

SHETH L.U.J AND SIR M.V COLLEGE
PRACTICAL- M2 1,2,3,4,5,6
SUBJECT - DATA ANALYSIS

PRAC 1

AIM- Generating descriptive statistics using summary() or describe()

OUTPUT -

The screenshot shows the RStudio IDE interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help, and Addins. The left sidebar has tabs for Source, Environment, History, Connections, and Tutorial. The Source tab displays a session log with R code and its output. The Environment tab lists various datasets loaded into the global environment, each with its size and variable count. The bottom right corner shows RStudio resources like Learning & Online, CRAN Task Views, and Getting Help with R.

```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
R 4.5.2 - ~/ ...
The downloaded binary packages are in
  C:\users\itlab\appdata\Local\Temp\rtmp4wvphx\downloaded_packages
> library(dplyr)

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':
  filter, lag

The following objects are masked from 'package:base':
  intersect, setdiff, setequal, union

> library(psych)
> print("---- 1. DESCRIPTIVE STATISTICS ----")
[1] "---- 1. DESCRIPTIVE STATISTICS ----"
> df <- read.csv("walmart_sales.csv")
>

> # PRE-PROCESSING: Create a grouping variable
> # Classify Holiday vs Non-Holiday weeks
> df$Holiday_Group <- ifelse(df$Holiday_Flag == 1, "Holiday", "Non-Holiday")
>
> # 1. PRACTICAL: Generating descriptive statistics using summary() or describe()
>
> print("---- 1. DESCRIPTIVE STATISTICS ----")
[1] "---- 1. DESCRIPTIVE STATISTICS ----"
>
> # A. Using base R summary()
> print("Summary of weekly sales:")
[1] "Summary of weekly sales:"
> summary(df$weekly_sales)
   Min. 1st Qu. Median Mean 3rd Qu. Max.
209986 55330 960746 1046965 1420159 3818687
>
> # B. Using psych::describe()
> print("Detailed Description of Temperature:")
[1] "Detailed Description of Temperature:"
> describe(df$Temperature)
   vars n mean sd median trimmed mad min max range skew kurtosis se
x1  1 6435 60.66 18.44  62.67  61.45 20.3 -2.06 100.14 102.2 -0.34 -0.61 0.23
>
```

Environment

Object	Description
category_pivot	6435 obs. of 46 variables
clean_exact	7 obs. of 3 variables
combined_data	10150 obs. of 2 variables
data	6435 obs. of 8 variables
data_processed	6435 obs. of 17 variables
data_science_jobs	42 obs. of 7 variables
dates_df	4 obs. of 2 variables
df	6435 obs. of 9 variables
df_clean	6435 obs. of 4 variables
duplicates_report	0 obs. of 4 variables
employee_salary_data	50 obs. of 9 variables
flower_clean	10000 obs. of 2 variables
flower_df	10000 obs. of 4 variables
iris	150 obs. of 5 variables
iris_clean	150 obs. of 2 variables
iris_iris	150 obs. of 5 variables
long_df	12870 obs. of 4 variables
orders_df	7 obs. of 3 variables
penguins	344 obs. of 9 variables

Files Plots Packages Help Viewer Presentation

Home Find in Topic

R Resources

- Learning & Online
- CRAN Task Views
- R on StackOverflow
- Getting Help with R

RStudio

- Posit Support
- Posit Community Forum for the RStudio IDE
- Posit Cheat Sheets
- Posit Packages
- Posit Products

