Introduction to R

Rob Williams

UNC-CH Political Science

January 15, 2019

Overview

- Textbook
  - Can download chapters 1 and 2 from http://qss.princeton.press/
  - Run the R code in the book!
    - You can also download it from the book website
- DataCamp
- First day survey
- Installing R
- Basic math in R

DataCamp

- You should have access by now
- Let me know if you don't
    - Check your spam folder!
- Only 12 people have enrolled in the course
- First assignment due **before** class Thursday

Survey Results

- Concerns over flipped classroom format
- Worries about general computer skills
- Concerns about programming in general and R specifically
- Plans to ask lots of questions
- Math anxiety
- Desire to understand what tables/figures actually **mean**
- Excitement to learn data skills

Installing R

- Download R
  - https://cran.r-project.org/
- Download RStudio
  - https://www.rstudio.com/products/rstudio/download/

Basic R: Math

You can use R as an overpowered calculator

```
3 + 2
```

```
## [1] 5
```

In addition to addition, R understands subtraction

```
3 - 2
```

```
## [1] 1
```

multiplication

```
3 * 2
```

```
## [1] 6
```

and division

```
3 / 2
```

```
## [1] 1.5
```

Basic R: Functions

You can also get fancier with exponentiation

```
3^2
```

```
## [1] 9
```

and roots

```
sqrt(3)
```

```
## [1] 1.732051
```

this last one is a **function** R has lots of functions that you'll come to know and love

Basic R: Vectors

We can use the `c()` function to concatenate (combine) two numbers into a **vector**

```
c(3, 2)
```

```
## [1] 3 2
```

a vector can be made out of any number of **elements**

```
c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

Basic R: Objects

We can assign any number to an **object** using the **assignment operator** <-

```
x <- 3
y <- 2
```

once we've assigned a number to a object, we can access it any time by using the object name

```
x
```

```
## [1] 3
```

```
y
```

```
## [1] 2
```

Basic R: Objects

Object names and can letters and numbers

```
spring2019 <- 1
```

Object names cannot start with a number

```
2019spring <- 1
```

```
## Error: <text>:1:5: unexpected symbol
## 1: 2019spring
##         ^
```

You can use special characters like _ or .

```
spring_2019 <- 1
```

but they can't contain **operators** like +, -, *, ^, or !

```
spring-2019 <- 1
```

```
## Error in spring - 2019 <- 1: object 'spring' not found
```

Basic R: Data types – numbers

Everything we've done so far as used numbers. R has multiple **data types**, of which numbers are just one. You can use the `class()` function to identify the data type of any object

```
class(x)
```

```
## [1] "numeric"
```

You can also do this for numbers you haven't assigned to an object

```
class(3)
```

```
## [1] "numeric"
```

Basic R: Data types – text

If something isn't a number, it's probably a **string**. Strings are how computers store and read text

```
'UNC'
```

```
## [1] "UNC"
```

Notice the quotes around UNC in the console output! You can use single or double quotes

```
"Chapel Hill"
```

```
## [1] "Chapel Hill"
```

but you can't mix them

```
"North Carolina'
```

```
## Error: <text>:1:1: unexpected INCOMPLETE_STRING
## 1: "North Carolina'
##    ^
```

Basic R: Data types – logicals

The last data type you need to know are **logicals**. These represent **true** and **false** in the logical sense

```
TRUE
```

```
## [1] TRUE
```

You can also use `T` and `F` to shorten them

```
F
```

```
## [1] FALSE
```

You will encounter logicals whenever you make a comparison between two objects

```
3 > 2
```

```
## [1] TRUE
```

Vector math

You can do all the standard arithmetic operations between a vector and a **scalar** (an object with only one element)

```
z <- c(2, 4, 6, 8)
z / 2
```

```
## [1] 1 2 3 4
```

You can only perform operations between vectors if the length of the longer object is a multiple of the shorter one

```
c(1, 2) * z
```

```
## [1]  2  8  6 16
```

Notice how R cycles through the elements of `c(1, 2)`; the first element of the result is $1 \times 2 = 2$, the second is $2 \times 4 = 8$, the third is $1 \times 6 = 6$, and the fourth is $2 \times 8 = 16$.

Accessing vectors

You can access the elements of a vector individually using the [] square bracket operator

```
z[1]
```

```
## [1] 2
```

```
z[3]
```

```
## [1] 6
```

Once you do this, R treats the result like you typed in that number yourself. This means you can do math with specific elements of vectors

```
z[2] * 3
```

```
## [1] 12
```

You can access multiple elements of a vector by combining square brackets with concatenation

```
z[c(3, 4)]
```

R quirks

R doesn't care about spaces. In fact, it ignores them entirely

```
3+2
```

```
## [1] 5
```

```
3 + 2
```

```
## [1] 5
```

```
3+        2
```

```
## [1] 5
```

R is case sensitive

```
color <- 'blue'
Color
```

```
## Error in eval(expr, envir, enclos): object 'Color' not four
```

Getting help

To find out what any function does in R, just type a question mark before the name of the function

```
?mean
```

Leaving notes to your future self, or someone who's grading your code, is super important. You can do this by writing **comments**. To write a comment, just use a #; R ignores anything after the # on a line

```
# you can write comments on their own line
mean(c(1, 2, 3)) # or at the end of a line
```

```
## [1] 2
```

```
# 3 * 2 won't produce anything
```

Hands on with R

- Download today's R script from Sakai and open it up in RStudio