

Challenge-3

Wang Renhe

28-08-2023

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: I will assign "character" to this variable, because the variable "postSentiment" falls within the realm of ordinal variables, the ratings of sentiments have natural order.

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: I will use "character" for the variable "postHashtags". For example, if student council in NUS wants to identify the most popular topic among students, they only need to search for a few words to identify which set of hashtags has been used the most. It helps to summarise the accumulative and repetitive data, so that it is easier to analyse the data.

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:

I will use numeric data type to represent the timestamp, to be more specific, it is the type "double". Because time includes specific number(integer) and possible decimal places.

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: I will assign numeric data type to it, specifically "double".

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: I think the list of nominated candidates fall under "character". Because each candidate has a name(non-numeric), but there is no natural ordering in the name list.

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: I will assign “character” to the variable “preferredChannel”.

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: I will choose “character” data type for the survey. The color name itself cannot change into numbers, and different colors represent different emotions among participants when they surf online. For example, warm red means excited, and cool blue means sad.

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:

variable1: average time spend on social media platform (numeric) variable2: number of friends on social media (numeric) variable3: username (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:

```
# Enter code here
ages<-c(25,30,22,28,33)
print(ages)
```

```
## [1] 25 30 22 28 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:

```
# Enter code here
student_info <- list(
  names = c("Alice", "Bob", "Catherine"),
  scores = c(85, 92, 78),
  pass_exam = c(TRUE, TRUE, FALSE)
)
student_info
```

```
## $names
## [1] "Alice"      "Bob"        "Catherine"
##
## $scores
## [1] 85 92 78
##
## $pass_exam
## [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:

```
# Enter code here
data=c(10,15.5,"20",TRUE)
typeof(data[1])
```

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

```
typeof(data[2])
```

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

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

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

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

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

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:

```
# Enter code here
prices <- c(20.5,15,"25")
prices <- as.numeric(prices)
print(prices)
```

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

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:

```
# Enter code here
x <- c(5,10,15)
y <- c("apple", "banana", "cherry")
combined_vector <- c(x,y)
typeof(x[1])
```

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

```
typeof(y[1])
```

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

```
typeof(combined_vector[1])
```

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

what happen: when I combined one numeric and one character vector, the type will automatically change to character, because all elements inside one vector must be the same data type.

implicit coercion: When R carries out the conversion of a vector from one data type to another based on our inputs

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 automatically handle the data type conversion, the result is uncomputable

```
# Enter code here
numbers <- c(7, 12.5, "15.7")
numbers <- as.numeric(numbers)
sum_of_numbers <- sum(numbers)
print(sum_of_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:

```
# Enter code here
grades <- c(85, 90.5, "75.2")
grades <- as.numeric(grades)
average <- mean(grades)
print(average)
```

```
## [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:

```
# Enter code here
mixed_data <- list(
  numeric_vector = c(10, 20, 30),
  character_vector = c("red", "green", "blue"),
  logical_vector = c(TRUE, FALSE, TRUE)
)

mean_numeric <- mean(mixed_data$numeric_vector)
print(mean_numeric)
```

```
## [1] 20
```

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:

```
# Enter code here
score_of_Bob <- student_info$scores[student_info$names == "Bob"]
print(score_of_Bob)
```

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

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:

```
# Enter code here
x<-c(1,3,5,7,9)
x[length(x)]
```

```
## [1] 9
```

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:

```
# Enter code here
apple_indices <- c("apple", "banana", "cherry", "apple")
apple_indices <- which(apple_indices == "apple")
print(apple_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:

```
# Enter code here
ages <- c(22, 47, 15, 33, 27, 42)
older_than_30 <- ages[ages > 30]
print(older_than_30)
```

```
## [1] 47 33 42
```

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:

```
# Enter code here
sequence1 <- 1:20
every_third_element <- sequence1[seq(from = 3, to = 20, by = 3)]
print(every_third_element)
```

```
## [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:

```
# Enter code here
x <- 1:10
x[4:8]
```

```
## [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:

```
# Enter code here
data <- c(10, NA, 15, 20)
missing <- is.na(data)
print(missing)
```

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

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:

```
# Enter code here
temperatures <- c(85, 66, 94, 55, 87, 99, 72, 96)
hot_days <- temperatures > 90
total_hot_days <- sum(hot_days)
print(total_hot_days)
```

```
## [1] 3
```

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:

```
# Enter code here
fruits <- c("apple", "banana", "watermelon", "pinapple")
long_names <- nchar(fruits) > 6
print(fruits[long_names])
```

```
## [1] "watermelon" "pinapple"
```

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:

```
# Enter code here
numbers <- c(5, 10, 15, 20, 25, 30, 35, 66)
divisible_by_5 <- numbers %% 5 == 0
print(numbers[divisible_by_5])
```

```
## [1] 5 10 15 20 25 30 35
```

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:

```
# Enter code here
vector1 <- c(20, 25, 14, 44)
vector2 <- c(18, 19, 23, 42)
comparison <- vector1 > vector2
print(comparison)
```

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