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

PRACTICAL NO:11
11 Reshaping data using pivot_longer()/pivot_wider() (R).

CODE:

```
library(dplyr)
library(tidyr)

#
================================================================================
======
# 1. SETUP: Read Your Dataset


df <- read.csv("C:/Users/info/Downloads/Retail Product_ - Retail Product.csv",
               na.strings = c("", "NA")) %>%
  mutate(ProductID = row_number()) %>%
  select(ProductID, Category, Price, Discount)

print("--- 1. Original Wide Data ---")
print(head(df))

#
================================================================================
======
# 2. PIVOT_LONGER (Wide → Long)
#
================================================================================
======

long_df <- df %>%
  pivot_longer(
    cols = c(Price, Discount),
    names_to = "Metric",
    values_to = "Value"
  )

print("--- 2. Long Format ---")
print(head(long_df, 6))

#
================================================================================
======
# 3. PIVOT_WIDER (Long → Wide)
```

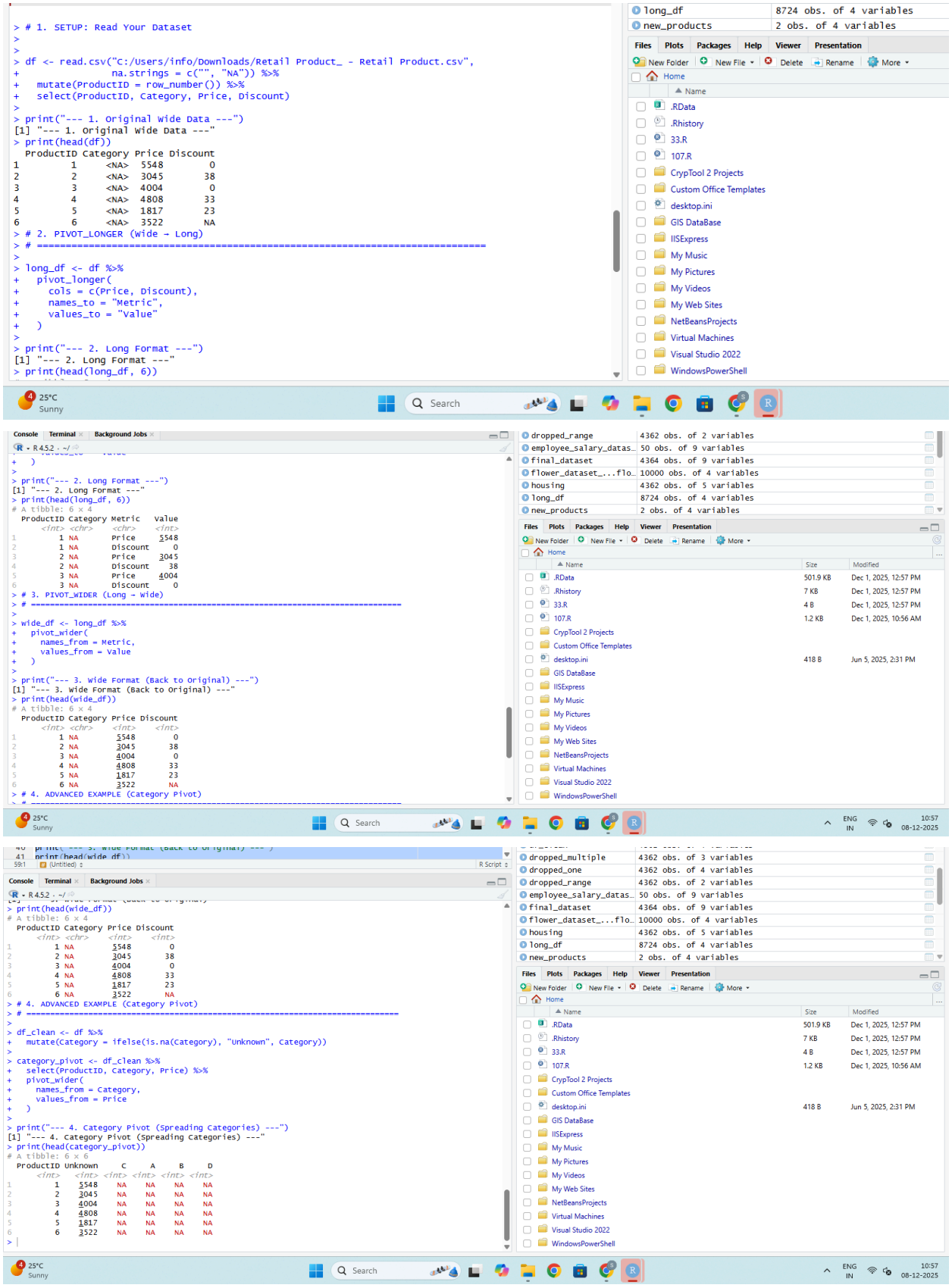
RIYA RANE
S107 SYCS

```
#
================================================================================
======

wide_df <- long_df %>%
  pivot_wider(
    names_from = Metric,
    values_from = Value
  )

print("--- 3. Wide Format (Back to Original) ---")
print(head(wide_df))

#
================================================================================
======
# 4. ADVANCED EXAMPLE (Category Pivot)
#
================================================================================
======

df_clean <- df %>%
  mutate(Category = ifelse(is.na(Category), "Unknown", Category))

category_pivot <- df_clean %>%
  select(ProductID, Category, Price) %>%
  pivot_wider(
    names_from = Category,
    values_from = Price
  )

print("--- 4. Category Pivot (Spreading Categories) ---")
print(head(category_pivot))
```

