**Exercises: Data Manipulation with R(Session 2)**

Create a new script (Ctrl + Shift + N) and give it a meaningful name. Work through the following exercises by typing them directly into the script—while you can copy and paste, typing helps reinforce learning. Run the script (Ctrl + Shift + Enter) and check for any errors. You can also run line-by-line (Ctrl + Enter) Try different key-combinations. Finally, save the script for future reference and reuse. Recommended to include comments, for example: # Exercise 1

### **Exercise 1: Basic Data Frame Operations**

1. Create a data frame students with the following columns:
   * Name (Character)
   * Age (Numeric)
   * Score (Numeric)
   * Passed (Logical)
2. Print the first few rows using head().
3. Check the structure of the data frame using str().

students <- data.frame(

Name = c("Alice", "Bob", "Charlie", "David", "Eve"),

Age = c(20, 22, 21, 23, 22),

Score = c(85, 90, 78, 88, 95),

Passed = c(TRUE, TRUE, FALSE, TRUE, TRUE)

)

head(students)

str(students)

### **Exercise 2: Selecting and Filtering Data**

1. Select only the Name and Score columns.
2. Filter the students who have scored more than 80.
3. Filter students who are older than 21 and have passed.

library(dplyr)

students %>% select(Name, Score)

students %>% filter(Score > 80)

students %>% filter(Age > 21, Passed == TRUE)

### **Exercise 3: Creating and Modifying Columns**

1. Create a new column Grade, where:
   * Score >= 90 → "A"
   * Score >= 80 & Score < 90 → "B"
   * Score < 80 → "C"
2. Modify the Age column by adding 1 to each value.

students <- students %>%

mutate(Grade = case\_when(

Score >= 90 ~ "A",

Score >= 80 ~ "B",

TRUE ~ "C"

))

students <- students %>% mutate(Age = Age + 1)

### **Exercise 4: Grouping and Summarizing Data**

1. Group the students by Grade and calculate the average Score for each group.
2. Count the number of students in each grade category.

students %>% group\_by(Grade) %>% summarize(Average\_Score = mean(Score))

students %>% count(Grade)

### **Exercise 5: Reshaping Data with tidyr**

1. Convert the dataset from wide format to long format using pivot\_longer().
2. Separate a new column Full\_Name into First\_Name and Last\_Name.

library(tidyr)

students\_long <- students %>% pivot\_longer(cols = c(Age, Score), names\_to = "Variable", values\_to = "Value")

students <- students %>% mutate(Full\_Name = paste(Name, "Smith"))

students %>% separate(Full\_Name, into = c("First\_Name", "Last\_Name"), sep = " ")

### **Exercise 6: Handling Missing Values**

1. Introduce some NA values in the Score column.
2. Replace missing values with the average Score.
3. Drop rows with missing values.

students$Score[c(2, 4)] <- NA

students <- students %>% mutate(Score = ifelse(is.na(Score), mean(Score, na.rm = TRUE), Score))

students <- students %>% drop\_na()