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.

The : operator generates a sequence of integers from -5 to 5.

sequence <- -5:5 print(sequence) Output:

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

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

x <- 1:7 print(x)

Output:

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.

sequence <- seq(1, 3, by = 0.2) print(sequence)

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

The output creates a sequence from 1 to 3 with a step size of 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?

Output: [1] 22

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

Output: [1] 28 36

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

Output:

[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 41 53 40 18 44 38 41 48 27 39 19

[43] 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.

It's creates a named vector, and accessing elements by their names returns the selected values

b. Write the code and its output.

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

print(x)

print(x[c("first", "third")]) Output:

>first second third

3 0 9

> print(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 second element of the sequence is modified

to 0.

b. Write the code and its output.

x <- -3:2 x[2]

<- 0 print(x)

Output:

-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-quntity Liter (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_per_liter <- c(52.5, 57.25, 60.00, 65.00, 74.25, 54.00) purchase_quantity <- c(25, 30, 40, 50, 10, 45)

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

print(diesel_data)

avg_expenditure <- weighted.mean(price_per_liter, purchase_quantity)

print(avg_expenditure) Ouput:

month price_per_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

> avg_expenditure <- weighted.mean(price_per_liter, purchase_quantity)

> print(avg_expenditure)

[1] 59.2625

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.

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

[1] 141.0000 83357.0000 591.1844 425.0000 243908.4086 493.8708 135.0000 3710.0000

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?

- **Length: Number of elements in the `rivers` dataset (141).**
- **Sum: Total length of all rivers.**

- **Mean: Average river length.**
- **Median: Middle value in the sorted list of river lengths.**
- **Variance: Measure of how far river lengths are spread from the mean.**
- **Standard deviation: Square root of variance, showing dispersion.**
- **Minimum: Shortest river. □ Maximum: Longest river.**

c. Write the R scripts and its outputs.

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

[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.

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, 75, 25, 39, 45, 32, 40, 31)

print(power_ranking)

print(celebrity_name)

print(pay) Output:

> print(power_ranking)

[1] 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

> print(celebrity_name)

[1] "Tom Cruise" "Rolling Stones" "Oprah Winfrey" "U2" "Tiger Woods"

[6] "Steven Spielberg" "Howard Stern" "50 Cent" "Cast of the Sopranos" "Dan Brown"

[11] "Bruce Springsteen" "Donald Trump" "Muhammad Ali" "Paul McCartney" "George Lucas"

[16] "Elton John" "David Letterman" "Phil Mickelson" "J.K. Rowling" "Brad Pitt"

[21] "Peter Jackson" "Dr. Phil McGraw" "Jay Leno" "Celine Dion" "Kobe Bryant"

> print(pay)

[1] 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

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.

power_ranking[which(celebrity_name == "J.K. Rowling")] <- 15

pay[which(celebrity_name == "J.K. Rowling")] <- 90 print(power_ranking)

print(pay) Output:

>print(power_ranking)

[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 15 20 21 22 23 24 25

> print(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

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?

celebrity_data <- data.frame(PowerRanking = power_ranking, CelebrityName = celebrity_name, Pay = pay) write.csv(celebrity_data, "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 <- imported_data[10:20,]

save(subset_data, file = "Ranks.RData")

print(subset_data) save(subset_data, file = "Ranks.RData")

print(subset_data) Output:

> subset_data <- imported_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  40
18          18    Phil Mickelson  47
19          15     J.K. Rowling  90
20          20        Brad Pitt  25
> 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  40
18          18    Phil Mickelson  47
19          15     J.K. Rowling  90
20          20        Brad Pitt  25
>
```

e. Describe its output.

The output will display rows 10 to 20 from the imported_data data frame, showing the selected celebrity names, their power rankings, and pay. Additionally, the data will be saved as Ranks.RData for future use. This will include celebrities like Dan Brown, Bruce Springsteen, Donald Trump, and others, along with their respective power rankings and pay values. 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)

hotel_data <- read_excel("path_to_hotels_vienna_file.xlsx") print(hotel_data)

b. How many dimensions does the dataset have?

What is the R script? WHat is its output?

dimensions <- dim(hotel_data)

print(dimensions) Ouput:

[1] 100 7

c. Select columns country, neighbourhood, price, stars,

accomodation_type, and ratings. Write the R script.

selected_columns <- hotel_data[, c("country", "neighbourhood", "price", "stars", "accommodation_type", "ratings")] print(selected_columns)

d. Save the data as **new.RData to your RStudio. Write the R 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("new.RData") head(selected_columns)

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 <- c("Carrot", "Broccoli", "Spinach", "Tomato", "Potato","Cabbage", "Cauliflower", "Pepper", "Cucumber", "Lettuce")

print(vegetables) Output:

rint(vegetables)

[1] "Carrot" "Broccoli" "Spinach" "Tomato" "Potato" "Cabbage" "Cauliflower" "Pepper" "Cucumber"

[10] "Lettuce"

b. Add 2 additional vegetables after the last vegetables

in the list. What is the R script and its output?

vegetables <- c(vegetables, "Onion", "Garlic")

print(vegetables) Output:

[1] "Carrot" "Broccoli" "Spinach" "Tomato" "Potato" "Cabbage" "Cauliflower" "Pepper" "Cucumber"

[10] "Lettuce" "Onion" "Garlic

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 <- c(vegetables[1:5], "Eggplant",

"Zucchini", "Pumpkin", "Beetroot",

vegetables[6:length(vegetables)])

print(vegetables) length(vegetables) Output:

> print(vegetables)

[1] "Carrot" "Broccoli" "Spinach" "Tomato" "Potato" "Eggplant" "Zucchini" "Pumpkin" "Beetroot"

[10] "Cabbage" "Cauliflower" "Pepper" "Cucumber" "Lettuce" "Onion" "Garlic"

> 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) Output:

> print(vegetables)

[1] "Carrot" "Broccoli" "Spinach" "Tomato" "Eggplant" "Zucchini" "Pumpkin" "Beetroot" "Cauliflower"

[10] "Pepper" "Cucumber" "Lettuce" "Garlic"

> length(vegetables)

[1] 13

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

Prepared by:

JOYCE F. JAMILE

Instructor

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