SHETH L.U.J AND SIR M.V COLLEGE

PRACTICAL NO: 1

Generating descriptive statistics using summary () or describe()(R)

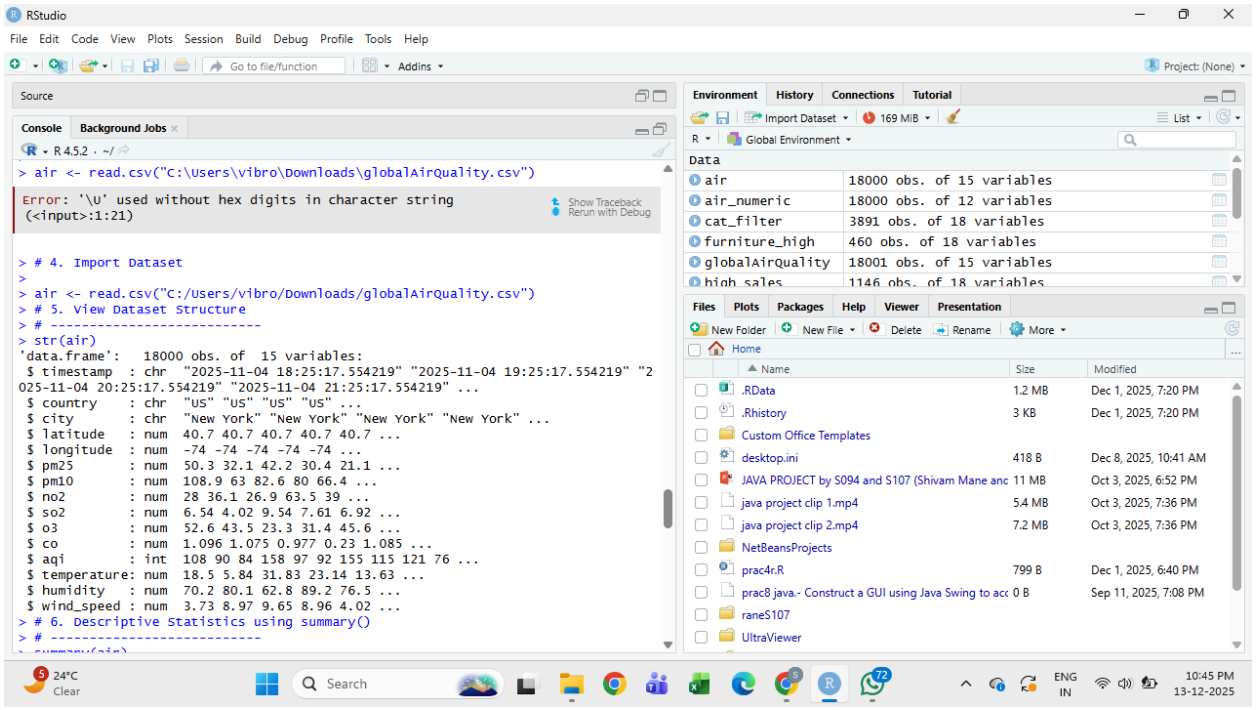
CODE:

```
#
================================================================================
# R Script: Descriptive Statistics
# Functions: summary(), describe()
# Dataset: globalAirQuality.csv
#
================================================================================

# ---------------------------
# 1. Install Required Packages
# ---------------------------
install.packages("psych")
install.packages("dplyr")

# ---------------------------
# 2. Load Libraries
# ---------------------------
library(psych)
library(dplyr)

# ---------------------------
# 3. Set Working Directory (CHANGE PATH IF NEEDED)
# Use forward slashes to avoid '\U' error


# ---------------------------
# 4. Import Dataset

air <- read.csv("C:/Users/vibro/Downloads/globalAirQuality.csv")

# ---------------------------
# 5. View Dataset Structure
# ---------------------------
str(air)

# ---------------------------
# 6. Descriptive Statistics using summary()
```

RIYA RANE
S107 SYCS

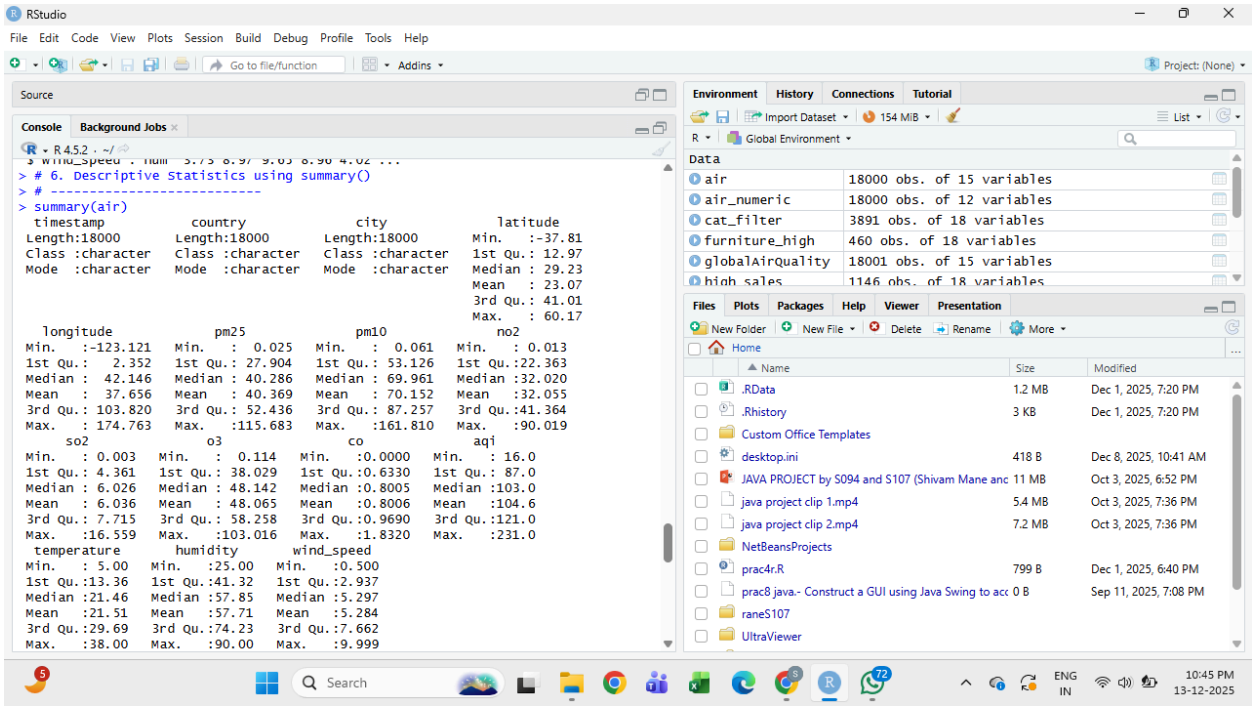
```
# ---------------------------
summary(air)

# ---------------------------
# 7. Select Only Numeric Columns
# ---------------------------
air_numeric <- air %>% select(where(is.numeric))

# ---------------------------
# 8. Descriptive Statistics (Numeric Data Only)
# ---------------------------
summary(air_numeric)

# ---------------------------
# 9. Detailed Descriptive Statistics using describe()
# ---------------------------
describe(air_numeric)
```

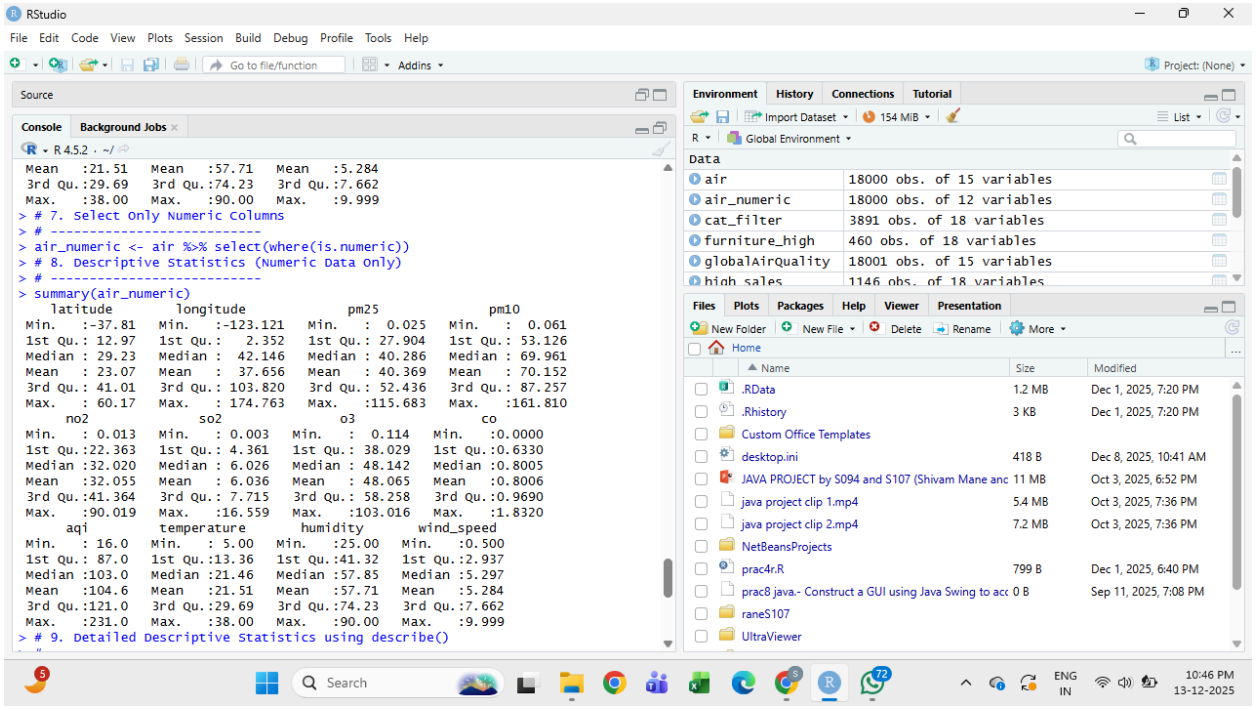
OUTPUT:



