

# RWorksheet\_Calzado#2.R

Michael Angelo S. Calzado

2024-09-22

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

```
vec <- -5:5  
vec
```

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

The output is a vector vec that contains a sequence of numbers from -5 to 5, inclusive.

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

```
x <- 1:7  
x
```

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

The value of x is a vector that contains the numbers 1 through 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.

```
vec <- seq(1, 3, by=0.2)  
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 vector vec that contains a sequence of numbers from 1 to 3, with a step size of 0.2. The seq() function generates a sequence of numbers starting from 1, incrementing by 0.2, until it reaches 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,

4.

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)  
third_element <- ages[3]  
third_element
```

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

The value of the 3rd element is 22.

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

```
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)  
second_element <- ages[2]  
fourth_element <- ages[4]  
c(second_element, fourth_element)
```

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

The values of the 2nd and 4th elements are 28 and 36, respectively.

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

```
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, 20)
all_but_4th_and_12th <- ages[-c(4, 12)]
all_but_4th_and_12th
```

```
## [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
```

The output is a vector containing all elements except the 4th and 12th elements.

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)
names(x)
```

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

The `names(x)` function returns the names of the elements in the vector `x`.

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

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

The output is a vector containing the values of the elements with names “first” and “third”. The values are 3 and 9, respectively.

b. Write the code and its output.

```
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.

```
x <- -3:2
x
```

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

The output is a sequence of numbers from -3 to 2.

```
x[2] <- 0
x
```

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

The output shows that the 2nd element has been changed to 0.

b. Write the code and its output.

```
x <- -3:2
x

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

```
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)

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

```
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)

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

average_expenditure <- weighted.mean(price_per_liter, purchase_quantity)
average_expenditure
```

```
## [1] 59.2625
```

The output is the average fuel expenditure of Mr. Cruz from Jan to June, which is approximately 59.2625 PhP per liter.

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 <- 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
```

```
## [7]    135.0000    3710.0000
```

b. What are the results?

```
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
## [7]    135.0000    3710.0000
```

The results are:

length(rivers): 141 (the number of elements in the rivers dataset) sum(rivers): 7335 (the sum of the lengths of all rivers) mean(rivers): 52.0234 (the mean length of all rivers) median(rivers): 35 (the median length of all rivers) var(rivers): 441.0167 (the variance of the lengths of all rivers) sd(rivers): 21.34509 (the standard deviation of the lengths of all rivers) min(rivers): 10 (the minimum length of all rivers) max(rivers): 692 (the maximum length of all rivers)

c. 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
## [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. Ranking, Power Celebrity Name, Pay

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

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

```
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", "Pay 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")
annual_pay <- c(67, 90, 225, 110, 90, 332, 302, 41, 52, 88, 55, 44, 55, 40, 233, 34, 40, 47, 75, 25, 39, 31, 45, 32, 40, 31)

power_ranking <- data.frame(ranking, celebrity_name, annual_pay)
power_ranking
```

##	ranking	celebrity_name	annual_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	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.

```
power_ranking$ranking[19] <- 15
power_ranking$annual_pay[19] <- 90

print(power_ranking)
```

##	ranking	celebrity_name	annual_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?
- d. Access the rows 10 to 20 and save it as Ranks.RData. Write the R script and its output.
- e. Describe its output.