Lab 5 – Introduction to working with data in R and RStudio

Note – some portions of this lab were originally written and delivered by WFCB graduate student George Maynard. Any errors or omissions are most certainly Erik's fault.

Overview

From here forward we'll be conduction all analyses in Program R, using the analysis management software RStudio. Most of you are probably at least passingly familiar with R and its capabilities. R is a Free and Open-Source (FOS) statistical programming language. When you think of FOS software, you should think of "free" as in "free speech" rather than "free" as in "free beer". Because R is FOS the users have the freedom to run, copy, distribute, study, change and improve the software. For more information about the FOS concept, check out https://www.gnu.org/philosophy/free-sw.html.

You can use R to manage data, run basic statistical tests, run advanced simulations and complex models, create publication-quality graphs, analyze spatial data, write manuscripts, and a lot of other things we won't cover here. The point is, because R is FOS, it can be modified to accomplish a lot of different tasks, and most of the packages people have written to do those things are available for free to anyone who wants to use them or improve on them.

Today we will really just barely scratch the surface of R and its capabilities, and our overall objective is to get everyone in the class to the same baseline starting point for conducting analyses in RMark in the coming weeks. R is a programming language and as with any language proficiency is gained only through regular use and practice. I highly, highly recommend that you force yourself to use R for any and all of your data analysis and management as you complete your graduate program.

As a general rule, the internet is your friend when working with R and there are a number of different sources for supporting resources for working in R. My typical approach, indeed how I developed much of the content for this lab, involves thinking through a particular problem (e.g. the step in a data management workflow) and then doing some googling to figure out how to address the issue. Nine times out of 10, someone else has already asked the world how to do the same thing I am contemplating, or at least something similar enough to nudge me in the right direction.

If you are the type who learns best when you have a comprehensive resource at your disposal (i.e. a book) then I recommend the "R for Fledglings" e-book from the Vermont Coop Unit:

https://www.uvm.edu/rsenr/vtcfwru/R/?Page=fledglings/fledglings.htm

There are no doubt dozens of other options out there, but this one seems to be a good beginner's guide for ecologists. The most commonly-cited online reference is stack overflow (https://stackoverflow.com/) which is a programming community in general; if you google a particular problem as I suggest above, odds are you'll end up at a stack overflow page.

When working in R there are a number of operations (we'll call them functions) that are housed in 'Base R', i.e. that come with the R software when you download it. For more specific applications, R users have developed 'packages' that contain addition, often more specialized functions to complete

any number of tasks. All R packages have their own reference manual in the form of a vignette, which can be accessed from the CRAN website (https://cran.r-project.org/). Sometimes interpreting the vignettes can be a challenge, but then that is where the internet becomes your friend. R can also be downloaded at the CRAN site, and RStudio can be obtained from https://www.rstudio.com/. Both software are free.

Format of this and future labs

In this and subsequent labs all of the operations will be completed via command line prompts to R, which we will organize as scripts writing in R studio. For each lab, I will provide a set of script as an RStudio file on Blackboard that contains all the information needed for a particular lab. This approach has the benefit that we do not need to worry about waiting long amounts of time for everyone to type in their code. The downside is that it will be very easy for you to just toggle through running the code without having to think carefully about what you are doing. I will try to design the labs, and associated homeworks, to foster the careful thinking, but you should also strive to take your time to achieve understanding. To help with this, I have added extremely detailed annotation (all comments preceded by '##') in the code that explains the operations the code is performing, offers rationale for the steps, and also gives some back story as relevant. These annotations are the equivalent of the step-by-step instructions from past labs, so, you should read them carefully.

First steps – Open RStudio

RStudio is what's known as an Integrated Development Environment (IDE). It is also FOS. You will need to have both programs installed on your computers to use RStudio. R is an object-oriented programming language. That means that data is passed to the computer in the form of "objects", each of which contain "attributes". The user can write "procedures", typically in the form of command statements, which tell the computer what operations to perform on objects. If you open up R, you'll notice that it's just a black box (console) that you can enter commands into, whereas in RStudio you'll find the same R console, and also three other panels. One of those panels is used to write scripts that are executed in the console, one houses your 'global environment' which contains all of the objects that you have created and procedures that you've defined during the current R session, another is a viewer window that will project graphs and other visual objects (e.g. package vignettes).

From here we are ready to start working through this week's code. Again, the script I have written is intended as a starting point for working in R to ensure that everyone has a common tool kit. Some who have a stronger R background may think I am omitting critically important R skills in what follows, which may be true, but in general what we'll cover today reflects what I find to be important baseline skills for doing demographic analysis in R. I anticipate that you will all need to expand your tool kit further, through self-learning, for your independent projects.

IMPORTANT - As a final disclaimer I do not claim to be an expert in working in R, nor is my coding eloquent or efficient. I fully expect that you or your classmates will know things I do not and I hope to learn from you all as much as you learn from me.

Assignment

You've been given access to a database, collected a long time ago in a galaxy far, far away, that details captures of Porgs, the endearing mammal-like birds that inhabit the planet of Ahch-To. The data file contains information on 17 years of mark-recapture that was conducted during the summer at the bird's island nesting colony. During captures birds were banded, sex was determined based on plumage (males have an orange cap) and a blood sample was taken to measure the Midi-Chlorian levels of each bird. The data file also contains information on the year of capture (one through seventeen), ordinal day of capture, and the bander. Your task for this homework is to create an encounter history file from this data file that could be read by RMark and that includes the capture history, the mean midi-chlorian value for each bird, and a grouping term for the sex of the bird. To complete your assignment, please upload an excel file with your finished history data file (yep you'll probably need to Google how to export an excel file, if you don't already know) AND your RStudio script file that demonstrates how you completed the assignment.