ENV 226 Lab Online R Manual

Sara Souther

2024-09-18

Contents

Chapter 1

Ecological data analysis in ENV 226 laboratory

This course surveys the central concepts in ecology: evolution, population dynamics, community interactions, biogeochemical cycling, and limiting factors, as well as how those factors are measured, quantified, and interact with drivers of global environmental change. This course is required for the B.S. in Environmental Sciences degree program, and also the B.S. in Environmental and Sustainability Studies degree. This course acquaints students with foundational concepts and theories in ecology and provides a broad basis for more advanced courses in subdisciplines and applications of ecology.

A critical part of ecological research is developing practical and analytic skills. Most ecologists and data scientists use R statistical software. Learning how to manage, manipulate and analyze data in R will serve your undergraduate career and beyond! In ENV 226 lab, we will ease you into using R for all your data needs!

1.1   How to use this resource

Each chapter in this online book corresponds to a lesson that will help you in lab. When starting a new chapter, create a new R script with a name that allows you to easily connect the content in the R script to the chapter. Then, copy and paste sections of code in the chapter into your R script to practice!

Chapter 2

Welcome to R!

R is an open source statistical software package commonly used by researchers and other folks, who crave a free way to manipulate, analyze, and visualize date. R uses it's own programming language, which is similar to S+ (the paid precursor to R). R employs an object-oriented programming (OOP) paradigm, specifically a type of OOP known as "class-based object-oriented programming," to manage and manipulate data. In R, objects are fundamental entities, and you work with data and functions through objects. Let's walk through the basics of installing and using R!

2.1   R and R studio installation

You will first want to download R statistical software and R studio, which is a powerful program that interfaces with R to make your coding experience more organized and enjoyable. Notice that you need to select a version of R depending on your operating system.

2.2   Download the R statistical software from the official R Project website.

Open your web browser and go to the official R Project website at https://www.r-project.org/.

Choose a CRAN Mirror: On the R Project website's main page, you'll see a section that says "Download and Install R." Click on the link that says "CRAN (Comprehensive R Archive Network)." This will take you to the CRAN website.

1. Select Your Mirror: On the CRAN website, you'll find a list of mirrors (servers) from which you can download R. Choose a mirror that is geographically close to your location, as this will generally provide faster download speeds. Click on the mirror's link.

2. Download R for Your Operating System: On the mirror's page, you'll see options to download R for various operating systems (e.g., Windows, macOS, Linux). Click on the appropriate link for your operating system.

3. Choose the Latest Version: You'll typically see multiple versions of R available for download. It's recommended to choose the latest stable version unless you have a specific reason to use an older version.

4. Download and Install: After clicking on the download link, the installation file for R will begin downloading. Once the download is complete, run the installer and follow the installation instructions for your operating system.

5. Start Using R: After the installation is complete, you can launch R from your computer. Depending on your operating system, you may also have an option to install RStudio, a popular integrated development environment (IDE) for R, to enhance your R programming experience.

2.3   Download R studio

Now download R studio!

1. Visit the RStudio Website: Open your web browser and go to the official RStudio website at https://www.rstudio.com/.

2. Download RStudio: On the RStudio website's main page, click on the "Products" menu at the top, and then select "RStudio" from the dropdown menu.

3. Choose the RStudio Edition: RStudio offers different editions, including RStudio Desktop (for use on your local machine), RStudio Server (for remote access), and RStudio Workbench (formerly known as RStudio Server Pro, designed for collaboration and sharing in enterprise environments). You will want to choose the free version, RStudio Desktop.

4. Download the Installer: After selecting the edition, you'll be directed to a page with download options. Click on the download link for your operating system (e.g., Windows, macOS, Linux).

5. Download and Install: The installation file for RStudio will begin downloading. Once the download is complete, run the installer and follow the installation instructions for your operating system.

Start Using RStudio: After the installation is complete, you can launch RStudio from your computer. You'll have access to a powerful IDE that provides a user-friendly interface for working with R, including code editing, interactive R console, data visualization, and more. RStudio greatly enhances your R programming and data analysis experience, and it's widely used by R users for its features and capabilities.

