Worksheet-2 in R

Worksheet for R Programming

Instructions:

- Use RStudio or the RStudio Cloud to accomplish this worksheet.
- Save the R script as RWorksheet_lastname #2.R.
- Commit and push the R script and your Rmarkdown file in html to your own repo. Do not forget to comment your Git 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.
[1] -5 -4 -3 -2 -1 0 1 2 3 4 5
Its output is a sequence of integers starting from -5 and ending at 5, wherein it increases 1.

b. x <- 1:7. What will be the value of 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 script and its output. Describe the output.
seq_vector <- seq(1, 3, by=0.2)
seq_vector
[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 a sequence of numbers starting from 1 to 3, with a step size of 0.2, resulting in a series of evenly spaced decimal values.

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?

[1] 22

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

[1] 28 36

c. Access all but the 4th and 12th element is not included. Write the R script and its output.

```
all_but_fourth_twelfth <- ages[-c(4, 12)]
all_but_fourth_twelfth
[1] 34 28 22 27 18 52 39 42 29 35 27 22 37 34 19 20 57 49 50 37 46 25 17 37 43 53
[27] 41 51 35 24 33 41 53 40 18 44 38 41 48 27 39 19 30 61 54 58 26 18
```

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

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

first second  third
    3      0      9
```

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

```
first third
   3     9
```

b.Write the code and its output.

```
> selected_elements <- x[c("first", "third")]
> selected_elements
first third
   3       9
```

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

```
x <- -3:2
```

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

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

Describe the output.

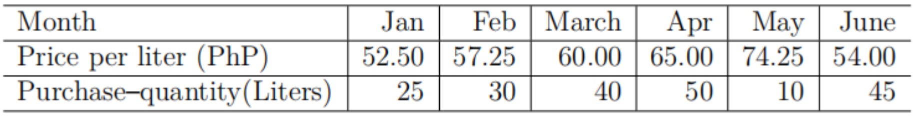
```
[1] -3  0 -1  0  1  2
```

The output shows the updated vector where the original 2nd element (-2) has been change by 0. The resulting vector now includes -3, 0, -1, 0, 1, and 2, demonstrating how the specific element was changed while the other elements remain unchanged.

b.Write the code and its output.

```
x <- -3:2
x[2] <- 0x
[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 R scripts and its output.

```
diesel_data <- data.frame(Month = month, Price_Per_Liter_Php =
price_per_liter, Purchase_Quantity_Liters = purchase_quantity)
diesel_data
Month Price_Per_Liter_Php Purchase_Quantity_Liters
1   Jan             52.50                    25
2   Feb             57.25                    30
3 March             60.00                    40
4   Apr             65.00                    50
5   May             74.25                    10
6  June             54.00                    45
```

b. What is the average fuel expenditure of Mr. Cruz from Jan to June? Note: Use 'weighted.mean (liter, purchase)'. Write the R scripts and its output.

```
average_expenditure <- weighted.mean(price_per_liter,purchase_quantity )
average_expenditure
[1] 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))
```

b.What are the results?

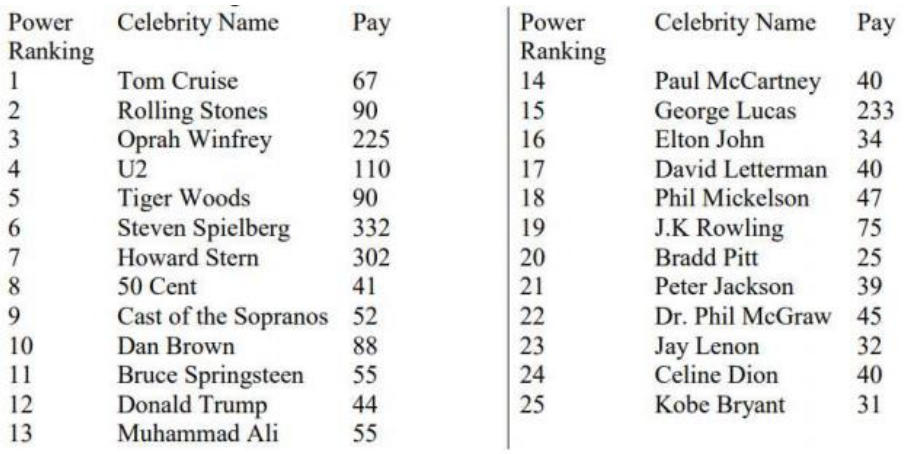
```
[1]  735  320  325  392  524  450 1459  135  465  600  330  336  280  315  870
 [16]  906  202  329  290 1000  600  505 1450  840 1243  890  350  407  286  280
 [31]  525  720  390  250  327  230  265  850  210  630  260  230  360  730  600
 [46]  306  390  420  291  710  340  217  281  352  259  250  470  680  570  350
 [61]  300  560  900  625  332 2348 1171 3710 2315 2533  780  280  410  460  260
 [76]  255  431  350  760  618  338  981 1306  500  696  605  250  411 1054  735
 [91]  233  435  490  310  460  383  375 1270  545  445 1885  380  300  380  377
[106]  425  276  210  800  420  350  360  538 1100 1205  314  237  610  360  540
[121] 1038  424  310  300  444  301  268  620  215  652  900  525  246  360  529
```

```
[136]  500  720  270  430  671 1770

  length        sum       mean     median    variance        sd        min 
   141.0000  83357.0000    591.1844    425.0000 243908.4086    493.8708    135.0000 
        max 
  3710.0
```

