**POLI 175 Lab Session**

*Getting Started in R: Lab Material #A*

1. **R as Calculator!**

   Addition: +
   Subtraction: -
   Multiplication: *
   Division: /

| *EXERCISE 1:* spend a few minutes doing some basic arithmetic. |
|---|

2. **"Object" Creation**

In R, we can create "***objects***" that contain a datum or data. Here is how we make objects:

Name of Object <- Content of Object

Note: the content of the object should have quotes (" ") if it contains text (aka string); however, quotes are not needed if it is a number. For example:

- object <- 2
- object <- "two"
- object <- "this object is named two"

| *EXERCISE 2:* spend a few minutes making different objects in R; some objects should be numerical, while others should be textual. |
|---|

In this lab session, we will focus mostly on numerical objects (although with some exceptions below). The next thing to know is that we can apply the math operators to numerical objects.

| *EXERCISE 3*: spend a few minutes doing some basic arithmetic using *objects* |
|---|

It is also possible to create new numerical objects by combining old numerical objects using the math operators.

| *EXERCISE 4*: spend a few minutes making new numerical objects using only other numerical objects and the math operators. [This may be a bit tricky but see if you can figure out how this works. If not, consult a classmate and see if you all can figure it out] |
|---|

A few notes about objects:

- If you give two objects the same name, R will overwrite the first object you created.
- You can remove objects from your R environment using the rm() function.

- For example, if I wanted to remove the object "object" (from the example above), I would enter rm(object) into R.

### 3. Using Functions

Once we have objects, we often want to do something with them!

Functions are commands in the R language that allow us to manipulate objects.
For instance, R has a square root function: sqrt(). If I entered sqrt(9) into R, it would return a value of 3.

| |
|---|
| *EXERCISE 5*: calculate the square root of a number. Now create an object that takes on the value of that number, and calculate the square root of the object. |

All functions have a basic structure:

*Function Name(Required Argument, Optional Argument Name = Optional Argument).*

We will ignore the optional arguments for now (they are optional, after all), and just focus on the required arguments.

A "required argument" is just information R needs to carry out the function. So, the required argument for the sqrt() function is a number (or numerical object). If you didn't include this information in the parentheses, R can't calculate the square root.

Here are a few other basic functions:

- Mean: mean()
- Median: median()
- Absolute value: abs()

| |
|---|
| *EXERCISE 6:* spend a few minutes trying out the mean, median, absolute value, and square root functions. *Optional challenge*: can you figure out how to use the square root and absolute value functions at the same time? |

Still feeling confused about functions? Take a few minutes to review section 1.6.3 of DASS and then come back here to repeat Exercise 6 before moving on.

### 4. Creating Vectors

A vector is just an object that can store more than one value at a time. So far we've only created objects with a single value, such as 2 or "hello".

We can use the **c()** command to create vectors. Here, **c** stands for *combine*. It looks like this:

Vector Name <- c(Value1, Value2,…Value*i*)

A vector can contain as many or as few values as you would like—I indicated this above by writing "….Value*i*" where *i* simply indicates whatever number of values you are including in your vector.

*EXERCISE 7*: create a vector that is filled with the numbers from 1 to 10.

We can also make vectors with text values instead of numeric values.

*EXERCISE 8*: create a vector that is filled with the letters of your first name (don't forget about quotes [" "] rule when working with textual values)

Just as we can analyze objects using functions, we can also analyze vectors using functions (after all, a vector is just an object with multiple values).

*EXERCISE 9:* calculate the mean and median of the vector you created in Exercise 7.

## 5. Creating Dataframes

We can combine vectors into a dataframe. Note: the vectors must contain the same number of values for this to work.

We create dataframes using the function: data.frame(). So, it looks like this:

Name of DataFrame <- data.frame(Vector1, Vector2,…Vector*i*)

Like with the creation of vectors, I use *i* to indicate that you can include as many vectors as you want—that is, up to *i* number of vectors (1, 2, 5, 100, 999, etc.).

*EXERCISE 10*: using what you've learned so far in this lesson, convert the data below into a dataframe in R (note: the variable names in the data should be *the names of the vectors*).

| id | name | age | partisanship | state | education | vote2024 |
|---|---|---|---|---|---|---|
| 1 | Ben | 34 | Independent | California | 6 | 0 |
| 2 | Irene | 77 | Republican | Wyoming | 1 | 1 |
| 3 | Antonio | 35 | Democrat | Oregon | 3 | 1 |
| 4 | Isaac | 43 | Independent | Nevada | 4 | 0 |
| 5 | Anthony | 29 | Republican | Illinois | 5 | 1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 6 | Emily | 20 | Democrat | California | 4 | 1 |
| 7 | Cecilia | 56 | Republican | Florida | 5 | 0 |

You can save data frames into csv file

write.csv(name of the data frame, "name.csv")

By default, R saves the file into your working directory.
To check your current working directory, type: getwd()

Later, we'll learn how to set your working directory so you can choose exactly where files are saved (for example, your Desktop or a project folder). For now, just remember:

getwd() = "Where am I working right now?"
write.csv() = "Save my data frame to that folder."

We can also import data into R using a function. For example, if we have a .csv file, we can use the read.csv() function. It would look like this:

Name of DataFrame <- read.csv('Pathname of .csv file')

*EXERCISE 11*: Save the data frame as .csv. Then using the read.csv command to import the data into R.