Reproducible Science with R

M.K. Lau

Class Time: 4.5 hours

Goals: Use R and other tools to conduct reproducible research.

Day 1

Overview: Why make your work reproducible and why R? (15 min)

Science is driven by the exchange of information and knowledge. Currently, there is a lot that we can do to make research more transparent and useful. A recent study (Stodden *et al.* 2018) demonstrated that only 26% of studies published in *Science* could be reproduced. This was even more striking given that the study was conducted after the *Science* had instituted it's open data policy.

What this and other studies point to is the need to provide well documented data as well as the software that were needed to conduct the study. Luckily, advances in open-source computer languages, such as **R** and **python**, provide a way to produce computations that can more easily document scientific research in a transparent, easily shared way.

In this course, we will cover how to conduct **reproducible** scientific research using the **R** programming language and supporting software that will enable researchers to more clearly and easily document

Reproducibility Framework (10 min)

Max(reproducibility) = Data * Software * Documentation

1. Setup your project so that there is a clear architecture (*RStudio*)
2. Work so that your computation from initial data to finished results will be coded (wherever possible), including data cleaning and processing steps (**R**)
3. Keep track of versions of your code (*git*)
4. Make initial data available (whenever possible, *git*)
5. Keep track of software dependencies (*packrat*, *git*)
6. Be organized, succinct in style, coding and documentation (**R**, *Markdown*, *formatR*)

If you're interested in more details on how to conduct reproducible research, see the **TEE** (Transparency in Ecology and Evolution) website http://www.ecoevotransparency.org/ and the **FAIR** reproducible research guide at https://www.go-fair.org/fair-principles/.

RStudio Tour (10 min)

It's time to take a break! (10 min)

Anatomy of an R Script (10 min)

- Typical flow: read-in, wrangle and then analyze
- comments
- function anatomy
- objects
- assignments
- basic plotting

Let's look a little more closely at basics.R

```r
# Basic R script

# MK Lau

# 15 April 2018

dat <- read.csv(file = "./data/data.csv")

cor.test(dat[,"x"], dat[,"y"])
```

Day 2

Project Architecture (10 min)

Max(reproducibility) = Data * Software * Documentation

In this section, we'll go over some project best practices that generally work for computational projects. We'll initiate a new project from *RStudio* and set up a file system with:

- **README**: describes the project and associated files
- **data**: folder to collect relevant data files or links
- **src**: where all of the **R** scripts should be kept
- **results**: this is where output from scripts can go
- **docs**: further documentation and relevant files (e.g. notes, papers)
- **bin**: ADVANCED - if you need to include other associated software

ACTIVITY: Ecological data project (10 min)

Setup a new project and outline a script that will:

Day 3

“Backing Up” Version Control (10 min)

Max(reproducibility) = Data * Software * Documentation

In this module, you’ll learn how to use the version control system, known as git, from RStudio. We will cover the basic topics of how to:

1. Initiate a project “repository”
2. Create, “add” and “commit” changes
3. View “diffs” and share a compressed project

We will not cover how to use the online git repository known as *github*. Using *github* provides a central server through which projects can be shared among collaborators in real-time via a system that keeps users from stepping on each other’s “digital toes”. To setup a free, private repository through *github.com*, go to their educational program webpage: https://education.github.com

Activity: Setup your ecological data project as a repository (10