Exercises for the First session

All the exercises need to be done in .Rmd file using R Studio ; you will see on your own discretion how to create it, work on the format using Markdown syntax and create a HTML page using the Knit function.

At any moment, the file could be asked and need to be submitted according to M.ROSARI's demand.

Part 1: Basic Operations with Numbers and Strings

Exercise 1: The Online Store Calculator

For this exercise, we will perform basic calculations to determine the final price of a shopping cart. This will involve using arithmetic operators and assigning values to variables.

Tasks:

1. Create a variable named `item_price` and assign it a value of `150`.
2. Create another variable named `item_quantity` and assign it a value of `3`.
3. Calculate the subtotal (price * quantity) and store it in a variable called `subtotal`.
4. A discount of 20% is applied to the subtotal. Calculate the discount amount and store it in `discount_amount`.
5. Calculate the new total after the discount and store it in `discounted_total`.
6. A tax of 5% is applied to the `discounted_total`. Calculate the tax amount and store it in `tax_amount`.
7. Calculate the final price by adding the tax to the `discounted_total`. Store this in a variable named `final_price`.
8. Print the value of `final_price`.

Exercise 2: Creating User Profiles

In this exercise, we will manipulate character strings to create and format user information for a fictional application.

Tasks:

1. Create a variable `first_name` and assign it the string "alex".
2. Create a variable `last_name` and assign it the string "doe".
3. Create a `full_name` variable by combining `first_name` and `last_name`. Make sure there is a space between them. (Hint: use the `paste()` function).
4. Create a `welcome_message` variable that combines the string "Welcome, " with the `full_name` and ends with an exclamation mark "!".
5. A username is generated from the first letter of the first name and the full last name. Create a variable `username` with this value (e.g., "adoe"). (Hint: look into the `substr()` function).
6. Print the `welcome_message` and the `username`.

Part 2: Operations with Vectors and Matrices

Exercise 3: Analyzing Student Grades

This exercise involves creating a vector of student grades to perform basic statistical analysis.

Setup:

First, create a vector containing the following grades and assign it to a variable named `grades`.
`grades_data <- c(88, 92, 85, 74, 95, 68, 85, 89, 91, 85)`

Operations on the vector:

1. Calculate the average grade. (Hint: use the `mean()` function).
2. Find the highest and lowest grades. (Hint: use `max()` and `min()`).
3. How many students are in the class? (Hint: use `length()`).
4. Select and display only the grades that are 90 or higher.
5. How many students scored exactly 85?

Exercise 4: Monthly Sales Report

For this exercise, you will work with two vectors representing monthly sales and expenses to calculate the monthly profit.

Setup:

Create two vectors:
`sales <- c(5000, 5500, 4800, 6200, 7000)expenses <- c(2200, 2400, 2100, 2800, 3100)`

Operations on the vectors:

1. Calculate the profit for each month (sales - expenses) and store the result in a new vector called `profit`.
2. Calculate the total profit for all months combined. (Hint: use `sum()`).
3. What was the average monthly profit?
4. Which month had the highest profit? (Hint: use `which.max()` on the `profit` vector).

Exercise 5: Store Inventory Management

In this exercise, you will create a matrix to represent the inventory of three products across four different stores.

Setup:

Create a matrix using the following data. The matrix should have 3 rows (for Products A, B, C) and 4 columns (for Stores 1-4). Assign it to a variable named `inventory_matrix`.
`inventory_data <- c(25, 150, 75, 40, 200, 110, 30, 180, 90, 55, 220, 130)`

R

```
# Use this code to create the matrix
inventory_matrix <- matrix(inventory_data, nrow = 3, byrow = TRUE)
```

Operations on the matrix:

1. Display the entire inventory matrix.
2. What is the inventory level of Product B in Store 3? (Access the element at row 2, column 3).
3. Display the inventory levels for all products in Store 4 only.
4. Calculate the total stock for Product A across all stores. (Hint: use `sum()` on the first row).
5. Which store has the highest stock of Product C? (Hint: use `which.max()` on the third row).

Part 3: Reading CSV and Applying Transformations

Exercise 6: Analyzing Employee Data

For this exercise, you will work with a small dataset of employees. You will first need to create a CSV file.

Dataset Preparation:

1. Open a plain text editor (like Notepad on Windows or TextEdit on Mac).
2. Copy and paste the following text exactly as it is shown below:Extrait de code

   ```
   employee_id,name,department,salary_usd,start_year
   101,John Smith,Sales,62000,2019
   102,Jane Doe,Marketing,68000,2020
   103,Peter Jones,IT,75000,2018
   104,Susan Lee,Sales,64000,2021
   105,David Chen,IT,82000,2017
   ```

3. Save this file as `employees.csv` in your R working directory.

Data Analysis Tasks:

1. Load the `employees.csv` file into an R data frame called `employees_df`. (Hint: use `read.csv()`).
2. Display the first few rows of the data frame to verify it loaded correctly. (Hint: use `head()`).
3. Inspect the structure of the data frame to see the column types. (Hint: use `str()`).
4. Calculate the average salary of all employees. (Access the salary column with `employees_df$salary_usd`).
5. Create a new column called `salary_eur` by converting the USD salary to EUR. Assume an exchange rate of 1 USD = 0.92 EUR.
6. Create a new data frame called `it_employees_df` that contains only the employees from the 'IT' department.
7. Display the `it_employees_df` data frame.