RIYA RANE
S107 SYCS

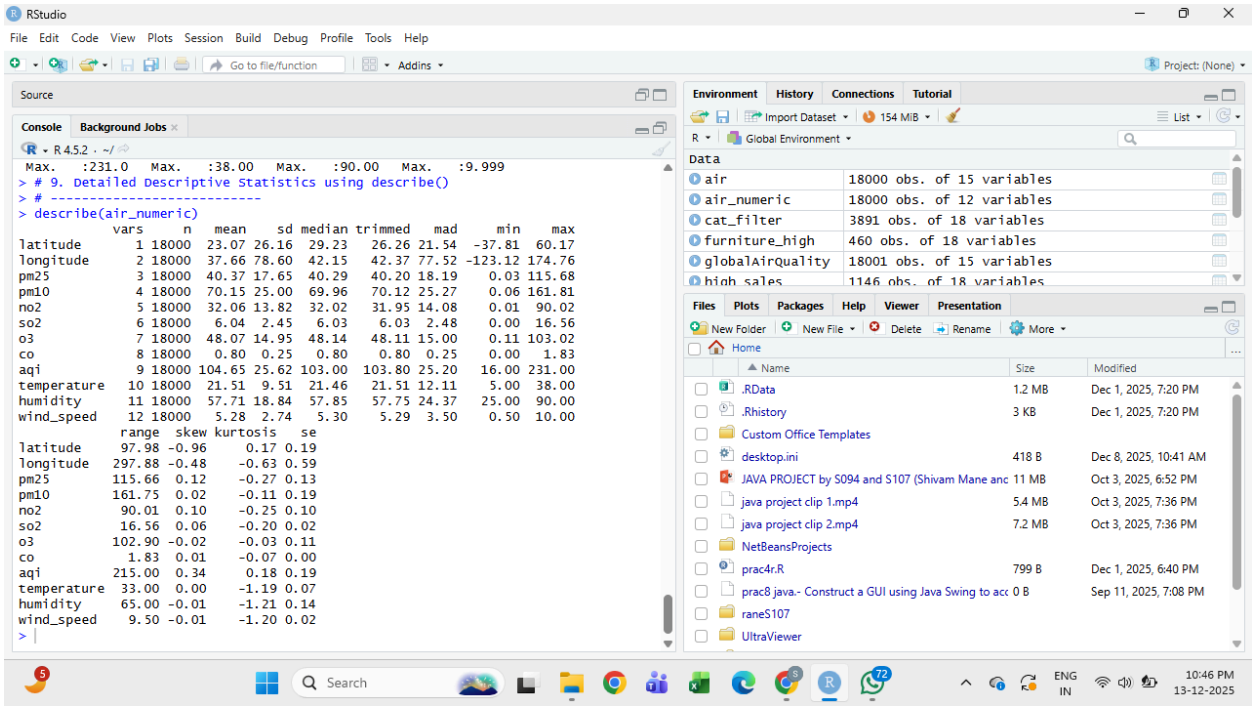
SHETH L.U.J AND SIR M.V COLLEGE





RIYA RANE
S107 SYCS

SHETH L.U.J AND SIR M.V COLLEGE



PRACTICAL NO:2

GENERATING FREQUENCY TABLE USING TABLE() OR COUNT().

CODE:

```
#
================================================================================
# R Script: Frequency Table Generation
# Functions Used: table(), count()
# Dataset: car_price_prediction_.csv
#
================================================================================

# ---------------------------
# 1. Install Required Package (Run Once)
# ---------------------------
install.packages("dplyr")

# ---------------------------
# 2. Load Library (Must Run Every Time)
# ---------------------------
library(dplyr)
```

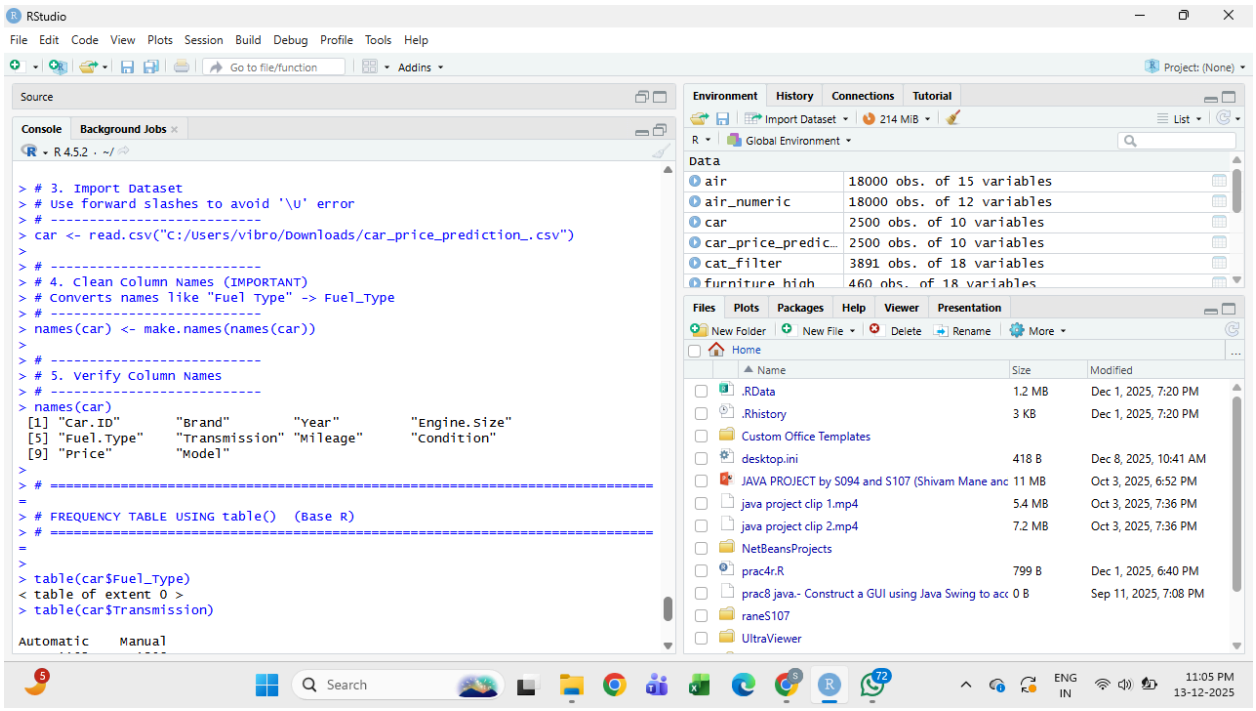
RIYA RANE
S107 SYCS

```
# ---------------------------
# 3. Import Dataset
# Use forward slashes to avoid '\U' error
# ---------------------------
car <- read.csv("C:/Users/vibro/Downloads/car_price_prediction_.csv")

# ---------------------------
# 4. Clean Column Names (IMPORTANT)
# Converts names like "Fuel Type" -> Fuel_Type
# ---------------------------
names(car) <- make.names(names(car))

# ---------------------------
# 5. Verify Column Names
# ---------------------------
names(car)

#
================================================================================
======
# FREQUENCY TABLE USING table()  (Base R)
#
================================================================================
======

table(car$Fuel_Type)
table(car$Transmission)
table(car$Owner)

#
================================================================================
======
# FREQUENCY TABLE USING count() (dplyr)
#
================================================================================
======

count(car, Fuel_Type)
count(car, Transmission)
count(car, Owner)
```

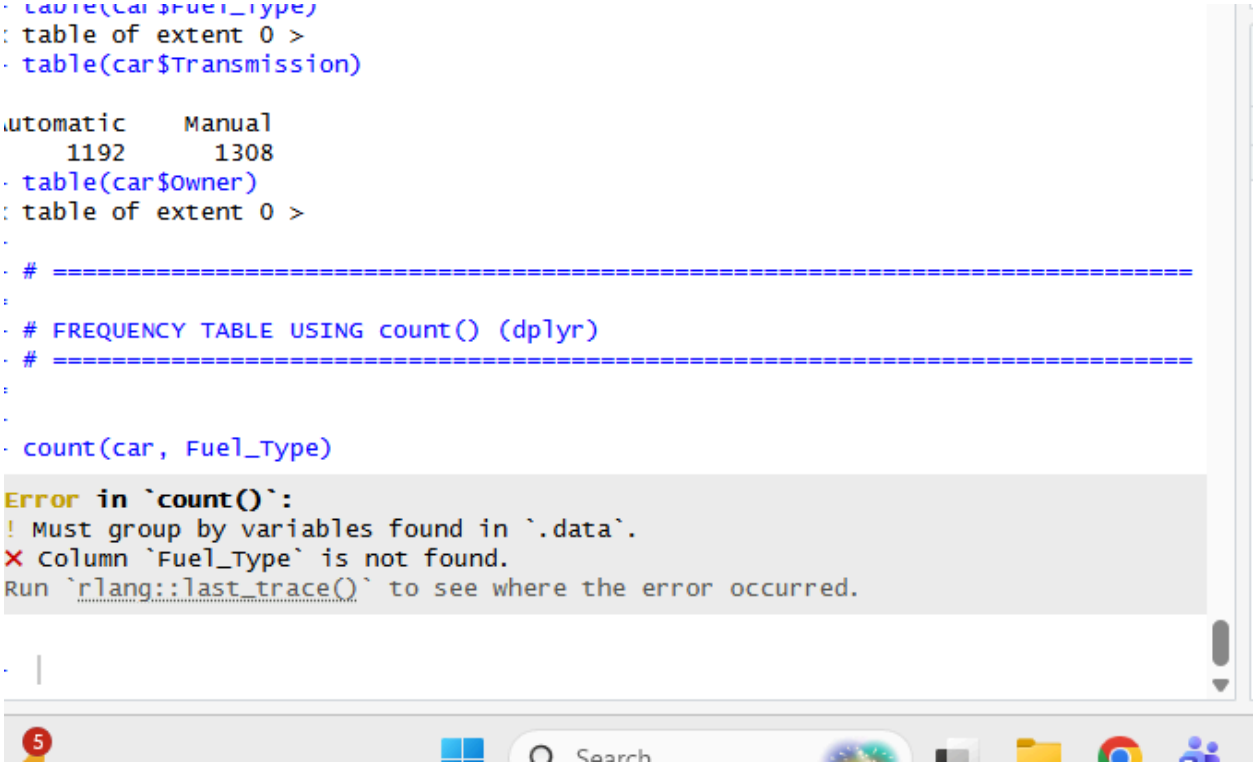
RIYA RANE
S107 SYCS

```
#
================================================================================
======
# SORTED FREQUENCY TABLE (Descending Order)
#
================================================================================
======

count(car, Fuel_Type, sort = TRUE)
count(car, Transmission, sort = TRUE)
count(car, Owner, sort = TRUE)
```

