RWorksheet_salve#2PDF

Ben Robert T. Salve.

2022-10-06

#1)

#a.

#R code:

```
operator <- seq(-5, 5)
operator
```

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

#Answer: It displayed the a sequence from -5 to 5.

#b.

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

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

#Answer: The value of x are 1, 2, 3, 4, 5, 6, 7.

#2)

#a.

#R code:

```
seq(1, 3, by=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
```

#Anser: It created a sequence from 1 to 3 by 0.2. It adds 0.2 until it reaches to 3.

#3)

#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,
18)
ages
```

```
##  [1] 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
## [26] 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
```

#Anwser: The value of the 3rd element is 22.

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

```
ages[2]
```

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

```
ages[4]
```

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

#Anwser: The values of the 2nd and 4th element are 28 and 36.

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

#R code:

```
ages
```

```
##  [1] 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
## [26] 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[2:50]
```

```
##  [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)

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

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

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

```
print(x)
```

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

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

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

#Answer: After print, it showed a table which has the first, second, third below it are the values assigned to the characters. Next is the same table but the second is now gone.

#b. Write the code and its output.

#R codes:

```
print(x)
```

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

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

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

#5)

```
x <- seq(-3:2)
x
```

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

#a. Modify 2nd element and change it to 0.

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

```
## [1] 1 0 3 4 5 6
```

#Describe the output.

#Answer: The sequence was 1 to 6 after changing x [2] into 0 instead of the # 1 2 3 4 5 6 sequence now it is 1 0 3 4 5 6. 2 is now replaced by 0.

#b. Write the code and its output.

#R code:

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

```
## [1] 1 0 3 4 5 6
```

#6)

#a. Create a data frame for month, price per liter (php) and purchase-quantity (liter). Write the codes.

#Data frame and codes:

```
Month <- c("Jan", "Feb", "March", "Apr", "May", "June")
Price_per_liter_php <- c(52.50, 57.25, 60.00, 65.00, 74.25, 54.00)
Purchase_quantity_liter <- c(25, 30, 40, 50, 10, 45)

data_frame <- data.frame(Month, Price_per_liter_php, Purchase_quantity_liter )
data_frame
```

```
##   Month Price_per_liter_php Purchase_quantity_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)

#R codes:

```
ave_fuel <- weighted.mean(Price_per_liter_php, Purchase_quantity_liter)
ave_fuel
```

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

#Answer: The average fuel expenditure of Mr. Cruz form Jan to June is [1] 59.2625.

#7)

#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
```

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

#b. What are the results?

#Answer: It gave me the length, sum, mean, median, var, sd, min, max of the rivers data.

#c. Write the code and its outputs.

#R codes:

```
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
```

#8) #a. Create vectors according to the above table. Write the codes.

#R codes:

```
Power_Ranking <- 1:25
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
```

#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.

#R codes:

```
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 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
```

```
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            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 Bryant  31
```

#c. Interpret the data.

#Answer: After successfully changing the Power_Ranking using this code Power_Ranking [19] <- 15. I couldn't understand why I can't change the Pay. Until I notice that [75] is a position their are only 25 who are in the rank not 75 so I change the Pay [75] <- 90 into Pay [19] <- 90 then I was able to change it without getting NAs.