Worksheet-1 in R

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 worksheet to your own repo.

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

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.

   a. How many data points?
   Answer: 12 data points

   b. Write the R code and its output.
   Answer:
   R code: 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)
   df <- data.frame(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))
   Output: [1] 34 28 22 36 27 18 52 39 42 29 35 31 27 22 37 34 19 20 57 49 50
   #[22] 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.

Answer: R code : reciprocal <- function(age) vec <- 1/age

rage <- reciprocal(age)

rage

Output:

```
[1] 0.02941176 0.03571429 0.04545455
 [4] 0.02777778 0.03703704 0.05555556
 [7] 0.01923077 0.02564103 0.02380952
[10] 0.03448276 0.02857143 0.03225806
[13] 0.03703704 0.04545455 0.02702703
[16] 0.02941176 0.05263158 0.05000000
[19] 0.01754386 0.02040816 0.02000000
[22] 0.02702703 0.02173913 0.04000000
[25] 0.05882353 0.02702703 0.02380952
[28] 0.01886792 0.02439024 0.01960784
[31] 0.02857143 0.04166667 0.03030303
[34] 0.02439024
```

3. Assign also new_age <- c(age, 0, age).
What happen to the new_age?
Answer: It displays the age and the 0 is in the center.

4. Sort the values for age.

   Write the R code and its output.

   Answer: R code: sort(age)

   Output:

```
[1] 17 18 19 20 22 22 24 25 27 27 28 29
[13] 31 33 34 34 35 35 36 37 37 37 39 41
[25] 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.

   Answer: R code:

```
max(age)
min(age)
```

   Output:

```
max(age)
 [1] 57
min(age)
 [1] 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.

   a. How many data points?

      Answer: 12 data points

   b. Write the R code and its output.

      R code:

```
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)
```

      Output:

```
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?

   Answer: The data is doubled when I use 2*Data

```
[1] 4.8 5.6 4.2 5.0 4.8 4.4 5.0 4.6 5.0
[10] 4.6 4.8 5.4
```

8. Generate a sequence for the following scenario:

   8.1 Integers from 1 to 100.

   Answer: seq(1:100)

   8.2 Numbers from 20 to 60

   Answer: seq(20,60)

   length(seq(20,60))

   *8.3 Mean of numbers from 20 to 60

   Answer: print(mean(20:60))

   *8.4 Sum of numbers from 51 to 91

   Answer: print(sum(51:91))

   *8.5 Integers from 1 to 1,000

   Answer: seq(1:1000)

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

      Answer: 43 data points are in 8.1 to 8.4

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

      Answer: R code:

      seq(1:100)

      seq(20,60)

      print(mean(20,60))

      print(sum(51:91))

      Output:

      number sequence from 1-100.

      numbers 1 -41.

      output is 40

      output is 2911

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

      Answer: m <- seq(1:10)

      max(m)

      Output: [1] 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.

Answer: R 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
[12] 26 29 31 32 34 37 38 41 43 44 46
[23] 47 52 53 58 59 61 62 64 67 68 71
[34] 73 74 76 79 82 83 86 88 89 92 94
[45] 97
```

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

Answer: R code:

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

Output:

```
  [1] 100  99  98  97  96  95  94  93
  [9]  92  91  90  89  88  87  86  85
 [17]  84  83  82  81  80  79  78  77
 [25]  76  75  74  73  72  71  70  69
 [33]  68  67  66  65  64  63  62  61
 [41]  60  59  58  57  56  55  54  53
 [49]  52  51  50  49  48  47  46  45
 [57]  44  43  42  41  40  39  38  37
 [65]  36  35  34  33  32  31  30  29
 [73]  28  27  26  25  24  23  22  21
 [81]  20  19  18  17  16  15  14  13
 [89]  12  11  10   9   8   7   6   5
 [97]   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.
    a. How many data points from 10 to 11?
       Answer: There are 101 data points from 10 to 11.

    b. Write the R code and its output from 10 and 11.
       Answer: R code:
       seq(from = 100, to = 1)
       sum((1:25)[((1:25)%%3 == 0) | ((1:25)%%5 == 0)])
       Output:
       numbers from 100 to 1
       output is 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 at the 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.

    Answer: The output is error because of unexpected undeclared variable of '}' in { x <- 0+ x + 5 + }

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.
    Answer: score <- c(72, 86, 92, 63, 88, 89, 91, 92, 75,75, 77)
    x[2] = 86
    x[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").
    b. Write the R code and its output. Describe the output.

Answer: R code:

```
a <- c(1,2,NA,4,NA,6,7)
print(a,na.print="-999")
```

Output:

```
[1]    1    2 -999    4 -999    6    7
```

The -999 was placed between 2 and 4, and in between of 4 and 6

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?

Answer: Output:

```
[1] "My name is Quennie and I am 19 years old."
```