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:12
12 Combining datasets vertically (concatenation) using rbind() (R).

CODE:

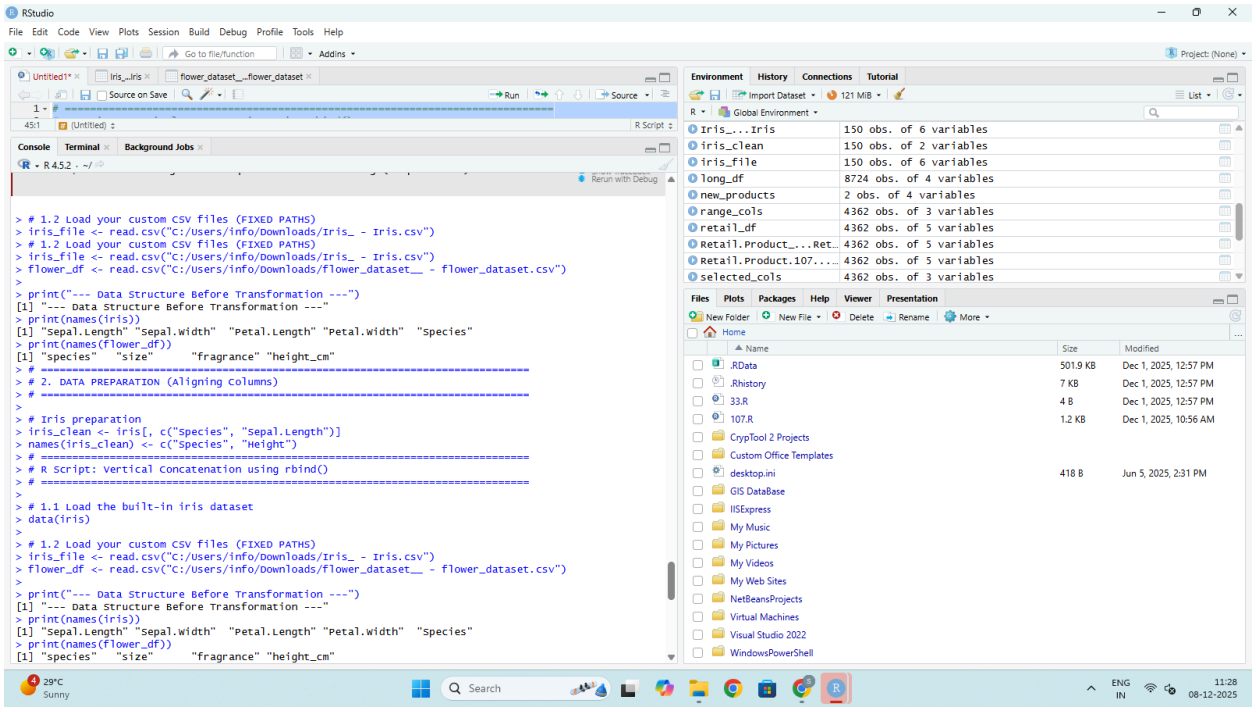
```
#
================================================================================
# R Script: Vertical Concatenation using rbind()
#
================================================================================

# 1.1 Load the built-in iris dataset
data(iris)

# 1.2 Load your custom CSV files (FIXED PATHS)
iris_file <- read.csv("C:/Users/info/Downloads/Iris_ - Iris.csv")
flower_df <- read.csv("C:/Users/info/Downloads/flower_dataset__ - flower_dataset.csv")

print("--- Data Structure Before Transformation ---")
print(names(iris))
print(names(flower_df))

#
================================================================================
# 2. DATA PREPARATION (Aligning Columns)
#
================================================================================

# Iris preparation
iris_clean <- iris[, c("Species", "Sepal.Length")]
names(iris_clean) <- c("Species", "Height")

# Flower dataset preparation
flower_clean <- flower_df[, c("species", "height_cm")]
names(flower_clean) <- c("Species", "Height")

# Convert to numeric
iris_clean$Height <- as.numeric(iris_clean$Height)
flower_clean$Height <- as.numeric(flower_clean$Height)
```