OUTPUT:



RIYA RANE
S107 SYCS



PRACTICAL NO: 3

CREATING CROSS TABULATION AND TWO WAY TABLES USING TABLES()(R)

CODE:

```
#
================================================================================
======
# R Script: Cross Tabulation using table()
#
================================================================================
======

# 1. Read the dataset
data <- read.csv("C:/Users/vibro/Downloads/data_date.csv", stringsAsFactors = FALSE)

# 2. Check column names
names(data)

# 3. Create Two-Way Cross Tabulation
# Replace Column1 and Column2 with actual column names
cross_tab <- table(data$Column1, data$Column2)
```

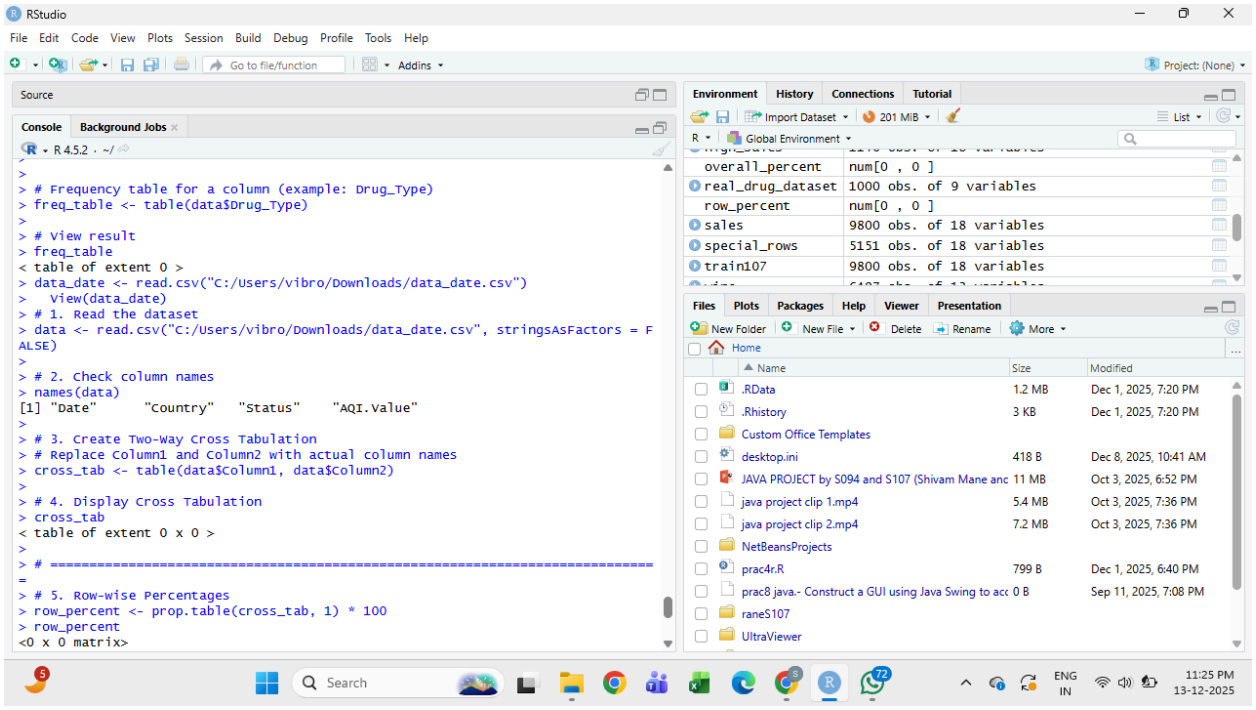
RIYA RANE
S107 SYCS

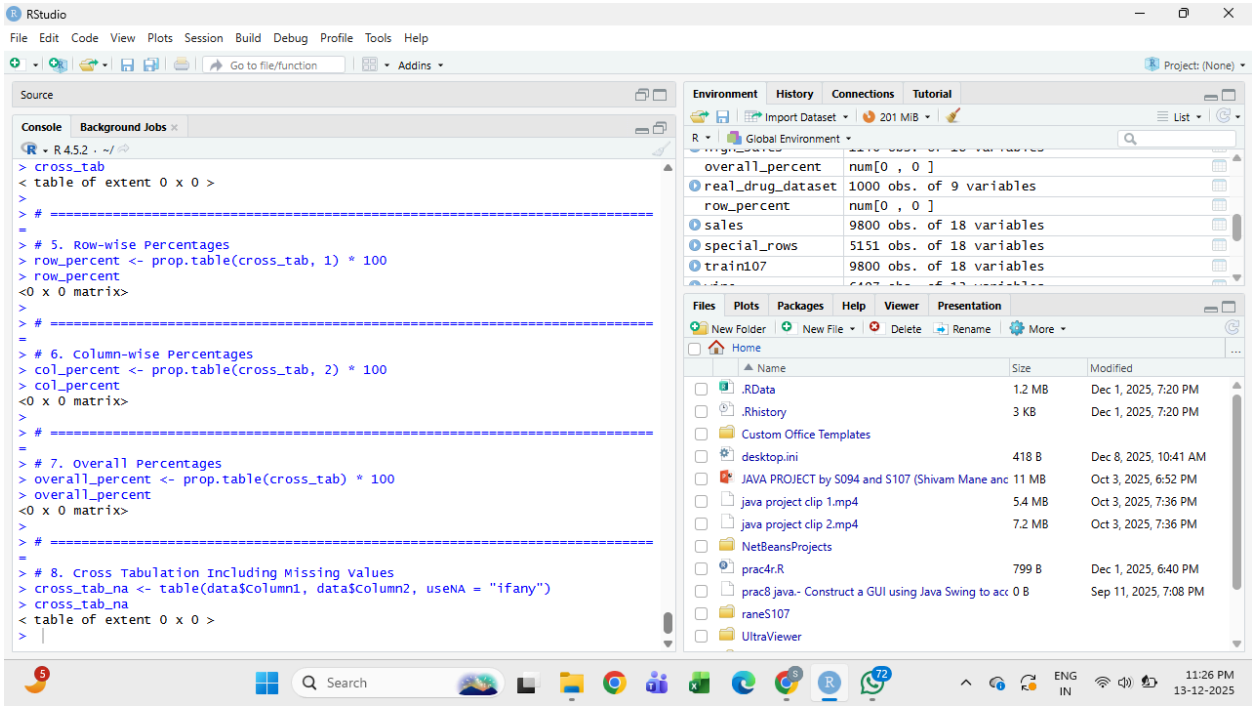
```
# 4. Display Cross Tabulation
cross_tab

#
==========================================================================
======
# 5. Row-wise Percentages
row_percent <- prop.table(cross_tab, 1) * 100
row_percent

#
==========================================================================
======
# 6. Column-wise Percentages
col_percent <- prop.table(cross_tab, 2) * 100
col_percent

#
==========================================================================
======
# 7. Overall Percentages
overall_percent <- prop.table(cross_tab) * 100
overall_percent

#
==========================================================================
======
# 8. Cross Tabulation Including Missing Values
cross_tab_na <- table(data$Column1, data$Column2, useNA = "ifany")
cross_tab_na
```

OUTPUT:

RIYA RANE
S107 SYCS

SHETH L.U.J AND SIR M.V COLLEGE





PRACTICAL NO: 4

PERFORMING ONE SAMPLE T-TESTS USING T.TESTS()..GIVE CODE

RIYA RANE
S107 SYCS

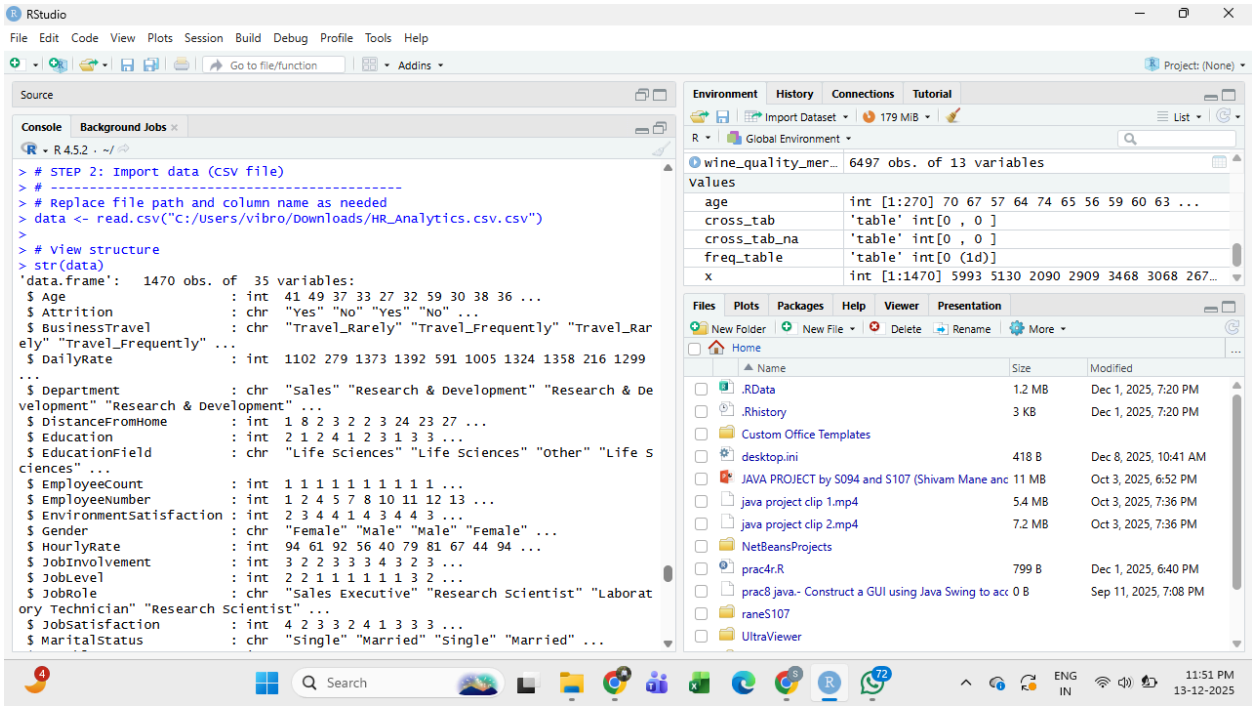
CODE:

