WorkSheets#2

John Rick Estoce

2025-09-29

#Worksheet#2 ##1. Create a vector using : operator a. Sequence from -5 to 5. Write the R code and its output.

```r
vec <- c(-5:5)
vec
```

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

output [1] -5 -4 -3 -2 -1 0 1 2 3 4 5

```r
 x <- c(1:7)
x
```

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

The value of x is [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.

```r
sequence <- seq(1, 3, by = 0.2)
sequence
```

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

Each value is evenly spaced by 0.2. Output: 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8 3.0

##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.

```r
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[3]
```

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

```r
ages[c(2,4)]
```

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

```r
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
```

output a. [1] 22 b. [1] 28 36 c. [1] [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 43 53 41 51 35 24 33 41 [34] 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). a. Print the results. Then access x[c("first", "third")]. Describe the output.

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

print(x)
```

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

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

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

b. Write the code and its output. first second third 3 0 9

first third 3 9

##5. Create a sequence x from -3:2. a. Modify 2nd element and change it to 0; b. Write the code and its output.

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

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

1st output[1] -3 -2 -1 0 1 2 2nd output: [1] -3 0 -1 0 1 2 The second element changed to zero after initializing the second element.

##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

```r
#a  Create a data frame for month, price per liter (php) and purchase-quantity (liter).
month <- c("Jan", "Feb", "Mar", "Apr", "May", "Jun")
price <- 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, price, liter)

fuel_data
```

```
##   month price liter
## 1   Jan 52.50    25
## 2   Feb 57.25    30
## 3   Mar 60.00    40
## 4   Apr 65.00    50
## 5   May 74.25    10
## 6   Jun 54.00    45
```

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

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

Output: a. month price liter 1 Jan 52.50 25 2 Feb 57.25 30 3 Mar 60.00 40 4 Apr 65.00 50 5 May 74.25 10 6 Jun 54.00 45

b. [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). b. what's the output? c. Code and its output

```r
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
```

```r
#b. What's the output?
# length    sum        mean  median  variance     sd      min     max
# 141.0   591952.0    591.2   425.0  98382.96   313.83   135.0  3710.0


data <- c(
  length = length(rivers),
  sum = sum(rivers),
  mean = mean(rivers),
  median = median(rivers),
  variance = var(rivers),
  sd = sd(rivers),
  min = min(rivers),
  max = max(rivers)
)

data
```

```
##       length          sum          mean        median     variance           sd
##     141.0000   83357.0000      591.1844      425.0000  243908.4086      493.8708
##          min          max
##     135.0000    3710.0000
```

```r
# length    sum        mean  median  variance     sd      min     max
# 141.0   591952.0    591.2   425.0  98382.96   313.83   135.0  3710.0
```

##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.

```r
#a

power_rank <- 1:25

celebrity <- 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", "Brad Pitt", "Peter Jackson",
  "Dr. Phil McGraw", "Jay Leno", "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 <- data.frame(power_rank, celebrity, pay)
forbes
```

```
##    power_rank            celebrity 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

jk_index <- which(forbes$celebrity == "J.K. Rowling")
forbes$power_rank[jk_index] <- 15
forbes$pay[jk_index] <- 90

forbes[jk_index, ]
```

```
##    power_rank    celebrity pay
## 19         15 J.K. Rowling  90
```

```
#c.Interpret the data
#The outpot shows celebrities ranked by power and pay. After updating, J.K. Rowling is ranked 15th with
```