EEEB UN3005/GR5005
Lab - Week 01 - 28 and 30 January 2019

Xun Zhao

Introduction to R and RStudio

All lab and homework assignments in this course will be distributed as R Markdown files. You can think of R Markdown files as text documents that can also feature portions of R code in specially designated areas called "chunks." R Markdown documents can be "knit" into other document types. During the knitting process, the R code within the R Markdown document is executed and both text and code output is woven together (knitted) to make a final document, such as a pdf file. This is incredibly useful for our course purposes since you will be able to write both R code and text in response to assignment prompts and generate a nicely formatted pdf file as your final product for submission.

This lab is meant to reinforce some basic concepts in R and demonstrate how you use R Markdown documents.

Exercise 1: Code Chunks and Simple Calculations

A primary difference between R Markdown files and regular R scripts is that in R Markdown files, the R code itself is confined to what are known as code chunks. Code chunks are created by entering three back ticks, followed by an "r" in braces, and another three back ticks. Within this code chunk, you can write normal R code. Within a code chunk, you can run code line by line, as you would with any R script, or you can run the entire chunk of code with the "Run Current Chunk" button at the right of the chunk (green "play button").

To get familiar, first, add the numbers 3 and 4 using R code in the chunk below. Next, put a new line in the chunk that multiplies the numbers 5 and 6. Make sure you can get output running the code line by line and as an entire chunk.

```
3 + 4
```

```
## [1] 7
```

```
5 * 6
```

```
## [1] 30
```

Create a vector named `x` that contains the numbers 1, 5, and 9. Remember that you can create vectors using the `c()` function. Note that if you simply assign a vector to `x`, there will be no immediate output. In order to see the values contained within `x`, you'll have to type `x` on a separate line of the chunk. Make sure you do that as well.

Now, assign `x` the values of 1 through 10 using the colon operator (`:`) which will create an integer series between a starting and ending value. Type `x` again to confirm that the vector `x` has taken on a new set of values.

```
x = 1:10
x
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

Using an R operation, return the squared values of every element in `x`.

```
x ^ 2
```

```
##  [1]   1   4   9  16  25  36  49  64  81 100
```

Generate a variable `y` that is equal to the sum of all elements in `x`. Show the value of `y`.

```
y = sum(x)
y
```

```
## [1] 55
```

Make a vector called `panda_name` with two text elements. The first element should read "Ailuropoda". The second element should read "melanoleuca".

```
panda_name = c('Aliuropoda', 'melanoleuca')
```

Use bracket subsetting (`[ ]`) to return the second element of `panda_name`.

```
panda_name[2]
```

```
## [1] "melanoleuca"
```

Exercise 2: Working Directories

In R scripts or R Markdown files, the working directory describes where on your computer R is "looking" to find and save files. Consequently, R will usually be trying to retrieve files from and save files into your working directory unless you specify other file locations within function calls.

To get your current working directory location, use the function `getwd()`.