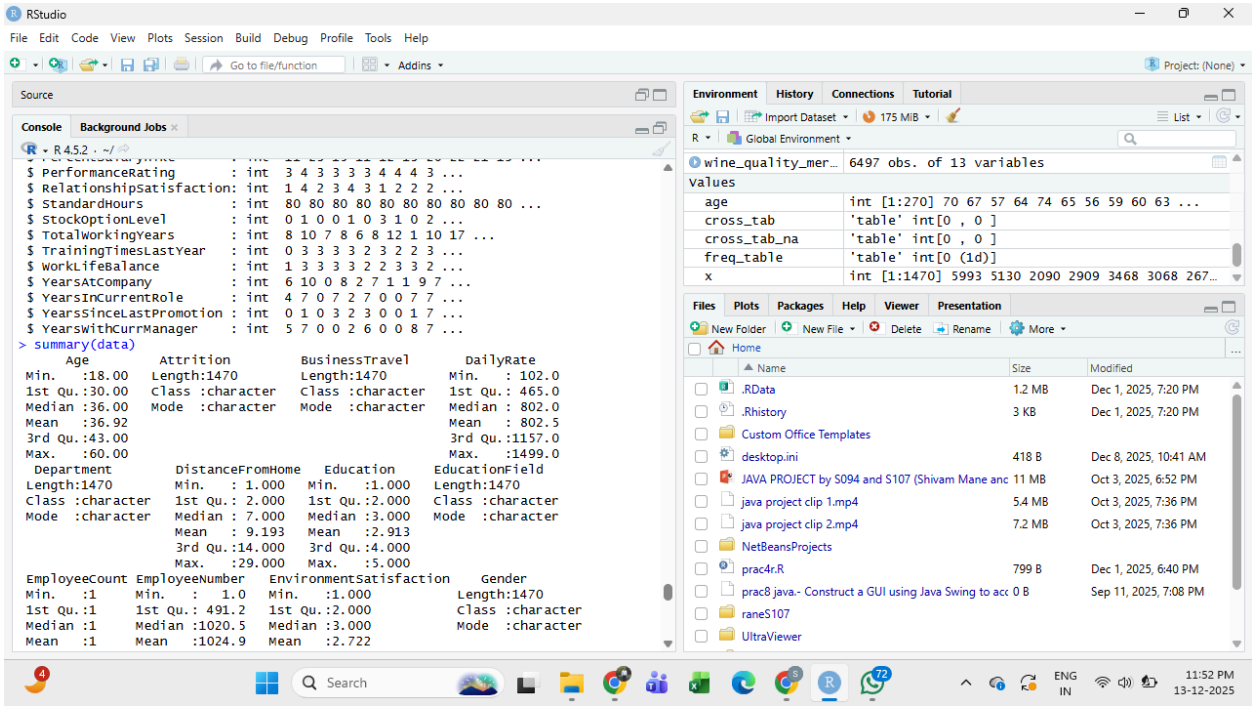
```
# --------------------------------------------
# ONE-SAMPLE T-TEST IN R
# --------------------------------------------

# Clear workspace
rm(list = ls())

# --------------------------------------------
# STEP 1: Load required libraries
# --------------------------------------------
# (No extra packages required for t.test)

# --------------------------------------------
# STEP 2: Import data (CSV file)
# --------------------------------------------
# Replace file path and column name as needed
data <- read.csv("C:/Users/vibro/Downloads/HR_Analytics.csv.csv")

# View structure
str(data)
summary(data)

# --------------------------------------------
# STEP 3: Select the variable for test
# --------------------------------------------
# Example: Monthly Income
x <- data$MonthlyIncome

# Remove missing values
x <- na.omit(x)

# --------------------------------------------

# --------------------------------------------
# STEP 5: One-Sample t-Test
# --------------------------------------------
# H0: Mean Monthly Income = 6500

t_test_re
```

OUTPUT:

