Lab Exercise #1

Lysyl Bess H. Talaban

2025-09-29

```
#1a.

vector1 <- c (-5:5)
vector1
```

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

```
#the output are numbers from -5 to 5

#1b.
x <- 1:7
x
```

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

```
#the output are numbers from 1 to 7
```

```
#2
seq1 <- seq(1, 3, by=0.2)
seq1
```

```
##  [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 sequence of numbers from 1 to 3 in steps of 0.2
```

```
#3
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)

#a. Access 3rd element, what is the value?
ages[3]
```

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

```
#the value is 22

#b. Access 2nd and 4th element, what are the values?
ages[c(2,4)]
```

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

```
#the values are 28 and 36

#c. Access all but the 1st element is not included. Write the R code and its output.
ages[c(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
#> # 3b.
#  > ages[c(2, 4)]
#[1] 28 36
#> # the values are 28 and 36
#  >
#  > # 3c.
#  > ages[c(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 43 53 41 51 35 24 33 41
#[39] 41 48 27 39 19 30 61 54 58 26 18
```

```
# 4.
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")]. Describe the output.
x[c("first", "third")]
```

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

```
# the output is displays a 2x2 table with the corresponding columns and rows

# b. Write the code and its output.
# OUTPUT
#x <- c("first"=3, "second"=0, "third"=9)
#> x
#first second  third
#3      0      9
#>
#  > # 4a.
#  > x[c("first", "third")]
#first third
#3     9
#> # the output is displays a 2x2 table with the corresponding columns and rows
#  >
#  > # 4b.
#  > x
#first second  third
#3      0      9
```

```
# 5. Create a sequence x from -3:2.
x <- seq(-3:2)

# a. Modify 2nd element and change it to 0; Describe the output.
x[2] <- 0
x
```

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

```
# 5b. Write the code and its output.
# OUTPUT
```

```
#> x[2] <- 0
#> x
#[1] 1 0 3 4 5 6
```

```
# 6.
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_Liters <- c(25, 30, 40, 50, 10, 45)

# a. Create a data frame for month, price per liter (php) and purchase-quantity (liter). Write the code
df <- data.frame(
  Month = Month,
  Price_per_liter_PhP = Price_per_liter_PhP,
  Purchase_quantity_Liters = Purchase_quantity_Liters
)

df
```

```
##   Month Price_per_liter_PhP Purchase_quantity_Liters
## 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, 
weighted.mean(Price_per_liter_PhP, Purchase_quantity_Liters)
```

```
## [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?
# the result are 141, 83357, 591.1844, 425, 243908, 493, 135,  and 3710 correspondingly

# 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 <- 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    135.0000   3710.0000
```

```
# 8.
# a. Create vectors according to the above table. Write the codes.
Power_Ranking <- c(1:25)
```

```r
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_Millions <- 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
)

# b. Modify the power ranking and pay of J.K. Rowling. Change power ranking to 15 and pay to 90. Write
Power_Ranking[19] <- 15
Pay_Millions[19] <- 90


# c. Interpret the data.

Modified_Data <- data.frame(
  Ranking = Power_Ranking,
  Celebrity = Celebrity_Name,
  Pay = Pay_Millions)

Modified_Data
```

```
##    Ranking            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      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
```