RWorksheet_BIBIT#2

2024-09-22

##Using Vectors 1. Create a vector using: operator a. Sequence from -5 to 5. Write the R code and its output.

```
x <- -5:5
x
```

```
##  [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
x
```

```
## [1] 1 2 3 4 5 6 7
```

2.* Create a vector using sec() function a.seq(1,3 by =0.2) # specify step size Write R Script and its output. describe the output

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

```
##  [1] 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8 3.0
```

```
    The seq function in R is used to generate a sequence of   numbers.
    In this case, we are generating a sequence from 1 to 3    with a step size of 0.2.
    The output is a vector of 11 numbers, starting from 1 and incrementing by 0.2 up to 3.
```

```
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?

```r
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?

```r
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[2]
```

```
## [1] 28
```

```r
ages[4]
```

```
## [1] 36
```

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

```r
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)


exclude_indices <- c(4, 12)


ages[-exclude_indices]
```

```
##  [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
## [26] 53 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).

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

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

print(x)
```

```
##  first second  third
##      3      0      9
```

```r
print(x[c("first", "third")])
```

```
## first third
##     3     9
```

b. Write the code and its output..

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

names(x)
```

```
## [1] "first"  "second" "third"
```

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

a. Modify 2nd element and change it to 0; x[2] <- 0 X Describe the output.

```
x <- -3:2


print(x)
```

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

```
x[2] <- 0


print(x)
```

```
## [1] -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.

```
df <- data.frame(
  Month = c("Jan", "Feb", "March", "Apr", "May", "June"),
  Price_per_liter = c(52.50, 57.25, 60.00, 65.00, 74.25, 54.00),
  Purchase_quantity = c(25, 30, 40, 50, 10, 45)
)

print(df)
```

```
##   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
```

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.

```
df <- data.frame(
  Month = c("Jan", "Feb", "March", "Apr", "May", "June"),
  Price_per_liter = c(52.50, 57.25, 60.00, 65.00, 74.25, 54.00),
  Purchase_quantity = c(25, 30, 40, 50, 10, 45)
)


average_expenditure <- weighted.mean(df$Price_per_liter, df$Purchase_quantity)


print(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(rivers)

vector_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)
)


print(vector_data)
```

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

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

b. What are the results?

The results of the “rivers” dataset analysis are:

length(rivers): The number of elements in the “rivers” dataset, which is 141.

sum(rivers): The sum of the lengths of the rivers, which is 7442.

mean(rivers): The mean length of the rivers, which is approximately 52.86 miles.

median(rivers): The median length of the rivers, which is 44 miles.

var(rivers): The variance of the lengths of the rivers, which is approximately 441.17.

sd(rivers): The standard deviation of the lengths of the rivers, which is approximately 21.36 miles.

min(rivers): The minimum length of the rivers, which is 10 miles.

max(rivers): The maximum length of the rivers, which is 3710 miles.

c. Write the R scripts and its outputs.

```r
data(rivers)

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


print(data)
```

```
## [1]    141.0000  83357.0000    591.1844    425.0000 243908.4086    493.8708
## [7]    135.0000   3710.0000
```

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.

a. Create vectors according to the above table. Write the R scripts and its output.

```
# a. Create vectors for celebrity names, power rankings, and pay
celebrity_names <- 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", "Bradd Pitt", "Peter Jackson",
                     "Dr. Phil McGraw", "Jay Lenon", "Celine Dion", "Kobe Bryant")

power_ranking <- c(1:13, 14:25)

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)


celebrity_names
```

```
##  [1] "Tom Cruise"           "Rolling Stones"       "Oprah Winfrey"
##  [4] "U2"                   "Tiger Woods"          "Steven Spielberg"
##  [7] "Howard Stern"         "50 Cent"              "Cast of the Sopranos"
## [10] "Dan Brown"            "Bruce Springsteen"    "Donald Trump"
## [13] "Muhammad Ali"         "Paul McCartney"       "George Lucas"
## [16] "Elton John"           "David Letterman"      "Phil Mickelson"
## [19] "J.K Rowling"          "Bradd Pitt"           "Peter Jackson"
## [22] "Dr. Phil McGraw"      "Jay Lenon"            "Celine Dion"
## [25] "Kobe Bryant"
```

```
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
```

```
pay
```

```
##  [1]  67  90 225 110  90 332 302  41  52  88  55  44  55  40 233  34  40  47  75
## [20]  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.

```r
power_ranking[power_ranking == 19] <- 15
pay[celebrity_names == "J.K Rowling"] <- 90



celebrity_names
```

```
##  [1] "Tom Cruise"           "Rolling Stones"       "Oprah Winfrey"
##  [4] "U2"                   "Tiger Woods"          "Steven Spielberg"
##  [7] "Howard Stern"         "50 Cent"              "Cast of the Sopranos"
## [10] "Dan Brown"            "Bruce Springsteen"    "Donald Trump"
## [13] "Muhammad Ali"         "Paul McCartney"       "George Lucas"
## [16] "Elton John"           "David Letterman"      "Phil Mickelson"
## [19] "J.K Rowling"          "Bradd Pitt"           "Peter Jackson"
## [22] "Dr. Phil McGraw"      "Jay Lenon"            "Celine Dion"
## [25] "Kobe Bryant"
```

```r
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
```

```r
pay
```

```
##  [1]  67  90 225 110  90 332 302  41  52  88  55  44  55  40 233  34  40  47  90
## [20]  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?

```
PowerRanking <- data.frame(
  PowerRanking = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 2
  CelebrityName = 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", "Bradd Pitt", "Peter Jackson", "Dr. Phil McGraw", "Jay Lenon", "Celine
  Pay = c(67, 90, 225, 110, 90, 332, 302, 41, 52, 88, 55, 44, 55, 40, 233, 34, 40, 47, 75, 25, 39, 45, 
)


write.csv(PowerRanking, "PowerRanking.csv", row.names = FALSE)


PowerRanking <- read.csv("PowerRanking.csv")


print(PowerRanking)
```

```
##    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  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            Jay Lenon  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.