RIYA RANE
S107 SYCS

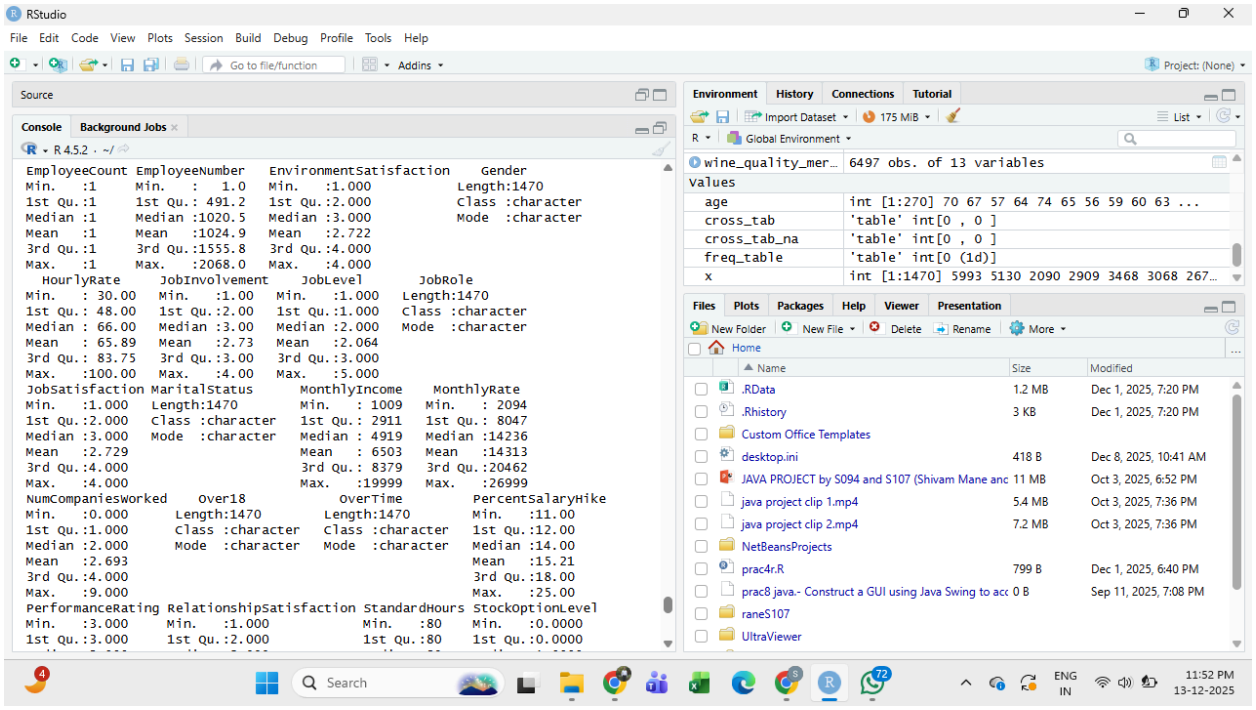
SHETH L.U.J AND SIR M.V COLLEGE

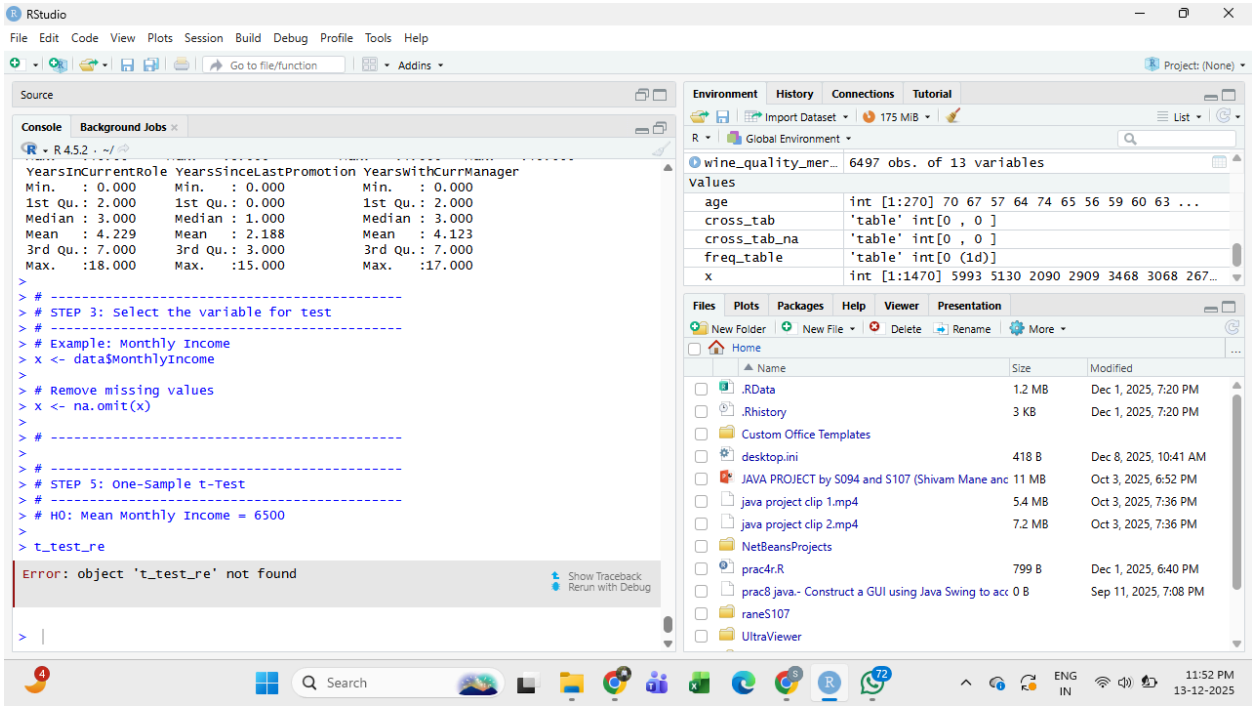




RIYA RANE
S107 SYCS

SHETH L.U.J AND SIR M.V COLLEGE





PRACTICAL NO:5

PERFORMING INDEPENDENT TWO SAMPLE T-TESTS USING T.TESTS() WITH GROUPING

RIYA RANE
S107 SYCS

SHETH L.U.J AND SIR M.V COLLEGE

CODE:

```
# ---------------------------------------------
# INDEPENDENT TWO SAMPLE T-TEST USING t.test()
# ---------------------------------------------

# 1. Create / Load Data
# Example dataset
data <- data.frame(
  Score = c(45, 50, 48, 52, 47, 60, 62, 58, 65, 61),
  Group = c("A", "A", "A", "A", "A", "B", "B", "B", "B", "B")
)

# View data
print(data)

# ---------------------------------------------
# 2. Check structure
# ---------------------------------------------
str(data)

# ---------------------------------------------
# 3. Descriptive statistics
# ---------------------------------------------
aggregate(Score ~ Group, data = data,
          FUN = function(x) c(
            Mean = mean(x),
            SD = sd(x),
            N = length(x)
          ))

# ---------------------------------------------
# 4. Normality Test (Shapiro-Wilk)
# ---------------------------------------------
shapiro.test(data$Score[data$Group == "A"])
shapiro.test(data$Score[data$Group == "B"])

# ---------------------------------------------
# 5. Equality of Variance Test (F-test)
# ---------------------------------------------
var.test(Score ~ Group, data = data)

# ---------------------------------------------
```

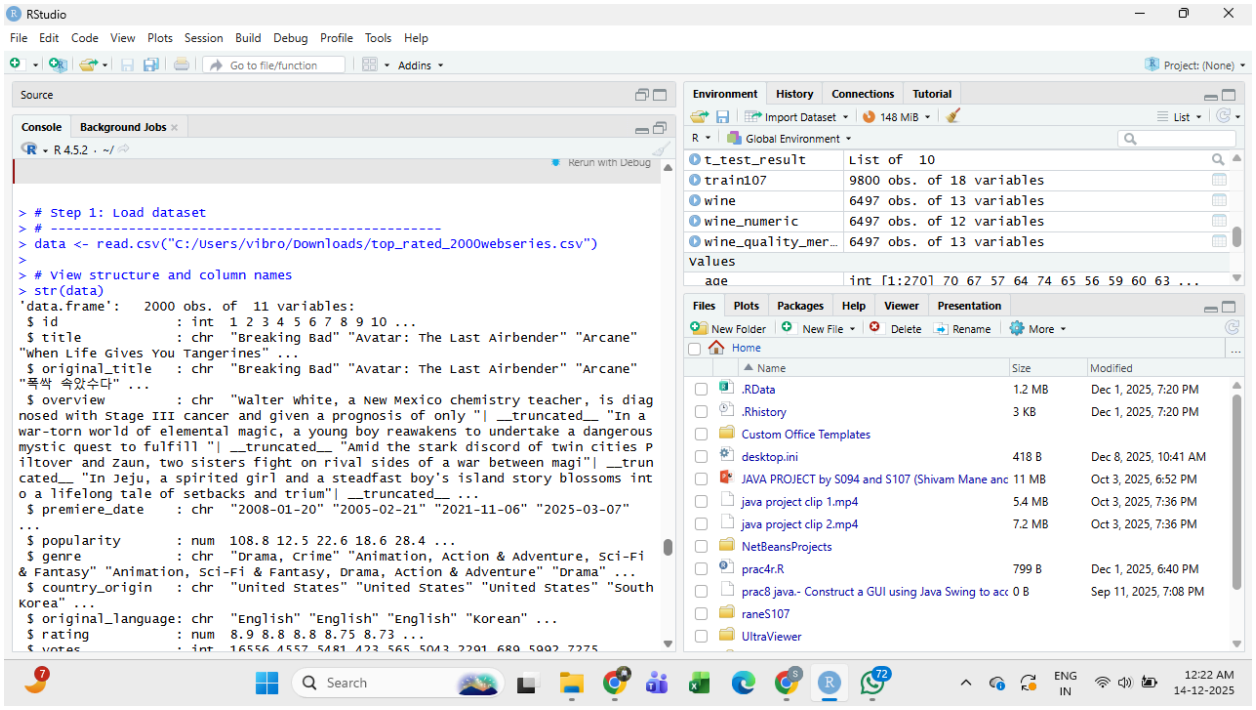
RIYA RANE
S107 SYCS

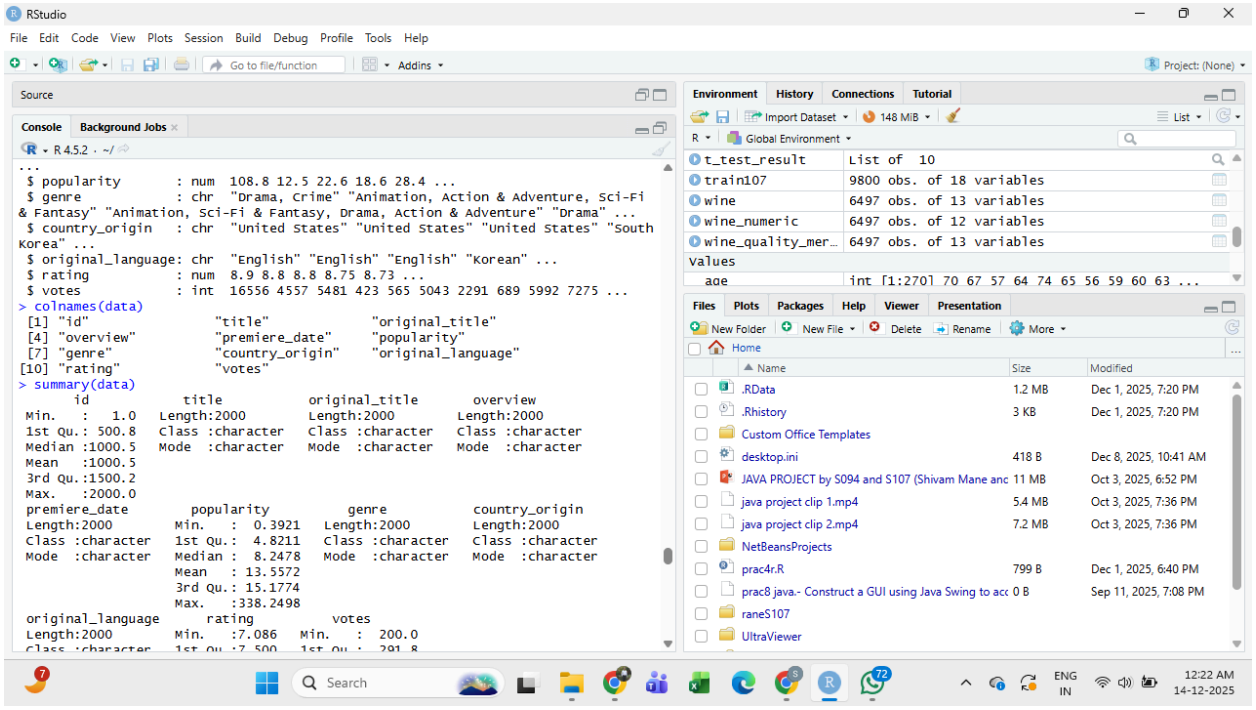
```
# 6. Independent Two Sample t-test
# ------------------------------------------

# 6a. Welch t-test (Unequal variance) - DEFAULT
t_test_result <- t.test(Score ~ Group, data = data)

# Print result
print(t_test_result)

# ------------------------------------------
# 6b. Equal variance assumed
# ------------------------------------------
t_test_equal_var <- t.test(Score ~ Group, data = data, var.equal = TRUE)

print(t_test_equal_var)

# ------------------------------------------
# 7. Manual Method (Subsetting)
# ------------------------------------------
group_A <- data$Score[data$Group == "A"]
group_B <- data$Score[data$Group == "B"]

t.test(group_A, group_B)

# ------------------------------------------
# END OF CODE
# ------------------------------------------
```

OUTPUT:

