Lab 1 R Fundamentals 1 Jackie Stephens

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

The outputs of the two lines are different because of the presence of the quotation marks. With the quotation marks around the line c(1,2,3) R sees that string of text as characters and not as something to be interpreted. The combine function c() does not get utilized.

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

c_1 is a variable because it is the output of the combine function c() assigned to it. I know this because of the paratheses that are associated with functions.

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

C_2 is also a variable because it is still an output of what is to the right of the equals symbol, no matter if there is quotation marks. The quotation marks turn the following into a string of text aka the c() no longer is seen as a computable function.

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

They have different values again because of the presence of the quotation marks creating a string of text instead of a computable numeric value.

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

The dimensions of the matrix are 3 rows and 2 columns (3x2).

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

Mat_1[3,1] locates row 3 column 1 which has the value 3.

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?

Depending on which longer (rows or columns), the values of my_vec were recycled.

(Answering this question was a little confusing because when searching recycling vectors in R online, the scenarios always involved two vectors being combined and the shorter one being recycled to match the longer one)

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 = 4, ncol = 5)

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

R gave a warning message saying "data length [6] is not a sub-multiple or multiple of the number of rows [4]". It recycled/distributed the values of my_vec top to bottom the rows and moving across the columns.

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    3    1    5
[2,]    2    6    4    2    6
[3,]    3    1    5    3    1
[4,]    4    2    6    4    2
```

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.

```
1 > my_list_1[[1]]

[1] 5.2
```

a. Value
b. Double brackets were used to subset
c. It subsetted by calling out the first (1) value of the list

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

[1] 5.2
```

d. Value
e. Double brackets were used to subset
f. It subsetted by calling out the first (1) value of the list and listing it as a number

```
3 > my_list_1[["1"]]

NULL
```

g. Null
h. Double brackets were used to attempt subsetting
i. There is no string of text name 1 in the list so it is undefined in the list.

```
4 > my_list_1[["one"]]

[1] "five point two"
```

j. Value
k. Double brackets were used to subset
l. It subsetted by finding the value in the list that was named "one" which corresponds with the second value "five point two"

```
5 > my_list_1$one
```

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

m. Value
n. The $ was used to subset
o. It was subsetted by calling out the name one in the list which corresponds with the second value "five point two"

```
6 > my_list_1$"one"
```

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

p. Value
q. The $ was used to subset
r. it was subsetted by calling out the name one in the list which corresponds with the second value "five point two"

```
7 > my_list_1$1
```

```
Error: unexpected numeric constant in "my_list_1$1"
```

s. Error
t. the $ was used to subset
u. NA

```
8> my_list_1$"1"
```

```
NULL
```

v. Null
w. The $ was used to subset
x. There is no string of text name 1 in the list

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

Lines 4, 5, and 6 produced the string output "five point two" because they all call out the second value in the list named "one" which corresponds with the string of text "five point two".

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

Lines 3 and 8 produce the NULL output because the string "1" is undefined or does not exist

Code:

```
c(1, 2, 3)

"c(1, 2, 3)"


c_1 = c(1, 2, 3)

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


my_vec = c(1, 2, 3, 4, 5, 6)
```

```
mat_1 = matrix(my_vec, nrow = 3)

mat_1[3,1]

mat_2 = matrix(my_vec, nrow = 2, ncol = 3)

mat_3 = matrix(my_vec, nrow = 3, ncol = 2)

mat_4 = matrix(my_vec, nrow = 4, ncol = 5)
mat_4

vec1 = c(0,1,2,3,4,5)

my_list_1 = list(5.2, "five point two", vec1)

my_list_1

names(my_list_1) = c("two", "one", "three")

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