

# Challenge-3

Youha Kim

August 28, 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:** character(ordinal) because they have a natural ordering from positive to negative

### 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(nominal) type because it makes it easier to search for specific hashtags. For instance, if there are three hashtags #coding, #programming, and #computer, and you want to only learn information on #coding, using the character(nominal) type will allow you to find that specific hashtag quickly.

### 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 because timestamps are expressed in the form 10:45:43.

### 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 character(ordinal) type date to represent the session date and time because the session date is expressed in the form YYYY-MM-DD, and the time is expressed in the form hh:mm.

### 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 variable because names of candidates are non-numeric

### 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 variable because words like "email", "phone", or "social media" are non-numeric

### 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 variable because colour of names are non-numeric

## 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: Screen Time - non-numeric Variable 2: Number of followers - numeric Variable 3: Number of posts - 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(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:**

```
student_info=list(student_names=c("Alice","Bob","Catherine"), scores=c(85,92,78), result=c(TRUE, TRUE, FALSE))
print(student_info)
```

```
## $student_names
## [1] "Alice"      "Bob"        "Catherine"
##
## $scores
## [1] 85 92 78
##
## $result
## [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(typeof(data[1]))
```

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

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

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

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

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

```
print(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:**

```
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:**

The implicit coercion is when R automatically converts values from one data type to another.

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

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

**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:** No, R will not automatically handle the data type conversion because “15.7” is of character data type. I will use explicit coercion to convert the data type from character to numeric and then calculate the sum.

```
numbers<-c(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:** If I directly calculate the mean using the mean() function, it will result in NA. I can ensure accurate calculation by using explicit coercion and changing the data type to numeric.

```
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(numeric=c(10,20,30), character=c("red","green","blue"), logical=c(TRUE,FALSE,TRUE))

print(mixed_data)
```

```
## $numeric
## [1] 10 20 30
##
## $character
## [1] "red" "green" "blue"
##
## $logical
## [1] TRUE FALSE TRUE
```

```
mean(mixed_data$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:**

```
student_info$scores[student_info$student_names=="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:**

```
cherry<-c(45,76,2000,54)
cherry[length(cherry)]
```

```
## [1] 54
```

### 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:**

```
necklace<-c("apple", "banana", "cherry", "apple")
which(necklace=="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:**

```
ages_individuals<-c(3,76,8,24,34,21)
ages_individuals[ages_individuals>30]
```

```
## [1] 76 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:**

```
stratocaster<-c(1:20)
stratocaster[seq(1, length(stratocaster),3)]
```

```
## [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:**

```
telecaster<-c(1:10)
telecaster[c(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:**

```
squier<-c(10, NA, 15, 20)
is.na(squier)
```

```
## [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:**

```
daily_temperatures<-c(91.4,95,0,57.2,39.2)
hot_days<-daily_temperatures>90
sum(hot_days)
```

```
## [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:**

```
fruit_names<-c("kiwi", "cherry", "blueberry", "orange")
over_six<-nchar(fruit_names)>6
print(fruit_names[over_six])
```

```
## [1] "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:

```
marshall<-c(40,89,3,56,35)
five_divisible<-marshall %% 5 == 0
print(marshall[five_divisible])
```

```
## [1] 40 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:

```
vector1<-c(45,73,3,0,65)
vector2<-c(4,7,5,3,2)
compare<-vector1>vector2
print(compare)
```

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