Introduction and Background

Graphical Analysis of Biological Data

This web page is available as a PDF file

Introductions

Who am I?

Dr. Mike Taylor
Department of Biology
Southeast Missouri State University
mtaylor@semo.edu

My educational background:

- B.S. in Biology (Chemistry minor) from Central Missouri State University.
- Ph.D. in Zoology from Louisiana State University in the lab of Dr. Mike Hellberg.
- Postdoctral Researcher at the University of Notre Dame in the lab of Dr. Jeff Feder.

My computer background:

My first computer was a Macintosh SE with 4 (!) megabytes (MB) of RAM, a 20 MB hard drive, and one 800 K 3.5" floppy disk drive (they weren't actually "floppy"). Compare that your laptop or desktop computer. Or your smart phone. Back then (ca. 1988), I ran a bulletin board system on that computer with a U.S. Robotics Courier 14.4K Dual Standard modem. In comparison, typical broadband wireless is at least 17X faster. Back then, my system was the shiznit.

I have formal training in the Pascal and C programming languages. I am self-taught in R and in Python. If you find yourself wanting to learn another language after R, learn Python.

Who are you?

You will tell me and the rest of the class a little bit about yourself as part of the first homework assignment.

Be curious

I expect that you will be curious. The notes for this course will be brief but they will have links scattered throughout. Follow the links and read the material. Do not follow the tl;dr way of thinking. Others have written tutorials, exercises, and other helpful information far better than I could write. Become a better you and take advantage of these resources. If you do not, you will not be successful in this course.

Follow the links! Read. Learn. Practice!

Data analysis requires coding. The code you will learn in this class is R, widely used by biologists across all disciplines. To learn proper coding, you will have to type R code. *Lots* of R code. Do not give in to the temptation to copy and paste from examples. You will not learn the R language well that way. Typing the code yourself will "drill" the language into your muscle memory and your brain memory. You will learn it better. You will retain it longer. You will have a better chance of getting a job. Just type it!

Reproducibility

A central tenet of science is reproducibility. Yet, often it is not. Read this introduction to reproducible science from rOpenSci. Follow the other links at your leisure. The tools you will use during this course emphasize the principle of open science. I feel strongly that you should adopt this philosophy in your scientific endeavors.

Make your work reproducible. Document it, test it, and leave a breadcrumb trail (version control), even when you are the only collaborator.

Use plain text

Use plain text files for all of your documents. Plain text files are small. Plain text files are portable. Text files can be opened and read on any computer plaform (e.g., OS X, PC, Linux) with any text editor, unlike (for example) Word and Excel documents that use proprietary formats, and are subject to change at any time. Text files are the common currency of reproducibility and open science.

This course, and in fact nearly everything I use for teaching, is built from plain text. Text files are the base that can be converted into a variety of formats, including web pages, PDF files, and even (gasp!) Word documents. This webpage was generated from this markdown source file on GitHub.

RStudio, which you will use throughout this course, has a built-in text editor, and it will suffice for this course. In the future, or even during this course, you may decide to install a stand alone text editor. Browse this list of free text editors to find one that works on your computer platform and suits your needs. Atom works on all platforms and has a good reputation. I used it to write this part of these notes. Nice features are that you can live preview your markdown documents and it can interact directly with Git and GitHub (more on both of these later).

I have used BBEdit for OS X since version 4 (now at version 12+) as a good general purpose text editor. You can use the full feature set free for 30 days. After that time, you can still use most of its important functions for free.

For PC, Notepad++ has a good reputation but you may want to try a few to find a text editor that suits your style and needs.

Do not use Microsoft Word to export plain text files. They are not truly plain text. Cut your umbilical cord. Leave Microsoft for the business world and the less enlightened SEMO community.

