Challenge-3

Chen Junyu Ryan

2023-08-30

I. Questions

Question 1: Emoji Expressions Imagine you're analyzing social media posts for sentiment analysis. If you were to create a variable named "postSentiment" to store the sentiment of a post using emojis (for positive, for neutral, for negative), what data type would you assign to this variable? Why? (*narrative type question, no code required*)

Solution: Emojis are a string of character hence i would assign character to the variable 'character'

Question 2: Hashtag Havoc In a study on trending hashtags, you want to store the list of hashtags associated with a post. What data type would you choose for the variable "postHashtags"? How might this data type help you analyze and categorize the hashtags later? (*narrative type question, no code required*)

Solution: Character, numebr of frequency of word would allow for identifying the most trenging hastag 'character string'

Question 3: Time Traveler's Log You're examining the timing of user interactions on a website. Would you use a numeric or non-numeric data type to represent the timestamp of each interaction? Explain your choice (*narrative type question, no code required*)

Solution: Numeric Value as it would be an integer value
' Integer '

Question 4: Event Elegance You're managing an event database that includes the date and time of each session. What data type(s) would you use to represent the session date and time? (*narrative type question, no code required*)

Solution: to represent the date i would use character string and to represent time i would use integer 'character string' & 'integer'

Question 5: Nominee Nominations You're analyzing nominations for an online award. Each participant can nominate multiple candidates. What data type would be suitable for storing the list of nominated candidates for each participant? (*narrative type question, no code required*)

Solution: 'Character string'

Question 6: Communication Channels In a survey about preferred communication channels, respondents choose from options like "email," "phone," or "social media." What data type would you assign to the variable "preferredChannel"? (*narrative type question, no code required*)

Solution: 'Character string'

Question 7: Colorful Commentary In a design feedback survey, participants are asked to describe their feelings about a website using color names (e.g., "warm red," "cool blue"). What data type would you choose for the variable "feedbackColor"? (*narrative type question, no code required*)

Solution: 'character string'

Question 8: Variable Exploration Imagine you're conducting a study on social media usage. Identify three variables related to this study, and specify their data types in R. Classify each variable as either numeric or non-numeric.

Solution: 3 variables is number of post , number of likes and comments on post number of post - integer numberic number of likes - integer numeric comments on post - character 'string' non numeric

Question 9: Vector Variety Create a numeric vector named "ages" containing the ages of five people: 25, 30, 22, 28, and 33. Print the vector.

Solution:

```
ages <- c(22,25,228,30,33)

print(ages)
```

```
## [1]  22  25 228  30  33
```

Question 10: List Logic Construct a list named "student_info" that contains the following elements:

- A character vector of student names: "Alice," "Bob," "Catherine"
- A numeric vector of their respective scores: 85, 92, 78
- A logical vector indicating if they passed the exam: TRUE, TRUE, FALSE

Print the list.

Solution:

```
student_info <- list(Student.names= c("Alice","Bob","Catherine"), Scores= c(85,92,78), Results= c(TRUE,
print(student_info)
```

```
## $Student.names
## [1] "Alice"     "Bob"       "Catherine"
##
## $Scores
## [1] 85 92 78
##
## $Results
## [1]  TRUE  TRUE FALSE
```

Question 11: Type Tracking You have a vector "data" containing the values 10, 15.5, "20", and TRUE. Determine the data types of each element using the typeof() function.

Solution:

```
data <- c(10,15.5,"20",TRUE)
print(data)
```

```
## [1] "10"   "15.5" "20"   "TRUE"
```

```
print(typeof(data[1]))
```

```
## [1] "character"
```

```
print(typeof(data[3]))
```

```
## [1] "character"
```

```
print(typeof(data[4]))
```

```
## [1] "character"
```

```
# if you notice although 10 and 15.5 are double "20" is a character so the whole vector becomes charact
```

Question 12: Coercion Chronicles You have a numeric vector "prices" with values 20.5, 15, and "25". Use explicit coercion to convert the last element to a numeric data type. Print the updated vector.

Solution:

```
prices <- c(20.5,15,"25")
prices_numeric <- as.numeric(prices)
typeof(prices_numeric[3])
```

```
## [1] "double"
```

```
print(prices_numeric)
```

```
## [1] 20.5 15.0 25.0
```

```
# vector can only contain one variable type, so u can't just change one element of the variable to becom
```

Question 13: Implicit Intuition Combine the numeric vector c(5, 10, 15) with the character vector c("apple", "banana", "cherry"). What happens to the data types of the combined vector? Explain the concept of implicit coercion.

Solution:

```
x <- c(5,10,15)
y <- c("apple","banana","cherry")
z <- c(x,y)
typeof(x)
```

```
## [1] "double"
```

```
typeof(y)
```

```
## [1] "character"
```

```
typeof(z)
```

```
## [1] "character"
```

```
# z <- (x,y) also works
# An implicit coercion is an automatic conversion of values from one datatype to another
```

Question 14: Coercion Challenges You have a vector "numbers" with values 7, 12.5, and "15.7". Calculate the sum of these numbers. Will R automatically handle the data type conversion? If not, how would you handle it?

Solution: 'R will not handle the data type conversion, i would use explicit coercion to change '15.7' into a double'

```
numbers <- c(7,12.5,"15.7")
"unable to add sum due to 15.7 being a character"
```

```
## [1] "unable to add sum due to 15.7 being a character"
```

```
print(numbers)
```

```
## [1] "7"    "12.5" "15.7"
```

```
numbers <- as.numeric(numbers)
sum(numbers)
```

```
## [1] 35.2
```

Question 15: Coercion Consequences Suppose you want to calculate the average of a vector "grades" with values 85, 90.5, and "75.2". If you directly calculate the mean using the mean() function, what result do you expect? How might you ensure accurate calculation?

