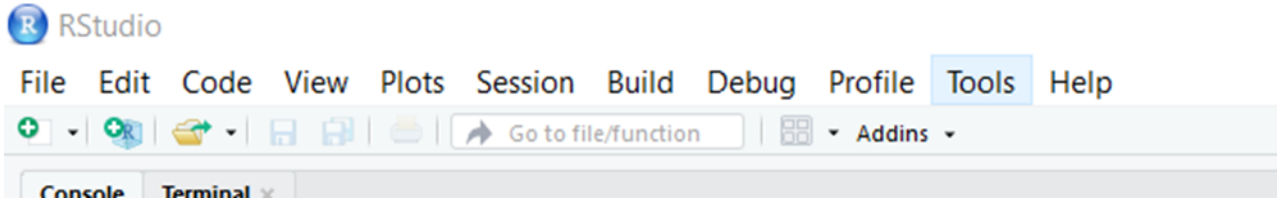
1) Download the newest version of R and follow the user-friendly installation process. Default installation is the recommended option.
2) Download the newest version of R studio and follow the user-friendly installation process. Default installation is the recommended option.
3) Once these steps are complete, you should have the most up-to-date version and be ready to begin working.

If you already have R and R Studio installed:

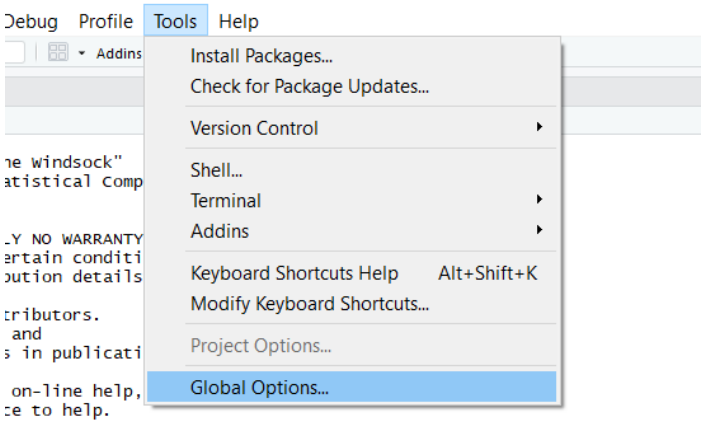
1) Open R and install *package installr.* This can be accomplished by using **install.packages('installr').**
2) Use either **require(installr)** or **library(installr)** at the command prompt to load this package.
3) Use **updateR()** at the command prompt. This will automatically check what version of R you are working with and will tell you if you are up to date. If you are not using the most recent version, you will be able to download the newest version. I suggest transferring all your previously downloaded and installed packages to the library that will be generated for the newest R version.



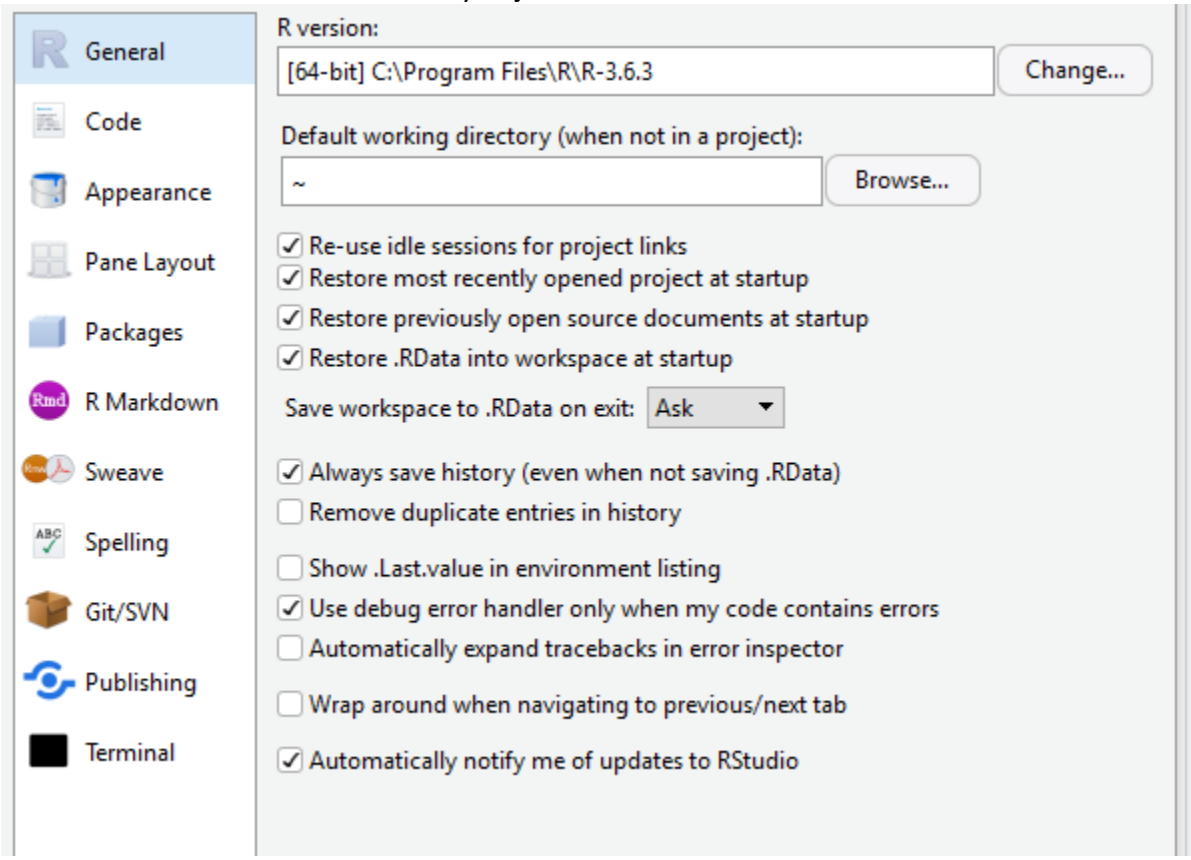
4) Once you have downloaded and installed the newest version of R, open R Studio. Once R Studio is open, go to "Tools" under the program's taskbar.