Manuals

25°C Sunny

ENGLISH IN 15-12-2025

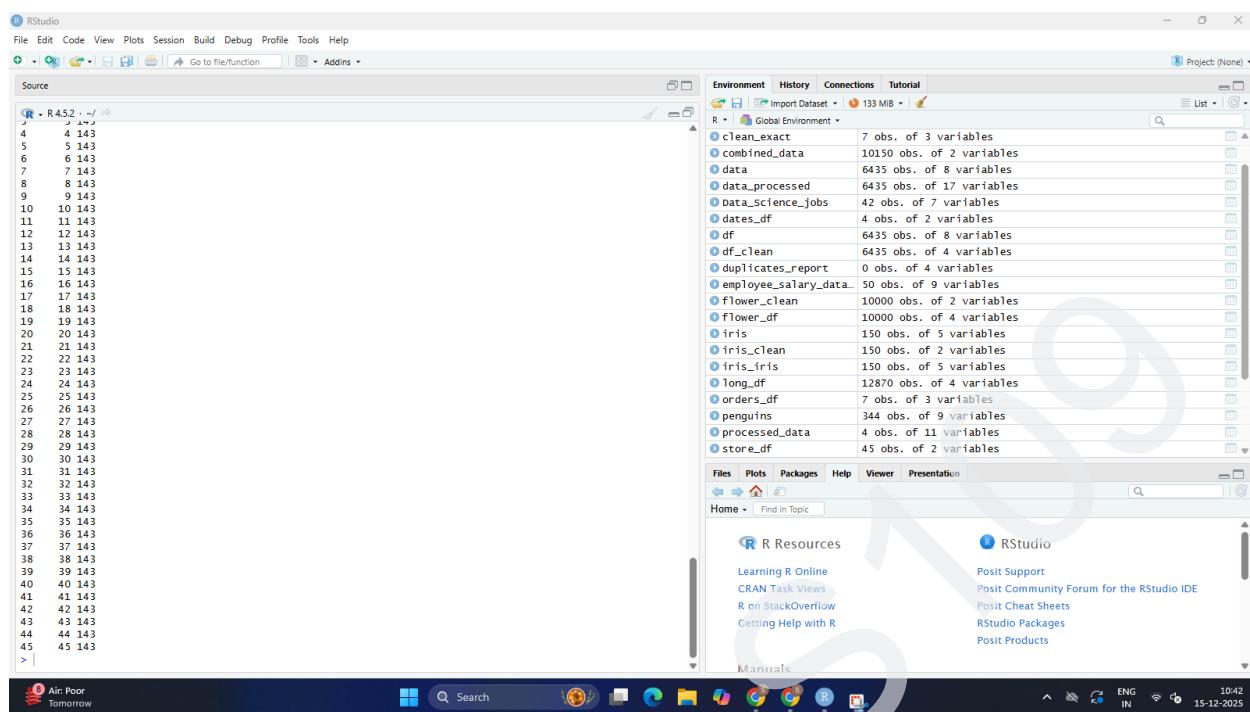
PRAC 2

AIM -Generating frequency tables using table() or count()

OUTPUT

NAME - UNNATI RATHOD
ROLL NO - S109

SHETH L.U.J AND SIR M.V COLLEGE
PRACTICAL- M2 1,2,3,4,5,6
SUBJECT - DATA ANALYSIS



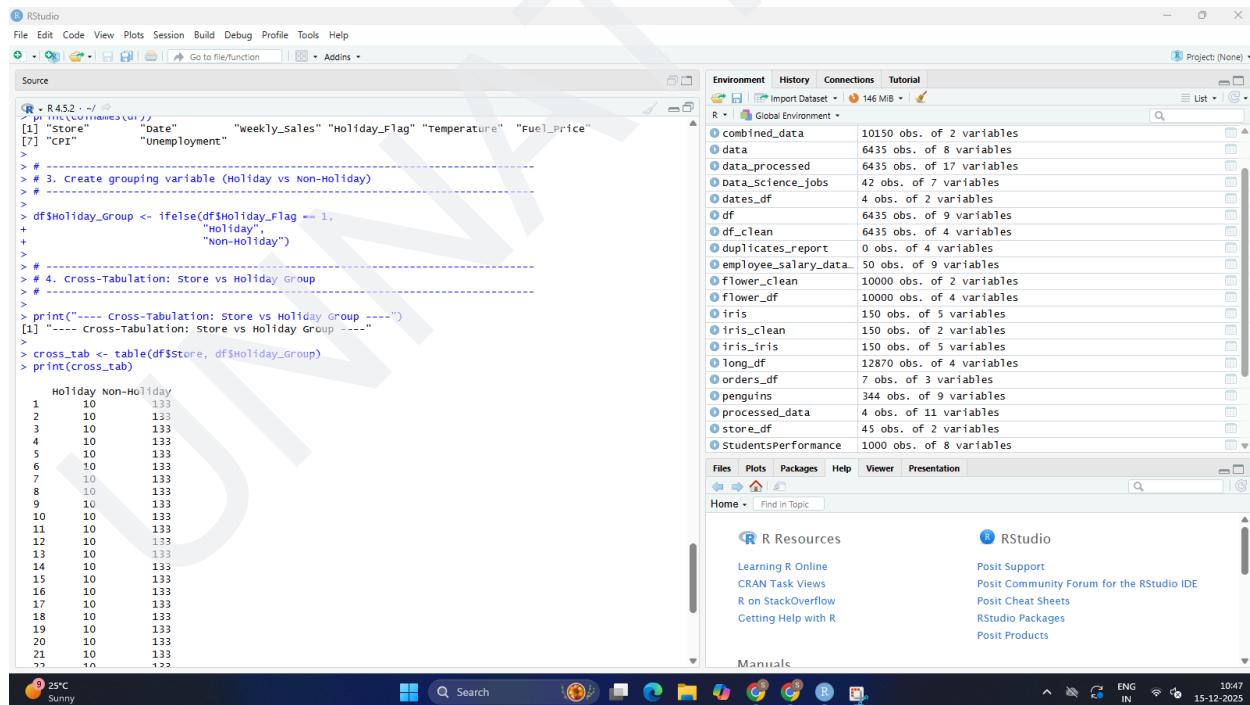
The screenshot shows the RStudio interface with the Global Environment pane open. It lists various datasets with their sizes and variable counts:

- clean_exact: 7 obs. of 3 variables
- combined_data: 10150 obs. of 2 variables
- data: 6435 obs. of 8 variables
- data_processed: 6435 obs. of 17 variables
- data_science_jobs: 42 obs. of 7 variables
- dates_df: 4 obs. of 2 variables
- df: 6435 obs. of 8 variables
- df_clean: 6435 obs. of 4 variables
- duplicates_report: 0 obs. of 4 variables
- employee_salary_data: 50 obs. of 9 variables
- flower_clean: 10000 obs. of 2 variables
- flower_df: 10000 obs. of 4 variables
- iris: 150 obs. of 5 variables
- iris_clean: 150 obs. of 2 variables
- iris_iris: 150 obs. of 5 variables
- long_df: 12870 obs. of 4 variables
- orders_df: 7 obs. of 3 variables
- penguins: 344 obs. of 9 variables
- processed_data: 4 obs. of 11 variables
- store_df: 45 obs. of 2 variables

PRAC3

AIM - Creating cross-tabulations and two-way tables

OUTPUT



The screenshot shows the RStudio interface with the Source pane containing R code and the Global Environment pane showing the resulting cross-tabulation.

```

R 4.5.2 : ~/practicals/m2
> # 1. Load required libraries
> library(tidyverse)
> library(readr)

# 2. Load dataset
> df <- read_csv("C:/Users/Unnati Rathod/Downloads/Store Sales Data.csv")

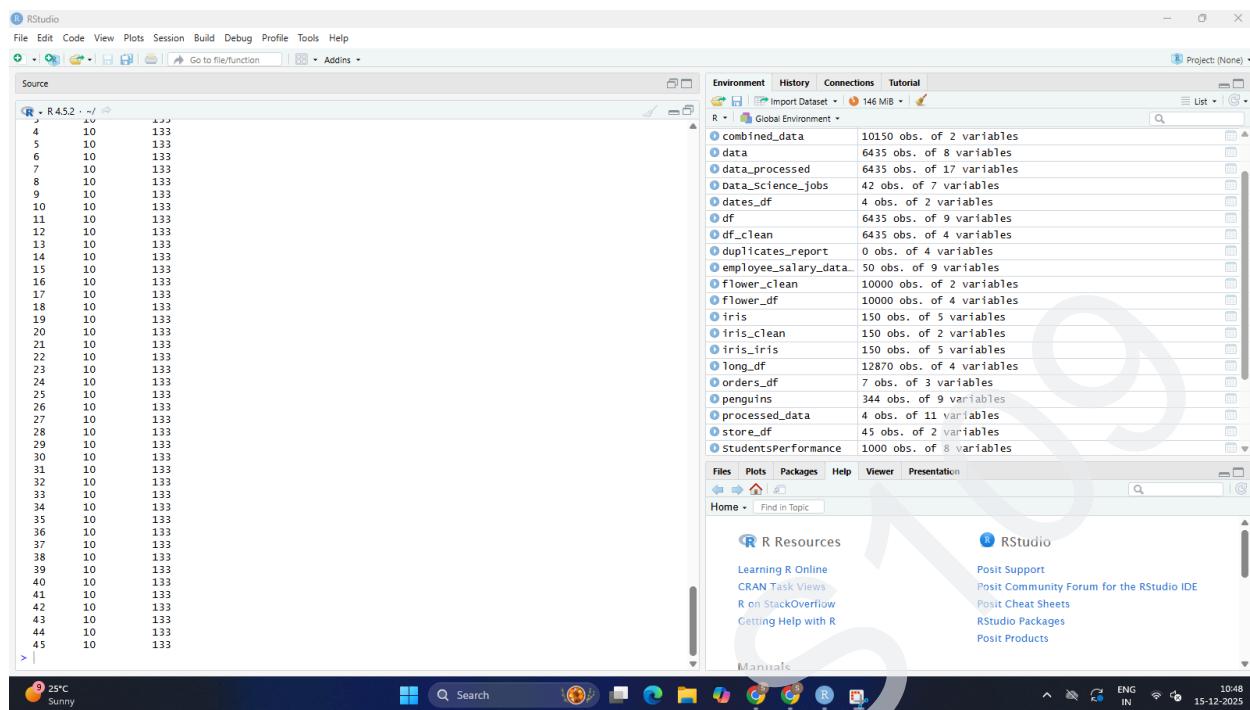
# 3. Create grouping variable (Holiday vs Non-Holiday)
> df$Holiday_Group <- ifelse(df$Holiday_Flag == 1,
+                             "Holiday",
+                             "Non-Holiday")

# 4. Cross-tabulation: Store vs Holiday Group
> cross_tab <- table(df$Store, df$Holiday_Group)
> print(cross_tab)

```

Holiday	Non-Holiday	Count
1	10	133
2	10	133
3	10	133
4	10	133
5	10	133
6	10	133
7	10	133
8	10	133
9	10	133
10	10	133
11	10	133
12	10	133
13	10	133
14	10	133
15	10	133
16	10	133
17	10	133
18	10	133
19	10	133
20	10	133
21	10	133
22	10	133

SHETH L.U.J AND SIR M.V COLLEGE
PRACTICAL- M2 1,2,3,4,5,6
SUBJECT - DATA ANALYSIS



RStudio Environment View showing the Global Environment pane with various datasets loaded. The 'data' dataset has 6435 observations and 8 variables.