c.Write the R scripts and its outputs.

```
print(rivers)
  [1]  735  320  325  392  524  450 1459  135  465  600  330  336  280  315  870
 [16]  906  202  329  290 1000  600  505 1450  840 1243  890  350  407  286  280
 [31]  525  720  390  250  327  230  265  850  210  630  260  230  360  730  600
 [46]  306  390  420  291  710  340  217  281  352  259  250  470  680  570  350
 [61]  300  560  900  625  332 2348 1171 3710 2315 2533  780  280  410  460  260
 [76]  255  431  350  760  618  338  981 1306  500  696  605  250  411 1054  735
 [91]  233  435  490  310  460  383  375 1270  545  445 1885  380  300  380  377
[106]  425  276  210  800  420  350  360  538 1100 1205  314  237  610  360  540
[121] 1038  424  310  300  444  301  268  620  215  652  900  525  246  360  529
[136]  500  720  270  430  671 1770
>
> data <- c(
   length = length(rivers),
   sum = sum(rivers),
   mean = mean(rivers),
   median = median(rivers),
   variance = var(rivers),
   sd = sd(rivers),
   min = min(rivers),
   max = max(rivers)
 )

> data
     length        sum       mean     median    variance        sd        min 
   141.0000  83357.0000    591.1844    425.0000 243908.4086    493.8708    135.0000 
        max 
  3710.0000
```

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 R scripts and its output.

```
power_ranking <- c(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)
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",
                    "Donald Trump", "Muhammad Ali", "Paul McCartney", "George Lucas",
                    "Elton John", "David Letterman", "Phil Mickelson", "J.K. Rowling",
                    "Brad Pitt", "Peter Jackson", "Dr. Phil McGraw", "Jay Leno",
                    "Celine Dion", "Kobe Bryant")
pay <- c(67, 90, 225, 110, 90, 332, 302, 41, 52, 88, 55, 44, 55, 40, 233, 34, 17, 47, 75,
20, 39, 45, 32, 40, 31)
forbes_data <- data.frame(PowerRanking = power_ranking,
                          CelebrityName = celebrity_name,
                          Pay = pay)
print(forbes_data)
```

```
PowerRanking        CelebrityName 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 Spielberg 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 McCartney  40
15           15         George Lucas 233
16           16           Elton John  34
17           17      David Letterman  17
```

```
18          18       Phil Mickelson  47
19          19         J.K. Rowling  75
20          20            Brad Pitt  20
21          21        Peter Jackson  39
22          22      Dr. Phil McGraw  45
23          23             Jay Leno  32
24          24          Celine Dion  40
25          25          Kobe Bryant  31
```

b. Modify the power ranking and pay of J.K. Rowling.
Change power ranking to 15 and pay to 90. Write the R scripts and its output.

```
forbes_data[forbes_data$CelebrityName == "J.K. Rowling", "PowerRanking"] <- 15
forbes_data[forbes_data$CelebrityName == "J.K. Rowling", "Pay"] <- 90
print(forbes_data)
```

```
PowerRanking       CelebrityName 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 Spielberg 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 McCartney  40
15          15         George Lucas 233
16          16           Elton John  34
17          17      David Letterman  17
18          18       Phil Mickelson  47
19          15         J.K. Rowling  90
20          20            Brad Pitt  20
21          21        Peter Jackson  39
22          22      Dr. Phil McGraw  45
23          23             Jay Leno  32
24          24          Celine Dion  40
25          25          Kobe Bryant  31
```

c. Create an excel file from the table above and save it as csv file (PowerRanking).
Import the csv file into the RStudio. What is the R script?

```
write.csv(forbes_data, file = "PowerRanking.csv", row.names = FALSE)
imported_data <- read.csv("PowerRanking.csv")
print(imported_data)
```

d. Access the rows 10 to 20 and save it as Ranks.RData
Write the R script and its output.

```
subset_data <- forbes_data[10:20, ]
save(subset_data, file = "Ranks.RData")
print(subset_data)
PowerRanking     CelebrityName Pay
10          10        Dan Brown  88
11          11      Bruce Springsteen  55
12          12     Donald Trump  44
13          13     Muhammad Ali  55
14          14   Paul McCartney  40
15          15     George Lucas 233
16          16       Elton John  34
17          17   David Letterman  17
18          18   Phil Mickelson  47
19          15     J.K. Rowling  90
20          20        Brad Pitt  20
```

e.Describe its output.
Each step modifies, saves, or retrieves data while maintaining a clear structure for understanding celebrity power and earnings.