2.4   Set up R studio

Alright, now that you've downloaded R and R studio, open R studio. You can customize the panes that you are visualizing in R.

In RStudio, the four panels or panes are commonly referred to as:

Source Pane: This is where you can write, edit, and save your R scripts and code files. It is typically used for script development and editing. You can open and create new R script files in this pane.

Console Pane: The console is where you interact with R directly. You can execute R commands and see their output here. It's an interactive environment where you can test and run R code line by line or in batches.

Environment Pane: The environment pane displays information about the objects, data frames, variables, and functions currently loaded in your R session. You can also use this pane to view data frames in a spreadsheet-like format and manage your workspace.

Files/Plots/Packages/Help Pane: This pane has multiple tabs and serves various purposes: *Files*: It shows the file system of your project, allowing you to navigate and manage files and directories. *Plots*: When you create plots in R, they will appear in this tab. You can interact with and export the plots from here. *Packages*: This tab displays information about installed packages, and you can use it to install, update, or load packages. *Help*: When you need documentation or help for R functions or packages, you can use the Help tab to search for and view documentation.

Typically, I select a structure in which I have my **Source pane** in the upper left, my **Console Pane** in the lower left position, my **Environment Pane** in the upper right corner, and the **Files/Plots/Packages/Help Pane** in the lower right position. You can select any position that you'd like, but if we create the same work environment, it will be easy for me to direct you when we are trouble-shooting code. To adjust the panels positions, use the pane layout function. Here's what that looks at for a Mac, but typically this arrangement is the default positioning for panels in R studio, so you likely won't have to adjust positioning!
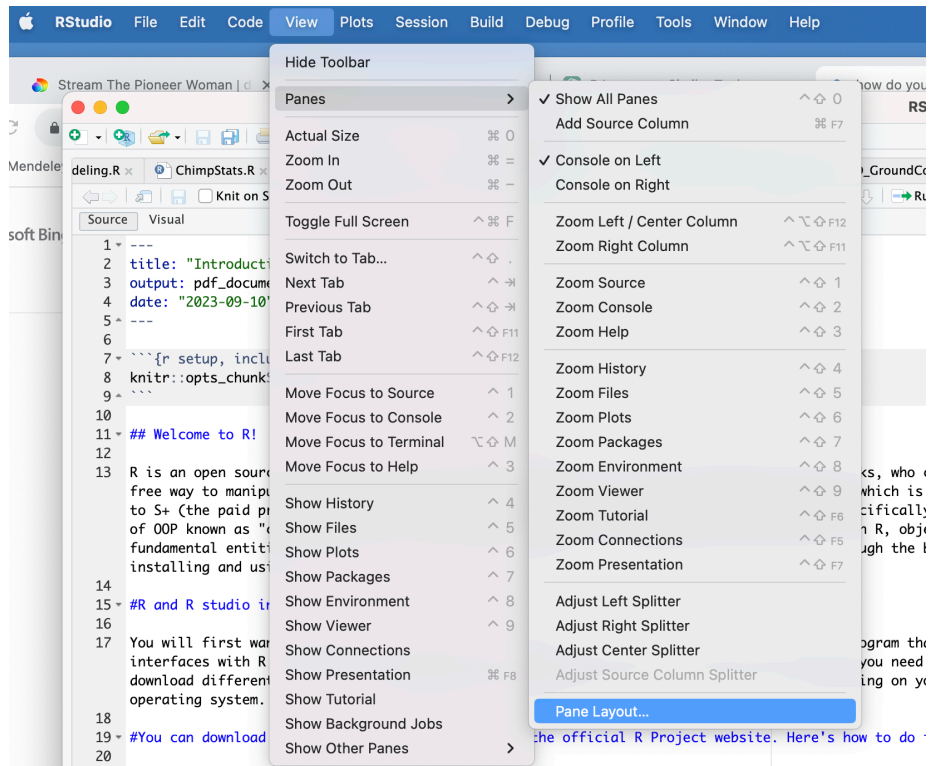
2.5   Good housekeeping

When you are coding in R, you will want to save your R or R markdown scripts and any other data files (e.g., .csv or spatial files) that you are analyzing in a common file. With good housekeeping, you will be able to seamlessly rerun your analyses at in point in time, allowing you to pick back up on projects that may have been dormant!