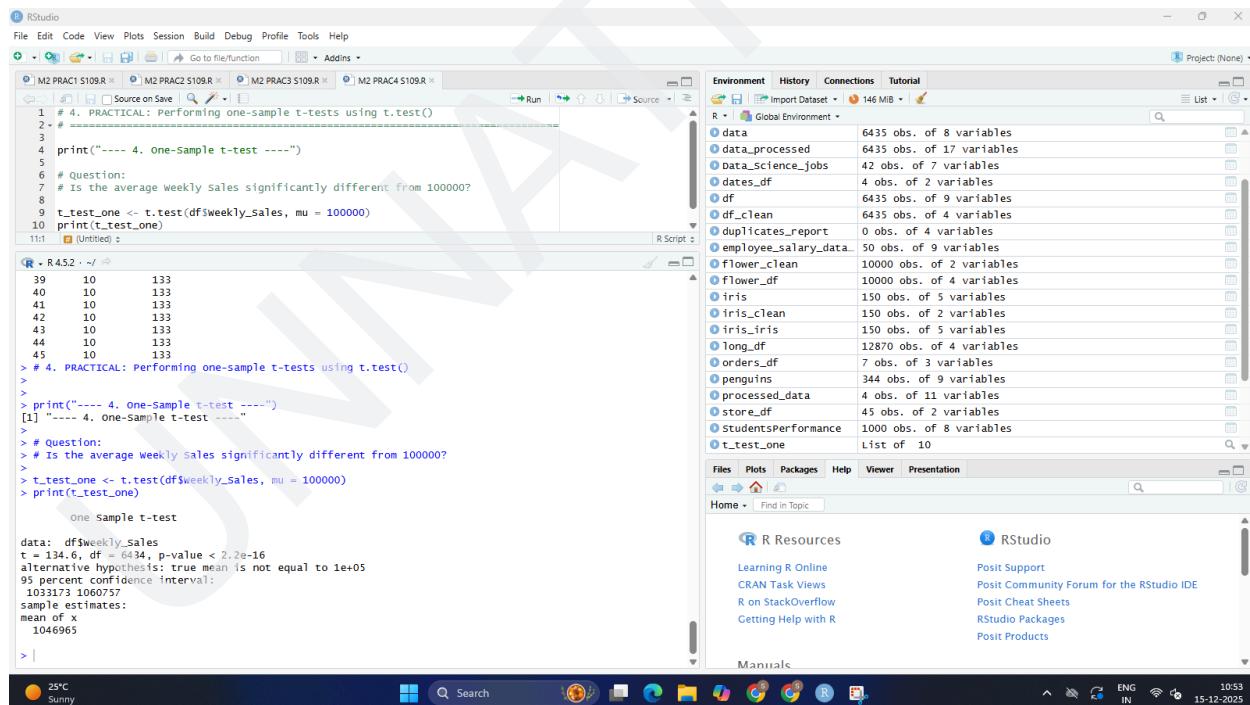
```

R 4.5.2 ~/x.csv
1 10 133
2 10 133
3 10 133
4 10 133
5 10 133
6 10 133
7 10 133
8 10 133
9 10 133
10 10 133
11 10 133
12 10 133
13 10 133
14 10 133
15 10 133
16 10 133
17 10 133
18 10 133
19 10 133
20 10 133
21 10 133
22 10 133
23 10 133
24 10 133
25 10 133
26 10 133
27 10 133
28 10 133
29 10 133
30 10 133
31 10 133
32 10 133
33 10 133
34 10 133
35 10 133
36 10 133
37 10 133
38 10 133
39 10 133
40 10 133
41 10 133
42 10 133
43 10 133
44 10 133
45 10 133

```

PRAC4

AIM - Performing one-sample t-tests using `t.test()`



RStudio Environment View showing the Global Environment pane with various datasets loaded. The 'data' dataset has 6435 observations and 8 variables.

