

The Basics of Coding

Election Data Science

Peter Licari

2020-09-08

Welcome!

**Y'all are gonna help me make my
lunch.**

The challenge:

- You guys are going to break out into two groups. Each will have:
 - An instruction writer.
 - An instruction reader.
- You have five minutes to write me instructions for how to make a peanut butter jelly sandwich.



Congratulations! You just wrote an algorithm!

Per Wikipedia: Algorithms are simply "finite sequences of well-defined steps typically used to solve a problem or some class of problems."

To understand coding in R,
there's a concept we need to
revisit.

$$f(x)$$

When you take a political science class and the instructor starts talking about algebra



I promise, it's gonna be ok.

Can't have "functions" without "fun"! (I'm so sorry)

- Normally when you see functions, it's often in the context of something like this:
 - $f(x) = x^2 - 1$
 - Say x is 2. Then $f(2) = 2^2 - 1 = 3$
- But we have to think **broader**.
- *Functions* are actually a set of steps that takes an inputted object and performs a transformation on it.
 - *transformations* are simply a sequence of pre-defined steps where actions are performed upon the input, resulting in an output.

Objects can also be generalized.

- **Single numbers aren't the only thing that can be input into a function.** - This is what happens when you make a graph out of a mathematical equation.
 - You're (usually) passing all possible values of x into $f(x)$ and plotting the outputted value.
 - Or at least all values within a set range.
- **But you can also make functions for non-numeric objects.**
 - A function that takes "cucumbers", "apples", and "bananas" and rearranges them alphabetically.
 - The list of fruit are an *object*; the function takes the object and then spits out a transformation.
- **You can also make functions for groups of objects (which are, themselves, objects)**
 - Function that takes a matrix $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$ and halves all odd numbers.
 - The matrix as a whole is an *object*; the function works on the elements within the object and transforms them.

To understand programming in R, two slogans are helpful:

- Everything that exists is an object.
- Everything that happens is a function call.

— John Chambers

When you do data science in R, your task is to use preexisting functions (and sometimes create your own!) in ways that take the desired objects (i.e., "data") and transforms them so that they are usable and useful for later analyses.

In so doing, you define the steps that you want the data to go through---the order and scope of the transformations. It is very easy for things to go awry and for you to end up with something completely different than what you thought when you sat down and started. So you have to be conscious and deliberate in your steps, and in your coding, or else your analyses will be plagued by otherwise avoidable errors.