Lab 1A: Installing R and RStudio

R and RStudio

R and RStudio are separate downloads and installations. R is the underlying statistical computing environment. RStudio is a graphical integrated development environment (IDE) that makes using R much easier and more interactive. You need to install R before you install RStudio. After installing both programs, you will need to install some specific R packages within RStudio. Follow the instructions below to install both:

Installation instructions

1. Follow the instructions to download R from the CRAN website for your operating system
2. Run the downloaded file to install R
3. Go to the RStudio download page and follow the instructions to download RStudio for your operating system

Getting Started with RStudio

The following gives a great overview on the differences between R and RStudio, familiarizing yourself with RStudio and starting a session in RStudio:

https://datacarpentry.org/R-ecology-lesson/00-before-we-start.html

Note: it's the start of an online tutorial, so it does mention a "code handout" that is relevant to that tutorial, but not this class.

Setting up RStudio

1. Create a folder where you are going to do all your lab work. Remember, organization is key! I recommend setting up a folder for all the labs, and then a sub-folder for each lab (e.g., ~/course1_labs/lab2/).
2. Start RStudio.
3. We need to set the working directory. This is where R will look for all your data files, scripts (more on those later) and save any of your work. There are a few ways to do this:
   - Go to Session -> Set Working Directory -> Choose Directory and browse to the folder you just created
   - In the Files Pane navigate to your lab folder, then click More -> Set as Working Directory
   - At the console prompt type `setwd('full/path/to/folder')`
4. Set Preferences to 'Never' save workspace in RStudio. To do this, go to Tools –> 'Global Options' and select the 'Never' option for 'Save workspace to .RData' on exit.'

Installing packages

The above instructions for installing R installs the **base** version of R. It has a bunch of default functions we can use. But there a lot of additional **packages** for R which contain even more functions for us to use.

```
install.packages("tidyverse")
```

Tips and Resources

Online book:

R for Data Science. Uses the tidyverse packages

A selection of beginner R courses from Data Carpentry and Software Carpentry:

R ecology lesson

R novice gapminder

R novice inflammation

R Markdown

This section is for those already familiar with R, but I'm including it in this lab script for anyone interested.

You can create documents that have text, figures and R code all in the same document using R Markdown. These R Markdown files (or R Notebooks - more later) are really useful for reproducible science because you can put all the information into one document in a clear, presentable way. I do all my research using these - I think of them as lab books, I don't keep a pen and paper lab book anymore. And all the lab scripts I've written for this course were done in R Markdown.

What is Markdown?

(adapted from markdownguide.org)

Markdown is a lightweight markup language that you can use to add formatting elements to plaintext text documents. Created by John Gruber in 2004, Markdown is now one of the world's most popular markup languages.

Using Markdown is different than using a WYSIWYG (what you see is what you get) editor. In an application like Microsoft Word, you click buttons to format words and phrases, and the changes are visible immediately. Markdown isn't like that. When you create a Markdown-formatted file, you add Markdown syntax to the text to indicate which words and phrases should look different.

For example, to create **bold** text you would type `**this text is bold**`.

To create your final document, you need to **render** your markdown file.

R Markdown and R Notebooks

In RStudio, you can create a R Notebook or a R Markdown file. In terms of how you write or code in them, they are the same. The difference comes in how they are rendered.

R Markdown files need to be **knitted**. When you knit a file, it runs all the code chunks, and formats all your text. You can knit a R Markdown file into a PDF or a html file or some other formats.

Whereas, a R Notebook file's default output is HTML. This means you can **Preview**, rather than knit, the file to render it. This is much faster because it doesn't run all the code, it just displays the output from the last time the code was run.

To start either a R notebook or R markdown file, press the symbol on the toolbar that has a green plus and white sheet of paper. Then select either 'R Notebook' or 'R Markdown'. In each case, the new file will have default/example text and information that explains how to create the file.

There's lots of information online about R Markdown, check out the RStudio R Markdown website, this reference sheet which shows examples of lots of different text formatting you might want to do, and this cheat sheet.