```

# 4. PRACTICAL: Performing one-sample t-tests using t.test()
# -----
# Question:
# Is the average weekly sales significantly different from 100000?
t_test_one <- t.test(df$weekly_Sales, mu = 100000)
print(t_test_one)

# 4. PRACTICAL: Performing one-sample t-tests using t.test()
# -----
# Question:
# Is the average weekly sales significantly different from 100000?
t_test_one <- t.test(df$weekly_Sales, mu = 100000)
print(t_test_one)

One Sample t-test
data: df$weekly_Sales
t = 134.6, df = 6434, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 1e+05
95 percent confidence interval:
1033173 1060757
sample estimates:
mean of x
1046905

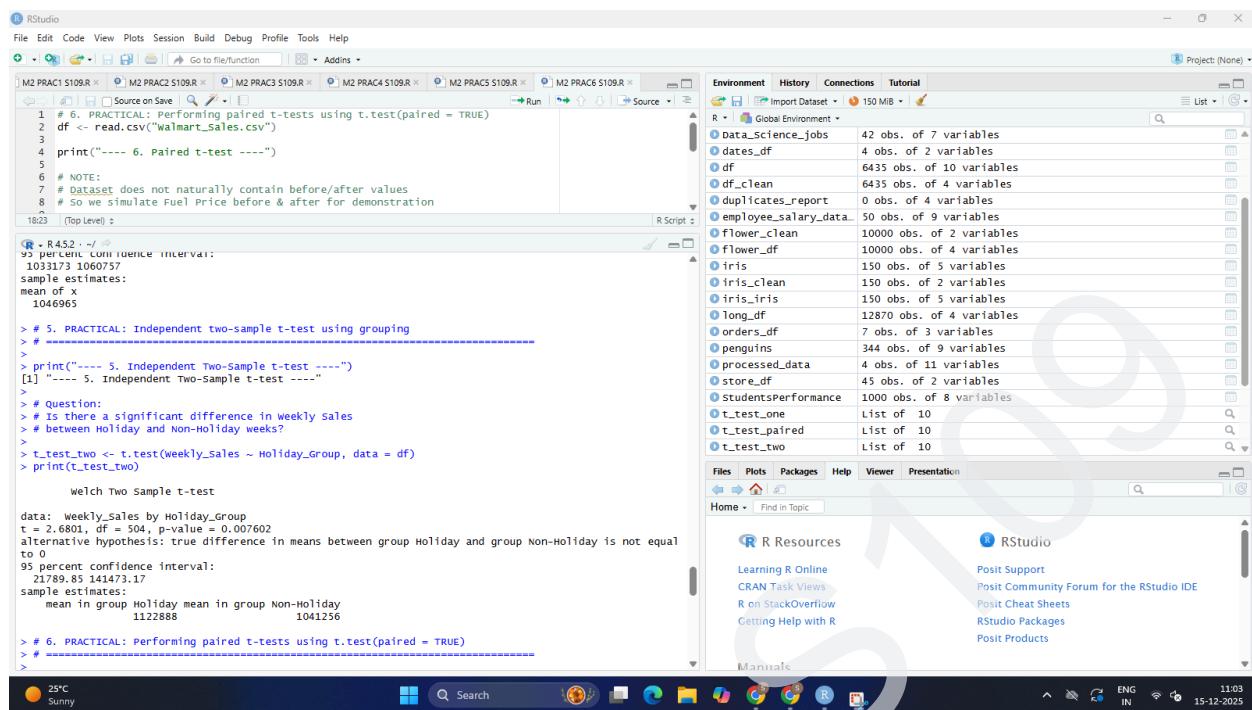
```

