Worksheet-1 in R

Name: Niel Andrew Callanga

YEAR/SECTION: BSIT 2A

Worksheet for R Programming

Instructions

- Use RStudio or the RStudio Cloud accomplish this worksheet. + Save the R script as *RWorksheet_lastname#1.R*.
- Create your own *GitHub repository* and push the R script as well as this pdf worksheetto your own repo.

Accomplish this worksheet by answering the questions being asked and writing the codemanually.

Using functions:

seq(), assign(), min(), max(), c(), sort(), sum(), filter()

1. Set up a vector named age, consisting of 34, 28, 22, 36, 27, 18, 52, 39, 42, 29,35, 31, 27, 22, 37, 34, 19, 20, 57, 49, 50, 37, 46, 25, 17, 37, 42, 53, 41, 51, 35, 24, 33, 41.

age <- c(34, 28, 22, 36, 27, 18, 52, 39, 42, 29,35, 31, 27, 22, 37, 34, 19, 20, 57, 49, 50, 37, 46, 25, 17, 37, 42, 53, 41,51, 35, 24, 33, 41)

a. How many data points?
34 data points

b. Write the R code and its output.
Code = age

Output =
[1] 34 28 22 36 27 18 52 39 42 29 35 31 27 22 37 34 19
[18] 20 57 49 50 37 46 25 17 37 42 53 41 51 35 24 33 41

2. Find the reciprocal of the values for age.Write the R code and its output.

Code =
recip_age <- 1/age

recip_age

Output =

[1] 0.02941176 0.03571429 0.04545455 0.02777778

[5] 0.03703704 0.05555556 0.01923077 0.02564103

[9] 0.02380952 0.03448276 0.02857143 0.03225806

[13] 0.03703704 0.04545455 0.02702703 0.02941176

[17] 0.05263158 0.05000000 0.01754386 0.02040816

[21] 0.02000000 0.02702703 0.02173913 0.04000000

[25] 0.05882353 0.02702703 0.02380952 0.01886792

[29] 0.02439024 0.01960784 0.02857143 0.04166667

[33] 0.03030303 0.02439024

3. Assign also new_age <- c(age,0,age).What happen to the new_age?

It will display random numbers, and it has 0 at the center.

4. Sort the values for age.

Write the R code and its output.

Code = sort(age)

Output =
[1] 17 18 19 20 22 22 24 25 27 27 28 29 31 33 34 34 35

[18] 35 36 37 37 37 39 41 41 42 42 46 49 50 51 52 53 57

5. Find the minimum and maximum value for age.Write the R code and its output.

Code = max(age)

Output = 57

Code = min(age)

Output = 17

6. Set up a vector named data, consisting of 2.4, 2.8, 2.1, 2.5, 2.4, 2.2, 2.5, 2.3,2.5, 2.3, 2.4, and 2.7.
data <- c(2.4, 2.8, 2.1, 2.5, 2.4, 2.2, 2.5, 2.3,2.5,2.3, 2.4,2.7)

a. How many data points?
12 data points

b. Write the R code and its output.
Code = data

Output =
[1] 2.4 2.8 2.1 2.5 2.4 2.2 2.5 2.3 2.5 2.3 2.4 2.7

7. Generates a new vector for data where you double every value of the data. | What happento the data?

To generate a new vector for data, I use (2* data) and this is the output:

4.8 5.6 4.2 5.0 4.8 4.4 5.0 4.6 5.0 4.6 4.8 5.4

It doubles the value of the given data.

8. Generate a sequence for the following scenario:

8.1 Integers from 1 to 100.
seq(1:100)

8.2 Numbers from 20 to 60
seq(20:60)

*8.3 Mean of numbers from 20 to 60

mean(20:60)

*8.4 Sum of numbers from 51 to 91

sum(51:91)

*8.5 Integers from 1 to 1,000
seq(1:1000)

a. How many data points from 8.1 to 8.4?
223 data points

b. Write the R code and its output from 8.1 to 8.4.

8.1
Code = seq(1:100)
Output is number sequence from 1-100.

8.2
Code=seq(20:60)
Output is number sequence from 20-60.

8.3
Code = mean(20:60)
Output is 40

8.4
Code = sum(51:91)
Output is 2911

c. For 8.5 find only maximum data points until 10.

Code = seq(1:10)

Output = 10

9. *Print a vector with the integers between 1 and 100 that are not divisible by 3, 5 and 7 using filter option.

filter(function(i) { all(i %% c(3,5,7) != 0) }, seq(100))

Write the R code and its output.

Code = Filter(function(i) { all(i %% c(3,5,7) != 0) }, seq(100))

Output=

[1] 1 2 4 8 11 13 16 17 19 22 23 26 29 31 32 34 37 38 41 43 44 46 47 52 53 58 59 61 62 64

[31] 67 68 71 73 74 76 79 82 83 86 88 89 92 94 97

10. Generate a sequence backwards of the integers from 1 to 100.Write the R code and its output.

Code = seq(from = 100, to = 1)

Output =

[1] 100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79

[23] 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57

[45] 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35

[67] 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13

[89] 12 11 10 9 8 7 6 5 4 3 2 1

11. List all the natural numbers below 25 that are multiples of 3 or

5. Find the sum of these multiples.

(1:25)
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Code = sum((1:25)[((1:25)%%3 == 0) | ((1:25)%%5 == 0)])

Output = 168

a. How many data points from 10 to 11?
125 data points

b. Write the R code and its output from 10 and 11.
10.
Code = seq(from = 100, to = 1)
Output is the numbers from 100 to 1

11.
Code = sum((1:25)[((1:25)%%3 == 0) | ((1:25)%%5 == 0)])
Output = 168

12. Statements can be grouped together using braces '{' and '}'. A group of statements is sometimes called a **block**. Single statements are evaluated when a new line is typed atthe end of the syntactically complete statement. Blocks are not evaluated until a new line is entered after the closing brace.

Enter this statement:
{ x <- 0+ x + 5 + }
Describe the output.

The code says "Error: unexpected '}' in "{ x <- 0+ x + 5 + }"

because object "x" was not assigned and also something missing after the operator "+".

13. *Set up a vector named score, consisting of 72, 86, 92, 63, 88, 89, 91, 92, 75,
75 and 77. To access individual elements of an atomic vector, one generally uses the x[i]
construction.

Find x[2] and x[3]. Write the R code and its output.

score <- c(72, 86, 92, 63, 88, 89, 91, 92, 75,75,77)

Code = score

Output = [1] 72 86 92 63 88 89 91 92 75 75 77
score[2] = 86
score[3] = 92

14. *Create a vector a = c(1,2,NA,4,NA,6,7).

a. Change the NA to 999 using the codes print(a,na.print="-999").
a <- c(1,2,999,4,999,6,7)

b. Write the R code and its output. Describe the output.
Code = a <- c(1,2,999,4,999,6,7)
print(a,na.print="-999")

Output = [1] 1 2 -999 4 -999 6 7
The NA changes its value to -999.

15. A special type of function calls can appear on the left hand side of the assignmentoperator as in > class(x) <- "foo".

Follow the codes below:

```
name = readline(prompt="Input your name: ")
age = readline(prompt="Input your age: ")
print(paste("My name is",name, "and I am",age ,"years old."))
print(R.version.string)
```

What is the output of the above code?

This code above is asking the user for data. Each piece of data the user inputs into the console will be recorded or saved in the environment pane. The final output will be displayed on the console pane and the output will be this [1] "My name is Niel Andrew Callanga and I am 20 years old."