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

Q1

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

The first line is a function and the second is a string of characters.

Q2

Is c_1 a variable, or a function?

Function

Q3

Is c_2 a variable, or a function?

Variable

Q4

If c_1 and c_2 have different values, why?

Yes, because the first one is the function c() that creates a vector, in this case as a string of numbers, and the other has " " which R reads as a string of characters.

Q5

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

3 rows 1 column

Q6

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

mat_1[3,]

Q7

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

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

Q8

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

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

Q9

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

columns

Q10

(1pt.): Create a matrix, mat_4, with several elements that is not a multiple of 3 and paste the code into the editor.

my_vec2<-c(1,2,3,4)

mat_4 = matrix(my_vec2)

Q11

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

It gave an error message: *the length of the data [3] is not a submultiple or multiple of the number of rows [4] in the array*

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.** 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.

- `my_list_1[[1]]`

```
A. value
B. [[]]
C. selected a component is using the numbered position of that
component.
```

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

```
A. value
```

B. [[]]

C. selected a component using the numbered position of that component.

- `my_list_1[["1"]]`

A. NULL

B. [[]],

C. selected a component using the numbered position of that component, but this is not in the format of the numerical position, the one looks like a string character.

- `my_list_1[["one"]]`

A. value

B. [[]]

C. refers to the name of the component

- `my_list_1$one`

A. value

B. $

C. refers to the name of the component

- `my_list_1$"one"`

A. value

B. $

C. refer to the names of the components

- `my_list_1$1`

A. **This gave an ERROR -** `Error: unexpected numeric constant in "my_list_1$1"`

- `my_list_1$"1"`

A. **NULL**

B. **$**

C. **there is no component named "1".**

Q13

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

- `my_list_1[["one"]]`

This gives the value because it is referring to the name "one"

which we assigned to the value "five point two" pulled by telling R that within the list there is something named "one" and want the value of that list component to be printed

- `my_list_1$one`

This gives the value because it is referring to the name "one"

which we assigned to the value "five point two". The same as above, but this time the '$' is directly telling R to pull the 'one' component from the list.

- `my_list_1$"one"`

This gives the value because it is referring to the name "one" which we assigned to the value "five point two". The same as above, but as we can see the

'$' reads one the same as "one". The quotations do not matter in this case, but they do matter in the [[]], which would have pulled the component in the first position of the line as opposed to the element named "one"

Q14

(2 pts): Identify which lines produced `NULL` output and explain why.

- `my_list_1[["1"]]`

NULL, [[]], selected a component is using the numbered position of that component, but this is not in the format of the numerical position, the one looks like a string character. And there is not a component named "1" in the list.

- `my_list_1$"1"`

There is no component named "1". Just like above. The $ directly pulls from the name of the components in list and does not consider positions like the [[]] do.

CODE used in R:

#LAB 1

#QUESTIONS 1-14

c(1, 2, 3)

"c(1, 2, 3)"

my_vec<-c(1,2,3)

mat_1 = matrix(my_vec)

dim(mat_1)

mat_1

mat_1[3,]

```
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=2)


"two"<-5.2

"one"<-"five point two"

"three"<- c(1:5)

my_list_1<-list("two"=5.2,"one"="five point two","three"=c(1:5))


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"
```