

Challenge-3

Nicole Lim

2023-11-23

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, what data type would you assign to this variable? Why? (*narrative type question, no code required*)

Solution: *Character. It's not a number*

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. The hashtags are non-numerical as well. The categorization can be done by types of hashtags (by their contents), while analysis can be done by length of hashtags.*

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. This would allow us to leverage on the inherently ordinal and numerical nature of the timestamps and allow us sort interactions from earliest to latest or group them based on timings.*

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: *The data type can be POSIXlt.*

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: *Characters if the nominated candidates are to be represented as names. Integers if they are to be represented as numbers (candidate 1,2, etc.)*

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*

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*

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: *Type of social media: Character, duration of social media platform usage in hours: integer, opinion towards their amount of social media usage: character*

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(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
n <- c("Alice", "Bob", "Catherine")
s <- c(85, 92, 78)
p <- c(TRUE, TRUE, FALSE)
studentlist = list(names=n, scores=s, pass=p)
print(studentlist)
```

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

```
prices4 <- c(20.5, 15, "25")
#prices4 <- c("25")
prices4 <- as.numeric(prices4)
print(prices4)
```

```
## [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
```

```
combined <- c(5, 10, 15, "apple", "banana", "cherry")
typeof(combined)
```

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

```
print("Implicit Coercion is where R infers and assigns a type to a vector based on the program's own in
```

```
## [1] "Implicit Coercion is where R infers and assigns a type to a vector based on the program's own in
```

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:

```
# Enter code here
numbers <- c(7, 12.5, "15.7")
print("R will not automatically handle the data type conversion. It gives the error that there is an in
```

```
## [1] "R will not automatically handle the data type conversion. It gives the error that there is an in
```

```
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:

```
# Enter code here
grades <- c(85, 90.5, "75.2")
print("Directly calculating the mean would give an error as the character data type contained is incom
```

```
## [1] "Directly calculating the mean would give an error as the character data type contained is incom
```

```
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:

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

mixed_data = list(numeric_vector, character_vector, logical_vector)
mean(mixed_data[[1]])
```

```
## [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
studentlist
```

```
## $names
## [1] "Alice"      "Bob"        "Catherine"
##
## $scores
## [1] 85 92 78
##
## $pass
## [1] TRUE TRUE FALSE
```

```
print(studentlist)
```

```
## $names
## [1] "Alice"      "Bob"        "Catherine"
##
## $scores
## [1] 85 92 78
##
## $pass
## [1] TRUE TRUE FALSE
```

```
print(studentlist[[2]][2])
```

```
## [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
values <- c(runif(10,1,100))
tail(values,1)
```

```
## [1] 85.01973
```

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

```
## [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(12, 43, 45, 73, 11, 9)
ages[ages>30]
```

```
## [1] 43 45 73
```

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
every_third <- seq(from=1, to=20, by=3)
print(every_third)
```

```
## [1] 1 4 7 10 13 16 19
```

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
numbers <- c(1:10)
print(numbers[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)
check <- c(is.na(data))
print(check)
```

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

```
print("Yes, the second element is missing.")
```

```
## [1] "Yes, the second element is missing."
```

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(90, 92, 47, 100, 49)
hot_days <- c(temperatures>90)
print(sum(hot_days == TRUE))
```

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

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("pomegranate", "apple", "kiwi", "blueberry")
longnames <- c(nchar(fruits)>6)
print(fruits[c(longnames)])
```

```
## [1] "pomegranate" "blueberry"
```

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
fives <- c(5, 10, 24, 59)
divisible_by_5 <- fives%%5==0
print(fives[c(divisible_by_5)])
```

```
## [1] 5 10
```

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(12, 43, 30)
vector2 <- c(24, 12, 33)
comparison <- c(vector1>vector2)
print(comparison)
```

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