Elyse McCormick
ECO 602
9/7/2022

Lab 1: R Fundamentals 1

Q1 (2 pts.): Explain why the outputs of the two lines are different.

The two outputs are different because of the placement of parentheses. The first line of code has no parentheses, which means that R reads it as an expression to be evaluated, whereas anything within quotes is considered as a string of literal text, and it will not be able to interpret it mathematically.

Q2 (1 pt.): Is c_1 a variable, or a function? How do you know?

c_1 is a variable that you're assigning a set string of integer data to. This can be used computationally.

Q3 (1 pt.): Is c_2 a variable, or a function? How do you know?

c_2 is also a variable; you're simply assigning a different type of data to it. Because since it's a string of text, it's not computational, but you can store that line of text as c_2.

Q4 (1 pt.): If c_1 and c_2 have different values, why?

c_1 and c_2 have different output values because of their differences as an expression and a string of text, respectively. c_1 gives you the list of numbers that could be used for computation, while c_2 gives you "c(1, 2, 3)" as the output.

Q5 (1 pt.): What are the dimensions of the matrix (i.e. how many rows and columns)?

There are three twos and two columns in this matrix.

Q6 (2 pts.): Write R code to retrieve the element of mat_1 that has a value of 3.

```
my_bool_vec <- my_vec == 3
my_bool_vec
data.frame(my_vec, my_bool_vec)
my_vec[my_bool_vec == TRUE]
```

Q7 (1 pt.): Paste the code you used to create mat_2.

```
mat_2 <- matrix(my_vec, nrow = 2, ncol = 3)
```

Q8 (1 pt.): Paste the code you used to create mat_3.

```
mat_3 <- matrix(my_vec, nrow = 3, ncol =2)
```

Q9 (1 pt.): Did R use rows or columns to recycle/distribute the values in my_vec?

It used columns to re-distribute the values in my_vec across the matrix.

Q10 (1 pt.): Using my_vec, create a matrix, mat_4. mat_4 must have a total number of elements that is not a multiple of 3.

mat_4 <- matrix(my_vec, nrow = 2, ncol = 5)

Q11 (1 pt.): How did R handle the recycling/distributing of values of my_vec in mat_4?

It didn't like it. It gave me the Warning message: " In matrix(my_vec, nrow = 2, ncol = 5) :   data length [6] is not a sub-multiple or multiple of the number of columns [5]"

It did run the code though, and it distributed them by column, and then recycled numbers it hadn't used yet (i.e. once it got to 6, it started over at 1).

Q12 (8 pts.): For each of the 8 lines, answer the following: A. Did the line return a 1: value, 2: error, or 3: NULL? B. What type of subsetting operation was used (or attempted)? C. If it did not return an error describe, in ordinary English, a plausible explanation of how R could have performed the subsetting.

A)

my_list_1[[1]]	5.2
my_list_1[[as.numeric("1")]]	5.2
my_list_1[["1"]]	NULL
my_list_1[["one"]]	value
my_list_1$one	value
my_list_1$"one"	value
my_list_1$1	Error: unexpected numeric constant in "my_list_1$1"
my_list_1$"1"	NULL

B + C) In subsetting, using brackets will find the first element of your vector. The double brackets used here asks R to simplify the subset as much as possible while it's searching for the element. Using a dollar sign allows R to search for named elements. That being said, this is what I think each piece of code did.

my_list_1[[1]]   5.2   using brackets to find the first element of the list

my_list_1[[as.numeric("1")]]   5.2   using brackets to find the element listed first via the character "1", but using as.numeric would then re-direct you to the numeric item 1 on the list

my_list_1[["1"]]   NULL   using brackets to find the element listed first via the character "1", but it can't find this item because the character "1" doesn't exist in this list.

my_list_1[["one"]]   "five point two"   using brackets to find the element listed first via the list name "one"

my_list_1$one   "five point two"   using the dollar sign to create a shortcut to find the name 'one' on the list

my_list_1$"one"   "five point two" using the dollar sign to create a shortcut to find anything with the character value "one" on the list

my_list_1$1   Error: unexpected numeric constant in "my_list_1$1" using the dollar sign to create a shortcut to find the numeric constant 1, but that doesn't exist

my_list_1$"1"   NULL   using the dollar sign to create a shortcut to find anything with the character value "1" on the list, but it doesn't exist.

Q13 (2 pts.): Identify which lines produced the string output "five point two" and explain why.

```
my_list_1[["one"]]
my_list_1$one
my_list_1$"one"
```

These three lines produced the output “five point two” because the naming convention of the list called the second item of the list “one”, which was equal to “five point two”. Based on the way R read the subsetting code, it knew to look for anything named “one” in various ways, giving the output “five point two” each time.

Q14 (1 pt.): Identify which lines produced NULL output and explain why.

```
my_list_1[["1"]]
my_list_1$"1"
```

These two lines of code produced the output NULL because there are no items in the list called “1”, and so there’s nothing for R to find, no matter what subsetting notation you use.