```
getwd()
```

```
## [1] "D:/OneDrive - mails.ucas.edu.cn/6th Term/IntroStatEcoEvol/Lab1"
```

Your working directory should be identical to the location of this R Markdown file on your computer. Therefore, R will assume all files you reference within the script are located in your working directory unless you specifically state otherwise through the use of appropriate filepaths.

With a typical R script, we can change the working directory with the function `setwd()` (short for "set working directory"). Alternatively, you can set your working directory within RStudio by clicking "Session" and selecting "Set Working Directory" in the menu. Unfortunately, modifying working directories is a bit more complicated with R Markdown files. For our purposes, it will be easiest to situate data and other files you may want to reference in your R Markdown script in a consistent location rather than altering your working directory.

That said, before moving onto the next exercise, create a new folder within your current working directory called `data` (unless you already have one). Note you can do this from your operating system itself (i.e., "Finder" on Mac operating systems) or via the file browser in RStudio.

Quick tangent: For your own ease of organization, it probably makes sense going forward to have something like a "Statistics for Ecology and Evolution" folder on the computer you plan to use for all class-related work. Within this folder, you could have, among others, `homework`, `lab`, and `data` folders to hold the relevant files. This will make your scripting more consistent throughout the semester since you'll be using similar filepaths the whole time. If you're a bit confused about all this, don't worry. The concept of directory structure and filepaths should make more sense following the next exercise.

Exercise 3: Importing Data

Often, we'll want to import external data into R. There are a number of functions that can be used for this purpose, but for files containing comma separated values (CSVs), `read.csv()` is an easy option. If you're unfamiliar with CSV files, they're a file type that serve similar purposes as XLS or XLSX files (and they can also be opened with Microsoft Excel), but they're also much simpler than Excel files. The primary argument needed for the `read.csv()` function is the filepath to the relevant file, input as a character string. For example, `read.csv("path/to/my_file.csv")` would read in a file called `my_file.csv` located in this particular filepath (but not assign it to an object).

The file `ebay_snake_captures.csv` can be found on CourseWorks or on the course GitHub repository. Download this file, move it into the folder called `data` that is in your working directory, and import it into R, using the `read.csv()` function. Note you'll probably need to move `ebay_snake_captures.csv` into your `data` folder depending on where the file downloads to your computer by default.

Once you import the data, can you report the column names of the resulting data frame? What functions might be used to inspect the data?

```
data = read.csv('ebay_snake_captures.csv')
colnames(data)
```

```
## [1] "Date"                  "Time"                  "TT"
## [4] "Species."              "X."                    "Comments.Observations"
```

```
str(data)
```

```
## 'data.frame':    457 obs. of  6 variables:
##  $ Date                 : Factor w/ 195 levels "1-Apr-03","1-Aug-03",..: 71 172 172 1
##  $ Time                 : int  1645 1400 1730 1045 1045 1030 1715 730 1645 1645 ...
##  $ TT                   : Factor w/ 3 levels "bucket","coffee can",..: 3 3 3 3 3 3 3
##  $ Species.             : Factor w/ 21 levels "Agkistrodon contortrix",..: 12 20 2 2
##  $ X.                   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Comments.Observations: Factor w/ 48 levels "","2 IN ONE!",..: 1 1 1 1 1 1 1 1 1 1
```

```
summary(data)
```

```
##         Date          Time               TT
##  14-Aug-03: 10   Min.   :  30   bucket    :159
##  28-Aug-03: 10   1st Qu.: 930   coffee can: 23
##  12-Aug-03:  9   Median :1630   snake trap:275
##  22-Aug-03:  9   Mean   :1388
##  15-Aug-03:  8   3rd Qu.:1745
##  29-Aug-03:  8   Max.   :2330
##  (Other)  :403
##                    Species.         X.           Comments.Observations
##  Seminatrix pygaea     :130   Min.   :1.000                   :297
##  Coluber constrictor   :125   1st Qu.:1.000   NEONATE         : 91
##  Agkistrodon piscivorus: 69   Median :1.000   JUV             : 11
##  Nerodia fasciata      : 38   Mean   :1.033   NEONATE UNMRKD:  7
##  Thamnophis sauritus   : 24   3rd Qu.:1.000   UNMRKD          :  4
##  Heterodon platirhinos : 12   Max.   :3.000   HATCHLING       :  3
##  (Other)               : 59                   (Other)         : 44
```

```
View(data)# It pops up a new window but shows nothing in the PDF.
```

Exercise 4: Knit To PDF

Now, we'd like to output the work we've done into a pdf file. Thankfully, all you need to do is click the "Knit" button in RStudio. Before you do so, make sure to change the `author` field at the top of this document to your name.

RStudio should pop up a preview of the resulting pdf file, but a copy will also be saved to your computer. Can you find it? Where was the file saved? What was it named?

Answer It is at the work directory shown above, which is the same folder as this Rmd file. It is named lab_week_01.pdf.

The R code `knitr::opts_chunk$set(echo = TRUE)` appears in the first code chunk of this document. Do you see it in your resulting pdf output? Can you figure out why or why not?

Answer It is because there is a parameter says 'include = FALSE', which makes the R codes not show in the PDF file.