RIYA RANE
S107 SYCS

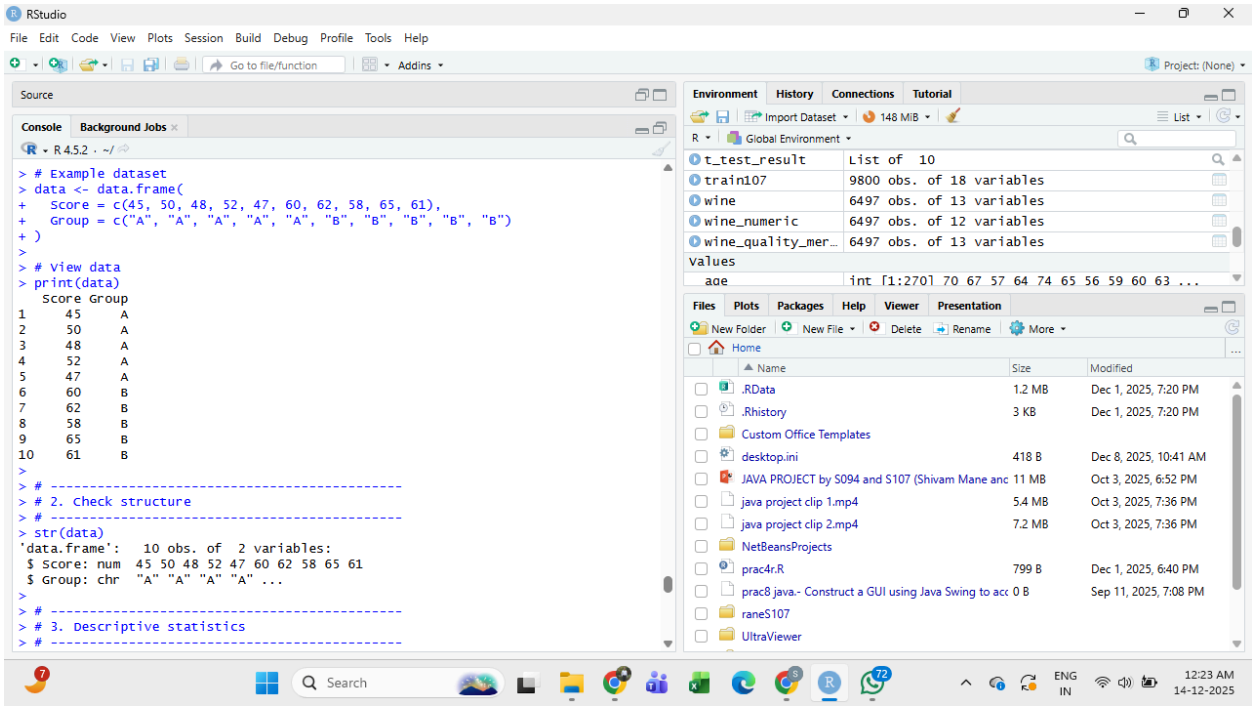
SHETH L.U.J AND SIR M.V COLLEGE



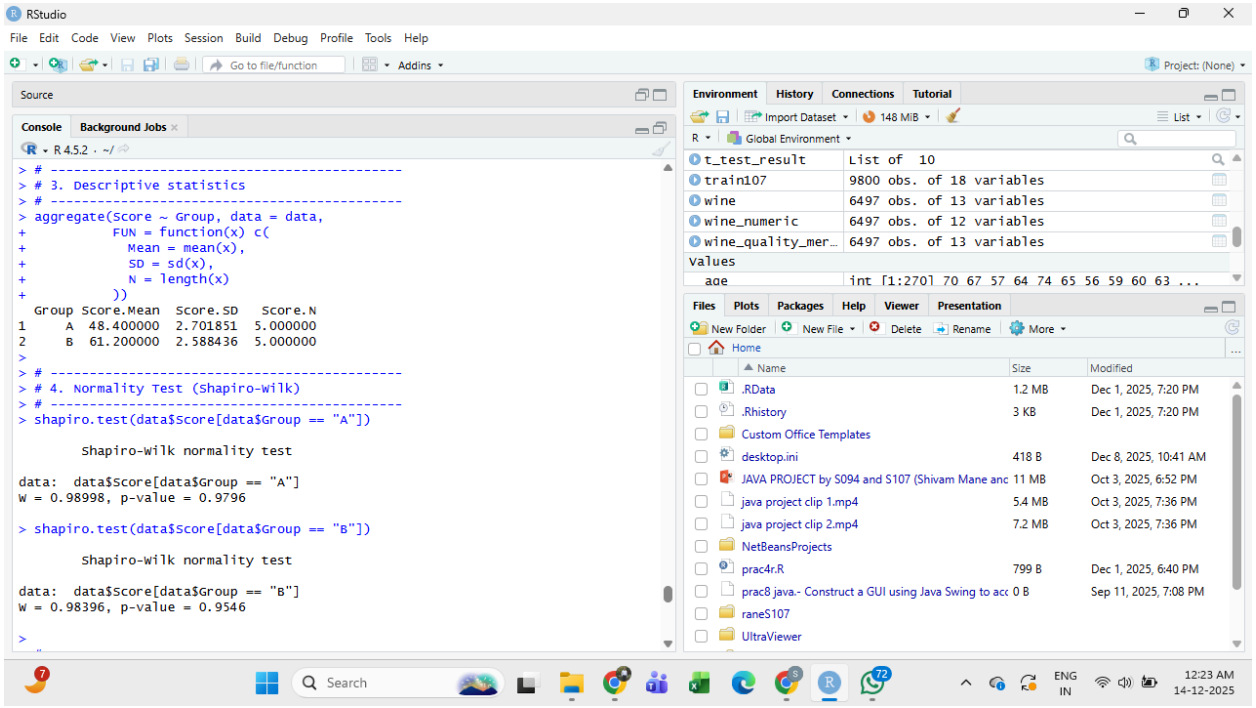


RIYA RANE
S107 SYCS

SHETH L.U.J AND SIR M.V COLLEGE



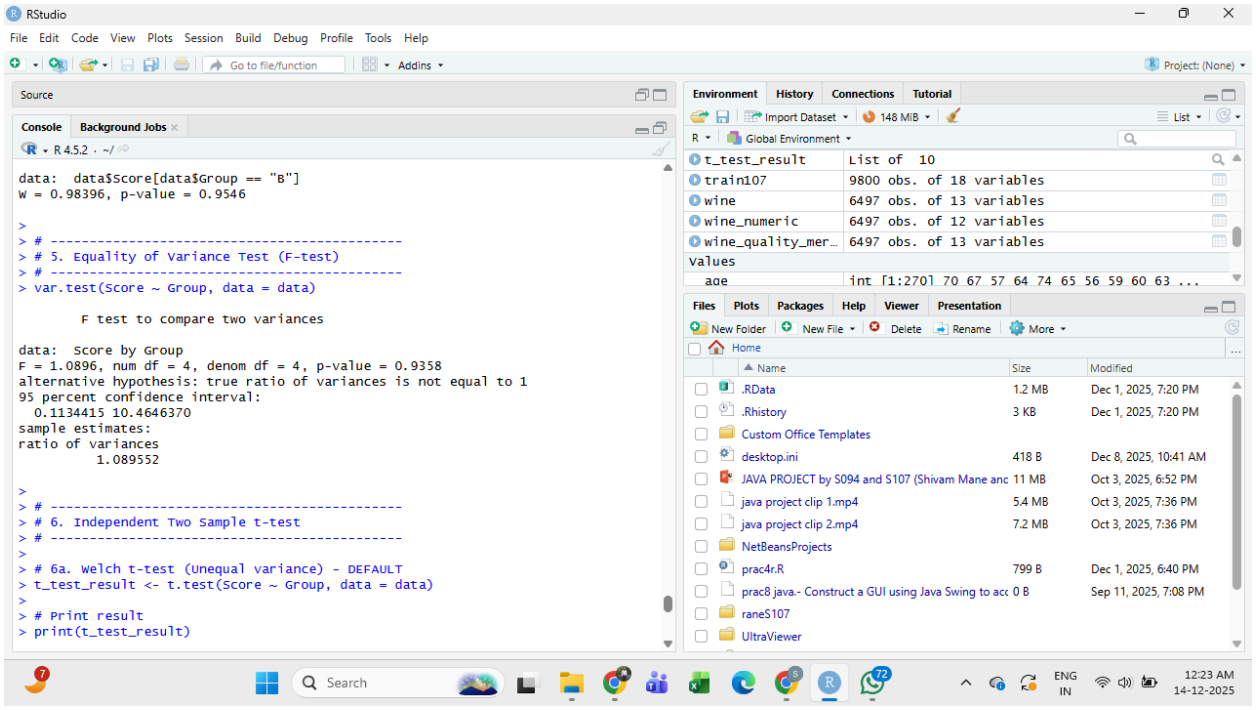
```
> # Example dataset
> data <- data.frame(
+   Score = c(45, 50, 48, 52, 47, 60, 62, 58, 65, 61),
+   Group = c("A", "A", "A", "A", "A", "B", "B", "B", "B", "B")
+ )
>
> # View data
> print(data)
   Score Group
1     45     A
2     50     A
3     48     A
4     52     A
5     47     A
6     60     B
7     62     B
8     58     B
9     65     B
10    61     B
>
> # ---------------------------------------------
> # 2. Check structure
> # ---------------------------------------------
> str(data)
'data.frame':   10 obs. of  2 variables:
 $ Score: num  45 50 48 52 47 60 62 58 65 61
 $ Group: chr  "A" "A" "A" "A" ...
>
> # ---------------------------------------------
> # 3. Descriptive statistics
> # ---------------------------------------------
```



