**Population Ecology (EEB319H1S) - Winter 2022**

**Lab 1 - Intro to *Excel & R***

**To do BEFORE Lab 1**:

Install *Excel* and *R* on your computer. As a UofT student, you may install the most recent version of *Excel* free-of-charge (*https://onesearch.library.utoronto.ca/ic-faq-categories/microsoft-365-proplus*). The lab is set up for the English version of *Excel*. The programming language *R* is open-source so you can download it for free (see instructions in *Part 2* below).

*During in-person labs we will have a limited number of laptops that you can borrow to do your labs. If you plan to borrow one of our laptops, make sure you figure out the best way to save your work.* *Please let your TA know ahead of time if you need to borrow a laptop*.

**Part 1: Intro to *Excel***

**Goals**: Learn how to use *Excel* to:

1. Enter data

2. Run basic data manipulation

* Writing a formula
* Copying formulae

3. Export (save) data as csv file *(will be used to input data into R programs)*

**1. Entering data**

Open *Excel* and you will see what is called a **worksheet** with letters that identify columns and numbers that identify rows. At the bottom there is a tab that says: “Sheet1”. You can change the name on the worksheet by right-clicking on the tab and using “Rename” - this becomes useful when you have several worksheets. Note that you could also add worksheets by clicking the “+” beside the tab(s). An Excel file with multiple worksheets is called a **workbook** (.xlsx).

Open the workbook we wrote for this lab, “EEB319\_Lab1\_ExcelExample.xlsx”, and take a few minutes to read the model and parameter descriptions in the “ModelDescription” worksheet. When you’re ready to simulate our model forward in time, click into the “ModelSimulation” (1-3) worksheets.

Time to enter data in your “ModelSimulation” worksheets. Position yourself in cell B3 (i.e., column B, row 3). The column names (“time”, “parasites”, and “immune\_cells” have already be set-up for you. Most programs, including R, are case-sensitive so later when you call a variable, make sure you enter upper- and lowercase letters exactly as in the column names in this example.

In column A, “time” refers to our time series; we’re interested in parasite and immune cell abundances over 50 time steps, evaluated every 1/10th of a time step. You could write these numbers manually in column A, but this becomes very tedious when you have large data sets and is prone to errors, so let’s do it with a **“Fill” function**. Note that the initial time in A2 is “0”, grab this “0” and a whole bunch of cells (>500 cells) below that. It does not matter exactly how many cells you grab. Then in the tabs above the worksheet go into Home/Fill (under Editing)/Series and a box will open. Leave “Step value:” as 0.1 (your time step) and under “Stop value:” write “50”. This should fill column A with all “time” values you need. As you might have guessed, you can change the start value, the step value and the stop value, so this is quite a useful function.

**2. Basic data manipulation**

Writing a formula

In column B, “parasites” refers to the parasite population abundance through time. In column C, “immune\_cells” refers to the host’s immune cell population abundance through time. You’ll notice in cells B2 and C2 the initial values of 1 (at time step = 0) for both “parasites” and “immune\_cells”. We’ll use our model to calculate population abundances forward in time.

Write the first equation in our model in B3, and the second equation in C3 (at time step = 0.1). Note that all formulae start with “=” and there is a wide range of signs and functions available in *Excel* (see list under Formulas/Insert Function). Basic operations are: *+* for additions, *-* for subtractions, *\** for multiplications and */* for divisions. As you write the equation, grab the cells with any value that is to be used in the model. That means click B2 and C2 when our equation call for *Pt* and *It*, respectively, as these terms refer to our last population abundance (at time *t*), and click on A3 when our equation calls for *ts* (time step). Enter parameter values provided in the “ModelDescription” worksheet (e.g. 0.001 for *k*). Double-check the equations you just entered and convince yourself that the results make sense - **ALWAYS double-check your calculations**!

Copying formulae

Once you are confident the equations you entered in B3 and C3 are correct, copy the formulae to the other rows of the time series. To do this, click B3, then hold down ctrl/command while clicking C3. Then, hold down the square in the bottom right of C3, and drag down our entire time series (to row 502). **Double-check that the formulae copied correctly and that the results** make sense.