PRAC5

AIM - Independent two-sample t-test using grouping
 OUTPUT -

NAME - UNNATI RATHOD
 ROLL NO - S109

SHETH L.U.J AND SIR M.V COLLEGE
PRACTICAL- M2 1,2,3,4,5,6
SUBJECT - DATA ANALYSIS



```

# 6. PRACTICAL: Performing paired t-tests using t.test(paired = TRUE)
df <- read.csv("walmart_sales.csv")
print("---- 6. Paired t-test ----")
# NOTE:
# Dataset does not naturally contain before/after values
# So we simulate Fuel Price before & after for demonstration
# 5. PRACTICAL: Independent two-sample t-test using grouping
# -----
# print("---- 5. Independent Two-Sample t-test ----")
#[1] "---- 5. Independent Two-Sample t-test ----"
# Question:
# Is there a significant difference in weekly sales
# between Holiday and Non-Holiday weeks?
t_test_two <- t.test(weekly_sales ~ Holiday_Group, data = df)
print(t_test_two)

Welch Two Sample t-test

data: weekly_sales by Holiday_Group
t = 2.6801, df = 504, p-value = 0.007602
alternative hypothesis: true difference in means between group Holiday and group Non-Holiday is not equal to 0
95 percent confidence interval:
 21789.82 141473.17
sample estimates:
 mean in group Holiday mean in group Non-Holiday
 1122888               1041256
# 6. PRACTICAL: Performing paired t-tests using t.test(paired = TRUE)
# -----
# 
# R 4.5.2 - / 
# 95 percent confidence interval:
# -0.3020415 -0.2964245
sample estimates:
mean difference
-0.299233

# 6. PRACTICAL: Performing paired t-tests using t.test(paired = TRUE)
df <- read.csv("walmart_sales.csv")
print("---- 6. Paired t-test ----")
#[1] "---- 6. Paired t-test ----"