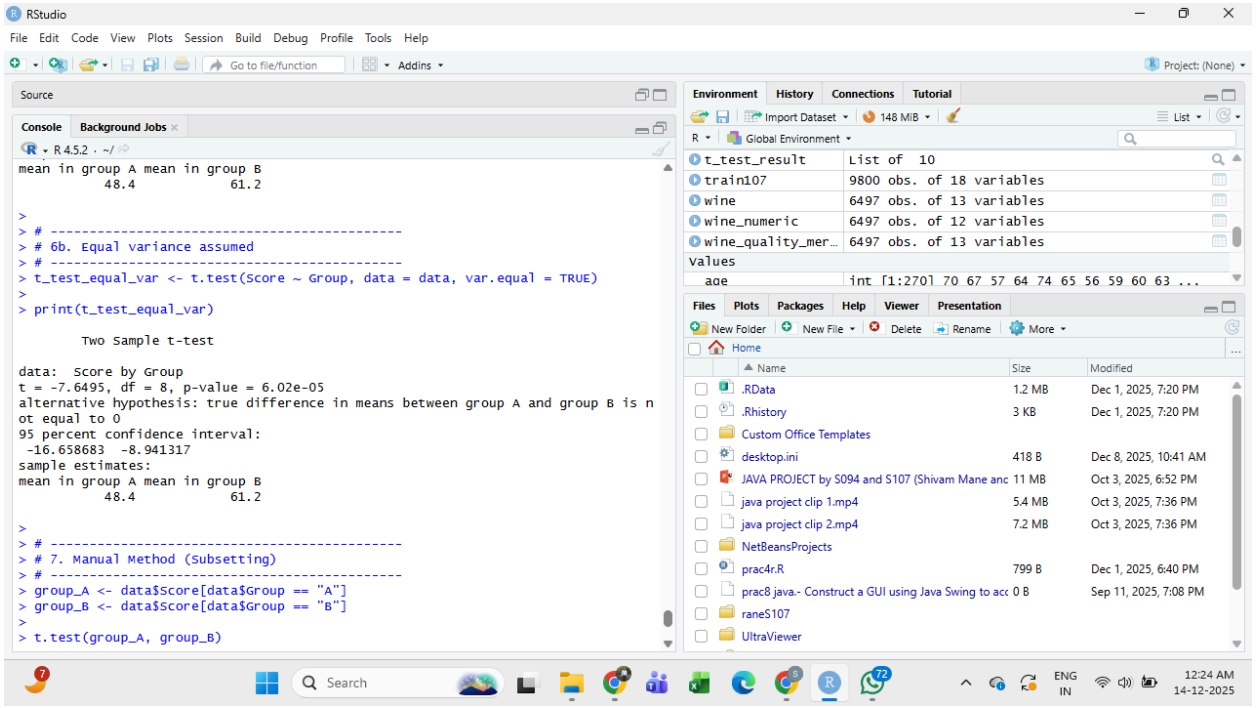
```
> # ---------------------------------------------
> # 3. Descriptive statistics
> # ---------------------------------------------
> aggregate(Score ~ Group, data = data,
+           FUN = function(x) c(
+             Mean = mean(x),
+             SD = sd(x),
+             N = length(x)
+           ))
  Group Score.Mean  Score.SD   Score.N
1     A  48.400000  2.701851  5.000000
2     B  61.200000  2.588436  5.000000
>
> # ---------------------------------------------
> # 4. Normality Test (Shapiro-Wilk)
> # ---------------------------------------------
> shapiro.test(data$Score[data$Group == "A"])

        Shapiro-Wilk normality test

data:  data$Score[data$Group == "A"]
W = 0.98998, p-value = 0.9796

> shapiro.test(data$Score[data$Group == "B"])

        Shapiro-Wilk normality test

data:  data$Score[data$Group == "B"]
W = 0.98396, p-value = 0.9546

>
```

RIYA RANE
S107 SYCS

SHETH L.U.J AND SIR M.V COLLEGE



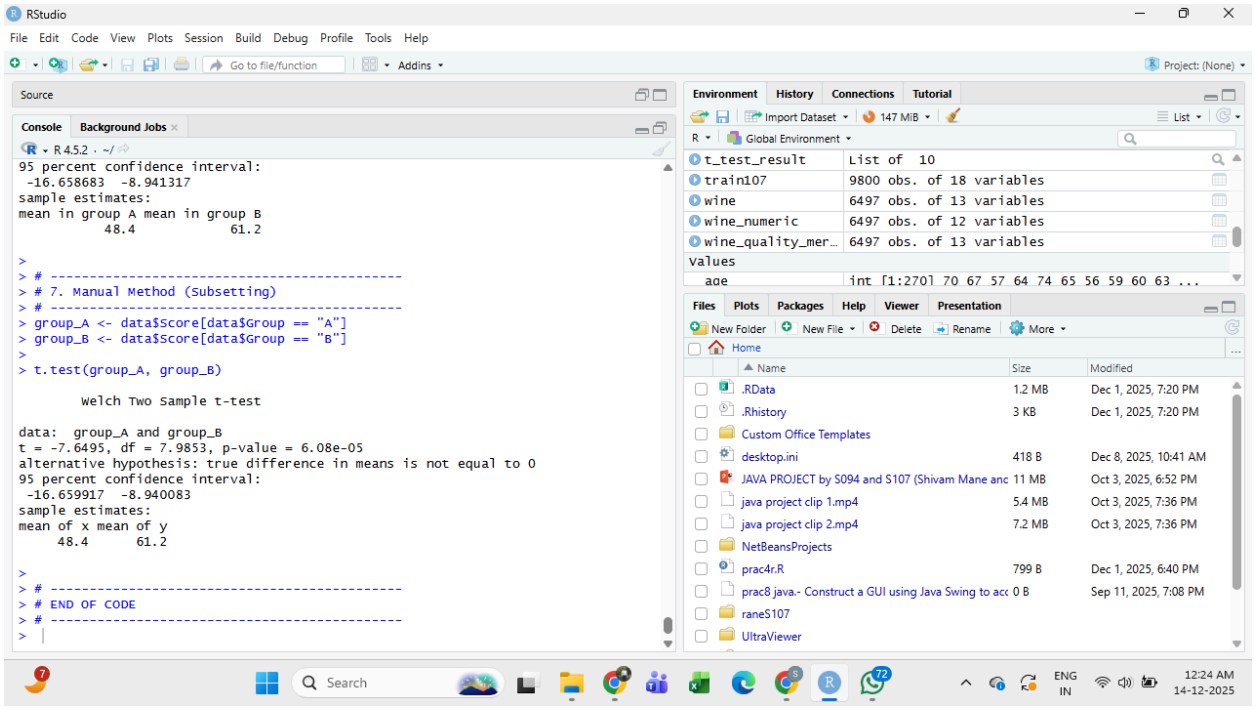
```
data:  data$Score[data$Group == "B"]
W = 0.98396, p-value = 0.9546

>
> # ------------------------------------------------
> # 5. Equality of Variance Test (F-test)
> # ------------------------------------------------
> var.test(Score ~ Group, data = data)

        F test to compare two variances

data:  Score by Group
F = 1.0896, num df = 4, denom df = 4, p-value = 0.9358
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
  0.1134415 10.4646370
sample estimates:
ratio of variances
          1.089552

>
> # ------------------------------------------------
> # 6. Independent Two Sample t-test
> # ------------------------------------------------
>
> # 6a. Welch t-test (Unequal variance) - DEFAULT
> t_test_result <- t.test(Score ~ Group, data = data)
>
> # Print result
> print(t_test_result)
```



```
mean in group A mean in group B
           48.4            61.2

>
> # ------------------------------------------------
> # 6b. Equal variance assumed
> # ------------------------------------------------
> t_test_equal_var <- t.test(Score ~ Group, data = data, var.equal = TRUE)
>
> print(t_test_equal_var)

        Two Sample t-test

data:  Score by Group
t = -7.6495, df = 8, p-value = 6.02e-05
alternative hypothesis: true difference in means between group A and group B is n
ot equal to 0
95 percent confidence interval:
 -16.658683  -8.941317
sample estimates:
mean in group A mean in group B
           48.4            61.2

>
> # ------------------------------------------------
> # 7. Manual Method (Subsetting)
> # ------------------------------------------------
> group_A <- data$Score[data$Group == "A"]
> group_B <- data$Score[data$Group == "B"]
>
> t.test(group_A, group_B)
```

RIYA RANE
S107 SYCS

SHETH L.U.J AND SIR M.V COLLEGE



PRACTICAL NO: 6

PERFORMING PAIRED T.TESTS USING T.TESTS(PAIRED=TRUE)

CODE:

```
# 1. Create Own Dataset (Before–After Example)
data <- data.frame(
  Subject = 1:10,
  Before = c(65, 70, 68, 72, 66, 75, 78, 74, 80, 77),
  After  = c(70, 75, 72, 78, 71, 80, 82, 79, 85, 83)
)

# View data
print(data)

# ----------
# 2. Check structure
# ----------
str(data)

# ----------
# 3. Descriptive statistics
# ----------
```

