Discovering Objects

Unit 1 - Lab 3

Directions: Follow along with the slides and answer the questions in **BOLDED** font in your journal.

What are objects?

- When data is loaded into R, it's stored as an *object*:
  - Don't have to type out all of the data's values each time you want to use it.
  - Just use the name of the object (our data in this case)
  - For example, `View(cdc)` is telling R to show you the object called *cdc*.
- Data is just **one** object that R can use.

Why are objects awesome?

- Using objects makes it easy to:
  - Store information
  - Manipulate information
  - And use information!

How about an example

- Let's suppose we wanted to find the sum of all integers between 5 and 14
- **Using RStudio, calculate the sum of all integers between 5 and 14**
- One way to find this sum might be

```
5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 + 13 + 14
```

- Now suppose you wanted to calulate the difference of the sum you calculated and each integer between 5 and 14.
  - Meaning,
    `95 minus 5`
    `95 minus 6`
    `95 minus 7` and so on ...
- **Go ahead and calculate these differences**
- You should notice by now that doing these calculations is tedious.
- Let's use objects to make our lives easier!

Creating objects

- Every *object* needs 3 things:
    1. A *name*
    2. The *assignment* symbol
    3. A *value*
- We'll start by creating a **vector** of values for integers between 5 and 14
    - A **vector** is a single *object* that can contain many *values*

Making vectors (tra-la-la!)

- Let's create our **vector** of integers between 5 and 14:
    - We'll *name* it `numbers`
    - We'll *assign* it values using `<-`
    - Our *values* will be the numbers 5 through 14
- Write the following code into your console.

```
numbers <- c(5, 6, 7, 8, 9, 10, 11, 12, 13, 14)
```

- Click on the *environment* pane.
    - Objects you create will show up there!

Let's review what we just did

```
numbers <- c(5, 6, 7, 8, 9, 10, 11, 12, 13, 14)
```

- This line of code tells R:

1. Create a single *object* called `numbers`
2. **Combine** the numbers 5 through 14 into a *single object* using the `c()` function
3. **Assign** the values we combined, using the `<-` symbol, to the object we created.

- The `numbers` object you created is an example of a *vector*

```
numbers <- c(5, 6, 7, 8, 9, 10, 11, 12, 13, 14)
```

- After running the code:
    - The *object* pops up in the *environment* tab.
    - Nothing is printed in the console.
- **What happens if you type `numbers` into the console and hit *enter*?**

Try the example again using objects

- Let's go back now and redo our previous example:
- Create the object

```
numbers <- c(5, 6, 7, 8, 9, 10, 11, 12, 13, 14)
```

- **What type of object is this again? Why?**
- R contains many convenient functions.
- One such function is the `sum()` function.
  - It takes vectors of numbers and adds all the values in them together.

```
sum(numbers)
```

- **Do you think using objects is any easier than just adding the numbers together in the console? Why or why not? Be honest!**
- Let's save the sum of all numbers between 5 and 15 as another new object.
- Run these lines of code:

```
total <- sum(numbers)
```

```
total
```

- **What is the *name* of the object? What is its value? How did you tell R to *assign* the value to the object?**
- Finally, let's calculate the differences of the sum and the numbers
  - Meaning, `95 minus 5`, `95 minus 6`, and so on
- We already have the sum (`total`) and all of the values (`numbers`)
- To calculate the differences of these two objects:

```
total - numbers
```

- **In your opinion, was using objects for this task easier than before? Why or why not?**

Other ways to save time!

- The `sum()` function was just one function that saves you time.
- Compare the outputs of the following:

```
c(5, 6, 7, 8, 9, 10, 11, 12, 13, 14)
```

```r
5:14
```

```r
seq(from = 5, to = 14, by = 1)
```

- **How are these 3 methods different?**
- **Explain how you think R interprets each line.**

. . . a twist

- Compare the following outputs:

```r
5:14
```

```r
14:5
```

- **What do you notice about the outputs?**

. . . a final twist

- Compare the following outputs:

```r
seq(from = 5, to = 14, by = 1)
```

```r
seq(from = 5, to = 14, by = 3)
```

```r
seq(5, 14, 1)
```

- **What do you notice about the outputs?**
- **For each line, explain how each part effects the output**

On your own:

- **Create a *vector* of numbers and save it as an *object***
- Run the commands with *your_object* (whatever you decide to name it):

```r
your_object + your_object
```

```r
your_object * 2
```

```r
cumsum(your_object)
```

- **Describe the computations that take place for each output**