The first step in good R housekeeping is to set a **working directory**. A working directory tells R where to look for and save files. As an example, let's say I am planning to save everything associated with this tutorial to a file called Class1_IntroToR. To do this, create a new folder on your desktop called 'R is great'. You can change your working directory using the R studio interface by selecting a working directory at the top of the console panel in the "Files" tab. If you want to change your working directory to a different location, click on the "..." (ellipsis) button in RStudio's "Files" tab. This will allow you to browse your file system and select a new directory as your working directory. That said, you will be *far* better served by including code in your R script that directs R to your working directory. I prefer to set it within the code in order to allow you to instantaneously be able to pick up work where you left off rather

than searching through files and trying to remember how you set up the code. You can view your working directory by running a simple bit of code (run code below).
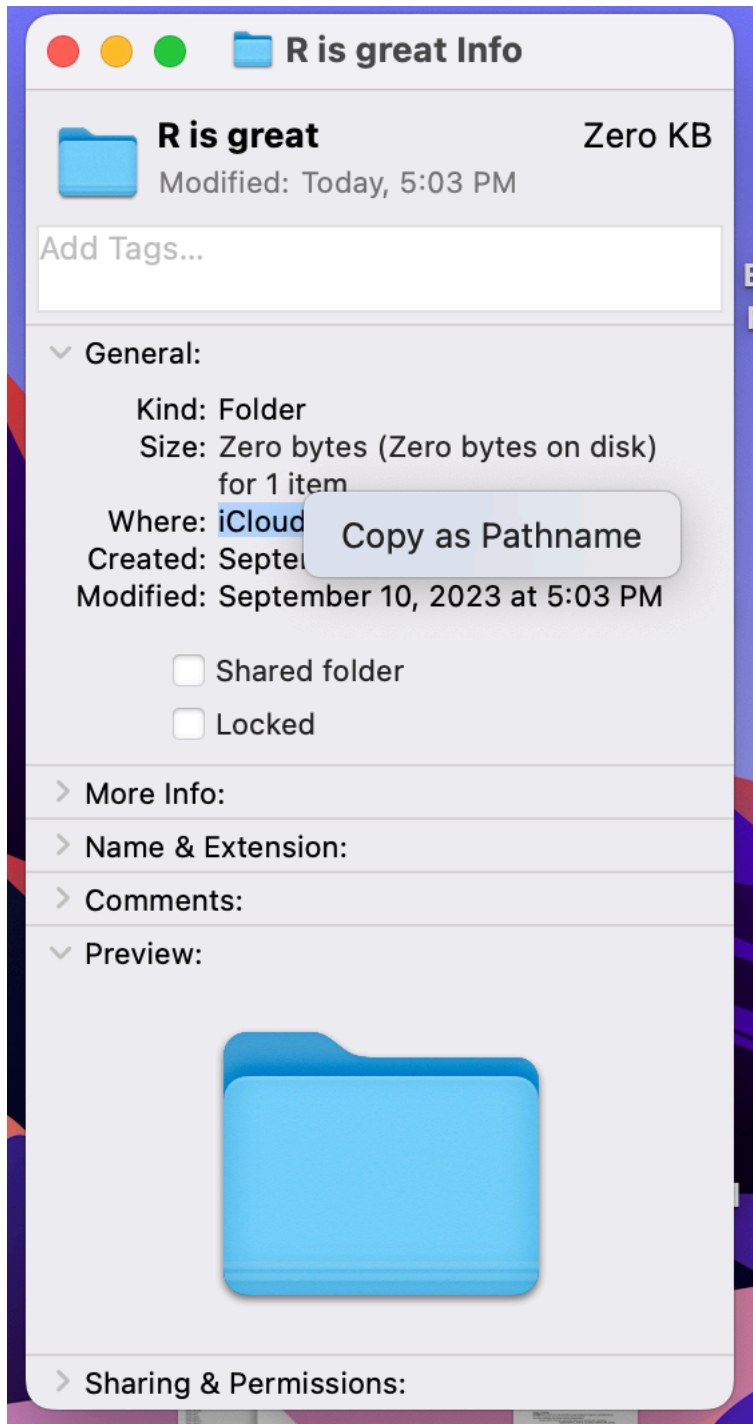
```
getwd() #display file path to R studio
```

```
## [1] "/Users/sks379/Desktop/ENV226LabRManual"
```