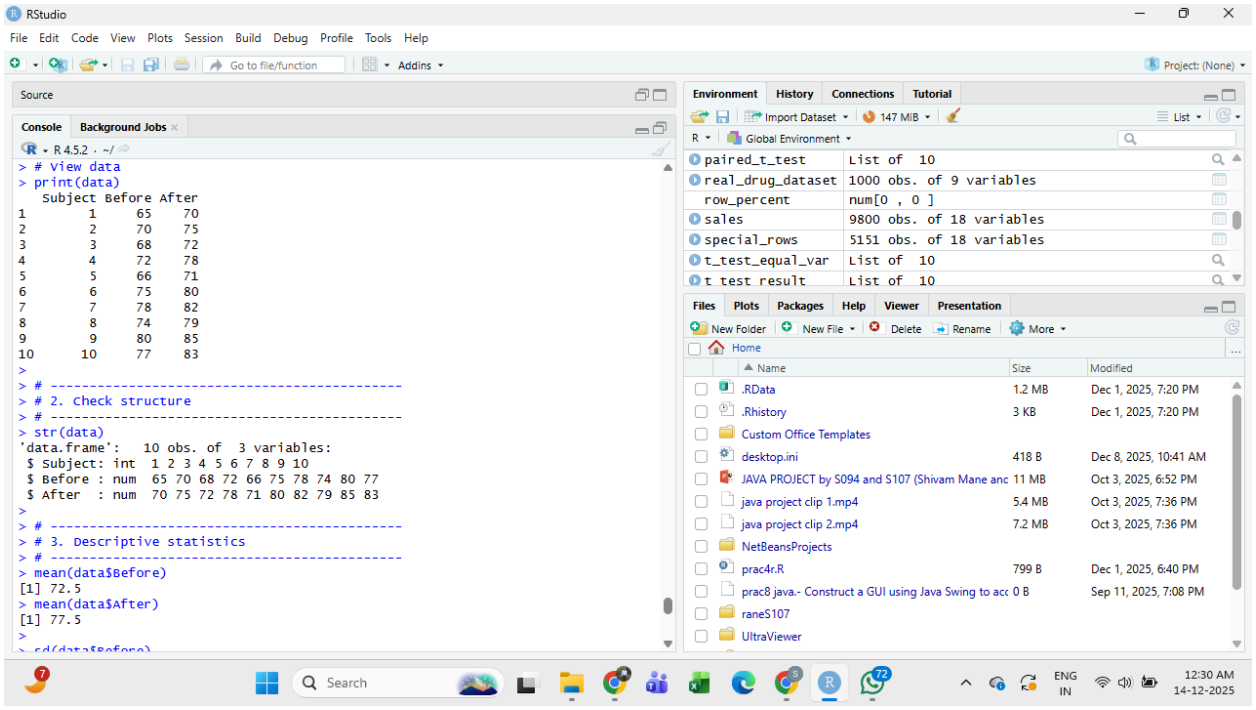
RIYA RANE
S107 SYCS

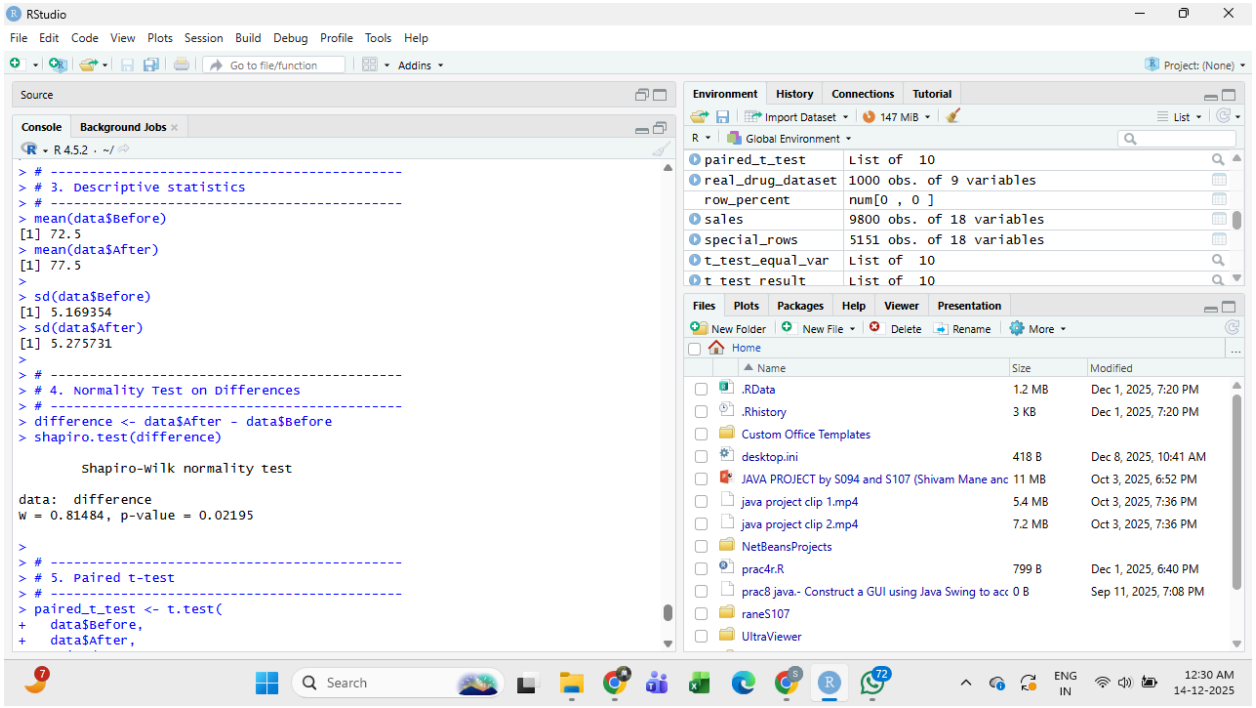
```
mean(data$Before)
mean(data$After)

sd(data$Before)
sd(data$After)

# ---------------------------------------------
# 4. Normality Test on Differences
# ---------------------------------------------
difference <- data$After - data$Before
shapiro.test(difference)

# ---------------------------------------------
# 5. Paired t-test
# ---------------------------------------------
paired_t_test <- t.test(
  data$Before,
  data$After,
  paired = TRUE
)

# Print result
print(paired_t_test)

# ---------------------------------------------
# 6. Alternative way (Formula method)
# ---------------------------------------------
# Convert to long format
long_data <- data.frame(
  Score = c(data$Before, data$After),
  Time = rep(c("Before", "After"), each = nrow(data))
)

t.test(Score ~ Time, data = long_data, paired = TRUE)

# ---------------------------------------------
# END OF CODE
# ---------------------------------------------
```

OUTPUT:

RIYA RANE
S107 SYCS

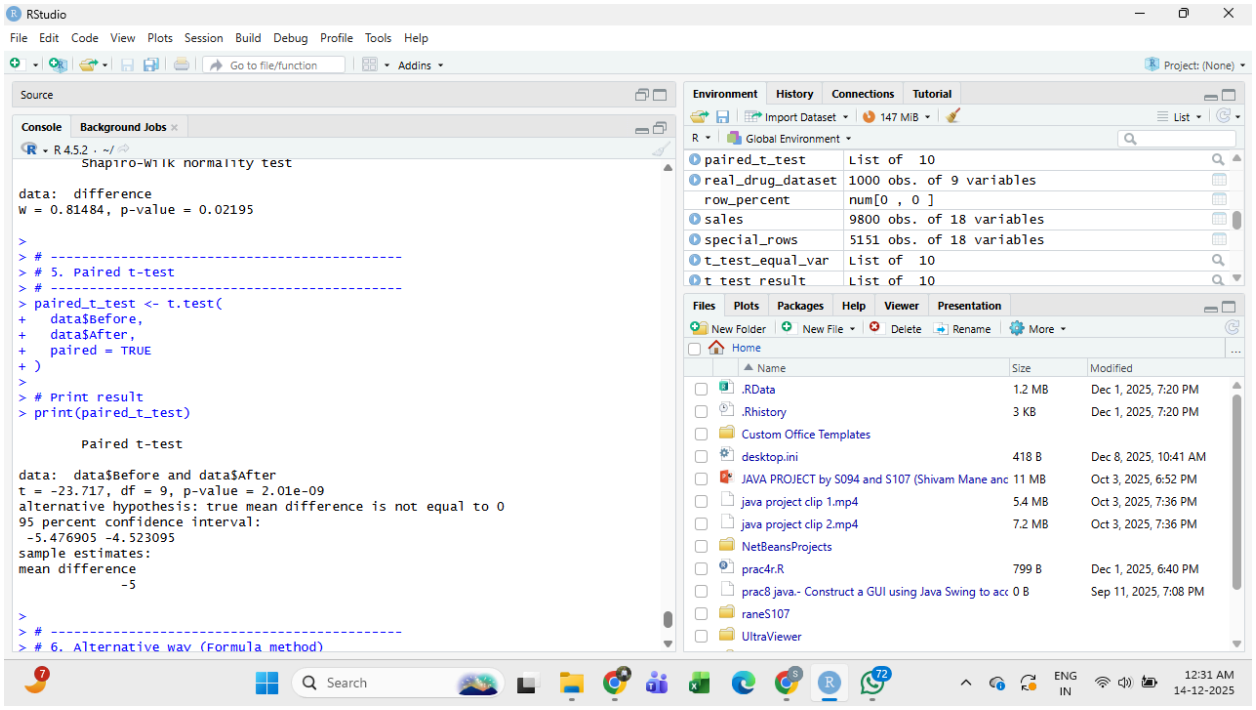
SHETH L.U.J AND SIR M.V COLLEGE

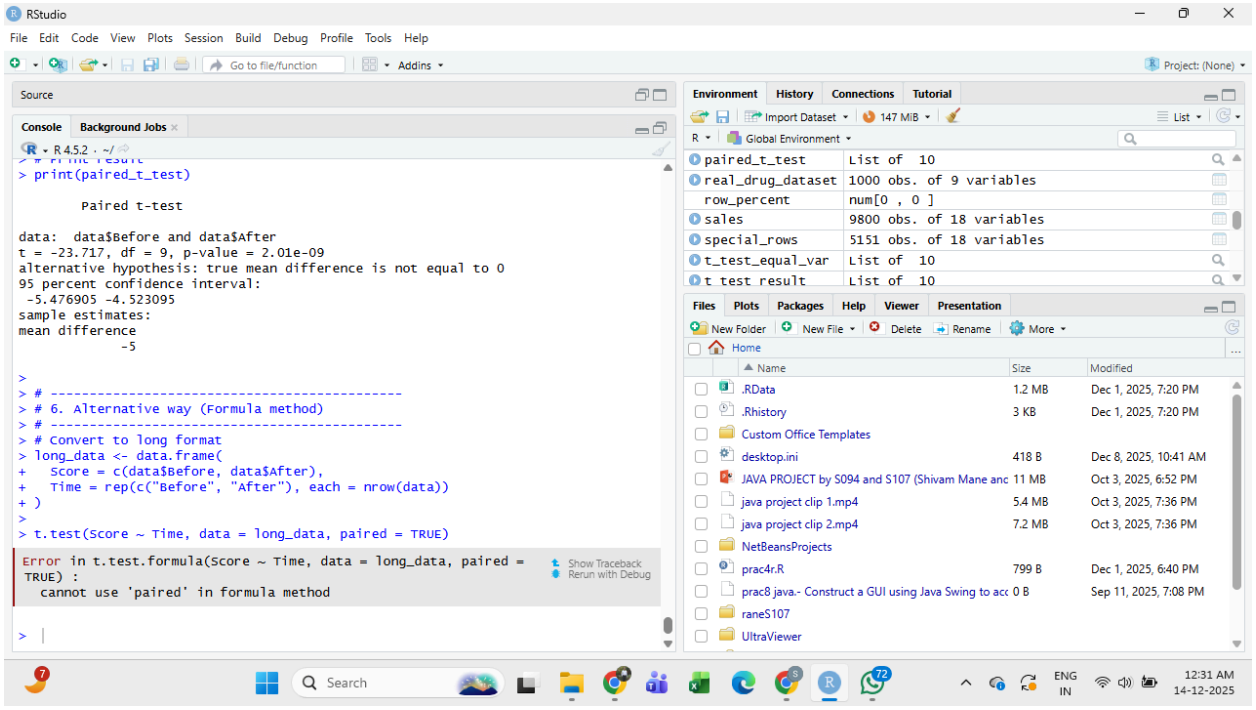




RIYA RANE
S107 SYCS

SHETH L.U.J AND SIR M.V COLLEGE





RIYA RANE
S107 SYCS