ELEC IT-E1

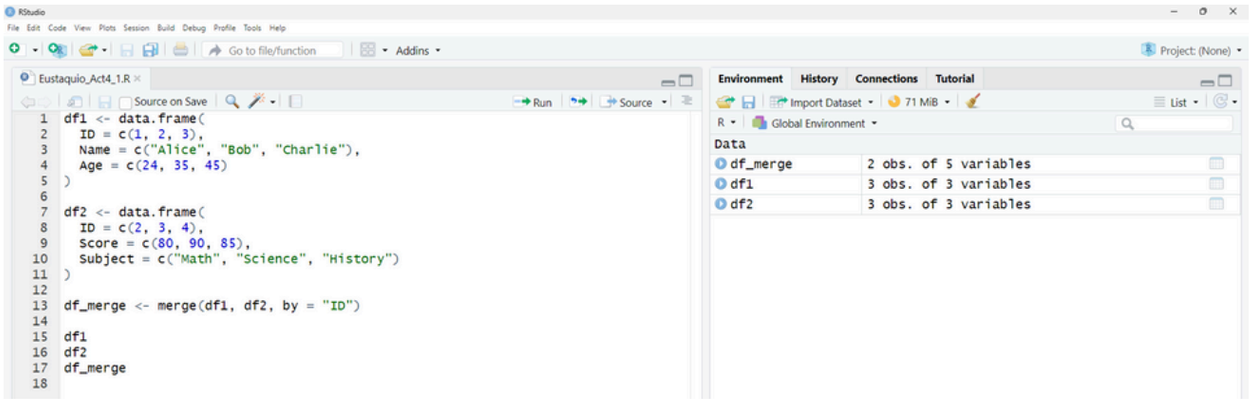
IT ELECTIVE 1

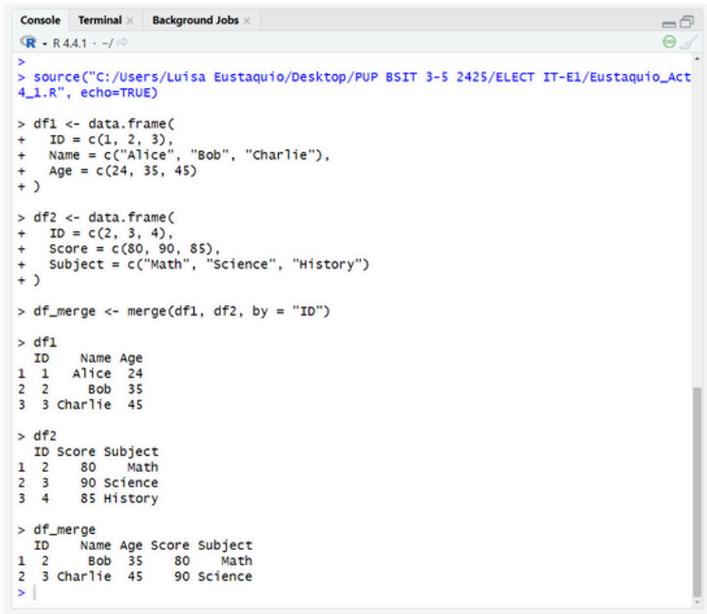
Name: Eustaquio, Maria Luisa O.	**Year & Section:** BSIT 3-5	**Date:** November 18, 2024

ACTIVITY #4 - MANIPULATING AND MERGING DATA INSTRUCTIONS

Instructions: Manipulate and combine datasets using R. Write and test your R code for each question. Provide explanations for your steps and results.

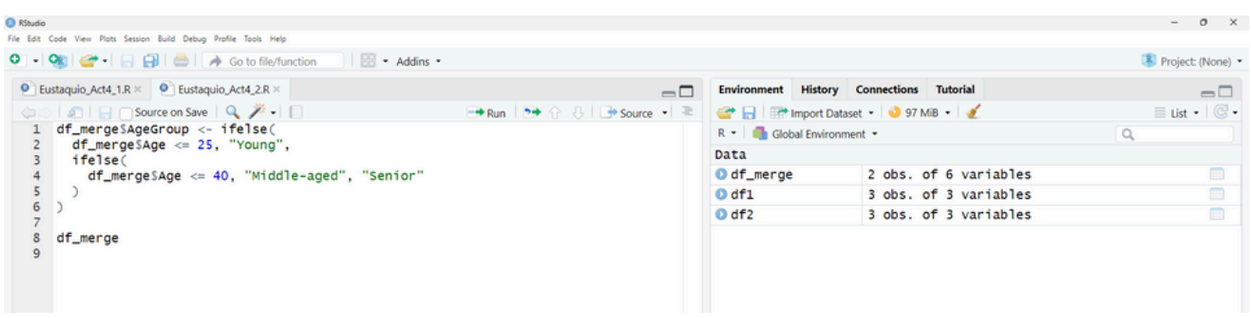
- Multi-Column Merge: Create two data frames: df1: Columns ID, Name, and Age. df2: Columns ID, Score, and Subject. Write R code to perform an inner join on ID and explain the resulting data frame structure.





