**Frias, Dazzele Mae** **BSIT2-A**

# 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:

```
  seq(-5:5)
[1]  1  2  3  4  5  6  7  8  9 10 11
```

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

Answer:

```
> x <- 1:7
> 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)
[1] 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8 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.

```
Wrkrs_age <- c(34,28,22,36,27,18,52,39,42,29,35,31,27,22,37,
               34,19,20,57,49,50,37,46,25,17,37,43,53,41,51,
               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:

```
Wrkrs_age[3]
      [1] 22
```

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

## Answer:

```
Wrkrs_age[2]
      [1] 28
Wrkrs_age[4]
      [1] 36
```

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

## R code:

```
Wrkrs_age[2:50]
```

## Output:

```
   Wrkrs_age[2:50]
 [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
[25] 37 43 53 41 51 35 24 33 41 53 40 18 44 38 41 48 27 39 19 30 61 54 58 26
[49] 18
```

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: The output are now arranged in row and column.

```
names <- c("first"=3, "second"=0, "third"=9)
 names
 first second  third
     3      0      9
```

b. Write the code and its output.
R CODE:
names [c("first", "third")]

OUTPUT:

```
    names [c("first", "third")]
          first third
            3     9
```

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

Answer:

```
 x <- c(-3:2)
 x
   [1] -3 -2 -1  0  1  2
```

a. Modify 2nd element and change it to 0;

```
x[2] <- 0
x
```

Describe the output.

ANSWER: THE SECOND ELEMENT WAS REPLACED BY THE ASSIGNED NEW VALUE WHICH IS "0"

```
[1] -3  0 -1  0  1  2
```

b. Write the code and its output.

R CODE:

```
x[2] <- 0
x
```

OUTPUT:

```
[1] -3  0 -1  0  1  2
```

6. *The following data shows the diesel fuel purchased by Mr. Cruz.

| Month | Jan | Feb | March | Apr | May | June |
|---|---|---|---|---|---|---|
| Price per liter (PhP) | 52.50 | 57.25 | 60.00 | 65.00 | 74.25 | 54.00 |
| Purchase–quantity(Liters) | 25 | 30 | 40 | 50 | 10 | 45 |

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

R CODES:

```
Month = c("Jan","Feb","March","Apr","May","June")
Php <- c(52.20,57.25,60.00,65.00,74.25,54.00)
Liters <- c(25,30,40,50,10,45)

Diesel_fuel <- data.frame (Month, Php, Liters)
Diesel_fuel
```

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

Answer:

```
weighted.mean(Php, Liters)
[1] 59.225
```

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))
```

   b. What are the results?

# Answer:

```
rivers
  [1]  735  320  325  392  524  450 1459  135  465  600  330  336  280  315
 [15]  870  906  202  329  290 1000  600  505 1450  840 1243  890  350  407
 [29]  286  280  525  720  390  250  327  230  265  850  210  630  260  230
 [43]  360  730  600  306  390  420  291  710  340  217  281  352  259  250
 [57]  470  680  570  350  300  560  900  625  332 2348 1171 3710 2315 2533
 [71]  780  280  410  460  260  255  431  350  760  618  338  981 1306  500
 [85]  696  605  250  411 1054  735  233  435  490  310  460  383  375 1270
 [99]  545  445 1885  380  300  380  377  425  276  210  800  420  350  360
[113]  538 1100 1205  314  237  610  360  540 1038  424  310  300  444  301
[127]  268  620  215  652  900  525  246  360  529  500  720  270  430  671
[141] 1770
>   #data
>     data
[1]    141.0000  83357.0000    591.1844    425.0000 243908.4086    493.8708
[7]    135.0000   3710.0000
```

   c. Write the code and its outputs.

# R code:

```
#rivers
  rivers
```

# Output:

```
[1]  735  320  325  392  524  450 1459  135  465  600  330  336  280  315
 [15]  870  906  202  329  290 1000  600  505 1450  840 1243  890  350  407
 [29]  286  280  525  720  390  250  327  230  265  850  210  630  260  230
 [43]  360  730  600  306  390  420  291  710  340  217  281  352  259  250
 [57]  470  680  570  350  300  560  900  625  332 2348 1171 3710 2315 2533
 [71]  780  280  410  460  260  255  431  350  760  618  338  981 1306  500
 [85]  696  605  250  411 1054  735  233  435  490  310  460  383  375 1270
 [99]  545  445 1885  380  300  380  377  425  276  210  800  420  350  360
[113]  538 1100 1205  314  237  610  360  540 1038  424  310  300  444  301
[127]  268  620  215  652  900  525  246  360  529  500  720  270  430  671
[141] 1770
```

# R code:

```
#data
  data
```

# output:

```
[1]    141.0000  83357.0000    591.1844    425.0000 243908.4086    493.8708
[7]    135.0000   3710.000
```

8. The table below gives the 25 most powerful celebrities and their annual pay as ranked by the editions of Forbes magazine and as listed on the Forbes.com website.

| Power Ranking | Celebrity Name | Pay | Power Ranking | Celebrity Name | Pay |
|---|---|---|---|---|---|
| 1 | Tom Cruise | 67 | 14 | Paul McCartney | 40 |
| 2 | Rolling Stones | 90 | 15 | George Lucas | 233 |
| 3 | Oprah Winfrey | 225 | 16 | Elton John | 34 |
| 4 | U2 | 110 | 17 | David Letterman | 40 |
| 5 | Tiger Woods | 90 | 18 | Phil Mickelson | 47 |
| 6 | Steven Spielberg | 332 | 19 | J.K Rowling | 75 |
| 7 | Howard Stern | 302 | 20 | Bradd Pitt | 25 |
| 8 | 50 Cent | 41 | 21 | Peter Jackson | 39 |
| 9 | Cast of the Sopranos | 52 | 22 | Dr. Phil McGraw | 45 |
| 10 | Dan Brown | 88 | 23 | Jay Lenon | 32 |
| 11 | Bruce Springsteen | 55 | 24 | Celine Dion | 40 |
| 12 | Donald Trump | 44 | 25 | Kobe Bryant | 31 |
| 13 | Muhammad Ali | 55 | | | |

Figure 1: Forbes Ranking

a. Create vectors according to the above table. Write the codes.

## R codes:

```
Power_Ranking <- c(1:25)

Celebrity_Name = c("Tom Cruise", "Rolling Stones",   "Oprah Winfrey",  "U2",  "Tiger
 Woods",  "Steven Speilberg",  "Howard Stern",  "50 Cent",  "Cast of the Sopranos",
 "Dan Brown",  "Bruce Springsteen", "Donald Trump",  "Muhammad Ali",   "Paul
 Mcartney",  "George Lucas", "Elton John",  "David Letterman",  "Phil Mickelson",  "J.K
 Rowling",  "Bradd Pitt",  "Peter Jackson",   "Dr. Phil McGraw",  "Ja 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)

Celebrities_pay <- data.frame( Power_Ranking, Celebrity_Name, Pay)

    Celebrities_pay
```

b. 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.

R CODES:

```
Celebrities_pay <- data.frame( Power_Ranking, Celebrity_Name,
                    Pay)
Celebrities_pay[Celebrities_pay == 19] <-15
Celebrities_pay[Celebrities_pay ==75] <-90
   Celebrities_pay
```

OUTPUT:

```
   Power_Ranking       Celebrity_Name Pay
1              1           Tom Cruise  67
2              2       Rolling Stones  90
3              3        Oprah Winfrey 225
4              4                   U2 110
5              5          Tiger Woods  90
6              6     Steven Speilberg 332
7              7         Howard Stern 302
8              8              50 Cent  41
9              9 Cast of the Sopranos  52
10            10            Dan Brown  88
11            11    Bruce Springsteen  55
12            12         Donald Trump  44
13            13         Muhammad Ali  55
14            14        Paul Mcartney  40
15            15         George Lucas 233
16            16           Elton John  34
17            17      David Letterman  40
18            18       Phil Mickelson  47
19            15          J.K Rowling  90
20            20           Bradd Pitt  25
21            21        Peter Jackson  39
22            22      Dr. Phil McGraw  45
23            23             Ja Lenon  32
24            24          Celine Dion  40
25            25          Kobe Bryant  31
```

c. Interpret the data.

```
Celebrities_pay <- data.frame( Power_Ranking, Celebrity_Name,
                                                          Pay)
```

```
 Celebrities_pay
    Power_Ranking       Celebrity_Name Pay
1               1           Tom Cruise  67
2               2       Rolling Stones  90
3               3        Oprah Winfrey 225
4               4                  U2 110
5               5          Tiger Woods  90
6               6     Steven Speilberg 332
7               7         Howard Stern 302
8               8              50 Cent  41
9               9 Cast of the Sopranos  52
10             10            Dan Brown  88
11             11    Bruce Springsteen  55
12             12         Donald Trump  44
13             13         Muhammad Ali  55
14             14        Paul Mcartney  40
15             15         George Lucas 233
16             16           Elton John  34
17             17      David Letterman  40
18             18       Phil Mickelson  47
19             19          J.K Rowling  75
20             20           Bradd Pitt  25
21             21        Peter Jackson  39
22             22      Dr. Phil McGraw  45
23             23             Ja Lenon  32
24             24          Celine Dion  40
25             25          Kobe Bryant  31
```