

**Bonnie Turek**

**Eco 634 – Lab 1 Submission**

**9/1/2021**

Q1. `c(1, 2, 3)`

`"c(1, 2, 3)"` These two lines have different results in R because the first line is a function. The combine function specifically which is used to create vectors. The first line creates a numeric vector with 1, 2, 3. Versus the second line which is in quotation marks which means it is recognized as characters or a string literal. The output is just the same thing, `"c(1,2,3)"`

Q2. `c_1` is a function. (the combine function)

Q3. `c_2` is a variable. It is a string literal. It is not a function because it is in quotes.

Q4. `c_1` and `c_2` have different values because they are typed slightly different and are therefore different data types. `C_1` is a combine function which creates a vector of numeric data type, whereas `C_2` is of string literal data type due to the quotation marks around it.

Q5. The matrix has one row and 3 columns, therefore it is a 1x3 matrix.

Q6. R code to retrieve value of 3 from matrix.

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

Q7. R code to create 2x3 matrix from `my_vec`:

```
> mat_2 <- matrix(my_vec, nrow = 2, ncol = 3, byrow = TRUE)
> mat_2
      [,1] [,2] [,3]
[1,]  1   2   3
[2,]  1   2   3
```

Q8. R code to create 3x2 matrix from my\_vec:

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

```
> mat_3
```

```
  [,1] [,2]
```

```
[1,]  1  2
```

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

```
[3,]  2  3
```

Q9. R used rows to recycle the values of my\_vec for the matrices. You can see this in the output of the matrices above. The values 1,2,3 are cycle from left to right in the rows.

Q10. R code for matrix 4 with a number of elements that is not a multiple of 3:

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

```
mat_4
```

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

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

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

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

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

Q11. R handled the recycling of values in the same way as matrix 1,2, and 3. The 1,2,3 values repeat as you move across the row and do not restart with the next new row.

Q12.

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

- A. VALUE (5.2)
- B. A single element of the list is selected using double square brackets
- C. R chose to extract the first element of the list which is the numeric value of 5.2

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

- A. VALUE (5.2)
- B. Double square brackets are used again to extract the first element of the list. As numeric function is also used to take the first element and make sure it is turned into a numeric, non-string value.
- C. R chose to extract the first element of the list again which is the numeric value of 5.2. The "1" is in quotes in case the first element is a

string literal but needs to be made numeric. This was not the case for 5.2

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

- A. NULL
- B. Double square brackets are used here to select the first element of the list that is a string literal.
- C. Since the first element of the list which was indexed is not a string literal, (rather it is a numeric 5.2), and the as.numeric function isn't used, the return value is NULL.

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

- A. VALUE ("five point two")
- B. The double brackets are used to select the element in the list which was named "one".
- C. R chose to extract "five point two" because per the lab instructions, the second element in the list was to be named "one"

```
my_list_1$one
```

- A. VALUE ("five point two")
- B. The dollar sign \$ is used to subset and select the second item in the list by its name, which we called "one". I assume Rstudio typically interprets names as string literals. In this case we did not put quotes around one to specify, but it is a string anyway.
- C. R chose to retrieve the second element in my list because we referenced the name of that element using the \$ symbol

```
my_list_1$"one"
```

- A. VALUE ("five point two")
- B. The dollar sign is used again to subset the second element of the list by the name we called it ("one"). Except here we put quotes around "one" to specify that it is a string literal.
- C. R chose to retrieve the second element in my list because we referenced the name of that element using the \$ symbol and its name "one."

```
my_list_1$1
```

- A. ERROR (unexpected numeric constant in "my\_list\_1\$1")
- B. It returned an error
- C. It returned an error

```
my_list_1$"1"
```

- A. NULL
- B. The dollar sign is used again to subset here by calling a list component name in quotes ("1"), however we have no list component that was named "1" so nothing can be returned.
- C. R would choose to retrieve the component in the list that was named "1" however we do not have that here so NULL is returned.

Q13. Which lines produced output "five point two" and explain why:

Lines 4, 5, and 6 produced output "five point two." This is because those lines either used the double brackets or the dollar sign to subset the second element from my list by its name, "one".

Q14. Which lines produced NULL output and explain why:

Lines 3 and 8 produced output NULL. This is because in line 3, the index we used to extract the first element in our list was in quotation marks. When indexing using double square brackets, the index number should not be in quotes. This should be numeric. And if R interpreted this line as to subset by the name of the elements in the list, we also do not have an element named "1". Similarly for line 8, the dollar sign is used to subset here by calling a list component name in quotes ("1"), however we have no list component that was named "1" so nothing can be returned.