When you run, getwd() you will see where R is looking for files. Now, let's tell R where we want it to access files from! First, you will need to identify your file path. To find the file path to the Class1_IntroToR on a mac, double click on the file and should see several option, including 'Get info' (check out picture below).
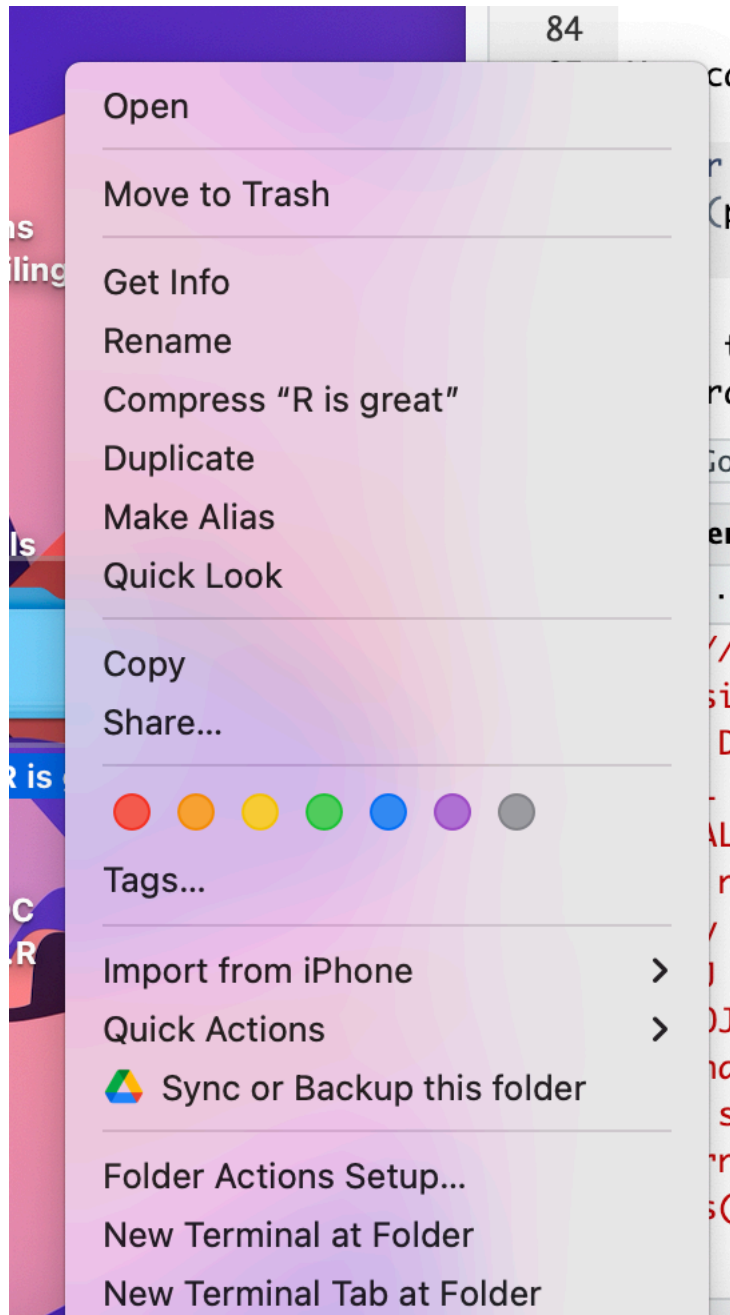
Then, select 'Get info'. Then, highlight the information after 'where' and copy it as a pathname (see picture).

