Week 1: Introduction to R

Monday, 7th January 2019

1 R installation

R is a powerful statistical and data visualization program that is open (i.e., free software) and can run on different platforms (Windows, Unix, Linux, or Mac computers). A current version of R is installed on the Student Terminal Services to which you should have access to; however, you may choose to download a version of R on a desktop or laptop computer for easier access to this package. To install and run R on any computer, go to http://cran.r-project.org/ and click on the appropriate link depending on which platform you work on.

2 References

There is a tremendous amount of online resources for R – here are two examples to get you started:

The R Book
ftp://ftp.kyb.mpg.de/kyb/bresciani/Crawley%20-%20The%20R%20Book.pdf

An Introduction to R
https://cran.r-project.org/doc/manuals/R-intro.pdf

3 Introduction to R

The goal of this document is to introduce you to R, which you will be using throughout the course, to learn aspects of commonly used statistical analyses, to analyze data for your course assignments, and to create high-quality graphics and plots for data visualization.

R is the name of the programming language and statistical package itself but there are two convenient interfaces that you may also use: RStudio and Tinn-R. For this course, however, only the standard R platform is required. This first exercise will introduce you how to use R as a basic tool and how to plot your data. Hence, you will learn how to:

- Create a data frame;
- Load your data frame;
- Save your data;
- Plot your data.

4 What to Remember

4.1

Before you start, do not forget these important points:

1. Create a folder to save your work (Please note: if you are working on a UNBC computer, save your work on your H: drive. It is also not a bad idea to save your workspace on a memory key too, in addition to the H: drive).
2. Save your R script into the folder by clicking on "Save Workspace".
3. Define the working directory (into the same folder). This can be done by selecting "Change dir" under the "File" tab at the top of the R interface.
4. Variable names are case sensitive, so `y` is not the same as `Y`.
5. Variable names do not start with either numbers (e.g. 2Y) or symbols (e.g. %Y).
6. Variable names should not contain blank spaces (e.g. use `hydro.data` not `hydro data`).

4.2

In addition, to complete your assignments it is important to know that:

- Round brackets () are for function arguments (or conditional statements or loops); e.g., when using functions such as `mean(1:10)`, round brackets are used to tell the function what information you want it to use. In this example, you are calculating the mean of 1 + 10, which is 5.5. At the R prompt, type:
  `> mean(1:10)` # then type return and you should see:
  `> [1] 5.5`

- Square Brackets [] are for object indices: Square brackets inform R what part of an object you want to look at or use.
  For example:
  `> x <- c(3, 90, 120)` # Define a vector `x` and assign it three numbers (Incidentally, c() is a function that combines or concatenates values)
  `> x[1]` # Let's look at the first object – here `[1]` implies index 1 or the first object in vector `x`. When you type return, you should see:
  `> [1] 3`
  `> x[c(1, 3)]` # What about the first and third objects? Type return and you should see:
  `> [1]  3 120`
- `<-` is called assignment (gets)
- `>` is the R prompt at which line commands must be typed
- `$` is list indexing (the 'elements name' operator)
- `:` creates a sequence
- `#` creates a comment
- `+` is the R prompt when the previous line statement is incomplete (e.g. due to a missing closing bracket; as an example, at the R prompt type `5+` and press return, then type `3` and type return again.)

5 Simple Exercises

5.1 Demo and Help

A good way of getting an idea of what can be done in R is to get a list of what demos are available. At the R prompt, type `demo(graphics)` to see some of the graphic capabilities of R. Cycle through the examples by pressing return after each example.

If you want to see all of the parameters of a function plus some description, you can use the `help()` function. For instance, if you want to know what the `help()` function does you can type:

```
> help(help)
```

If you type `help.search("standard deviation")` a window will pop up with a list of functions. You can scroll through them to find the one you want to use.

```
> help.search("standard deviation")
```

If you try `help(sd)` you will get a documentation window describing the function.

```
> help(sd)
```

5.2 Creating Data

To create a vector (here with 8 elements) you can assign data to a vector name:

```
> vecname=c(3, 5, 2, 6, 1, 8, 9, 7)
```

or

```
> vecname<-c(3, 5, 2, 6, 1, 8, 9, 7)   # (<- is recommended although both are fine)
```

This assigns the values "3, 5, 2, 6, 1, 8, 9, 7" using the function c(), to the vector called "vecname".
You can check your data by typing the vector name:

```
> vecname
```

5.3 Creating a Data Frame

To create a data frame (or a matrix/table of data) you can use the following code:

```
> datname<-data.frame(column1=c(1,2,3,4),column2=c(2,4,6,8))
```

To view the contents of your data frame, then type:

```
> datname
```

Note how the columns are labeled "column1" and "column2" following the instructions in the data.frame command.

5.4 Creating Tables and Saving Data

To create tables, we have many options; however we explore only two options here:

Option 1:

```
> write.table(datname, file="Datname.csv")
```

Option 2:

```
> write.table(datname, file="Datname2.csv", row.names = F, sep = ",")
```

5.5 Loading Data

To do this exercise, first you need to download the "Temperature data (.csv file)" file with filename "temp.csv" from the website (http://weather.unbc.ca/310/). Then save it to the folder that you have already created (your working directory).

Load your data by using the following command:

```
> temp <-read.csv("temp.csv")
```

You can display the information by using the `str` function

```
> str(temp)
```

Check the summary of your data (e.g. mean, maximum, minimum, etc.) by using the following code:

```
> summary (temp)
```

Use the head and tail function to see your start and end timestamp and values by using the following code:

```
> head(temp, n=5)          # shows the data for the first five indexes (1-5)
> tail(temp, n=2)          # shows the data for the last two indexes (30-31)
```

5.6 Package Activation

Many packages exist to plot the data; however one of the most powerful packages is the "ggplot2". To install the package go to the packages tab and install the package. Activate the package by using the following code:

```
library(ggplot2)
```

For the most part, use of this package is not required to perform basic graphics or plots. Instructions will be given in the assignments if any special packages are needed to complete special analyses or graphs.

5.7 Plotting Your Data

Now let us perform a quick plot by using the following code:

```
> plot(x=temp$Year, y=temp$Winter)
```