

IDS LAB INTERNAL-1 ANSWERS

1. Write an R program to create a **list** containing:

- Numbers from 1 to 5
- A character vector with 3 strings
- A logical vector with TRUE/FALSE values
- V: Name some popular R packages for data visualization.

CODE:

```
my_list <- list(  
  numbers = 1:5,  
  strings = c("Data", "Visualization", "R Programming"),  
  logicals = c(TRUE, FALSE, TRUE, FALSE)  
)  
print(my_list)  
popular_packages <- c("ggplot2", "lattice", "plotly", "shiny", "highcharter")  
cat("\nPopular R packages for data visualization:\n")  
print(popular_packages)
```

2. Demonstrate the concept of **factors** by creating a factor variable for fruit names ("Apple", "Mango", "Apple", "Orange").

- Display the number of levels.
- Convert factor back to character.
- V: **What are the basic data structures in R?** (e.g., vectors, lists, matrices, data frames, arrays, factors). Explain the differences between them

CODE:

```
# Create a factor variable
```

```
fruit_factor <- factor(c("Apple", "Mango", "Apple", "Orange"))
```

```
# Display the factor variable
```

```
print(fruit_factor)
```

```
# Display the number of levels
```

```
num_levels <- nlevels(fruit_factor)
```

```
cat("\nNumber of levels:", num_levels, "\n")
```

```
# Convert factor back to character
```

```
fruit_character <- as.character(fruit_factor)
```

```
print(fruit_character)
```

```
# V: Basic data structures in R and differences
```

```
cat("\nBasic Data Structures in R:\n")
```

```
cat("1. Vector: A sequence of elements of the same type (numeric, character, logical).\n")
```

```
cat("2. List: A collection of elements which can be of different types (numeric, character, etc.).\n")
```

```
cat("3. Matrix: 2-dimensional, homogeneous data structure (all elements of the same type).\n")
```

```
cat("4. Data Frame: 2-dimensional heterogeneous data structure where each column can have a different type.\n")
```

```
cat("5. Array: Multi-dimensional homogeneous data structure (extends matrix to more than 2 dimensions).\n")
```

```
cat("6. Factor: Used to represent categorical data with predefined levels.\n")
```

3. Write an R program to perform set operations on two vectors:

- `A <- c(1,2,3,4,5)`
- `B <- c(4,5,6,7,8)`
Find **union, intersection, and set difference**.
- V: What is a data frame in R?

CODE:

```
A <- c(1, 2, 3, 4, 5)
```

```
B <- c(4, 5, 6, 7, 8)
```

```
union_result <- union(A, B)
```

```
intersection_result <- intersect(A, B)
```

```
set_difference_A_B <- setdiff(A, B)
```

```
set_difference_B_A <- setdiff(B, A)
```

```
cat("Union of A and B:\n")
```

```
print(union_result)
```

```
cat("Intersection of A and B:\n")
```

```
print(intersection_result)
```

```
cat("Set Difference A - B:\n")
```

```
print(set_difference_A_B)
```

```
cat("Set Difference B - A:\n")
```

```
print(set_difference_B_A)
```

```
cat("\nV: A data frame in R is a two-dimensional, heterogeneous data structure where each column  
can contain elements of different types (numeric, character, logical, etc.). It is like a table in which  
columns represent variables and rows represent observations.\n")
```

4 i).Write an R program to **concatenate** two vectors of strings into one.

ii)V:What is the purpose of the apply() function in R?

CODE:

```
vector1 <- c("Hello", "World")
vector2 <- c("Data", "Science")
concatenated_vector <- c(vector1, vector2)
print(concatenated_vector)

cat("The apply() function in R applies a function to the rows or columns of a matrix or data frame.")

mat <- matrix(1:9, nrow=3)
result <- apply(mat, 1, sum)
print(result)
```

5.Import a **text file** containing numbers.

- Find the sum and mean of all numbers.
- Save the results in another text file
- V:How do you create a scatter plot in R?

CODE:

```
numbers_to_write <- c(10, 20, 30, 40, 50)
cat(numbers_to_write, file = "numbers.txt", sep = "\n")
numbers <- scan("numbers.txt")
total_sum <- sum(numbers)
mean_value <- mean(numbers)
result <- c(Sum = total_sum, Mean = mean_value)
cat("Sum:", total_sum, "\nMean:", mean_value, "\n", file = "results.txt")
print(result)

x <- 1:10
y <- x^2

plot(x, y, main = "Scatter Plot Example", xlab = "X values", ylab = "Y values", pch = 19)
```

6. Load an **Excel file** in R and extract only one specific sheet.

- Convert the sheet into a dataframe.
- Save it back as CSV
- V: How do you calculate the mean of a vector in R?

CODE:

```
install.packages("readxl")
install.packages("writexl")
library(readxl)
library(writexl)

sample_data <- data.frame(Name = c("Alice", "Bob"), Score = c(95, 85))
write_xlsx(sample_data, "data.xlsx")

data <- read_excel("data.xlsx", sheet = "Sheet1")
df <- as.data.frame(data)
write.csv(df, "output.csv", row.names = FALSE)

vec <- c(1, 2, 3, 4, 5)
mean_value <- mean(vec)
print(mean_value)
```

7. On the dataset **iris**:

- Check for duplicate rows and remove them.
- Convert the categorical column **Species** into numerical encoding.
- Apply sampling: randomly select 20% of data for testing.
- V: How do you check if a value is NA in R?

