Worksheet-2 in R

John Martin S. Pabriaga

Worksheet for R Programming

Instructions:

- Use RStudio or the RStudio Cloud to accomplish this worksheet.
- Save the R script as Worksheet_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.

```
seq_vec <- -5:5
seq_vec
```

```
 [1] -5 -4 -3 -2 -1  0  1  2  3  4  5
```

The output creates a sequence starting from -5 and ending at 5, incrementing by 1.

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

```
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 script and its output. Describe the output.

```
 seq_vec <- seq(1, 3, by = 0.2)
seq_vec
```

```
 [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 is a numeric vector starting from 1 to 3, with values incremented by 0.2

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?

```
ages[3]
```

```
[1] 22
```

b. Access 2nd and 4th element, what are the values?
ages[c(2, 4)]

[1] 28 36

c. Access all but the 4th and 12th element is not included. Write the R script and its output.
ages[-c(4, 12)]

[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 41 51 35 24 33
[32] 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)
x

first second third
3 0 9

a. Print the results. Then access x [c("first", "third")].
Describe the output.
The output of x[c("first", "third")] returns the values corresponding to "first" and "third", which are 3 and 9, respectively. The names "first" and "third" are also shown in the output to indicate which elements were accessed. The name "second" is not shown in the output.

b. Write the code and its output.
x[c("first", "third")]

first third
3 9

5. Create a sequence x from -3:2.
a. Modify 2nd element and change it to 0;
x [2] <- 0
x
Describe the output.
Initially, the sequence x is created as [-3, -2, -1, 0, 1, 2]. Then, The second element (which was -2) is modified and changed to 0. The output is [-3, 0, -1, 0, 1, 2].

b.Write the code and its output.
x <- -3:2
x[2] <- 0
x

[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.

```
fuel_data <- data.frame(
Month = month,
 Price_per_Liter = price_per_liter,
 Purchase_Quantity = purchase_quantity)
fuel_data
```

```
  Month   Price_per_Liter   Purchase_Quantity
1   Jan          52.50                  25
2   Feb          57.25                  30
3   Mar          60.00                  40
4   Apr          65.00                  50
5   May          74.25                  10
6   Jun          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.

```
avg_expenditure <- weighted.mean(price_per_liter, purchase_quantity)
avg_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))

c. What are the results?

```
[1]    141.0000  83357.0000    591.1844    425.0000  243908.4086    493.8708    135.0000
3710.0000
```

Length: 141, Sum: 83357, Mean: 591.1844, Median: 425 Variance: 243908.4086 Standard Deviation: 493.8708 , Minimum: 135, Maximum: 3710

d. Write the R scripts and its outputs.

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

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

b. 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)
```

	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

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.

(a) It creates three separate vectors for the PowerRanking, CelebrityName, and Pay columns. They are combined into a data frame.
(b) It successfully updates the power ranking of J.K. Rowling to 15 and her pay to 90.
(c) It saves the table as a CSV file and imports it back into R, maintaining the data structure.
(d) It extracts rows 10 through 20 from the data and saves them as an RData file for future use. The subset contains celebrities like Dan Brown, Bruce Springsteen, etc.

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

```
install.packages("readxl")
library(readxl)

file_path <- "/Users/johnmartin/Desktop/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?

```
dimensions <- dim(hotels_data)
print(dimensions)
```

[1] 428 24

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

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

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

```
save(selected_columns, file = "new.RData")
```

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

```
load("~/Desktop/CS101/new.RData")
print(head(selected_columns))
print(tail(selected_columns))
```

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("Squash", "Carrot", "String Beans", "Moringa leaves", "Cabbage",
                   "Bitter Gourd", "Potato", "Eggplant", "Lettuce", "Water spinach")
print(vegetables)
```

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

[[2]]
[1] "Carrot"

[[3]]
[1] "String Beans"

[[4]]
[1] "Moringa leaves"

[[5]]
[1] "Cabbage"

[[6]]
[1] "Bitter Gourd"

[[7]]
[1] "Potato"

[[8]]
[1] "Eggplant"

[[9]]
[1] "Lettuce"

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

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

```
vegetables <- append(vegetables, c("Cauliflower", "Peas"))
print(vegetables)
```

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

[[2]]
[1] "Carrot"

[[3]]
[1] "String Beans"

[[4]]
[1] "Moringa leaves"

[[5]]
[1] "Cabbage"

[[6]]
[1] "Bitter Gourd"

[[7]]
[1] "Potato"

[[8]]
[1] "Eggplant"

[[9]]
[1] "Lettuce"

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

[[11]]
[1] "Cauliflower"

[[12]]
[1] "Peas"
```

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

```
vegetables <- append(vegetables, c("Spinach", "Broccoli", "Mushroom", "Bell Pepper"), after = 5)
print(vegetables)
length(vegetables)
```

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

[[2]]
[1] "Carrot"

[[3]]
[1] "String Beans"

[[4]]
[1] "Moringa leaves"

[[5]]
[1] "Cabbage"

[[6]]
[1] "Spinach"

[[7]]
[1] "Broccoli"

[[8]]
[1] "Mushroom"

[[9]]
[1] "Bell Pepper"

[[10]]
[1] "Bitter Gourd"

[[11]]
[1] "Potato"

[[12]]
[1] "Eggplant"

[[13]]
[1] "Lettuce"

[[14]]
[1] "Water spinach"

[[15]]
[1] "Cauliflower"

[[16]]
[1] "Peas"

> length(vegetables)
[1] 16
```

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

vegetables <- vegetables[-c(5, 10, 15)]
print(vegetables)
length(vegetables)

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

[[2]]
[1] "Carrot"

[[3]]
[1] "String Beans"

[[4]]
[1] "Moringa leaves"

[[5]]
[1] "Spinach"

[[6]]
[1] "Broccoli"

[[7]]
[1] "Mushroom"

[[8]]
[1] "Bell Pepper"

[[9]]
[1] "Potato"

[[10]]
[1] "Eggplant"

[[11]]
[1] "Lettuce"

[[12]]
[1] "Water spinach"

[[13]]
[1] "Peas"

> length(vegetables)
[1] 13
```

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.