Challenge-3

Phua Zong Yao

2023-08-28

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 would assign the data type 'character' to the variable 'postSentiment'. In this case, emojis are considered to be textual characters. Using the 'character' data type allows me to easily store and move the emojis as if they were groups of text.

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 would use the 'character' data type for the variable 'postHashtags'. Using this data type allows me to store many hashtags for each post as elements within a single list. Using this data type also allows for categorization by looking for recurring words or phrases. By going through the list, patterns can be identified which make it easier to group and later analyse posts.

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 would use a non-numeric data type POSIXct.This data type supports the computation of time for various interval periods.

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 would use a non-numeric data type POSIXct for the time variable. For the date variable, 'Date' can be used.

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: A 'character' data type would be suitable for storing the list of nominated candidates for each participant. The names of nominated candidates consist of non-numeric string of characters, which would make a numeric data type unsuitable.

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 would assign the ‘character’ data type 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 would assign the ‘character’ data type to the variable ‘feedbackColor’.

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:

Variable 1: Age Data Type: Numeric (Integer)

Age is a numeric value, used to measure the age of a respondent. To ensure consistent and easy analysis, an integer value should be used, measuring the respondent’s age in years.

Variable 2: PostsPerDay Data Type: Numeric (Double)

PostsPerDay is a numeric value, used to measure the average number of posts made by a respondent on a daily basis. Since it is an average value, it can and most likely will include decimal values. Thus, an integer would not be suitable and a double should be used instead.

Variable 3: TimeSpent Data Type: Numeric (Double)

TimeSpent is a numeric value, used to measure the average time spent on various social media platforms on a daily basis. Since it is an average value, it can and most likely will include decimal values. Thus, an integer would not be suitable and a double should be used instead.

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:

```r
# 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(studentname=c("Alice","Bob","Catherine"), studentscore=c(85,92,78), studen
tpassed=c(TRUE,TRUE,FALSE))

print(student_info)
```

```
## $studentname
## [1] "Alice"     "Bob"       "Catherine"
##
## $studentscore
## [1] 85 92 78
##
## $studentpassed
## [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)

typeof(prices)
```

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

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

x1 <- c(5,10,15)
x2 <- c("apple","banana","cherry")
x3 <- c(x1,x2)

print(x3)
```

```
## [1] "5"      "10"     "15"     "apple"  "banana" "cherry"
```

An implicit coercion is an automatic conversion of values by r 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:

```
# Enter code here

numbers <- c(7,12.5,"15.7")

numbers <- as.numeric(numbers)

sum(numbers)
```

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

R will not automatically handle the data type conversion. The 3rd element in the vector ‘numbers’ is “15.7”, which is a character and not a double. As a result, an error message is seen when a non-numeric argument is used for calculating the sum of elements in a vector.

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)

mean(grades)
```

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

You will receive a ‘NA’ message. R will not be able to calculate the mean of the vector ‘grades’ as the 3rd element, “75.2” is a character, which is non-numeric.

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(number=c(10,20,30), colour=c("red","green","blue"), logic=c(TRUE,FALSE,TRU
E))

mean(mixed_data$number)
```

```
## [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

student_info<-list(studentname=c("Alice","Bob","Catherine"), studentscore=c(85,92,78), studen
tpassed=c(TRUE,TRUE,FALSE))

print(student_info$studentscore[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

nnumeric <- c(1,2,3,4,5,6,7,8)

print(nnumeric[length(nnumeric)])
```

```
## [1] 8
```

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(28,29,30,31,32,33,34)

ages[ages>30]
```

```
## [1] 31 32 33 34
```

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

sequence <-1:20
sequence[seq(from=3,to=20,by=3)]
```

```
## [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

sequence <-1:10
seq(from=4,to=10,by=4)
```

```
## [1] 4 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)
print(is.na(data[2]))
```

```
## [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:

```
# Enter code here

temperatures <- c(90,100,60,70,120,130)
hot_days <- temperatures>90
sum(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("banana","strawberry","watermelon","pear","pineapple")

long_names <- nchar(fruits)>6

fruits[long_names]
```

```
## [1] "strawberry" "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:

```
# Enter code here

numeric <- c(1,2,3,4,5,10,20,40,100,99)
divisible_by_5 <- (numeric %% 5) == 0
numeric[divisible_by_5]
```

```
## [1]   5  10  20  40 100
```

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(2,4,6,51,69,99)
vector2 <- c(3,12,15,27,31,84)
greater_vec <- vector1 > vector2
print(greater_vec)
```

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