RIYA RANE
S107 SYCS

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

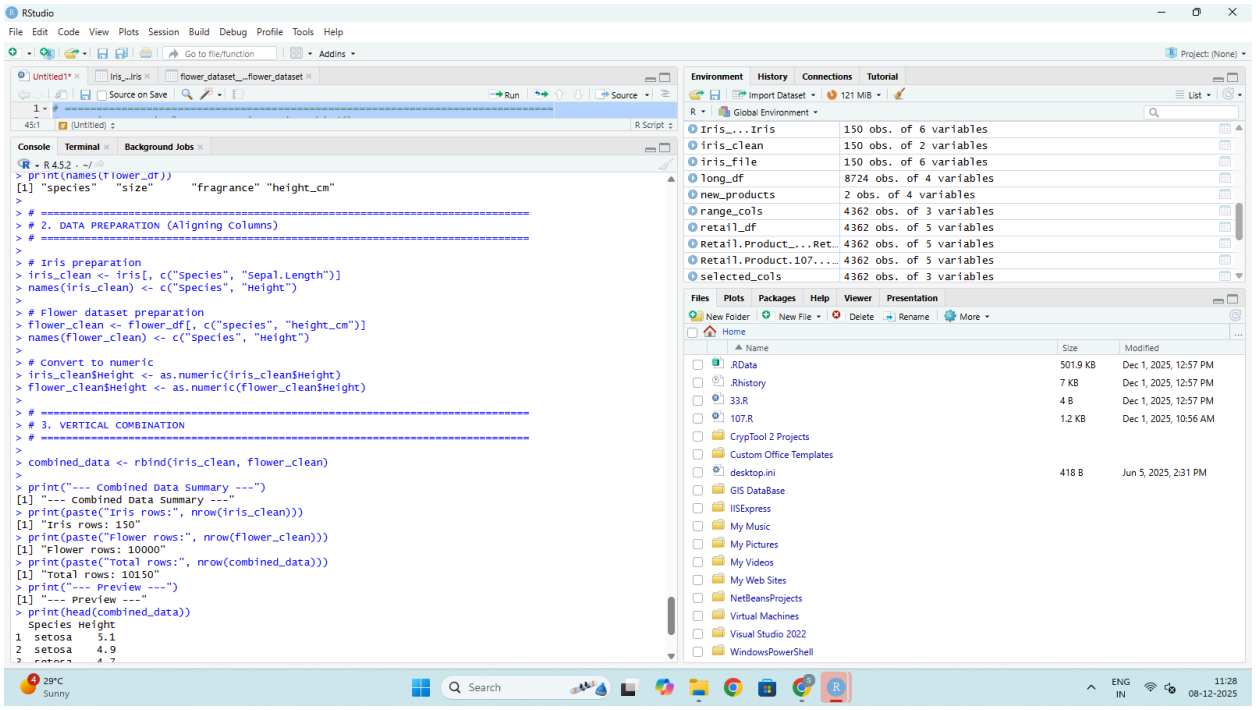
```
#
================================================================================
======
# 3. VERTICAL COMBINATION
#
================================================================================
======

combined_data <- rbind(iris_clean, flower_clean)

print("--- Combined Data Summary ---")
print(paste("Iris rows:", nrow(iris_clean)))
print(paste("Flower rows:", nrow(flower_clean)))
print(paste("Total rows:", nrow(combined_data)))
print("--- Preview ---")
print(head(combined_data))
print(tail(combined_data))
```

