ECO 634- Analysis of Environmental Data Lab

Lab Exercise 1- DataCamp

Expressions and Strings

Question 1

Run the following two lines of code in the console. Explain why the outputs of the two lines are different.

```
c(1, 2, 3)
```

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

```
"c(1, 2, 3)"
```

```
## [1] "c(1, 2, 3)"
```

The two lines of code produce different outputs because one (line 12) is a command to R that elicits a response, while the other (line 13) simply is a quoted text that R does not see as a command. The command in line 12 prompts R to create a vector containing the three elements (1, 2, and 3) in the stated order. This is seen in the console's output as it prints the values of the elements within the vector (1, 2, 3). Line 13 is not seen as a command by R, but instead appears to R as any sort of quoted text. This explains why the console's output is the same as the script's input- R is simply restating what you inputted as it would with any other quoted text. For example, you could simply type your name in quotes on the script and run the line and it'll elicite a similar response- R will just print out the quoted text similar to how it reacted to line 13.

```
"Mercy"
```

```
## [1] "Mercy"
```

```
"c(1,2,3)"
```

```
## [1] "c(1,2,3)"
```

Variables

```
c_1 = c(1, 2, 3)
c_2 = "c(1, 2, 3)"
```

Question 2

Is c_1 a variable, or a function? How do you know?

c_1 is a function because it is composed of more than one element stored in a vector. The vector contains multiple elements because line 28 acts as a command for R, asking it to put the 3 values into a vector. Since c_1 therefore contains multiple elements, it is not a variable (which may only contain a single element).

Question 3

Is c_2 a variable, or a function? How do you know?

c_2 is a variable as it has been assigned the value of a single element. Since c(1,2,3) is in quotation marks, R sees this as a single element of quoted text. Since c_2 has only been assigned a single element, we can state that it is a variable.

Question 4

If c_1 and c_2 have different values, why?

```
c_1
```

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

```
c_2
```

```
## [1] "c(1, 2, 3)"
```

c_1 and c_2 do indeed have different values. This is because R recognizes the command “c()” in the c_1 line and promptly follows the command to create a vector with the values of 1,2,and 3. When we print c_1, R returns the output which is a vector of the 3 elements. Since R does not recognize any commands in the c_2 line, printing c_2 returns only the quoted text that we assigned to the variable c_2 rather than a vector of the specified numbers.

Matrices 1

Create a numeric vector of length 3 called my_vec. It should contain the integers from 1 to 3. Build a matrix using the following code:

```
my_vec <- c(1, 2, 3)
mat_1 = matrix(my_vec)
mat_1
```

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

Question 5

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

The matrix is 3 rows by 1 column

Question 6

Write R code to retrieve the element of mat_1 that has a value of 3.

```
mat_1[3,1]
```

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

Matrices 2

Create a matrix mat_2 that has two rows and three columns using my_vec. Do not use the c() or rep() functions.

Create a matrix mat_3 that has three rows and two columns using my_vec. Do not use the c() or rep() functions.

```
mat_2 <- matrix(my_vec, nrow = 2, ncol = 3)
mat_3 <- matrix(my_vec, nrow = 3, ncol = 2)

mat_2
```

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

```
mat_3
```

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

Question 7

Paste the code you used to create mat_2.(See above)

Question 8

Paste the code you used to create mat_3.(See above)

Question 9

Did R use rows or columns to recycle the values in my_vec?

R used columns to populate the values in my_vec.

Question 10

Create a matrix, mat_4, with a number of elements that is not a multiple of 3 and paste the code into the editor.

```
mat_4 <- matrix(my_vec, nrow = 4, ncol = 5)
```

```
## Warning in matrix(my_vec, nrow = 4, ncol = 5): data length [3] is not a sub-
## multiple or multiple of the number of rows [4]
```

```
mat_4
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    2    3    1    2
## [2,]    2    3    1    2    3
## [3,]    3    1    2    3    1
## [4,]    1    2    3    1    2
```

Question 11

How did R handle the recycling of values of my_vec in mat_4?

R used columns to populate the values from my_vec, but threw an error message because the “data length is not a sub-multiple or multiple of the number of rows”. This is due to R being unable to evenly fill all elements of the matrix evenly (the last element was a 2 instead of a 3).

List Subsetting

Create a list, named my_list_1 with following three elements:

first element is numeric: 5.2 second element is a string “five point two” third element is a vector of all integers from 0 to 5.

Do recall how to do this from the DataCamp course? Yes:

```
elem_1 <- 5.2
elem_2 <- "five point two"
elem_3 <- c(0:5)

elem_1
```

```
## [1] 5.2
```

```
elem_2
```

```
## [1] "five point two"
```

```
elem_3
```

```
## [1] 0 1 2 3 4 5
```

```
my_list_1 <- list(elem_1, elem_2, elem_3)

my_list_1
```

```
## [[1]]
## [1] 5.2
##
## [[2]]
## [1] "five point two"
##
## [[3]]
## [1] 0 1 2 3 4 5
```

Name the elements in my_list_1:

“two” “one” “three”

```
names(my_list_1) <- c("two", "one", "three")

my_list_1
```

```
## $two
## [1] 5.2
##
## $one
## [1] "five point two"
##
## $three
## [1] 0 1 2 3 4 5
```

Run the following lines of code.

```
my_list_1[[1]]
```

```
## [1] 5.2
```

```
my_list_1[[as.numeric("1")]]
```

```
## [1] 5.2
```

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

```
## NULL
```

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

```
## [1] "five point two"
```

```
my_list_1$one
```

```
## [1] "five point two"
```

```
my_list_1$"one"
```

```
## [1] "five point two"
```

```
#my_list_1$1
my_list_1$"1"
```

```
## NULL
```

Question 12

For each of the 8 lines, answer the following: A. Did the line return a 1: value, 2: error, or 3: NULL? B. If it did not return an error, what type of subsetting operation was used? C. If it did not return an error, explain how R chose which element to retrieve.

Line 1:

A. value
B. double square brackets were used to specify the first element of the list because it was the first element [1]
C. R chose the entire contents of the first element in the list, which was specified by the double square brackets

Line 2:

A. value
B. the “as.numeric” command turned the “1” into the numeric value of 1. This allowed line 2 to act functionally the same as line 1.
C. R chose the first element because the number 1 was specified which R corresponded to the first element on the list

Line 3:

A. NULL
B. R was unable to subset because there was no element of the list named “1”
C. R was unable to choose an element

Line 4:

A. value
B. double square brackets were used to subset the element that was named “one” (which was the 2nd element in the list)
C. R chose “five point two” because it was the only value in the element named “one”

Line 5:

A. value
B. the dollar sign was used to specify the element named “one”
C. R chose “five point two” because it was the only value in the element named “one”

Line 6:

A. value
B. very similar to line 5, a dollar sign was used to specify the element that was named “one”. Quotation marks aren’t required since the name itself is a vector of another value, but they don’t interfere either.
C. R chose “five point two” because it was the only value in the element named “one”

Line 7:

A. error

Line 8:

A. NULL
B. similar to line 3, R was unable to subset because there was no element of the list named “1”
C. R was unable to choose an element

Question 13

Identify which lines produced output “five point two” and explain why.

Lines 4, 5, and 6 produced an output of “five point two” because the command called for the element named “one” to be subsetted. The element named “one” was the 2nd element in the list, which contained the value “five point two”.

Question 14

Identify which lines produced NULL output and explain why.

Lines 3 and 8 produced an NULL output because the command called for the element named “1” to be subsetted. Since none of the elements were named “1” (they were named “one”, “two”, and “three”), R did not have any elements to print.