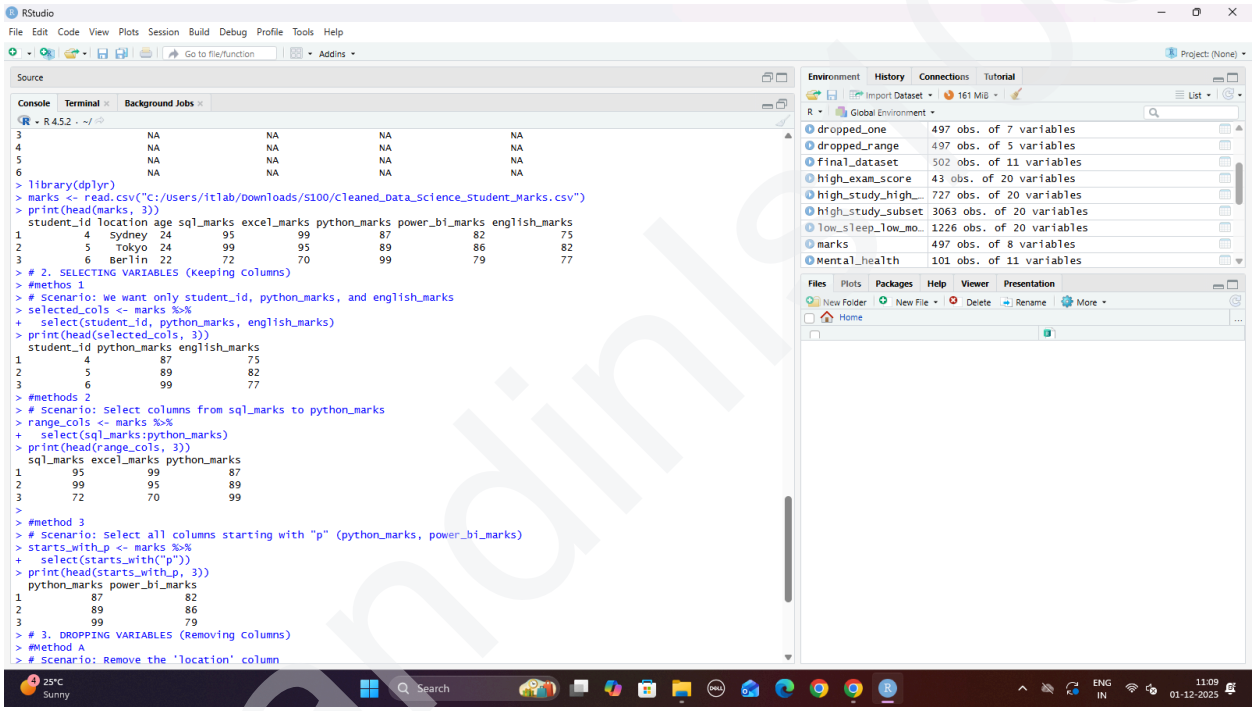
PRACTICAL NO :- 07
AIM :- Selecting and dropping variables using select() in R. import dataset.