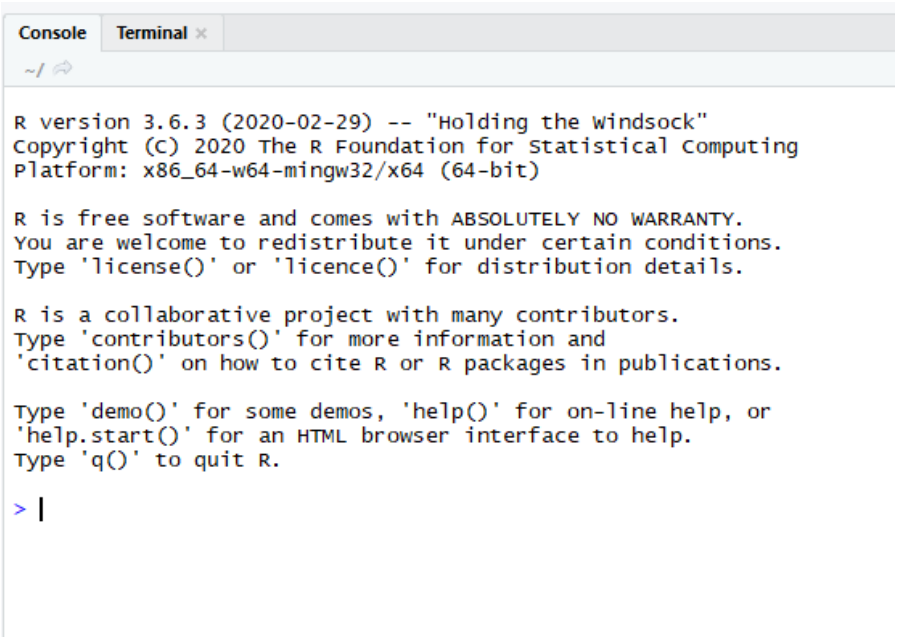
5) Select "Global Options" to open a new dialogue window.



6) The first option should allow you to select the version of R you would like to use. Make sure to direct it to the most recent version you just installed.

