Worksheet-2 in R

Worksheet for R Programming Instructions:

- Use RStudio or the RStudio Cloud accomplish this worksheet. + Save the R script as RWorksheet_lastname#2.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 Vectors

1. Create a vector using : operator a. Sequence from -5 to 5. Write the R code and its output. Describe its output.

Answer: **s <- 5:5**

Output: [1] -5 -4 -3 -2 -1 0 1 2 3 4 5

When we type s <- 5:5, the output will be numbers from -5 to 5.

b. x <- 1:7. What will be the value of x?

Answer:

> x

[1] 1 2 3 4 5 6 7

2. Create a vector using seq() function a. seq(1, 3, by=0.2) # specify step size Write the R code and its output. Describe the output.

Answer:

seq(1, 3, by = 0.2)

Output: [1] 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8 3.0

The output shows the numbers from 1 to 3 adding 0.2 to the previous number until it reaches 3.0.

3. A factory has a census of its workers. There are 50 workers in total. The following list shows their ages: 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, 43, 53, 41, 51, 35, 24,33, 41, 53, 40, 18, 44, 38, 41, 48, 27, 39, 19, 30, 61, 54, 58, 26, 18.

a. Access 3rd element, what is the value?

Answer: **The value of 3rd element is 22.**

b. Access 2nd and 4th element, what are the values?

Answer: **The value of 2nd and 4th elements are 28 and 36.**

c. Access all but the 1st element is not included. Write the R code and its output.

Answer:

```
> workers[2:49]
[1] 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 43 53 41 51 35 24 33 41 53
[35] 40 18 44 38 41 48 27 39 19 30 61 54 58 26
```

4. *Create a vector x <- c("first"=3, "second"=0, "third"=9). Then named the vector, names(x).

a. Print the results. Then access x[c("first", "third")]. Describe the output.

Answer:

```
x <- c("first"=3, "second"=3, "third"=9) names(x)
output: [1] "first"  "second" "third"
```

the output are “first”, “second”, “third” as we command to display the names stored in variable x.

b. Write the code and its output.

Answer:

```
x <- c("first"=3, "second"=3, "third"=9) names(x)
output: [1] "first"  "second" "third"
```

5. Create a sequence x from -3:2.

a. Modify 2nd element and change it to 0; x[2] <- 0 x Describe the output.

Answer**: The output showed 3 and 2**

b. Write the code and its output.

Answer: **x <- 3:2**

X

Output:

> x

[1] 3 2

6. a. Create a data frame for month, price per liter (php) and purchase-quantity (liter).

Write the codes. **month <- c("Jan", "Feb", "Mar", "Apr", "May", "June")**

price_per_liter <- c(52.50, 57.25, 60.00, 65.00, 74.25, 54.00) purchase_quantity <- c(25, 30, 40, 50, 10, 45)

frame <- data.frame(month, price_per_liter, purchase_quantity) frame

c. What is the average fuel expenditure of Mr. Cruz from Jan to June? Note: Use weighted.mean(liter, purchase).

Answer: The average fuel expenditure of Mr. Cruz from Jan to June is 59.2625.

7. R has actually lots of built-in datasets. For example, the rivers data “gives the lengths (in miles) of 141 “major” rivers in North America, as compiled by the US Geological Survey”.
   a. Type “rivers” in your R console.

Create a vector data with 7 elements, containing the number of elements (length) in rivers, their sum (sum), mean (mean), median (median), variance (var) standard deviation (sd), minimum (min) and maximum (max). data <- c(length(rivers), sum(rivers), mean(rivers), median(rivers), var(rivers), sd(rivers), min(rivers), max(rivers))

a. data <- c(length(rivers), sum(rivers), mean(rivers), median(rivers), var(rivers), sd(rivers), min(rivers), max(rivers)) data

b. What are the results?

Answer:

```
141.0000  83357.0000    591.1844    425.0000 243908.4086    493.8708    135.0000 3710.0000
```

c. Write the code and its outputs.

```
data <- c(length(rivers), sum(rivers), mean(rivers), median(rivers), var(rivers),
sd(rivers), min(rivers), max(rivers)) data output:
 > data
[1]    141.0000  83357.0000    591.1844    425.0000 243908.4086    493.8708    135.0000 3710.0000
```

8. A. Create vectors according to the above table. Write the codes.

Answer:

```
power_ranking
<- c(1:25)
celebrity_name <- c("Tom Cruise", "Rolling Stones", "Oprah Winfrey", "U2", "Tiger Woods", "Steven Spielberg",
                "Howard Stern", "50 Cent", "Cast  of the sopranos", "Dan Brown", "Bruce Springsteen",
                "Donal Trump", "Muhammad Ali", "Paul McCartney", "George Lucas", "Elton John",
                "David Letterman", "Phil Mickelson", "J.K Rowling", "Bradd Pitt", "Peter Jackson",
                "Dr. Phil McGrow", "J Lenon", "Celine Dion",
                "Kobe Bryant")
pay <- c(67, 90, 225, 110, 90, 332, 302, 41, 52, 88, 55, 44, 55, 40, 233, 34, 40, 47, 75, 25, 39, 45, 32, 40, 31)
 data_ranking <- data.frame(power_ranking, celebrity_name, pay) data_ranking
```

c. Modify the power ranking and pay of J.K. Rowling. Change power ranking to 15 and pay to 90. Write the codes and its output.

Answer: power_ranking

[19] <- 15 power_ranking

pay [19] <- 90 pay

output:

```
> pay
 [1]  67  90 225 110  90 332 302  41  52  88  55  44  55  40 233  34  40  47  90  25  39  45  32
40  31
```

D . Interpret the data.

Answer:

The data shows the power ranking of each celebrity and the pay. When we made a data frame that includes the three of them, the results will be, each celebrity has corresponding rank and pay. When we modified the 19th celebrity in the ranking, it was changed into rank number 15 and got the 90 pay.