The code creates two data frames, df1 and df2, and performs an inner join on the common ID column using the merge() function. The resulting data frame, df_merge, includes only rows where the ID values exist in both df1 and df2. In this case, the overlapping IDs are 2 and 3. The structure of df_merge combines columns from both data frames: ID, Name, and Age from df1, and Score and Subject from df2. Specifically, the output contains two rows (IDs 2 and 3). Rows with IDs 1 (from df1) and 4 (from df2) are excluded as they do not exist in both data frames.

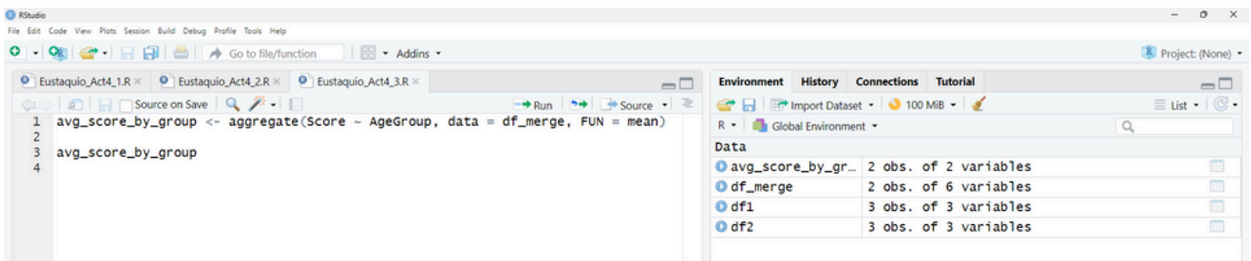
- Add a column to the merged data frame from Task 1 that dynamically categorizes Age into three groups: "Young": Age <= 25 "Middle-aged": Age > 25 and <= 40 "Senior": Age > 40



```
> source("C:/Users/Luisa Eustaquio/Desktop/PUP BSIT 3-5 2425/ELECT IT-E1/Eustaquio_Act4_2.R", echo=TRUE)

> df_merge$AgeGroup <- ifelse(
+   df_merge$Age <= 25, "Young",
+   ifelse(
+     df_merge$Age <= 40, "Middle-aged", "Senior"
+   )
+ )

> df_merge
  ID    Name Age Score Subject    AgeGroup
1  2     Bob  35    80    Math Middle-aged
2  3 Charlie  45    90 Science      Senior
>
```

The df_merge data frame now includes a new column, AgeGroup, which categorizes individuals based on their age. Each row in df_merge is updated with the appropriate age group based on the Age column.

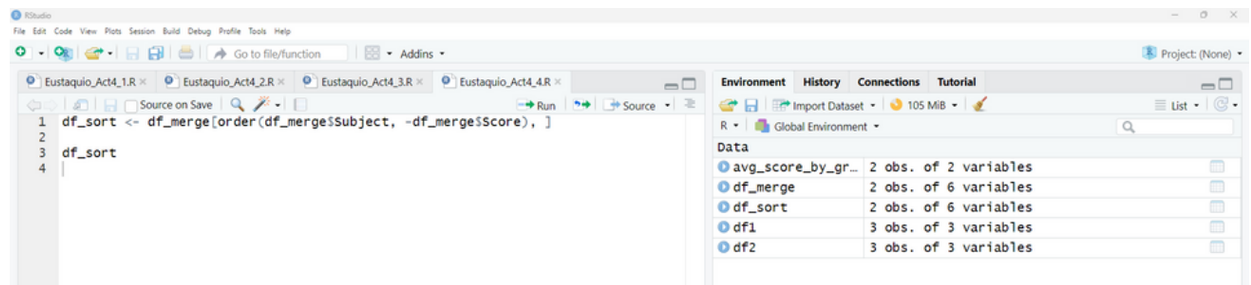
- Using the merged data, compute the average Score for each Age group created in Task 2. Explain the steps and provide the R code.




```
> source("C:/Users/Luisa Eustaquio/Desktop/PUP BSIT 3-5 2425/ELECT IT-E1/Eustaquio_Act4_3.R", echo=TRUE)

> avg_score_by_group <- aggregate(Score ~ AgeGroup, data = df_merge, FUN = mean)

> avg_score_by_group
     AgeGroup Score
1 Middle-aged    80
2      Senior    90
>
```