Use Git and GitHub

During this course, you will write lots of R code. Your code will run. Eventually. But, you think you can do better. You pursue a thread of an idea, and edit your code. You follow that thread down the rabbit hole, and write even more code. You test your code and realize it does not do what you wanted. The Mad Hatter laughs at you. You try to back out of the rabbit hole but you are lost. You have a jumble of non-working code that even a hookah-smoking caterpillar cannot understand.

Git is a way of protecting yourself from the maddest of the Hatters: you. With Git, you create several "safety nets" while you write your code. You can continue to work on your code but, if (when?) something goes wrong, you can fall back at any time on to one of your safety nets, and start fresh.

You may not realize it now, but having a safety net for your R code, data, and other projects, opens up the doors to your programming creativity. You can safely try ideas, fail miserably, then try again, eventually to success. Failing miserably is one of the best ways to improve your skills. Ask me how I know.

Git protects your files on your local computer. GitHub provides a cloud-based solution to protect your files, but also allows you to share your code with others, a key part of the open-access philosophy that we embrace for this course.

You will use Git and GitHub as part of your assignments. At first, you will not necessarily understand what you are doing or why. But, once you have done it a couple of times, I think it will then be easier for you to understand the what and why of the process. You will gain more understanding in an upcoming assignment.

Save that which is real

As you interact with Git and GitHub, you will be tempted to make "safety nets" for all of your files. Do you really need to do that? As long as you have the source files, like the R scripts and R Notebooks, you can generate the output. If you lose the source, then you lose the output. Think in terms of source and output, and remember that *source is real.* Output can be recreated from the source. You only need safety nets for source files.

If you find yourself tempted to protect your output files (e.g., HTML files), that is OK. You will get over it. Eventually.

The terminal or shell

We will avoid the **terminal** or **shell** when we can, but you will at least need it to install *git* as part of the first assignment. Here is the minimum need-to-know about using them:

- They are a (powerful) way to control your computer.
- OS-specific language!
  - Mac/linux/unix are about the same. In fact, Mac OS X, at its heart, is a BSD UNIX-based operating system called Darwin.
  - Windows has its own commands based on the Microsoft Disk Operating System (MS-DOS).
- Navigate with `cd`, `pwd`, and `ls`. `.` and `..`.

A great introduction to this is in happygitwithr: Appendix A.

You can even use the shell from within RStudio (`Tools > Terminal > New Terminal`), which opens a terminal window in the lower left console panel. **PC Users: If this does not work, please let me know.**

First reading assignment

As a reminder, R4ds is our textbook, *R for Data Science.* The book is written for people who want to develop a career in data science, which applies computer science and scientific methods to data analysis, often to so-called "big data" sets.

We will not be going that far down the rabbit hole. We will peer into the hole, But, this is a good book written by the authors of the R packages we will use throughout this course and it provides many examples and exercises that we will apply to biological data.

R4ds: Chapter 1

We will focus on making data tidy, transforming and wrangling data, and visualizing it (section 1.1). We will do just a little bit of modeling, leaving most of it to the data scientists and your own future growth. Here are a few things to note from this chapter.

- Section 1.3.3: Rectangular data refers to rows and columns, like a spreadsheet. The data you will most often come across for biological analysis will be in rectangular form.
- Section 1.4: Prerequisites mentions that having some programming skills is helpful, it is not essential for this course. You will learn those programming skills as part of this course. I will try to clarify places where I think it might get a bit too far down the rabbit hole. You can also ask questions in the internal discussion repo on the course website.
- Section 1.4.3: We will use Tidyverse throughout this course so install it now. You can type the command given in the text into your RStudio console window (lower left) or use the `Tools > Install Packages...` menu. Enter `tidyverse` into the space provided, be sure `Install dependencies` is checked, then click the `Install` button.
- Section 1.4.4: Use the same process to install only the `nycflights` package. Use the `Install Packages...` menu or enter `install.packages("nycflights13")` in the console. You can enter the slightly different command given in the chapter but you'll install packages that you do not need for this course.

R4ds: Chapter 2

This chapter is very short.