9. Download the Hotels-Vienna https://tinyurl.com/ Hotels- Vienna
a. Import the excel file into your RStudio.
What is the R. script?

```
file_path <- "C:/Users/Kea Joy/Documents/CS101/hotels-vienna.xlsx"

hotels_data <- read_excel(file_path)
hotels_data
```

b.How many dimensions does the dataset have?
What is the R script? What is its output?

```
dataset_dimensions <- dim(hotels_data)
dataset_dimensions
[1] 428  24
```

c.Select columns country, **neighbourhood,**
price, stars, accomodation_type, and
ratings. Write the R script.

```
selected_data <- hotels_data[, c("country", "neighbourhood", "price", "stars",
"accommodation_type", "rating")]
print (head(selected_data))
```

d. Save the data as **new.RData to your RStudio. Write the

R. script.

```
save(selected_data, file = "new.RData")
dir()
```

e.Display the first six rows and last six rows of the new.RData. What is the R script?

```
head(selected_data)
# A tibble: 6 x 6
  country neighbourhood price stars accommodation_type rating
  <chr>   <chr>         <dbl> <dbl> <chr>              <chr>
1 Austria 17. Hernals      81     4 Apartment          4.4000000000000004
2 Austria 17. Hernals      81     4 Hotel              3.9
3 Austria Alsergrund       85     4 Hotel              3.7
4 Austria Alsergrund       83     3 Hotel              4
5 Austria Alsergrund       82     4 Hotel              3.9
6 Austria Alsergrund      229     5 Apartment          4.8
> tail(selected_data)
# A tibble: 6 x 6
  country neighbourhood price stars accommodation_type rating
  <chr>   <chr>         <dbl> <dbl> <chr>              <chr>
1 Austria Wieden           73   3   Hotel              3.4
2 Austria Wieden          109   3   Apartment          5
3 Austria Wieden          185   5   Hotel              4.3
4 Austria Wieden          100   4   Hotel              4.4000000000000004
5 Austria Wieden           58   3   Hotel              3.2
6 Austria Wieden          110   3.5 Apartment          4
```

10.Create a list of ten (10) vegetables you ate during your lifetime. If none, just list down.

a. Write the R scripts and its output.

```
vegetables <- list("Carrot", "Broccoli", "Spinach", "Mushroom", "Cucumber",
              "Spring Bean", "Cabbage", "Squash", "Eggplant", "Okra")
vegetables

[[1]]
[1] "Carrot"

[[2]]
[1] "Broccoli"

[[3]]
[1] "Spinach"

[[4]]
[1] "Mushroom"

[[5]]
[1] "Cucumber"

[[6]]
[1] "Spring Bean"

[[7]]
[1] "Cabbage"

[[8]]
[1] "Squash"
```

```
[[9]]
[1] "Eggplant"

[[10]]
[1] "Okra"
```

b. Add 2 additional vegetables after the last vegetables in the list. What is the Rscript and its output.

```
vegetables <- append(vegetables, list("Malunggay", "Lettuce"))
vegetables
[[1]]
[1] "Carrot"

[[2]]
[1] "Broccoli"

[[3]]
[1] "Spinach"

[[4]]
[1] "Mushroom"

[[5]]
[1] "Cucumber"

[[6]]
[1] "Spring Bean"

[[7]]
[1] "Cabbage"

[[8]]
[1] "Squash"

[[9]]
[1] "Eggplant"

[[10]]
[1] "Okra"

[[11]]
[1] "Malunggay"

[[12]]
[1] "Lettuce"
```

c. Add 4 additional vegetables after index 5. How many datapoints does your vegetable list have? What is the R script and its output?

```
[1] 20
> new_vegetables <- list("Water spinach", "Bitter Gourd", "Broccoli"
, "Bottle gourd")
> vegetables <- append(vegetables, new_vegetables, after = 5)
> vegetables
[[1]]
[1] "Carrot"

[[2]]
[1] "Broccoli"

[[3]]
[1] "Spinach"

[[4]]
[1] "Mushroom"

[[5]]
[1] "Cucumber"
```

```
[[6]]
[1] "Water spinach"

[[7]]
[1] "Bitter Gourd"

[[8]]
[1] "Broccoli"

[[9]]
[1] "Bottle gourd"

[[10]]
[1] "Water spinach"

[[11]]
[1] "Bitter Gourd"

[[12]]
[1] "Broccoli"

[[13]]
[1] "Bottle gourd"

[[14]]
[1] "Spring Bean"

[[15]]
[1] "Cabbage"

[[16]]
[1] "Squash"

[[17]]
[1] "Eggplant"

[[18]]
[1] "Okra"

[[19]]
[1] "Malunggay"

[[20]]
[1] "Lettuce"
```

Remove the vegetables in index 5, 10, and 15. How many vegetables were left? Write the codes and its output.

```
> num_remaining_vegetables <- length(vegetables)
> num_remaining_vegetables
[1] 17
```

Note: Do not forget to push into your GitHub repo.

Prepared by:
JOYCE F. JAMILE
Instructor

Without ethical considerations, Al becomes a tool of chaos and harm.