🔵🟡🟢 📁 **R is great Info**

📁 **R is great**   Zero KB
    Modified: Today, 5:03 PM

Add Tags...

⌄ General:

      Kind: Folder
      Size: Zero bytes (Zero bytes on disk)
            for 1 item
    Where: iCloud   Copy as Pathname
  Created: Septe
  Modified: September 10, 2023 at 5:03 PM

      ☐ Shared folder
      ☐ Locked

› More Info:

› Name & Extension:

› Comments:

⌄ Preview:

📁

› Sharing & Permissions:

For PCs, start by opening your File Explorer: Press the Windows key + E on your keyboard. Alternatively, you can click the "File Explorer" or "This PC" icon on your taskbar or Start menu. Navigate to the Folder: Use the File Explorer to navigate to the folder where the file is located. You can click on folders to open them and view their contents.

1. Find the File: Locate the file you are interested in within the folder.

2. View the File Path: Once you've found the file, you can see its full file path in the address bar at the top of the File Explorer window. The file path will be displayed as a sequence of folder and file names separated by backslashes. You can click in the address bar and copy the file path to the clipboard by pressing Ctrl + C after selecting it.

3. Once you have copied your file path, paste that path name into the following code and set your working directory: setwd("/Users/sks379/Desktop/R is great/")

Now R studio is directed to upload and save work to this folder.

2.6   Annotating your code

Notice anything about the code in the section you just ran? You can use the hast tag symbol to tell R not to run a section of code. Whenever I am generating code, I try to add lots of notes to myself, so that that future me knows what code I created and why. Annotating your code is just good practice for coding! Alternatively, you can create R markdown files (what this tutorial has been created in), but R markdown, while generating pretty PDFs and websites, adds an extra layer of complexity that you generally don't want or need while coding, so I typically recommend creating an R script and annotating your that file! One thing that you might want to include in your code description is the version of R that you are using (you may need to load older versions of R if your scripts stop working due to updates to the program). To check the version of R that you are using, paste this in the command line: R.version.string
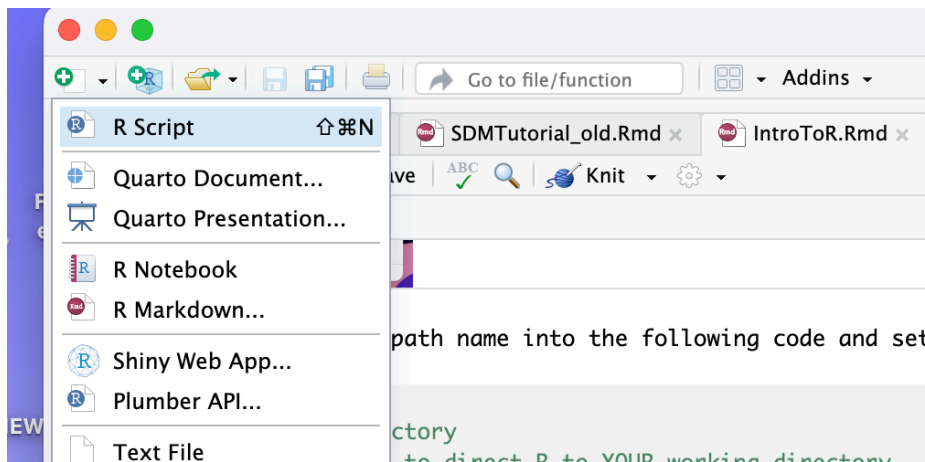
2.7 Libraries

Base R, the fundamental, built-in set of functions, data structures, and libraries that come with the R programming language without the need for additional packages or extensions, including basic math functions, statistical analyses and visualization tools. However, one amazingly cool think about R is that folks are out there creating 'libraries', or a collection of R functions, data sets, and documentation bundled together into a single package, to do specialized analyses. For most tasks in R, you will need to install and load libraries. There are two methods to install libraries. Let's start by installing an important library (or package) for data manipulation, called 'tidyverse' (actually several packages - hence why the name references a universe). To do this, run this code: install.packages("tidyverse").

