

RWorksheet_urdas#2

Cindy Urdas

2022-10-11

Using Vectors

1. Create a vector using : operator

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

```
> x <- -5:5
> x
[1] -5 -4 -3 -2 -1 0 1 2 3 4 5
```

The output of this code is the 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
```

the value of x is the numbers 1 to 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.

```
> z <- seq(1, 3, by=0.2)
> print("Specify step size:")
[1] "Specify step size:"
> "Specify step size:"
[1] "Specify step size:"
> z
[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 of this code is the numbers from 1 to 3 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__age[3]
[1] 22
```

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

```
> workers_age[2]
[1] 28
> workers_age[4]
[1] 36
```

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

```
workers_age[-1]
[1] 28 22 36 27 18 52 39 42 29 35 31 27 22 37 34 19 20 57
[19] 49 50 37 46 25 17 37 43 53 41 51 35 24 33 41 53 40 18
[37] 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.

Because I have named the vector of the statement `names(x)` the result of the statement `x <- c("first"=3, "second"=0, "third"=9)`. Is only "first" "second" "third". And when I access `x[c("first", "third")]`. The output includes the numbers 3 and 9.

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.

The output of this code changes the second element's value from -2 to 0.

b. Write the code and its output.

```
> x <- seq(-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 codes.

```

> month<- c("Jan","Feb","March","April","May","June")
> month
[1] "Jan" "Feb" "March" "April" "May" "June"

> PhP<- c(52.50,57.25,60.00,65.00,74.25,54.00)
> PhP
[1] 52.50 57.25 60.00 65.00 74.25 54.00

```

```

> Liters<- c(25,30,40,50,10,45)
> Liters
[1] 25 30 40 50 10 45

```

```

> data.frame(month,PhP,Liters)
  month  PhP Liters
1  Jan 52.50    25
2  Feb 57.25    30
3 March 60.00    40
4 April 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)`

```

> average_fuel <-weighted.mean(PhP,Liters)
> average_fuel
[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))

```

b. What are the results?

```

[1] 141.0000 83357.0000 591.1844 425.0000
[5] 243908.4086 493.8708 135.0000 3710.0000

```

c. Write the code and its outputs.

```

> 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
[5] 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.

a. Create vectors according to the above table. Write the codes.

```
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", "Elthon John", "David Letterman", "Phil Mickelson", "J.K Rowling", "Bradd Pitt", "Peter Jackson", "Dr. phil McGraw", "Jay Lenon", "Celine Dion", "Kobe Bryant")
```

```
> Celebrity_name
[1] "Tom Cruise"           "Rolling Stones"
[3] "Oprah Winfrey"        "U2"
[5] "Tiger Woods"          "Steven Spielberg"
[7] "Howard Stern"         "50 Cent"
[9] "Cast of the Sopranos" "Dan Brown"
[11] "Bruce Springsteen"    "Donald Trump"
[13] "Muhammad Ali"         "Paul McCartney"
[15] "George Lucas"         "Elthon John"
[17] "David Letterman"      "Phil Mickelson"
[19] "J.K Rowling"          "Bradd Pitt"
[21] "Peter Jackson"        "Dr. phil McGraw"
[23] "Jay Lenon"            "Celine Dion"
[25] "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)
> pay
[1] 67 90 225 110 90 332 302 41 52 88 55 44 55
[14] 40 233 34 40 47 75 25 39 45 32 40 31
```

```
> data.frame(power_ranking, Celebrity_name, pay)
  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     Elthon 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

b.Modify the power ranking and pay of J.K. Rowling. Change power ranking to 15 and pay to 90. Write the codes and its output.

```
> power_ranking[19] <- 15
> power_ranking
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
[19] 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
[14] 40 233 34 40 47 90 25 39 45 32 40 31
```

c.Interpret the data.

```
> data.frame(power_ranking,Celebrity_name,pay)
  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   Elthon 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
```