CODE:

```
data(iris)

iris <- iris[!duplicated(iris), ]
iris$Species <- as.numeric(factor(iris$Species))

set.seed(123)

sample_indices <- sample(1:nrow(iris), size = 0.2 * nrow(iris))
test_data <- iris[sample_indices, ]
print(head(test_data))

x <- c(1, NA, 3)
is_na <- is.na(x)
print(is_na)
```

8. Use **reshape2** to transform the **airquality** dataset:

- Melt the dataset on **Month** and **Day**.
- Cast it back to find the mean values of variables by month
- V: How do you calculate the p-value in R?

CODE:

```
library(reshape2)

data(airquality)

melted <- melt(airquality, id.vars = c("Month", "Day"))

casted <- dcast(melted, Month ~ variable, mean, na.rm = TRUE)

print(casted)

x <- c(1, 2, 3, 4, 5)

y <- c(2, 4, 6, 8, 10)

test <- cor.test(x, y)

print(test$p.value)
```

9. Create visualizations using **ggplot2**:

- Plot histogram of **Sepal.Length** (iris dataset).
- Scatter plot of **mpg** vs **hp** (mtcars dataset) with regression line.
- Bar plot of average **Sepal.Width** grouped by species.
- V: How do you create a histogram in R?

CODE:

```
library(ggplot2)

data(iris)

ggplot(iris, aes(x = Sepal.Length)) +
  geom_histogram(binwidth = 0.5, fill = "blue", color = "black") +
  ggtitle("Histogram of Sepal.Length")

data(mtcars)

ggplot(mtcars, aes(x = hp, y = mpg)) +
  geom_point() +
  geom_smooth(method = "lm", col = "red") +
  ggtitle("Scatter Plot of mpg vs hp with Regression Line")

ggplot(iris, aes(x = Species, y = Sepal.Width)) +
  stat_summary(fun = mean, geom = "bar", fill = "green") +
  ggtitle("Bar Plot of Average Sepal.Width by Species")
```

10. Using **data.table**, read the **mtcars** dataset and:

- Display cars with **mpg > 25**.
- Find the mean horsepower (**hp**) by number of cylinders (**cyl**).
- V: What is the role of R packages like **dplyr** and **ggplot2** in data science?

CODE:

```
library(data.table)

dt <- as.data.table(mtcars)

high_mpg <- dt[mpg > 25]

print(high_mpg)

mean_hp_by_cyl <- dt[, .(Mean_HP = mean(hp)), by = cyl]

print(mean_hp_by_cyl)

cat("\nV: Role of R packages in data science:\n")

cat("1. dplyr: Provides easy-to-use functions for data manipulation such as filtering, selecting columns, grouping, and summarizing large datasets in a fast and readable way.\n")

cat("2. ggplot2: Allows creation of advanced and customizable data visualizations using the Grammar of Graphics framework, making it easier to produce high-quality plots for exploratory data analysis and reporting.\n")
```

11. Create a list in R that contains numbers from 1 to 10.

- Extract first 5 elements from that list
 - Extract elements from position 4 to 8.
- V: What is the role of **dplyr** and **ggplot2** in data science with R?

CODE:

```
my_list <- as.list(1:10)

first_5 <- my_list[1:5]

print(first_5)

pos_4_to_8 <- my_list[4:8]

print(pos_4_to_8)

cat("\nV: Role of dplyr and ggplot2 in data science with R:\n")

cat("dplyr: Simplifies data manipulation tasks like filtering rows, selecting columns, arranging data, and summarizing datasets with an intuitive syntax that works well with large data.\n")

cat("ggplot2: Provides a powerful and flexible system for creating high-quality data visualizations based on the Grammar of Graphics, enabling clear exploration and communication of data insights.\n")
```

12. Create a function that takes a vector of sentences as input and:

- Finds the longest word.
- Finds all words that start with a vowel.
- Returns a dictionary (named list) with results.

V: What are the types of graphs in R

CODE:

```
find_words_info <- function(sentences) {  
  words <- unlist(strsplit(tolower(sentences), "\\W+"))  
  longest_word <- words[which.max(nchar(words))]  
  words_starting_with_vowel <- words[grepl("^[aeiou]", words)]  
  result <- list(  
    longest_word = longest_word,  
    words_starting_with_vowel = unique(words_starting_with_vowel)  
  )  
  return(result)  
}  
  
sentences <- c("Data science is amazing", "R programming is useful", "Artificial intelligence evolves")  
output <- find_words_info(sentences)  
print(output)  
  
cat("\nV: Types of graphs in R:\n")  
  
cat("1. Histogram\n2. Scatter Plot\n3. Line Chart\n4. Bar Plot\n5. Box Plot\n6. Pie Chart\n7. Density Plot\n8. Heatmap\n9. Area Chart\n10. Correlation Plot\n")
```

13. Using the dataset **iris**, perform the following with **dplyr**:

- Select only **Sepal.Length** and **Species**.
- Filter rows where **Sepal.Length > 6**.
- Group by **Species** and find the average **Petal.Length**.
- V: How to access data in dataframe.

CODE:

```
library(dplyr)
data(iris)
selected <- iris %>%
  select(Sepal.Length, Species)
filtered <- iris %>%
  filter(Sepal.Length > 6)
grouped_avg <- iris %>%
  group_by(Species) %>%
  summarise(Avg_Petal_Length = mean(Petal.Length))
print(selected)
print(filtered)
print(grouped_avg)
cat("\nV: How to access data in a dataframe:\n")
cat("1. Using $ operator: iris$Sepal.Length\n")
cat("2. Using square brackets: iris[1, 2] (row 1, column 2)\n")
cat("3. Using column name: iris[, 'Sepal.Length'] or iris[, 1]\n")
cat("4. Using dplyr functions: select(), filter(), mutate(), summarise(), etc.\n")
```