The weird thing about including install packages code is that you don't want to re-install packages every time that you use R (in fact, it caused my R markdown code to freak out, which is why I've included the install function in the text). You will need to load packages, but you update R and R studio, you won't need to install packages after you have done it once. You can either then install the packages and delete the install code OR you can install through the R studio interface by going to packages, selecting install and searching for and installing the packages that you are interested in. When prompted, be sure to install dependencies - this will make sure that you have any pieces of code that the library that you are installing needs to operate.
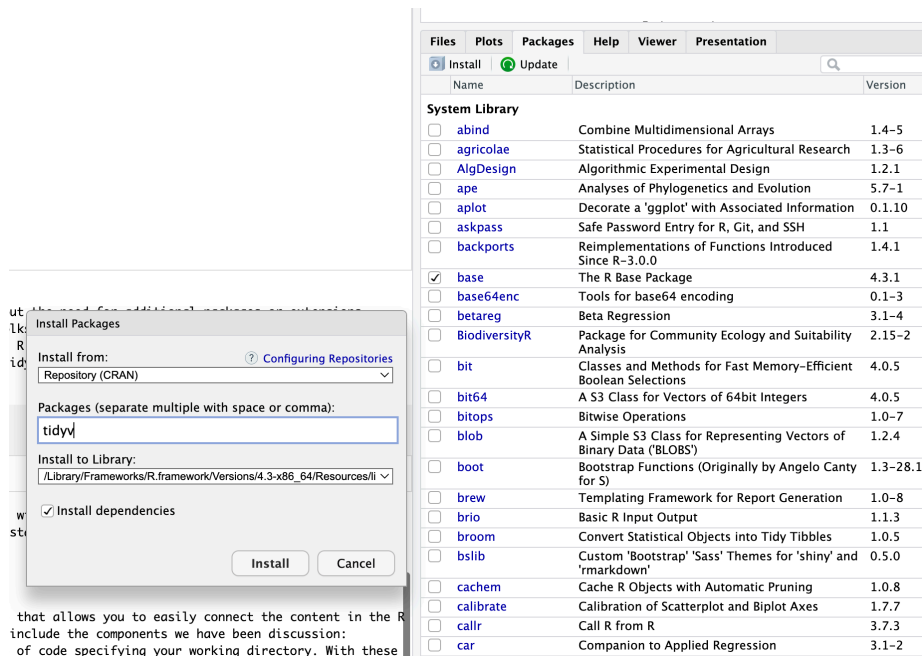
Excellent! You have installed a library! Now, we need to load it. AND you will need to load R packages everytime that you use R. Any functions associated with your package won't work, unless the package is loaded, so I suggest keeping the load library code in your R script. The code is simple (see below).

```
library(tidyverse)
```

Wonderful, you now have the essential knowledge base that you need to start working efficiently and effectively in R!

2.8   Organizing your code

A well-written R script will include the components we have been discussion: Annotated notes on what the script is doing and potentially even the version of R, loading commands for your libraries, and a line of code specifying your working directory. With these elements in place, you are ready to code your heart out!
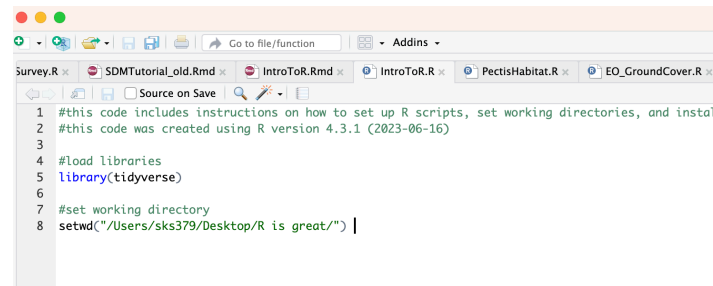
Here is a glimpse at what your R scripts should look like.

And with that, welcome to the wonderful world of coding in R!