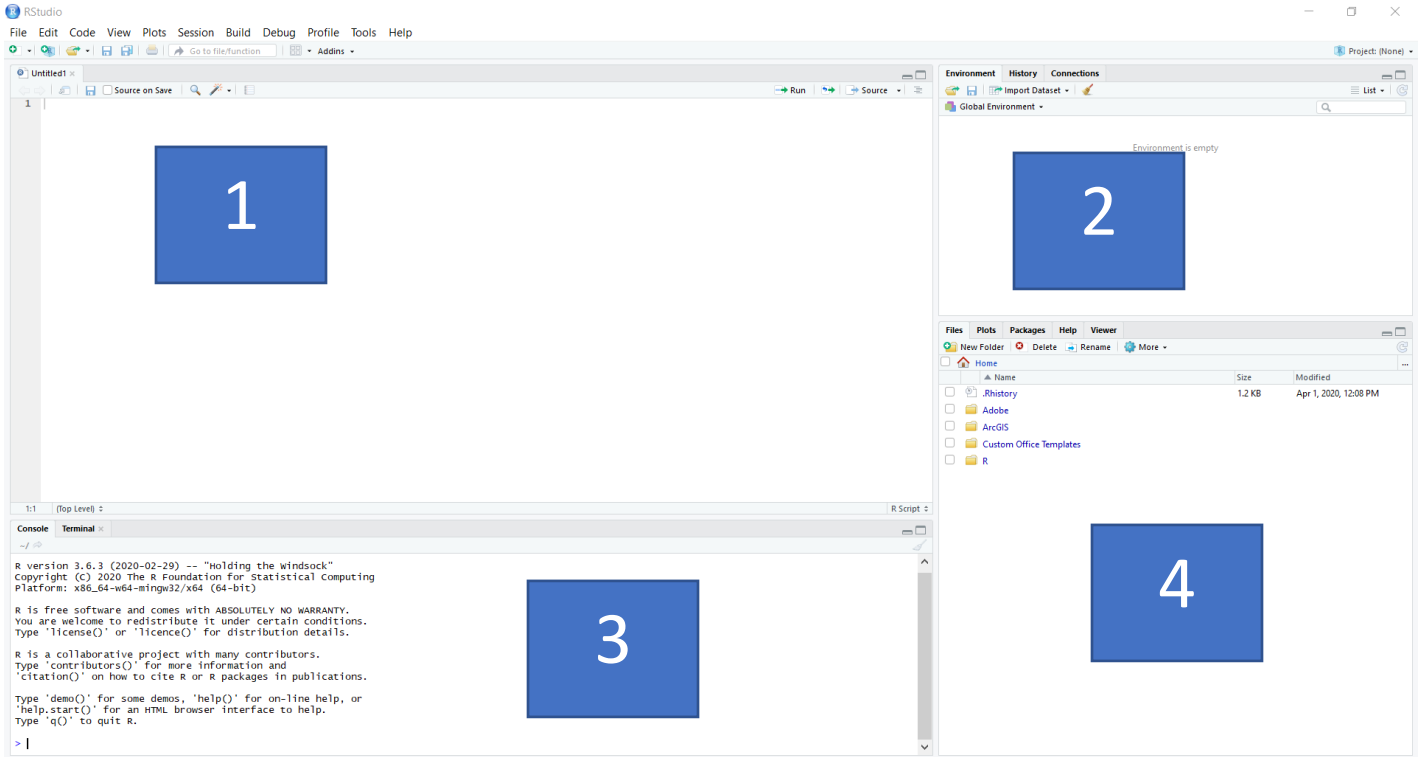


7) That should be it! You'll probably need to restart R Studio for any of these changes to take effect. Once you've restarted, you should see the following information that shows you are using the most recent version!



Using R Studio – The Basics

Now that you've downloaded R and R Studio, or updated both programs, let's look at what we'll be working with. Open R Studio – you should see the following screen (minus the blue squares containing numbers):



This is your basic home screen that is composed of 4 different panels. These panels are:

Panel 1 – This is where you will be able to write and save R scripts. As opposed to regular R, R Studio allows you to write code here, then run it all in one batch. Super convenient and user-friendly – just ask anyone that has used regular R!

Panel 2 – This panel shows you multiple pieces of information. The "Environment" tab shows you what data and information you are using, as well as what objects you have created with your code. The "History" tab will show you what code you've previously run in the current session, as well as previous sessions. I rarely look at or use the "Connections" tab.

Panel 3 – This is the panel where the magic happens! All your R code will run in this panel. You can also directly enter code here at the command prompt, or that ">" symbol!

Panel 4 – This panel shows multiple pieces of information depending on the selected tab. I rarely use the "Files" tab, but you can use this as a point-and-click means to direct yourself to specific directories. The "Plots" tab will contain any graphs or figures that you might generate – you can also zoom in on these visualizations as well as export them. The "Packages" tab will allow you to point-and-click select any packages you might want to install or update. The "Help" tab will show information for each package or function – I'll discuss in a bit how to use this tab. Last is the "Viewer" tab, which I rarely use.

BEFORE THIS WEEK'S SESSION:

Before we begin this week session, please make sure you have:

- Downloaded the data we will be working with from the UofI Box link I have shared
- Installed the following packages in R Studio: packages '*sf','ggplot2','tidyr','spdep','readxl*'
    - Remember, you use the command **install.packages()** to accomplish this
    - You can do each package individually, or you can do all at once by using:
        - **install.packages(c(***'sf','ggplot2','tidyr','spdep', 'readxl'***))**

The goals of this session will be to learn how to:

1) Familiarize yourselves with R Studio and its accessibility/functionality
2) Import data from Excel (.csv or .xlsx formats)
3) Import GIS data
4) Visualize these data and produce publication quality figures
5) Reconcile data from multiple sources into easily used objects
6) Manipulate data
7) Save data in multiple formats so it can be used in other programs (ArcGIS, Excel)

I realize this is a rather cursory overview of R, but I do not want to immediately jump into the deep end of R's full analytical capabilities without having some idea of how you all can envision using it. On that note, I would like to be able to tailor future R sessions to meet your needs or interests. Please feel free to contact me at schartel@illinois.edu and provide me with some analyses you are working on, functions or capabilities you would like to explore with R, or just general questions about the software.