Worksheet-2 in R
RWorksheet_Sorenio#2

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.

```
> x <- -5:5
> x
 [1] -5 -4 -3 -2 -1  0  1  2  3  4
[11]  5
> # b
> x <- 1:7
> x
[1] 1 2 3 4 5 6 7
>
```

The output (-5:5) generates a vector of 11 integers starting from -5 and ending at 5.

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

[1] 1 2 3 4 5 6 7
This means x is a vector containing the integers from 1 to 7. The colon (:) operator in R creates a sequence, incrementing by 1 between the specified start and end points. So in this case, the values range from 1 to 7, inclusive.

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 <- seq(1, 3, by=0.2)
> seq
 [1] 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6
[10] 2.8 3.0
```

The output of the R function seq(1, 3, by=0.2) is a sequence of numbers starting at 1 and ending at 3, incrementing by 0.2. This creates a vector with the values: 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, and 3.0. The sequence is inclusive of the end value (3.0) because the step size divides evenly within the range from 1 to 3. The output is a continuous, evenly spaced set of numbers in a vector format.

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 <- 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)
>
> ages[3]
[1] 22
```

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

```
> # b
> 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
[13] 37 34 19 20 57 49 50 37 46 25 17 37
[25] 43 53 41 51 35 24 33 41 53 40 18 44
[37] 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).

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

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

The output shows that the values associated with first and third are 3 and 9.

b. Write the code and its output.

```
> # b
> x <- c("first"=3, "second"=0, "third"=9)
> names(x)
[1] "first"  "second" "third"
> 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.

The initial sequence x <- -3:2 creates a vector with the elements -3, -2, -1, 0, 1,

2. After changing the 2nd element (-2) to 0, the modified vector becomes -3, 0, -1, 0, 1, 2.

b. Write the code and its output.

```
> x <- -3:2
>
> # a
> 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.

```
> month <- c("Jan", "Feb", "March", "Apr",
"May", "June")
> price_liter <- c(52.50, 57.25, 60.00, 65.0
0, 74.25, 54.00)
> purchase_quantity <- c(25, 30, 40, 50, 10,
45)
>
> fueldata <- data.frame(month, price_liter,
purchase_quantity)
>
> print(fueldata)
  month price_liter purchase_quantity
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_fuelprice <- weighted.mean(price_l
iter, purchase_quantity)
> print(average_fuelprice)
[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))

```
> data(rivers)
> data <- c(length(rivers), sum(rivers), mea
n(rivers), median(rivers), var(rivers), sd(r
ivers), min(rivers), max(rivers))
> data
```

b. What are the results?

Length: 141.000 (number of rivers)
Sum: 83357.000 (total length of all rivers in miles)
Mean: 591.184 (average river length)
Median: 425.0000 (middle value of river lengths)
Variance: 243908.4086 (variance of the river lengths)
Standard Deviation: 493.8708 (standard deviation of river lengths)
Minimum: 135.000 (shortest river length)
Maximum: 3710.000 (longest river length)

c. Write the R scripts and its outputs.

```
> data(rivers)
> data <- c(length(rivers), sum(rivers), mea
n(rivers), median(rivers), var(rivers), sd(r
ivers), min(rivers), max(rivers))
> data
[1]    141.0000  83357.0000    591.1844
[4]    425.0000 243908.4086    493.8708
[7]    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, 40, 47, 25, 25, 39, 45, 32, 40, 31)

celeb_data <- data.frame(power_ranking, celebrity_name, pay)

celeb_data
```