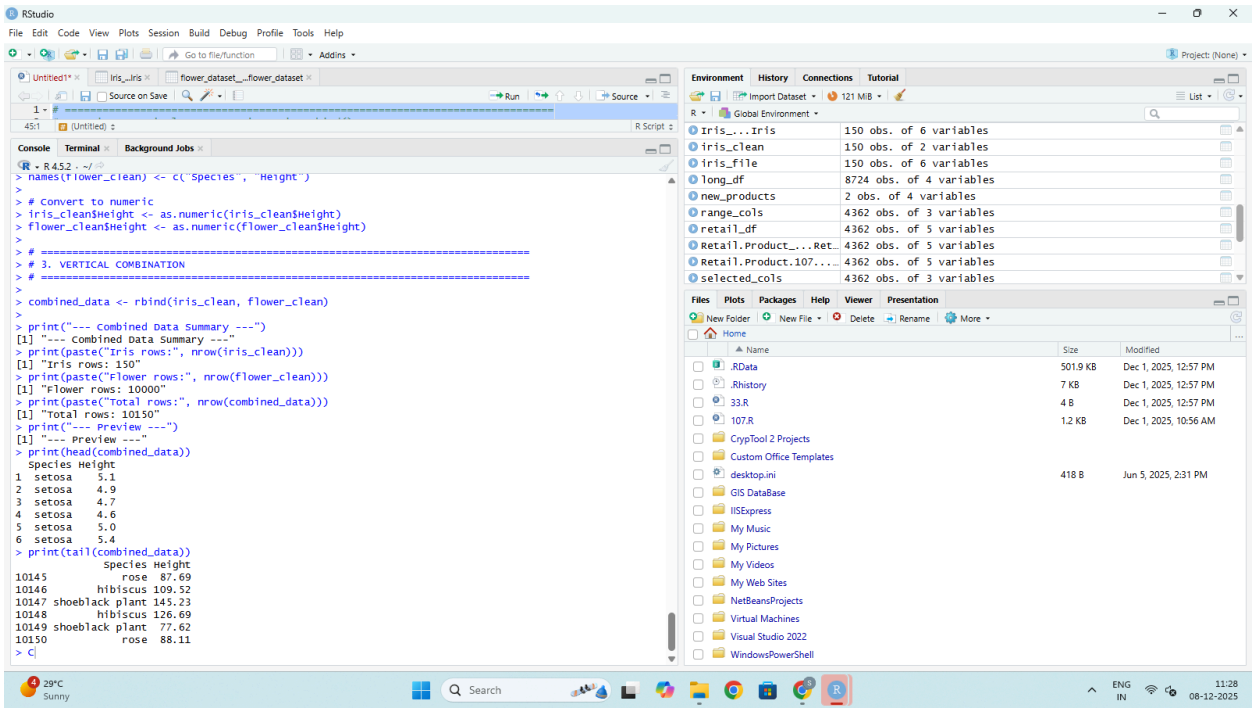
OUTPUT:



RIYA RANE
S107 SYCS

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





PRACTICAL NO:13

13. Identifying and handling duplicates using distinct() (R studio).

CODE:

RIYA RANE

S107 SYCS

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

```
#
================================================================================
======
# R Script: Identifying and Handling Duplicates
# Function: distinct() from the dplyr package
#
================================================================================
======

# Load library
library(dplyr)

#
================================================================================
======
# 1. SETUP: Create a Dataset with Intentional Duplicates


orders_df <- data.frame(
  OrderID = c(101, 102, 102, 103, 104, 101, 104),
  Customer = c("NISHITA", "YUKTA", "SIYA", "NANDINI", "SIMRAN", "PURVI", "GAZALA"),
  Product = c("Laptop", "Phone", "Phone", "Tablet", "Monitor", "Laptop", "Mouse")
)

print("--- 1. Original Dataset (Note 7 rows) ---")
print(orders_df)

#
================================================================================
======
# 2. IDENTIFYING DUPLICATES
#
================================================================================
======

duplicates_report <- orders_df %>%
  group_by(OrderID, Customer, Product) %>%
  count() %>%              # Count occurrences
  filter(n > 1)           # Keep only rows that appear more than once

print("--- 2. Identification Report (Rows that are duplicated) ---")
print(duplicates_report)
```

