QTM 150

Week 2 – R Basics I

Umberto Mignozzetti
Feb 05

Today's Agenda

- DataCamp.
- Why is it so hard to program?
- Getting Started with R and R Studio.
- Computations with R.
- Working with Vectors.
- Functions and Arguments.

DataCamp

DataCamp

- We partnered with DataCamp to provide you the first two R classes.
- These classes are going to teach you the very basics.
- The advantage of DataCamp, in these first days, is that you don't need to have R in your machine to get started
- Let's check my DataCamp?!

Questions?

Why is it so hard to program?

- Can you read this sentence: 4 exampel in Ingli$h you kin get prackicly evRiThing rong-rong-rong and sti11 be undr3stud.
- You can do it because English is a Natural language.
- R, Python, and other programming languages are formal languages.
- This means that you should pass the exact instructions for the computer to it do what you want to get done.
- Learning how to program is to learn how to speak a very formal language with a thing that doesn't understand informal writing.
- In this sense, as with learning any language, practice makes it perfect!

Why is it so hard to program?

- To understand R, you have to learn the syntax of the program.
- In the beginning, it is going to be hard. Then, with practice, you will see that there is a logic behind the program.
- This will be equivalent to fluency in a language: you will be able to strategize your coding even when you don't exactly know what to do.
- In my own experience, it took me around two months to become familiar with most of the rules.

R and R Studio

- R is the programming language that we are studying here.
- You probably by now googled R and know its history and how widely it is used.
- R Studio is **not** R! It is an Integrated Development Environment. Sort of a nice front-end to R.
- Let's check out R and R Studio!

R and R Studio

- Source editor x console.
- Files and working directory.
- Packages: expanding our R.
- How to get help?

Questions?

R as a Calculator

- We can use R as a fancy calculator
- Check out the operations: `+`, `-`, `*`, `/`, `^`, `%%`, `%/%`.
- Creating and operating with variables. Syntax:

```
var ← things_I_want_here
```

- **Your turn**: Compute the average of the following numbers: 1, 2, 3, 4, 5, 6

R as a Calculator

- In R, as in all other computer languages, we `do things` with `stuff`.
- The variables that we created are the `stuff`.
- The operations and functions are the ways that we `do things` with the variables.
- The `stuff` in R we call **objects**. In DataCamp, you will learn the most common.

Questions?

Variables and Vectors

- Vectors are collections of numbers.
- We use the function `c(.)` in R to put them together.
- Let's create some vectors in R.
- To do things in R (apply functions), we need to understand how they work.
- The syntax for functions is:

```
func(arg1, arg2, ...)
```

- **Your turn**: Compute the average of the following numbers: 1, 2, 3, 4, 5, 6. But now, save them in an object called `my_numbers` and use the function `mean()`.

Sequences and Repetitions

- We can also create vectors with sequences and repetitions.
- Sequences can be built with `:` or the function `seq`.
- Repetitions can be built with the function `rep`.
- **Your turn**: Compute the average of the following numbers: 1, 2, 3, 4, 5, 6. Save them in an object called `my_numbers`, but create them with sequences. Then, use the function `mean()`.

Questions?

Have a great weekend!