Context

In the 21st century, biologists-in-training are asked to be myriad things including naturalists, statisticians, wet lab technicians, science communicators, and software engineers to name only a few. The toolbox for an ecologist in training is only growing, and it's essential to develop the skillsets to make yourself marketable no matter what field you decide to pursue. One of the greatest new (and constantly evolving) skill sets that a biologist can learn is how to make their work Open. While there are a number of challenges to doing so, Open Science is full of opportunities for young biologists to make their mark, as highlighted by Allen & Mehler (2019 - PLoS Biology).

Open science practices are a nebulous set of methods put forward to help address some of the biggest pervasive problems in biology (and science more generally), including the replication/reproducibility crises and increasing numbers of fraud.

This table from Allen & Mehler (2019) nicely summarizes *what* Open Science methods are: drawing

Open science methods allow for more rapid scientific progress; the more open science is, the more widely it is cited and re-used. However, while we all might aspire to perfectly open science, nothing is ever perfect, and we must be willing to have conversations with ourselves, our collaborators/supervisors, data stakeholders, and institutions about what exactly “open” means and how to do it. You can find more on the different aspects of Open Science in this book.

For this course, we're most interested in the first and last items in the Allen & Mehler table, "Resources" and "Methodology". Throughout the course, you will be required to interact with a variety of softwares that are used today by biologists to create reproducibile and rigorous research products. For our purposes, this will be split into two main parts:

Programatic Analyses

It is becoming increasingly less acceptable in biology to publish work that has not been done in a reproducible, programatic way. That means that (mostly) gone are the days where using point-and-click software like Excel or SPSS is an acceptable way to do scientific analysis. Not only are these softwares proprietary (which is deeply antithetical to the principles of knowledge sharing in science for the benefit of society), but they are also innaccurate, and most importantly **hard/impossible to reproduce analyses in.** We believe *very* strongly that part of your training as burgeoning scientists and researchers, should be a) how to do **rigorous** science first and foremost, and b) how to do that rigorous science in a way that others can reproduce. For that reason, this class will be using programatic analyses **ONLY** (i.e. no Excel!!), and you will be expected to do all of your analyses *and* your write ups in the open-source statistical programming language R.

Version Control

In this course, we will also be using version control, a system that tracks changes to files through time and keeps "versions" of each of those changes, allowing you to see exactly how someone proceeded through their work, and also allowing you to revert back to a previous version of your files at any time. Version control is an essential aspect of open science, and in the sciences is usually implemented through Git/GitHub. Git is a mature, open source version control system developed by Linus Torvalds (who also developed the Linux operating system kernel), that interfaces with GitHub, an (unfortunately) for-profit company that acts as a git repository hosting service. Git is a command line-based tool, but GitHub, is the web-based graphical interface that lets us interact with our repositories.

Note: the Resources page has extensive links and videos that help explain Git/GitHub and help you get going with both.

Assignment Guidelines (10 points)

Throughout the course, we will use other assignments as a way to get experience with the fundamental open science practices including:

1. Performing reproducibile analyses in R
2. Writing all course material in RMarkdown documents
3. Submitting all assignments via GitHub repositories

While all assignments will be submitted via a GitHub repository, you will receive a formal grade on 3 components:

1 - GitHub Classroom Assignment (2 points)

As your first assignment of the course, you will perform a completion-based assignment to ensure you can `fork` a repository from GitHub Classroom, make changes to the repo, and `push` it back. More details are provided in the assignment itself.

2 - Population Model Code & Repository (4 points)

For this assignment, separate from the actual population model assignment, you will be marked on whether or not you followed the repository structure guidelines (2 points) and the code structure guidelines (2 points).

3 - Field Sampling Project Code & Repository (4 points)

For this assignment, identical to the above, separate from the actual field sampling assignment, you will be marked on whether or not you followed the repository structure guidelines (2 points) and the code structure guidelines (2 points).

1: Curious what markdown is? Read more: https://en.wikipedia.org/wiki/Markdown, https://jaantollander.com/post/scientific-writing-with-markdown/

Submission Details

Use the GitHub classroom link here: https://classroom.github.com/a/rZDwW2zV to accept the assignment and view submission instructions.