

SPEC REU: Introduction to R

Alix Ziff

Summer 2020

Ready or not

Today we will explore R and learn the very *very basics*. We will go over what R is, how to write a line of code, basic arithmetic, and getting help.

This section would not be possible without the previous efforts and brilliance of Miriam Barnum and Therese Anders. Except for the alliteration and puns... that's all me.

R'nt you going to explain what R is?

R is a programming language for statistical computing and data visualization. It's open source (free and frequently updated) unlike Stata or SPSS. R is maintained and developed by a vibrant community of programmers and statisticians (and Therese!) and offers packages for just about any statistical task imaginable.

For political science and IR scholars, R has become the dominant programming language in the field and is used for everything from regression analysis to data management or even making pretty plots.

R we gonna do this or what?!

If you have not done this: <https://www.rstudio.com/products/rstudio/download/...> do it. You will need both 'R' and R studio.

R Studio

Generally, students and scholars type in an **R script** that they save to keep track of their code and operations. The **Console** is where the computing is done. The **Environment** keeps track of the data you are adding or manipulating. The **Files, Plots, Packages, Help and Viewer** is where you can view visualizations, get information on functions and packages, or load a file. **You will type into either the R script or the console.**

I command you to...

To make a command (also called a function) you can either highlight your code and click 'Run' at the top of the R script or you can highlight and press the 'ctrl' and 'enter' keys simultaneously.

You can make comments that will not run as code using a '#' before any word or line

I object!

In R, you can assign values to an **object**. This is how R stores information and any data you create, use, or manipulate. You can name objects almost whatever you want. **Except** you cannot start them with numbers, use special characters, or anything that doubles as an operator or function (EX: add).

For example: if I wanted to store 10102020 as my favorite number I would do the following

```
luckyuno <- 10102020
```

Then I can see that value using the function 'print'.

```
print(luckyno)
```

Arithmetic in R

You can also use R as a calculator.

```
5*2
```

You can use these objects instead of writing out the value itself.

```
luckyno/2
```

Or make new objects

```
haha <- 1601.6 * 5
```

Or change the value of existing objects

```
haha <- 8007393-742
```

There are several logical operators you can use that will result in TRUE or FALSE Less than < Less than or equal to <= Greater than > Greater than or equal to >= Exactly equal to == Not equal to != Not x !x x or y x | y x and y x & y %in% Testing whether a value is contained within a set.

It's like asking R a question. For example: *is 8 in the vector (a collection of ordered information) 2,3,4,5?*

```
8 %in% c(2,3,4,5)
```

Vectors & Variables

A vector is the simplest type of data you can work with in R. It is essentially a list of data of a single variable type. You should be familiar with Numerical, Character, Logical and Factor variables. • Numerical: Well, any number. • Character: This is what data (and other programming languages such as Python) calls a string. We typically store any alphanumeric data that is not ordered as a character vector. • Logical: A collection of TRUE and FALSE values. • Factor: Think about it as an ordinal variable, i.e. an ordered categorical variable.

To create a vector we use the function `c()` (“concatenate”) to combine separate data points. The general format for creating a vector in R is as follows:

```
name_of_vector <- c("what you want to put into the vector")
```

The “Oh No!”s: arrows, quotes, and typos

Let's avoid some errors:

1. Make sure your operator is facing the right direction <-
2. Make sure you do not have a # before a command. That makes it a comment which won't run in R
3. R is case sensitive
4. Make sure you have both sides of the quotation mark or parenthesis
5. CHECK FOR TYPOS: typos are often the cause of your ‘Oh No!’ moment
6. Make sure all your objects have unique names (you may have overwritten the object)

How to get HELP

Google. No seriously. If you are struggling and it isn't any of the above, then someone has likely had the same struggle. There are several websites like linked phrase that know everything about everything R.

R Studio Help Menu It is the tab between ‘Packages’ and ‘Viewer’ and it can help with how to order things, what a package does, etc.

SPEC Statistics Consulting Hours There’s at least an hour every single day with a brilliant SPEC student or doctoral affiliate. Take advantage.