1. The **aggregate()** function groups data by the AgeGroup column from Task 2.
2. **Score ~ AgeGroup** specifies grouping by AgeGroup to compute statistics for Score.
3. **FUN = mean** calculates the average score for each age group.
4. The output, avg_score_by_group, has two columns:
    - AgeGroup: Age categories ("Young", "Middle-aged", "Senior")
    - Score: The average score for each group.

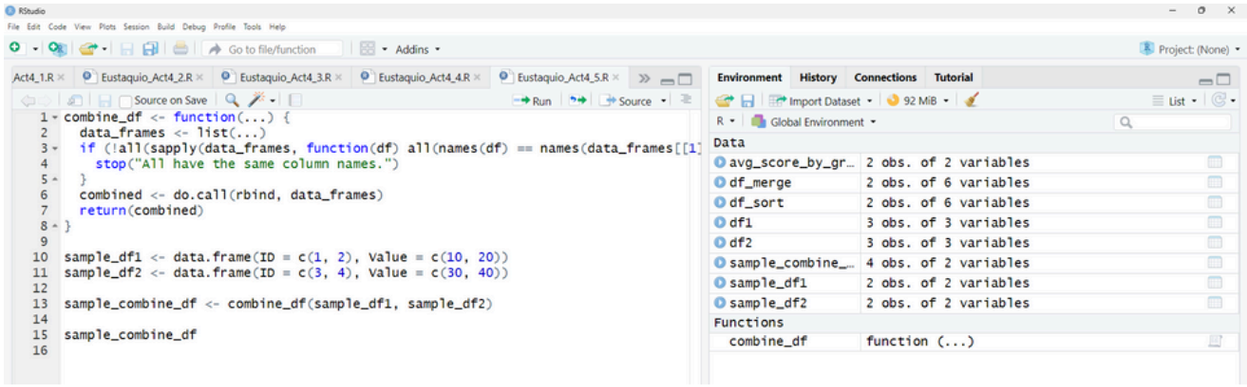
- Sort the merged data frame by Subject (ascending) and Score (descending). Write and explain the R code used.

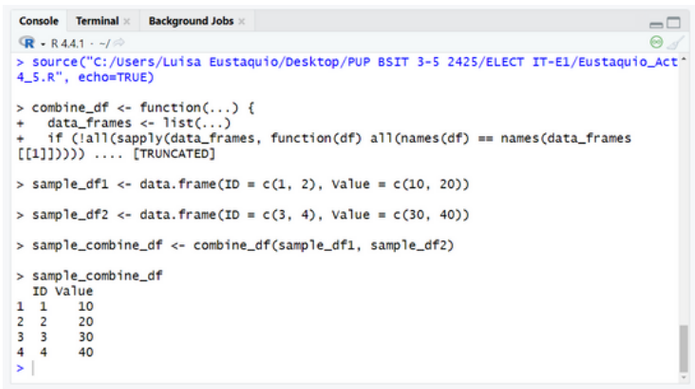



```
> source("C:/Users/Luisa Eustaquio/Desktop/PUP BSIT 3-5 2425/ELECT IT-E1/Eustaquio_Act4_4.R", echo=TRUE)

> df_sort <- df_merge[order(df_merge$Subject, -df_merge$Score), ]

> df_sort
  ID    Name Age Score Subject    AgeGroup
1  2     Bob  35    80    Math Middle-aged
2  3 Charlie  45    90 Science      Senior
>
```

The merged data frame is sorted using order(), first by Subject (ascending) and then by Score (descending with -df_merge$Score), creating df_sort with rows arranged by subject and scores ranked highest to lowest within each subject.

- Write an R function that accepts multiple data frames as input and combines them using rbind() after ensuring all have the same column names. Demonstrate the function with sample inputs.





combine_df

- Accepts multiple data frames as input.
- Checks that all data frames have the same column names.
- Combines the data frames row-wise using rbind().

sample_df1 and **sample_df2**

- Both have the same column names (ID and Value), which ensures compatibility with combine_df.

combine_df(sample_df1, sample_df2)

- Correctly combines the two data frames into sample_combine_df.

sample_combine_df

- Should contain all rows from both sample_df1 and sample_df2.