Work_Sheet #2

Syndric James Z. Junsay

2025-09-29

Exercise 1 Create a vector using : operator

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

```
y <- c(-5:5)
print(y)
```

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

#The output generates a sequence of integers starting from -5 and increasing by 1 up to 5.

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

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

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

The value of x is: [1] 1 2 3 4 5 6 7 ## Exercise 2 Create a vector using seq() function #a. seq(1, 3, by=0.2) # specify step size

```
vector <- seq(1, 3, by=0.2)
print (vector)
```

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

output: [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 generates a numeric sequence starting at 1 and ending at 3, incrementing by 0.2, resulting in the values 1.0, 1.2, 1.4, up to 3.0.

exercise 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?

```
ages <- c(4, 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)
thirdage <- ages[3]
print (thirdage)
```

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

the value is [1] 22

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

```
ages <- c(4, 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)
second_and_fourth <- ages[c(2,4)]
print (second_and_fourth)
```

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

the values of the 2nd & 4th elements are [1] 28 36

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

```
ages <- c(4, 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)
all_but_one <- ages[c(2:50)]
print (all_but_one)
```

```
##  [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
```

Rcode: ages <- c(4, 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) all_but_one <- ages[c(2:50)] print (all_but_one)

Output:[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 [28] 41 51 35 24 33 41 53 40 18 44 38 41 48 27 39 19 30 61 54 58 26 18

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

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

the output prints out the values of the names “first” & “third”.

b. code: x <- c(3, 0, 9) names(x) <- c(“first”, “second”, “third”) x[c(“first”, “third”)] output: first third 3 9

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

a. Modify 2nd element and change it to 0; x[2] <- 0 x

Describe the output.

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

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

the output shows that instead of -2 the value was changed to 0

b. code: x <- c(-3:2) x[2] <- 0 x output: [1] -3 0 -1 0 1 2

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

```r
month <- c("Jan", "Feb", "March", "Apr", "May", "June")
price <-c(52.50, 57.25, 60.00, 65.00, 74.25, 54.00)
purchase_Quantity <-c(25, 30, 40, 50, 10, 45)

fuel_purchase <- data.frame(month, price, purchase_Quantity)
fuel_purchase
```

```
##   month price 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)

```
average_fuel <- weighted.mean(price, purchase_Quantity)
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)

```
data <- c(length(rivers), sum(rivers), mean(rivers), median(rivers), var(rivers), sd(rivers), min(river
data
```

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

b. What are the results? [1] 141.0000 83357.0000 591.1844 425.0000 243908.4086 [6] 493.8708 135.0000 3710.0000

c. Write the code and its outputs. code: data <- c(length(rivers), sum(rivers), mean(rivers), median(rivers), var(rivers), sd(rivers), min(rivers), max(rivers)) data outputs: [1] 141.0000 83357.0000 591.1844 425.0000 243908.4086 [6] 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.

```
power_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",
                    "Paul McCartney", "Elon Musk", "Elton John", "David Letterman",
                    "Phil Mickelson", "J.K. Rowling", "Brad Pitt", "Peter Jackson",
                    "Dr. Phil McGraw", "Jay Leno", "Celine Dion", "Kobe Bryant")

# Pay in million USD
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)
celebrities <- data.frame(power_ranking, celebrity_name, pay)
celebrities
```

```
##    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            Elon Musk 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 codes and its output.

```
index <- which(celebrity_name == "J.K. Rowling")

# Modify power ranking and pay
power_ranking[index] <- 15
pay[index] <- 90

celebrity_name[index]
```

```
## [1] "J.K. Rowling"
```

```
power_ranking[index]
```

```
## [1] 15
```

```
pay[index]
```

```
## [1] 90
```

c. Interpret the data.

The data shows the top 25 highest-paid or most powerful celebrities according to Forbes. power_ranking indicates their influence, while pay indicates their annual earnings in million USD.

Example interpretations:

Highest-paid celebrity: Steven Spielberg ($332M) Celebrity with lowest pay: Brad Pitt ($25M)

J.K. Rowling's new ranking and pay indicate a significant increase in both influence and earnings.

This data set can be used to analyze trends between pay and influence, e.g., do higher pay always mean higher ranking?