Data Scientifique

GradProSkills

Introduction to R

Instructor: Yara Abu Awad

Data Scientifique

HOME

Data Scientifique fosters data science initiatives among researchers

Our mission is to help researchers develop and strengthen their analytical, computational, & programming techniques to facilitate health knowledge mobilization.

Data Scientifique offers opportunities for researchers to leverage advances in data management/integration, quantitative statistical methods, and data visualization through Educational Workshops, Enrichment Awards, and Data Clinic. Data Scientifique is a professional development catalyst for faculty, post-docs, and graduate students. Resources include tools to optimize research collaboration, archival data repositories, and open-science initiatives.

In September 2015, Data Scientifique was founded by Jennifer J McGrath, PhD MPH (PERFORM Chair, Childhood Preventive Health & Data Science) and Muhammed Idris, PhD (former PERFORM Postdoctoral Scholar, current TED Resident: www.ted.com/speakers/muhammed_idris). Housed within the PERFORM Centre, in February 2018 Data Scientifique expanded with a satellite office in the Vanier Library to make data services and workshops accessible to the wider community. Since January 2019, Data Scientifique is under the direction of Jennifer J McGrath, PhD MPH and Yara Abu Awad, ScD MS MBA (current PERFORM & Horizon Postdoctoral Scholar).

www.datascientifique.ca

Workshop Schedule

- In this workshop, the following topics will be covered:
  - Mastering R basics: will include an introduction to the R environment, packages and data types
  - Describing data: will demonstrate how to generate descriptive statistics, table outputs and simple statistical tests (i.e. t tests)
  - Visualizing data: will show participants how to generate multiple types of plots and charts

Objectives

- Give you the tools to start using R
- Show you where to get information when you need it
- Be patient, learning how to code takes time!

Plan for today

- I talk for about an hour
- Then you work in pairs to carry out exercises provided
- Why work in pairs? Peer to peer education can be very effective (I am not trying to torture you). I am also here to answer questions.

Why R?

- It's free
- There is a package for everything
- Beautiful visuals:

  https://plot.ly/r/

  https://ggplot2.tidyverse.org/
- A LOT of help online (google is your best friend):

  https://stackoverflow.com/

  www.rpubs.com

R and R Studio

Is the Engine

Is the graphical user interface (GUI for short).
aka: the nice limo that gets your there in comfort

What is code?

- Code is a set of instructions you give to a machine in a language it understands so it can do what you want
- Your instructions have to be clear and in the correct syntax (just like the English language!)
- Unlike humans, machines are very precise
- You need to watch out for spaces, capital letters and punctuation.

Code has to be planned and thought out

- Exercise
- Imagine you are giving someone instructions on how to make tea:
- Step 1:
- Step 2:
- Step 3:
- Step 4:

Types of Data

- Numeric:
  - e.g: 1, 2.13908, 100000
- Factor: categorical data
  - e.g: education variable has values ‘high school’ ‘University’ ‘less than high school’
- Character: text
  - e.g: ‘this is character data’
  - Note the use of quotations. If you don’t use quotations, then R will assume that you are referring to an object

Other types of data: Special values

- NA stands for missing
- Inf and –Inf stand for infinity and –infinity (i.e. after dividing a number by zero)
- TRUE & FALSE are logical values and can be used in Boolean statements
- &, | are the symbols of 'and' and 'or' in Boolean statements
- == is a test, is the value of the left equal to the value on the right?
- != is also a test but in this case 'is not equal to'

What is an Object?

- An object can be thought of as a file or a variable
- It is a *container* of data (or code) that is stored in R's 'working environment'
- The working environment is like your computer's RAM. A file can sit there until it is saved
- We can save data / code in objects and then refer to them in our code
- Objects are easy to create!

Creating an object

- I want to create an object named x which equals the number 2:

x = 2
Alternatively
x <- 2

Now, there is an object named x in R's working memory.
We do the same when we import / create datasets

How data is organized and stored

- Vectors
- Lists
- Matrices
- Data frames

Vectors

- Are a collection of multiple pieces of information. You can think of them as carriages linked together in a train.
- They can be made of numeric, character or factor data
- But each vector can only contain one type of data

numeric

9 19 200 30 45

character

'a' 'an' 'is' 'by' 'hot'

factor

'blue' 'yellow' 'green' 'orange' 'purple'

Lists

- Like vectors are a collection of multiple pieces of information.
- They can be made up of individual pieces of data or collections of data: vectors, data frames, matrices, other lists
- Unlike vectors, lists can contain many types of data
- Example: one list made up of 3 vectors:

numeric

9 19 200 30 45

character

'a' 'an' 'is' 'by' 'hot'

factor

'blue' 'yellow' 'green' 'orange' 'purple'

Matrices & Data Frames

Store data in rows and columns and therefore are 2 dimensional (like tables)

	Column 1	Column 2	Column 3	Column 4
Row 1				
Row 2				
Row 3				
Row 4				
Row 5				

Data frame vs matrix

- Matrix can only contain numbers therefore uses up less space when stored
- Data frame can contain many types of variables (factor numeric and character) in addition to column names (very useful!)
- Some R models will only accept matrices (i.e. elasticnet)

Manipulating data

- All of these types of data collections (vectors, lists, data frames & matrices) must be objects in R before we can do anything with them
- We can create them in R or we can import them if they exist already
- We can also export them and save them on our hard drive (or somewhere outside of R)
- We use R functions to do something with our data such as run a regression, generate a plot or carry out a t test

Functions & Packages

- A function is an object that contains code to do something
- It is a word followed by round brackets. For example, I can use the summary function to summarize my dataset named df which I have imported into the working environment as follows

```
summary(df)
```

- The R installation (base R) comes with many useful functions If you need to do something that is not covered by a function in base R, you can install an R package that contains the function you need
- R Packages contain a collection of functions

Manipulating data: using brackets

- Square brackets [] grab a slice of an object (example numeric vector we saw earlier)
- Round brackets () contain information needed by a function (also known as arguments which are separated by commas). The summary function we saw earlier needed the name of the data frame we wanted to summarize
- Curly brackets {} enclose code in a function or a loop (we will not use these in this workshop)

Manipulating data: using mathematical operators

- In R you can use the standard symbols to:
- Add +
- Subtract –
- Multiply *
- Divide /
- To the power of ^
- Order of execution is outlined in multiple places online but when in doubt use brackets

Your Working Directory

- A directory is a folder
- A path is the location of your folder
  - Windows: C:\Users\vlad\notes.txt
  - Mac: /Users/vlad/notes.txt
- Your working directory is the folder that all files are imported from or saved to by default, unless otherwise specified
- Let's have a tour of R studio & set the working directory