**3. Exporting worksheets as .csv (*comma separated values*) file:**

Comma-separated values (.csv) files are text files that can be checked and edited easily, and that are compatible with all programming languages (e.g. *R*). So you will save the worksheets you just created into .csv files.

Before you do this, **make sure you have saved your data file as an *Excel* workbook file** (EEB319\_Lab1\_ExcelExample.xlsx). This is important when you have multiple worksheets since a single worksheet is saved in a .csv file and you will lose all other worksheets.

Position your cursor somewhere on the “ModelSimulation\_x” worksheet of the EEB319\_Lab1\_ExcelExampleworkbook, go to **File / Save as**, select the directory on your computer where you want to save this file (best to create a directory called “*R\_Projects/EEB319*” where you can store all files for this course), enter **File name:** ModelSimulation\_x, select **File save as:** csv format and click the **Save** button. Check that a file called “ModelSimulation\_x*.csv*” has been added to your working directory. Double left-click on it to open it and make sure it contains your data. Repeat this process for all three of your simulations (“ModelSimulation\_1”, “ModelSimulation\_2”, “ModelSimulation\_3”).

**Part 2 - Intro to *R***

*Written by: M. Jarvis-Cross (Last Updated: January 2021)*

**Content**:

1. Introduction:

* What is R?
* Downloading R and R Studio
* R Markdown
* Making Your Own Markdown Document

2. Let's Get Started!

* Setting your Working Directory
  + setwd()
  + Start a Project
* An Example for Today’s Lab: Within-Host Disease Dynamics
  + Importing data
  + Exploring data
  + Using Functions
  + Plotting data

3. After getting started…

* *For* loops
* Combining *For* loops and *If... else* Statements
* Writing Functions

4. Learning Resources

5. Lab 1 Assignment and Marking Scheme

**What is R?**

R is a computer programming language and environment for statistical computing and graphics. R is open-source, which means you can use it for free, and is maintained and updated by its users, who can write and upload "packages", which are units of sharable code that extend R's basic toolkit. As of June 2019, CRAN (Comprehensive R Archive Network), hosts over 14,000 packages, which means that if you're working on a problem, another R user likely has a solution!

For example, you can use a package called "*lme4*" to fit linear or generalized linear mixed-effects models, "*popbio*" to construct and analyse projection matrix models, "*ggmap*" to visualize spatial data on maps, and "*wesanderson*" to make your plots look like they were taken from the set of Moonrise Kingdom!

Resources:  
*R Project:*  <https://www.r-project.org/about.html>  
*Big Book of R:*  <https://www.bigbookofr.com/index.html>

**Downloading R and R Studio *[\*\*to do before first lab]***

Download and install ***R*** on your computer from: <https://utstat.toronto.edu/cran>

After downloading/installing *R*, you'll want to download/install ***RStudio***, which is just *R* with a graphical user interface (GUI) that is much easier to use: <https://www.rstudio.com/products/rstudio>

Try to open *R Studio* to make sure it works. In *R Studio* you'll see four windows. The top-left window is your editor, where you can write, run, and save scripts. The bottom-left window is your console/terminal, where you can run code without saving it in a script. The top-right window is your environment, where you can find any objects you've defined. The bottom-right window is where you can navigate between files in your working directory, view plots you've generated, or read package documentation. No panic - we’ll go through all this in detail during lab.

**R Markdown**

This document, along with most of the labs in this course, and your assignment submissions, will be written using R Markdown. R Markdown is a file format for making dynamic documents (documents that can be continually edited and updated) with R. R Markdown documents are written in markdown (an easy-to-write plain text format, similar to HTML and LaTex (pronounced "law-tech")) and contain chunks of embedded R code. More generally, writing in markdown allows you to create documents that smoothly integrate chunks of text, mathametical equations, hyperlinks, and chunks of code. You can open and edit R Markdown files (.Rmd) in R Studio.

To learn more about *R Markdown*, go to: <https://rmarkdown.rstudio.com/articles_intro.html>

<https://cran.r-project.org/web/packages/stationery/vignettes/Rmarkdown.pdf>

\*\*AN IMPORTANT NOTE\*\*: When we write Markdown documents, we can export .Rmd files to PDF, Word, or HTML via a process called "knitting". You should be able to knit this file ("*EEB319IntroR.Rmd*") to HTML by clicking "knit" at the top left of this page. Once R has finished knitting this file to HTML, a preview window will pop up. You can view the knitted document in HTML by clicking "Open in Browser". Opening this document in a browser will also show you how the formatting of a .Rmd file maps onto the knitted version of the document.