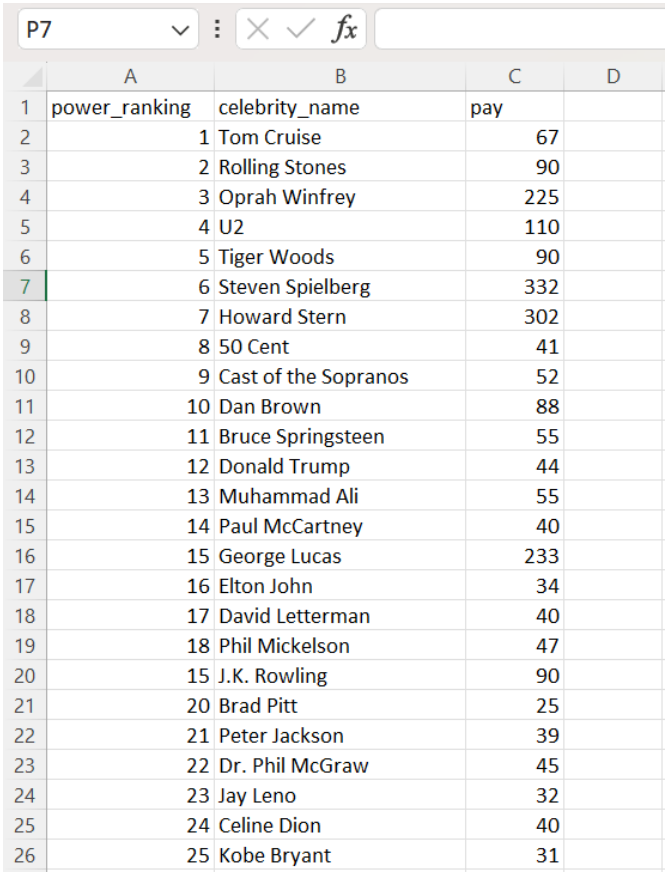
```
> celeb_data
   power_ranking       celebrity_name pay
1              1           Tom Cruise  67
2              2       Rolling Stones  90
3              3        Oprah Winfrey 225
4              4                   U2 110
5              5          Tiger Woods  90
6              6     Steven Spielberg 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  40
18            18       Phil Mickelson  47
19            19         J.K. Rowling  25
20            20            Brad Pitt  25
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.

```
> celeb_data[celeb_data$celebrity_name == "J.K. Rowling", "power_ranking"] <- 15
> celeb_data[celeb_data$celebrity_name == "J.K. Rowling", "pay"] <- 90
>
> celeb_data
   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 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  40
18            18       Phil Mickelson  47
19            15          J.K. Rowling  90
20            20            Brad Pitt  25
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(celeb_data, "PowerRanking.csv", row.names = FALSE)



	A	B	C	D
1	power_ranking	celebrity_name	pay	
2	1	Tom Cruise	67	
3	2	Rolling Stones	90	
4	3	Oprah Winfrey	225	
5	4	U2	110	
6	5	Tiger Woods	90	
7	6	Steven Spielberg	332	
8	7	Howard Stern	302	
9	8	50 Cent	41	
10	9	Cast of the Sopranos	52	
11	10	Dan Brown	88	
12	11	Bruce Springsteen	55	
13	12	Donald Trump	44	
14	13	Muhammad Ali	55	
15	14	Paul McCartney	40	
16	15	George Lucas	233	
17	16	Elton John	34	
18	17	David Letterman	40	
19	18	Phil Mickelson	47	
20	15	J.K. Rowling	90	
21	20	Brad Pitt	25	
22	21	Peter Jackson	39	
23	22	Dr. Phil McGraw	45	
24	23	Jay Leno	32	
25	24	Celine Dion	40	
26	25	Kobe Bryant	31	

```
> imported_data <- read.csv("PowerRanking.csv")
>
> imported_data
   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 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  40
18            18        Phil Mickelson  47
19            15         J.K. Rowling  90
20            20            Brad Pitt  25
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.

```
> ranks_subset <- celeb_data[10:20, ]
>
> save(ranks_subset, file = "Ranks.RData")
>
> ranks_subset
   power_ranking    celebrity_name 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  40
18            18    Phil Mickelson  47
19            15      J.K. Rowling  90
20            20         Brad Pitt  25
>
```

e. Describe its output.

The output includes rows 10 to 20 from the original dataset. It displays the power rankings, celebrity names, and their corresponding annual pay. These rows feature well-known celebrities such as Dan Brown, Bruce Springsteen, Donald Trump, Muhammad Ali, and others. Each row represents a specific celebrity along with their power ranking and pay.

This subset will be saved as Ranks.RData, and it can be later loaded back into R using the load() function.

9. Download the Hotels-Vienna https://tinyurl.com/Hotels-Vienna

a. Import the excel file into your RStudio. What is the R script?

