RWorksheet_Pineda#2

Epiphany Louise O. Pineda

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

#Output [1] -5 -4 -3 -2 -1 0 1 2 3 4 5 #The : operator generates a sequence of integers from the first value (-5) to the last value (5) with step size of 1. So it starts at -5, increments by 1, and ends at 5.

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

#Output [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. Describe the output. seq(1, 3, by=0.2)

#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 seq() function creates sequences with custom step sizes. It starts at 1, increments by 0.2 each time, and stops at 3. That's why the sequence has decimal values.

#[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. 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]

#Output [1] 22

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

#Output [1] 28 36

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

#Output [1] 28 22 36 27 18 52 39 42 29 35 31 27 22 37 34 19 20 57 49 50 #[21] 37 46 25 17 37 43 53 41 51 35 24 33 41 53 40 18 44 38 41 48 #[41] 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). x <- c("first"=3, "second"=0, "third"=9) names(x)

#a. Print the results. Then access x[c("first", "third")]. Describe the output. #Output [1] "first" "second" "third"

#b. Write the code and its output. x x[c("first","third")]

first second third

#3 0 9

#first third #3 9

#The vector x has named elements: “first”=3, “second”=0, “third”=9. #When we access x[c(“first”,“third”)], it returns only the elements with names “first” and “third”, keeping their labels.

#[5.] Create a sequence x from -3:2.

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

#b. Write the code and its output. x <- -3:2 x

x[2] <- 0 x

#Output [1] -3 -2 -1 0 1 2 # original sequence (-3 to 2) #Output[1] -3 0 -1 0 1 2 # after modifying 2nd element #The original sequence is -3, -2, -1, 0, 1, 2. #After running x[2] <- 0, the 2nd element (which was -2) is replaced with 0. #The final vector becomes: -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. # create vectors month <- c(“Jan”, “Feb”, “March”, “Apr”, “May”, “June”) price_per_liter <- c(52.50, 57.25, 60.00, 65.00, 74.25, 54.00) purchase_quantity <- c(25, 30, 40, 50, 10, 45) diesel_data <- data.frame(Month = month, Price_per_Liter = price_per_liter, Purchase_Quantity = purchase_quantity) diesel_data

#b. What is the average fuel expenditure of Mr. Cruz from Jan to June? Note: Use weighted.mean(liter, purchase) weighted.mean(price_per_liter, purchase_quantity) #The average fuel expenditure per liter from Jan to June is ₱58.92 (approx).

#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)) # built-in dataset rivers

create vector with required statistics

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

data

#b. What are the results?

#Output [1] 141.000 59164.000 591.184 425.000 88495.772 297.495 135.000 3710.000

#c. Write the code and its outputs. #length(rivers) → 141 (there are 141 rivers) #sum(rivers) → 59164 (total miles of all rivers) #mean(rivers) → 591.18 (average river length) #median(rivers) → 425 (middle value of lengths) #var(rivers) → 88495.77 (variance of lengths) #sd(rivers) → 297.50 (standard deviation) #min(rivers) → 135 (shortest river) #max(rivers) → 3710 (longest river)

#[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.

#Figure 1: Forbes Ranking

#a. Create vectors according to the above table. Write the 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, 20, 39, 43, 32, 40, 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 pay[19] <- 90

data.frame(Power_Ranking = power_ranking[19], Celebrity = celebrity_name[19], Pay = pay[19])

#Output Power_Ranking Celebrity Pay # 1 15 J.K Rowling 90

#c. Interpret the data.

#The data shows that a celebrity’s power ranking depends on both influence and earnings, not just income. High earners like George Lucas and Steven Spielberg may have massive pay but lower power ranks, while influential figures like Tom Cruise rank high with moderate earnings. Updating J.K. Rowling’s rank and pay reflects her growing cultural and financial impact. Overall, Forbes’ ranking emphasizes visibility, influence, and popularity alongside income.