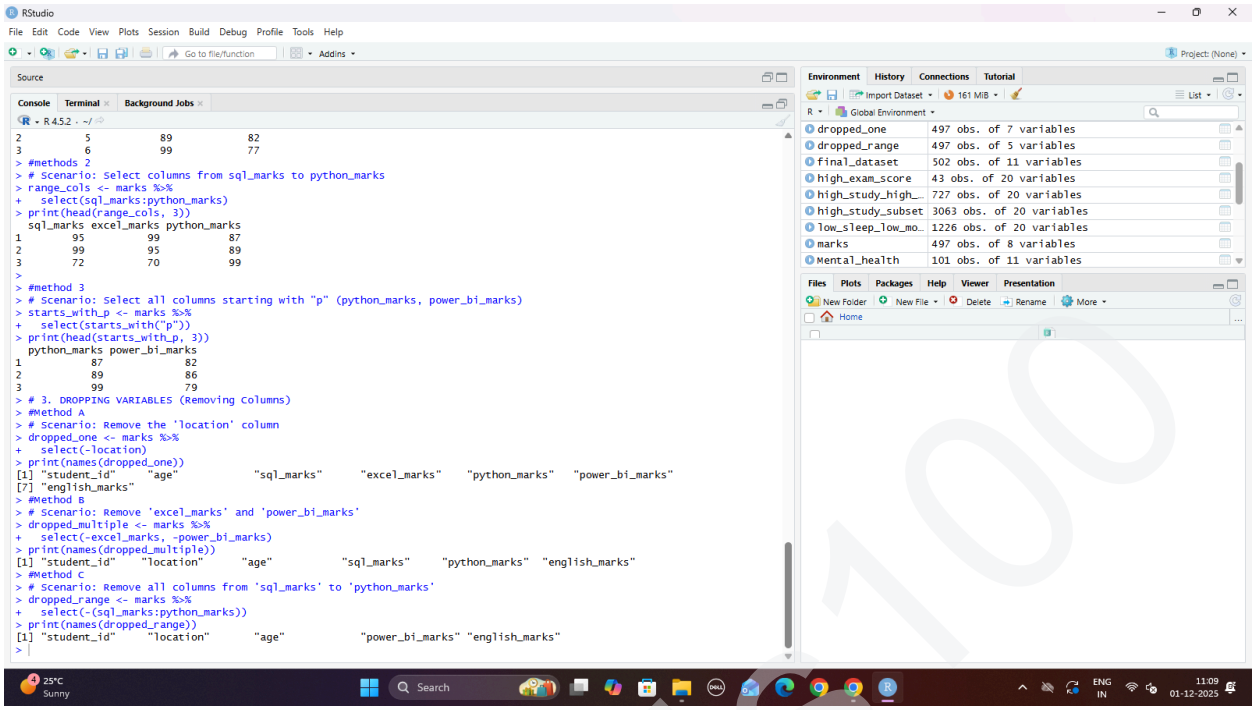
CODE :-

```
library(dplyr)

marks <- read.csv("C:/Users/itlab/Downloads/S100/Cleaned_Data_Science_Student_Marks.csv")

print("--- Original Dataset (First 3 rows) ---")
print(head(marks, 3))

# 2. SELECTING VARIABLES (Keeping Columns)
#method 1
# Scenario: We want only student_id, python_marks, and english_marks
selected_cols <- marks %>%
  select(student_id, python_marks, english_marks)

print("--- Selected Specific Columns ---")
print(head(selected_cols, 3))

#methods 2
# Scenario: Select columns from sql_marks to python_marks
range_cols <- marks %>%
  select(sql_marks:python_marks)
print("--- Selected Range of Columns (sql_marks to python_marks) ---")
print(head(range_cols, 3))

#method 3
# Scenario: Select all columns starting with "p" (python_marks, power_bi_marks)
starts_with_p <- marks %>%
  select(starts_with("p"))


print("--- Selected columns starting with 'p' ---")
print(head(starts_with_p, 3))


# 3. DROPPING VARIABLES (Removing Columns)
#Method A
# Scenario: Remove the 'location' column
dropped_one <- marks %>%
  select(-location)
print("--- Dataset with 'location' dropped ---")
print(names(dropped_one))

#Method B
```

```
# Scenario: Remove 'excel_marks' and 'power_bi_marks'
dropped_multiple <- marks %>%
  select(-excel_marks, -power_bi_marks)
print("--- Dataset with 'excel_marks' and 'power_bi_marks' dropped ---")
print(names(dropped_multiple))


#Method C
# Scenario: Remove all columns from 'sql_marks' to 'python_marks'
dropped_range <- marks %>%
  select(-(sql_marks:python_marks))
print("--- Dataset with columns 'sql_marks' to 'python_marks' dropped ---")
print(names(dropped_range))
```



MVLU COLLEGE



Nandini Pandit S100

DATA ANALYSIS WITH SAS/SPSS/R