Solution: you would expect it to fail because

```
grades <- c(85,90.5,"75.2")
grades <- as.numeric(grades)
mean(grades)
```

```
## [1] 83.56667
```

Question 16: Data Diversity in Lists Create a list named "mixed_data" with the following components:

- A numeric vector: 10, 20, 30
- A character vector: "red", "green", "blue"
- A logical vector: TRUE, FALSE, TRUE

Calculate the mean of the numeric vector within the list.

Solution:

```
mixed_data <- list(A= c(10,20,30), B= c("red","green","blue"), D= c(TRUE, FALSE, TRUE))
print(mixed_data)
names(mixed_data)
'list can contain multiple vectors with diff variable thats the useful part so i can just filter the li
mean(mixed_data$A)
```

Question 17: List Logic Follow-up Using the "student_info" list from Question 10, extract and print the score of the student named "Bob."

Solution:

```
print(student_info)
```

```
## $Student.names
## [1] "Alice"     "Bob"       "Catherine"
##
## $Scores
## [1] 85 92 78
##
## $Results
## [1]  TRUE  TRUE FALSE
```

```
which(student_info$Student.names == "Bob") # this gives me the place of where bob name is at
```

```
## [1] 2
```

```
print(student_info$Scores[which(student_info$Student.names == "Bob")])
```

```
## [1] 92
```

```
# we are trying to automate the process and not manually finding, so bobs score is linked to the placem
# indices are

# vector contains variables of only one data type, list contains vectors with variables that can be of
# indices are the positions of the vector
```

Question 18: Dynamic Access Create a numeric vector values with random values. Write R code to dynamically access and print the last element of the vector, regardless of its length.

Solution:

```r
set.seed(123)
q <- sample(-5:20, size = 5)
print(q)
```

```
## [1]  9 13  8 -3  4
```

```r
q[length(q)]
```

```
## [1] 4
```

Question 19: Multiple Matches You have a character vector words <- c("apple", "banana", "cherry", "apple"). Write R code to find and print the indices of all occurrences of the word "apple."

Solution:

```r
words <- c("apple","banana","cherry","apple")
indices <- which(words == "apple")
print(indices)
```

```
## [1] 1 4
```

Question 20: Conditional Capture Assume you have a vector ages containing the ages of individuals. Write R code to extract and print the ages of individuals who are older than 30.

Solution:

```r
ages <- c(25,42,19,36,51,29,33,28,40)
older_than_30 <- ages[ages > 30]
print(older_than_30)
```

```
## [1] 42 36 51 33 40
```

Question 21: Extract Every Nth Given a numeric vector sequence <- 1:20, write R code to extract and print every third element of the vector.

Solution:

```r
sequence <- 1:20
print(sequence)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

```r
every_third <- sequence[seq(from = 3, to = length(sequence), by = 3)]
print(every_third)
```

```
## [1]  3  6  9 12 15 18
```

Question 22: Range Retrieval Create a numeric vector numbers with values from 1 to 10. Write R code to extract and print the values between the fourth and eighth elements.

Solution:

```
Range <- 1:10
print(Range)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
fourth_to_eigth <- Range[4:8]
print(fourth_to_eigth)
```

```
## [1] 4 5 6 7 8
```

Question 23: Missing Matters Suppose you have a numeric vector data <- c(10, NA, 15, 20). Write R code to check if the second element of the vector is missing (NA).

Solution:

```
# hint use is.na
data <- c(10, NA, 15, 20)
missing_element <- is.na(data[2])
print(missing_element)
```

```
## [1] TRUE
```

Question 24: Temperature Extremes Assume you have a numeric vector temperatures with daily temperatures. Create a logical vector hot_days that flags days with temperatures above 90 degrees Fahrenheit. Print the total number of hot days.

Solution:

```
daily_temperatures <- c(50,60,70,80,90,100,110)
print(daily_temperatures)
```

```
## [1]  50  60  70  80  90 100 110
```

```
hot_days <- daily_temperatures[daily_temperatures>90]
print(length(hot_days))
```

```
## [1] 2
```

```
# use length
```

Question 25: String Selection Given a character vector fruits containing fruit names, create a logical vector long_names that identifies fruits with names longer than 6 characters. Print the long fruit names.

Solution:

```
fruits <- c("apple", "banana", "watermelon", "pineapple")
long_names <- nchar(fruits) > 6
print(long_names)
```

```
## [1] FALSE FALSE  TRUE  TRUE
```

```r
#' this prints out which one is more than 6 but it doesnt print out the names '
long_fruit_names <- fruits[long_names]
print(long_fruit_names)
```

```
## [1] "watermelon" "pineapple"
```

Question 26: Data Divisibility Given a numeric vector numbers, create a logical vector divisible_by_5 to indicate numbers that are divisible by 5. Print the numbers that satisfy this condition.

Solution:

```r
numbers <- c(10,12,2,1,8,9,15,25)
divisible_by_5 <- numbers %% 5 == 0
# %% will show the remainder so 9 %% 4 will give me the remainder of 9 divided by 4 which is 1 so by sa

# == compares every single element in the variable of the vector
divisible_numbers <- numbers[divisible_by_5]
print(divisible_numbers)
```

```
## [1] 10 15 25
```

Question 27: Bigger or Smaller? You have two numeric vectors vector1 and vector2. Create a logical vector comparison to indicate whether each element in vector1 is greater than the corresponding element in vector2. Print the comparison results.

Solution:

```r
vector1 <- c(1,2,3,4,11)
vector2 <- c(5,7,8,9,10)
comparison <- vector1 > vector2
print(comparison)
```

```
## [1] FALSE FALSE FALSE FALSE  TRUE
```