

RWorksheet_Freires#2

2024-09-21

1. Create a vector using : operator

- a) Sequence from -5 to 5. Write the R code and its output. Describe its output.

```
num <- -5:5  
print(num)
```

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

- The output would be sequenced by -5 until 5.

- b) What will be the value of x ?

```
x <- 1:7  
print(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.

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

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

Describe the output.

- The output is a sequenced vector that starts with 1 until 3.

3. A factory has a census of its workers. There are 50 workers in total. The following list shows their ages:

```
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, 29)
```

- a. Access 3rd element, what is the value?

```
third <- ages[3]
print(third)
```

```
## [1] 22
```

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

```
sec_fourth <- ages[c(2,4)]  
print(sec_fourth)
```

```
## [1] 28 36
```

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

```
all <- ages[-c(4,12)]  
print(all)
```

```
## [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)`.

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

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

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

```
names <- x[c("first", "third")]
print(names)
```

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

Describe the output.

- The output is showing 3 vectors and then accessing elements of the first and third by giving names.

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

- a. Modify 2nd element and change it to 0; `x[2] <- 0`

```
x <- -3:2
x[2] <- 0
print(x)
```

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

Describe the output.

- The output shows the sequence from -3 until 2, After accessing the second element, the -2 has changed to 0.

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")
php <- c(52.50, 57.25, 60.00, 65.00, 74.25, 54.00)
liter <- c(25, 30, 40, 50, 10, 45)

fuel_data <- data.frame(Month = month, Php = php, Liter = liter)
print(fuel_data)
```

```
##   Month   Php Liter
## 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_exp <- weighted.mean(php, liter)
print(average_exp)
```

```
## [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))
print(data)
```

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

b. What are the results?

- [1] 141.0000 (length of rivers) 83357.0000(sum of rivers) 591.1844(mean of rivers) 425.0000(median of rivers) 243908.4086 (variance of rivers) 493.8708(standard deviation of rivers) 135.0000(minimum length of rivers) 3710.0000(maximum length of rivers)

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.

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

```
celebrity <- c("Tom Cruise", "Rolling Stones", "Oprah Winfrey", "U2", "Tiger Woods", "Steven Spielberg")
pay <- c(67, 90, 225, 110, 90, 332, 302, 41, 52, 88, 55, 44, 55, 40, 233, 34, 40, 47, 75, 25, 39, 45, 31)
```

```
celebrity_data <- data.frame(Celebrity = celebrity, Pay = pay)
print(celebrity_data)
```

```
##           Celebrity Pay
## 1           Tom Cruise 67
## 2       Rolling Stones 90
## 3         Oprah Winfrey 225
## 4                U2 110
## 5           Tiger Woods 90
## 6       Steven Spielberg 332
## 7         Howard Stern 302
## 8              50 Cent 41
## 9   Cast of the Sopranos 52
## 10            Dan Brown 88
## 11      Bruce Springsteen 55
## 12           Donald Trump 44
## 13          Muhammad Ali 55
## 14         Paul McCartney 40
## 15         George Lucas 233
## 16            Elton John 34
## 17      David Letterman 40
## 18          Phil Mickelson 47
## 19           J.K Rowling 75
## 20           Brad Pitt 25
## 21        Peter Jackson 39
## 22      Dr. Phil McGraw 45
## 23             Jay Lenon 32
## 24          Celine Dion 40
## 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.

```
celebrity_data[celebrity_data$celebrity == "J.K. Rowling", "Pay"] <- 90
celebrity_data$Power_ranking <- 1:nrow(celebrity_data)
celebrity_data[celebrity_data$celebrity == "J.K. Rowling", "Power_ranking"] <- 15
print(celebrity_data)
```

```
##           Celebrity Pay Power_ranking
## 1      Tom Cruise  67             1
## 2    Rolling Stones  90             2
## 3      Oprah Winfrey 225             3
## 4              U2 110             4
## 5      Tiger Woods  90             5
## 6    Steven Spielberg 332            6
## 7      Howard Stern 302             7
## 8           50 Cent  41             8
## 9  Cast of the Sopranos 52             9
## 10         Dan Brown  88            10
## 11   Bruce Springsteen 55            11
## 12        Donald Trump 44            12
## 13        Muhammad Ali 55            13
## 14        Paul McCartney 40           14
## 15        George Lucas 233           15
## 16         Elton John  34            16
## 17   David Letterman  40            17
## 18        Phil Mickelson 47           18
## 19         J.K Rowling  75           19
## 20         Brad Pitt  25            20
## 21        Peter Jackson 39           21
## 22     Dr. Phil McGraw  45           22
## 23          Jay Lenon  32           23
## 24         Celine Dion  40           24
## 25         Kobe Bryant  31           25
```

- 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(celebrity_data, "power_ranking.csv", row.names = FALSE)
import_data <- read.csv("power_ranking.csv")
print(import_data)
```

```
##           Celebrity Pay Power_ranking
## 1      Tom Cruise  67             1
## 2    Rolling Stones  90             2
## 3      Oprah Winfrey 225             3
## 4              U2 110             4
## 5      Tiger Woods  90             5
## 6    Steven Spielberg 332            6
## 7      Howard Stern 302             7
## 8           50 Cent  41             8
## 9  Cast of the Sopranos 52             9
## 10         Dan Brown  88            10
## 11   Bruce Springsteen 55            11
## 12        Donald Trump 44            12
## 13        Muhammad Ali 55            13
```

## 14	Paul McCartney	40	14
## 15	George Lucas	233	15
## 16	Elton John	34	16
## 17	David Letterman	40	17
## 18	Phil Mickelson	47	18
## 19	J.K Rowling	75	19
## 20	Brad Pitt	25	20
## 21	Peter Jackson	39	21
## 22	Dr. Phil McGraw	45	22
## 23	Jay Lenon	32	23
## 24	Celine Dion	40	24
## 25	Kobe Bryant	31	25

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

```
ranks <- celebrity_data[10:20, ]
save(ranks, file = "Ranks.RData")
print(ranks)
```

##	Celebrity	Pay	Power_ranking
## 10	Dan Brown	88	10
## 11	Bruce Springsteen	55	11
## 12	Donald Trump	44	12
## 13	Muhammad Ali	55	13
## 14	Paul McCartney	40	14
## 15	George Lucas	233	15
## 16	Elton John	34	16
## 17	David Letterman	40	17
## 18	Phil Mickelson	47	18
## 19	J.K Rowling	75	19
## 20	Brad Pitt	25	20

e. Describe its output.

- The output shows the ranking and pay of top celebrities.