

“Worksheet_Bernardo#2”

Nikkko Bryan Bernardo

2022-11-03

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

```
numseq <- -5:5  
numseq
```

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

#The output is a ordered list of numbers from -5 to 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 seq() function a. seq(1, 3, by=0.2) # specify step size Write the R code and its output. Describe the output.

```
seq(1, 3, 0.2)
```

```
## [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 ordered list of numbers but it is by 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?

```
workers <- 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)  
ThirdElement <- c(workers[3])  
ThirdElement
```

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

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

```
SecondFourthElement <- workers[c(2,4)]
SecondFourthElement
```

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

c. Access all but the 1st element is not included. Write the R code and its output

```
AllElement <- workers[c(2:50)]
AllElement
```

```
## [1] 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
## [26] 43 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)
x
```

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

a. Print the results. Then access `x[c("first", "third")]`.
b. Write the code and its output.

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

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

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

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

#Describe the output. - Only first and third element appeared

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

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

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

a. Modify 2nd element and change it to 0;
b. Write the code and its output.

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

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

#Describe the output. - The 2nd element changes from 2 to 0. 6. *The following data shows the diesel fuel purchased by Mr. Cruz. a. Create a data frame for month, price per liter (php) and purchase-quantity (liter).Write the codes.

```
diesel <- data.frame(Month = c("Price per liter(Php)","Purchase-quantity (Liters)"),
  Jan = c("52.50","25"),
  Feb = c("57.25","30"),
  March = c("60.00","40"),
  Apr = c("65.00","50"),
  May = c("74.25","10"),
  June = c("54.00","45"))
diesel
```

```
##               Month   Jan   Feb March   Apr   May   June
## 1   Price per liter(Php) 52.50 57.25 60.00 65.00 74.25 54.00
## 2 Purchase-quantity (Liters)   25   30   40   50   10   45
```

b. What is the average fuel expenditure of Mr. Cruz from Jan to June? Note: Use `weighted.mean(liter, purchase)`

```
purchase <- c(25, 30, 40, 50, 10, 45)
purchase
```

```
## [1] 25 30 40 50 10 45
```

```
liter <- c(52.50, 57.25, 60.00, 65.00, 74.25, 54.00)
liter
```

```
## [1] 52.50 57.25 60.00 65.00 74.25 54.00
```

```
weighted.mean(liter, purchase)
```

```
## [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),
data
```

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

- b. What are the results?
- c. Write the code and its outputs.

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 codes.

```
magazine <- 1:25
CelebName <- 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, 32, 40, 31)
Data_Ranking <- data.frame(magazine, CelebName, pay)
Data_Ranking
```

##	magazine	CelebName	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	31

- b. Modify the power ranking and pay of J.K. Rowling. Change power ranking to 15 and pay to 90.

```
magazine[19] <- 15
magazine
```

```
## [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
```

```
pay [19] <- 90
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
```

```
Magazine_Ranking <- data.frame(magazine, CelebName, pay)
```

```
Magazine_Ranking
```

```
##   magazine      CelebName 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    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      31
```