# NOTE:
# Dataset does not naturally contain before/after values
# So we simulate Fuel Price before & after for demonstration
set.seed(123)
df$Fuel_Price_Before <- df$Fuel_Price + runif(nrow(df), min = 0.1, max = 0.5)
df$Fuel_Price_After <- df$Fuel_Price
# Perform paired t-test
t_test_paired <- t.test(
+ df$Fuel_Price_Before,
+ df$Fuel_Price_After,
+ paired = TRUE
)
print(t_test_paired)

Paired t-test

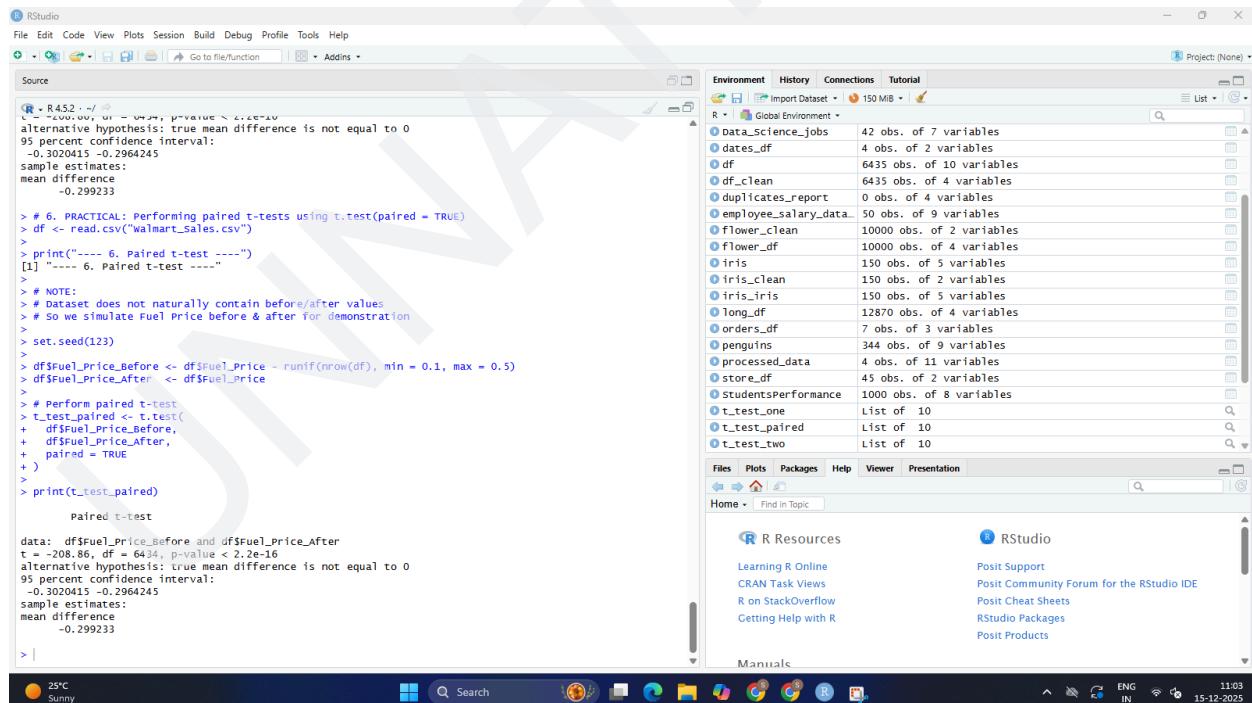
data: df$Fuel_Price_Before and df$Fuel_Price_After
t = -208.86, df = 6434, p-value < 2.2e-16
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
-0.3020415 -0.2964245
sample estimates:
mean difference
-0.299233

```

PRAC6

AIM - Performing paired t-tests using `t.test(paired = TRUE)`

OUTPUT-



```

# 6. PRACTICAL: Performing paired t-tests using t.test(paired = TRUE)
df <- read.csv("walmart_sales.csv")
print("---- 6. Paired t-test ----")
#[1] "---- 6. Paired t-test ----"
# NOTE:
# Dataset does not naturally contain before/after values
# So we simulate Fuel Price before & after for demonstration
set.seed(123)
df$Fuel_Price_Before <- df$Fuel_Price + runif(nrow(df), min = 0.1, max = 0.5)
df$Fuel_Price_After <- df$Fuel_Price
# Perform paired t-test
t_test_paired <- t.test(
+ df$Fuel_Price_Before,
+ df$Fuel_Price_After,
+ paired = TRUE
)
print(t_test_paired)

Paired t-test

data: df$Fuel_Price_Before and df$Fuel_Price_After
t = -208.86, df = 6434, p-value < 2.2e-16
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
-0.3020415 -0.2964245
sample estimates:
mean difference
-0.299233

```