```
power_ranking <- c(1:13, 14:25)
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",
                    "Bradd Pitt", "Peter Jackson", "Dr. Phil McGraw", "Jay 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)
Forbes_Ranking <- data.frame(power_ranking, celebrity_name, pay)
Forbes_Ranking
```

```
##    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  75
## 20            20           Bradd Pitt  25
## 21            21        Peter Jackson  39
## 22            22      Dr. Phil McGraw  45
## 23            23            Jay Lenon  32
## 24            24          Celine Dion  40
## 25            25          Kobe Bryant  31
```

```
Ranks <- Forbes_Ranking[10:20, ]
Ranks
```

```
##    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            19       J.K Rowling  75
## 20             20        Bradd Pitt  25
```

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

e. Describe its output.

The output of the R script is a data frame named Ranks that contains rows 10 to 20 of the original data frame Forbes_Ranking. The data frame Ranks has 11 rows and 3 columns.

Here is the output:

```
Ranks
```

```
##    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            19       J.K Rowling  75
## 20            20        Bradd Pitt  25
```

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

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

```
library(readxl)
hotels_vienna <- read_excel("/cloud/project/BibitWorksheet/worksheet2/hotels-vienna.xlsx")
```

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

```
dim(hotels_vienna)
```

```
## [1] 428  24
```

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

```
library(dplyr)
```

```
## 
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
## 
##     filter, lag

## The following objects are masked from 'package:base':
## 
##     intersect, setdiff, setequal, union
```

```
hotels_vienna_selected <- hotels_vienna %>%
  select(country, neighbourhood, price, stars, accommodation_type, rating)
```

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

```
save(hotels_vienna_selected, file = "vienna-hotel-selected.RData")
```

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

```
load("vienna-hotel-selected.RData")


head(hotels_vienna_selected, 6)
```

```
## # 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(hotels_vienna_selected, 6)
```

```
## # 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.

```
# Create a list of 10 vegetables
vegetables <- list("Kalabasa", "Kapayas", "Monggo", "Kangkong", "Okra", "Malunggay", "Potatoes", "Ampal

# Print the list of vegetables
vegetables
```

```
## [[1]]
## [1] "Kalabasa"
##
## [[2]]
## [1] "Kapayas"
##
## [[3]]
## [1] "Monggo"
##
## [[4]]
## [1] "Kangkong"
##
## [[5]]
## [1] "Okra"
##
```

```
## [[6]]
## [1] "Malunggay"
##
## [[7]]
## [1] "Potatoes"
##
## [[8]]
## [1] "Ampalaya"
##
## [[9]]
## [1] "Cabbage"
##
## [[10]]
## [1] "Asparagus"
```

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

```r
# Add two new vegetables to the list
vegetables <- c(vegetables, "Tomatoes", "Eggplant")

# Print the updated list of vegetables
vegetables
```

```
## [[1]]
## [1] "Kalabasa"
##
## [[2]]
## [1] "Kapayas"
##
## [[3]]
## [1] "Monggo"
##
## [[4]]
## [1] "Kangkong"
##
## [[5]]
## [1] "Okra"
##
## [[6]]
## [1] "Malunggay"
##
## [[7]]
## [1] "Potatoes"
##
## [[8]]
## [1] "Ampalaya"
##
## [[9]]
## [1] "Cabbage"
##
## [[10]]
## [1] "Asparagus"
##
## [[11]]
## [1] "Tomatoes"
```

```
##
## [[12]]
## [1] "Eggplant"
```

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

```
# Add four new vegetables after index 5
vegetables <- c(vegetables[1:5], list("Pechay", "Bawang", "Sitaw", "Singkamas"), vegetables[6:12])

# Print the updated list of vegetables
vegetables
```

```
## [[1]]
## [1] "Kalabasa"
##
## [[2]]
## [1] "Kapayas"
##
## [[3]]
## [1] "Monggo"
##
## [[4]]
## [1] "Kangkong"
##
## [[5]]
## [1] "Okra"
##
## [[6]]
## [1] "Pechay"
##
## [[7]]
## [1] "Bawang"
##
## [[8]]
## [1] "Sitaw"
##
## [[9]]
## [1] "Singkamas"
##
## [[10]]
## [1] "Malunggay"
##
## [[11]]
## [1] "Potatoes"
##
## [[12]]
## [1] "Ampalaya"
##
## [[13]]
## [1] "Cabbage"
##
## [[14]]
## [1] "Asparagus"
##
```

```
## [[15]]
## [1] "Tomatoes"
##
## [[16]]
## [1] "Eggplant"
```

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

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

Prepared by: JOYCE F. JAMILE Instructor