\*\*ANOTHER IMPORTANT NOTE\*\*: As previously mentioned, Markdown documents are neat because they can integrate chunks of text with chunks of code. To learn how we integrate a chunk of code, check of the example we use to set a working directory, below. Take a look at the "eval" argument. If we set \*\*eval = TRUE\*\*, R runs that chunk of code when knitting a document. If we set \*\*eval = FALSE\*\*, R will include the chunk in the knitted document, but won't run the code. When you submit assignments, you should make sure your code chunks are path independent (can be run on computers other than your own (you shouldn't have to worry about this if you've set your working directory correctly, which we'll go over in just a minute)), and set eval = TRUE, so that we can run your script quickly. You can check if your script runs smoothly from top to bottom by clicking the drop-down menu,"Run", in the top-right of the editor window (top-left window in default layout), and finding "Run All". In the "setwd()" example below, we've set eval = FALSE because we have different computers, and as such, the line we've written won't run on your computer. In upcoming labs, you'll notice that we've set eval = FALSE in some of our examples, and have included an empty code chunk, where we're set eval = TRUE, for you to write your own code below.

**Making Your Own Markdown Document**

You can make your own Markdown document by clicking \*\*File > New File > R Markdown\*\*. After you've opened that blank Markdown document, you can start typing, using this document, and the following resources as a guide.

Markdown Example: <https://www.math.mcgill.ca/yyang/regression/RMarkdown/example.html>

Formatting Cheatsheet: <https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf>

**Don't forget to fill in your name and student number at the top of the document**, next to "author:"!

**Let's Get Started!** Setting Your Working Directory:

We use working directories to tell R where it should look for and store files on our computer. We can start by creating a folder called "*R\_Projects*", and a subfolder called "*EEB319*". If you right-click on that subfolder, you should be able to see its pathname (e.g. on my computer: /Users/mjarviscross/Desktop/R\_Projects/EEB319). As a side-note, your working directory is similar to how you think about any folder. For example, let's download the course files from *Quercus* into our "EEB319" folders, and create subfolders for different labs.

There are two ways to set your working directory:

**setwd()**:

Use \*\*setwd()\*\* to set your working directory by including the pathname of your EEB319 folder:

setwd("*[copy path of your directory here]*")

This is the easier option, but not the best...

**Start a Project**:

If you set your working directory using *setwd()*, you'll be able to save all your files in the same place, but won't be able to save any of your objects or outputs. If you set your working directory by creating an R Project:

<https://r4ds.had.co.nz/workflow-projects.html>

you'll be able to save all your files in the same place \*and\* save all your objects and outputs. This will save you a lot of time in the long-run, and make your work reproducible.

Go to the top-right corner of your R Studio window and click "Project: (None)". Then, click the first option in the drop-down menu, "New Project". A new window will pop-up, and give you three options: "New Directory", "Existing Directory", and "Version Control". Click "Existing Directory", and use the browse button to navigate to your "EEB319" folder. Finally, click "Create Project" in the bottom-right of the pop-up window, and you're ready to go! As a side-note, you should be able to see all the files and subfolders in your working directory in the "Files" tab of your "Files, Plots, Packages, Help, and Viewer" window of R Studio (bottom-right window in default layout).

**Time to switch to R Studio:**

By now you have noticed that this handout is just repeating what is in the R Markdown file *EEB319IntroR.Rmd*, so please continue this Intro lab in RStudio.

**Lab 1 Assignment and Marking:** *(see end of EEB319IntroR.Rmd)*

Assignment:

1. Plot with new parameters??? And some esthetic changes

2. Answer all interpretation questions in *EEB319IntroR.Rmd* (*highlighted in red*)

Submit your assignment as an R Markdown file on *Quercus*.

Marking scheme:

\* Code with clear and concise commenting: 40%

\* Does your code run? 10%

\* Plotting: 25%

\* Interpretation questions: 25%