```
# 9. a
file_path <- "C:/Users/User/Downloads/hotels-vienna.xlsx"
hotels_data <- read_excel(file_path)

head(hotels_data)
```

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

```
> # b
> 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", "accom
modation_type", "rating")]

head(selected_data)
```

```
> selected_data <- hotels_data[, c("country", "neighbourhood",
"price", "stars", "accommodation_type", "rating")]
>
> head(selected_data)
# A tibble: 6 × 6
  country neighbourhood price stars
  <chr>   <chr>         <dbl> <dbl>
1 Austria 17. Hernals      81     4
2 Austria 17. Hernals      81     4
3 Austria Alsergrund       85     4
4 Austria Alsergrund       83     3
5 Austria Alsergrund       82     4
6 Austria Alsergrund      229     5
```

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

```
> # d
> save(selected_data, file = "new.RData")
>
> file_exists <- file.exists("new.RData")
> print(file_exists)
[1] TRUE
```

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

```
> # e
> load("new.RData")
> head(selected_data)
# A tibble: 6 × 6
  country neighbourhood price stars accommodation_type rating
  <chr>   <chr>         <dbl> <dbl> <chr>              <chr>
1 Austria 17. Hernals      81     4 Apartment          4.400…
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 × 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              4.400…
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 <- c("Cabbage", "Squash", "Beans", "Lettuce", "Papaya", "Malunggay","Potato", "Eggplant", "Mushroom", "Brocolli")

print(vegetables)

```
> vegetables <- c("Cabbage", "Squash", "Beans", "Lettuce", "Papaya", "Malungga
y", "Potato", "Eggplant", "Mushroom", "Brocolli")
> print(vegetables)
 [1] "Cabbage"   "Squash"    "Beans"     "Lettuce"   "Papaya"    "Malunggay"
 [7] "Potato"    "Eggplant"  "Mushroom"  "Brocolli"
```

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

```
> vegetables <- c(vegetables, "Carrot", "Green Peas")
> print(vegetables)
 [1] "Cabbage"    "Squash"     "Beans"      "Lettuce"    "Papaya"
 [6] "Malunggay"  "Potato"     "Eggplant"   "Mushroom"   "Brocolli"
[11] "Carrot"     "Green 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("Radish", "Peanut", "Garlic", "Onion"), af
ter = 5)
> print(vegetables)
 [1] "Cabbage"    "Squash"     "Beans"      "Lettuce"    "Papaya"
 [6] "Radish"     "Peanut"     "Garlic"     "Onion"      "Radish"
[11] "Peanut"     "Garlic"     "Onion"      "Malunggay"  "Potato"
[16] "Eggplant"   "Mushroom"   "Brocolli"   "Carrot"     "Green Peas"
> length(vegetables)
[1] 20
```

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

```
> # d
> vegetables <- vegetables[-c(5, 10, 15)]
> print(vegetables)
 [1] "Cabbage"    "Squash"     "Beans"      "Lettuce"    "Radish"
 [6] "Peanut"     "Garlic"     "Onion"      "Potato"     "Eggplant"
[11] "Mushroom"   "Brocolli"   "Green Peas"
> length(vegetables)
[1] 13
```

Codes:

```
# a
vegetables <- c("Cabbage", "Squash", "Beans", "Lettuce", "Papaya", "Malunggay",
"Potato", "Eggplant", "Mushroom", "Brocolli")

print(vegetables)

# b
vegetables <- c(vegetables, "Carrot", "Green Peas")

print(vegetables)

# c
vegetables <- append(vegetables, c("Radish", "Peanut", "Garlic", "Onion"), after =
5)

print(vegetables)

length(vegetables)

# d
vegetables <- vegetables[-c(5, 10, 15)]

print(vegetables)

length(vegetables)
```

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

GitHub Repository:
https://github.com/AMSorenio/RWorksheet.Sorenio.CS101-repo.git

GitHub Account Link:
https://github.com/AMSorenio

Prepared by:

JOYCE F. JAMILE
Instructor

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