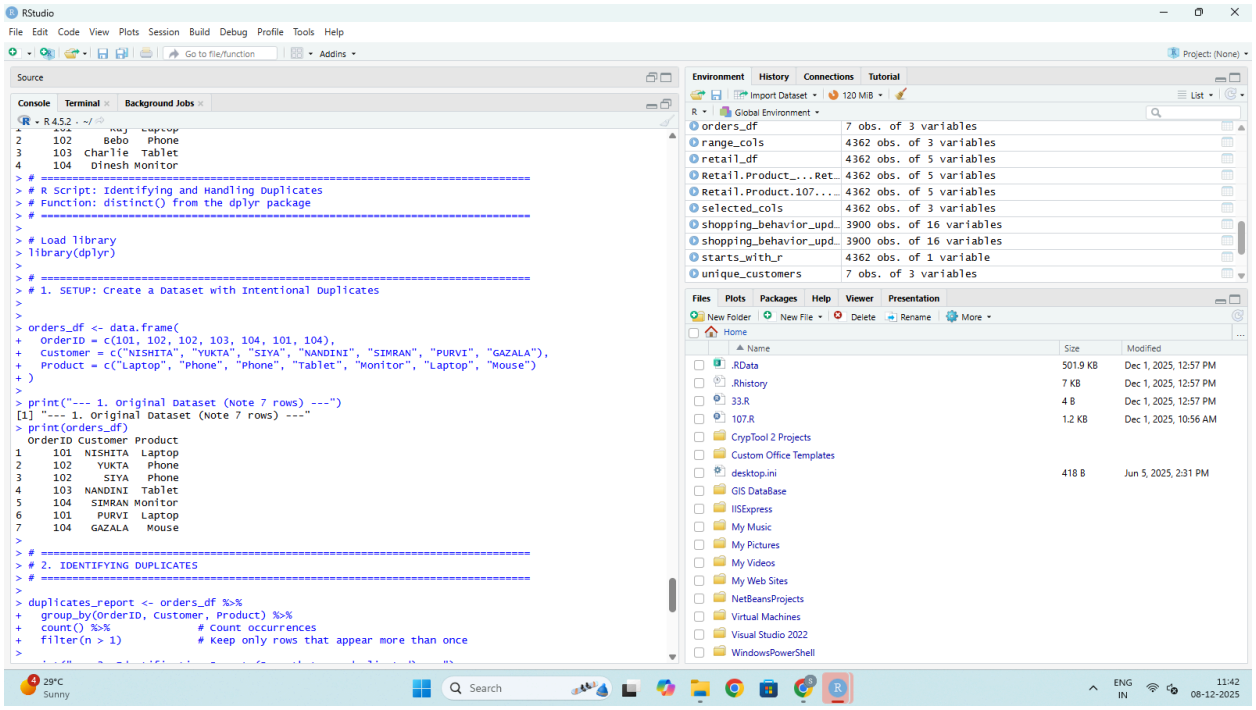
RIYA RANE
S107 SYCS

```
#
================================================================================
======
# 3. HANDLING EXACT DUPLICATES
#
================================================================================
======

clean_exact <- orders_df %>%
  distinct()              # Removes only exact row matches

print("--- 3. Removed Exact Duplicates (distinct) ---")
print(clean_exact)

#
================================================================================
======
# 4. HANDLING DUPLICATES BASED ON SPECIFIC COLUMN
#
================================================================================
======

unique_customers <- orders_df %>%
  distinct(Customer, .keep_all = TRUE)   # Keeps first appearance of each customer

print("--- 4. Unique Customers Only (Partial duplicates removed) ---")
print(unique_customers)
```

OUTPUT:

RIYA RANE
S107 SYCS

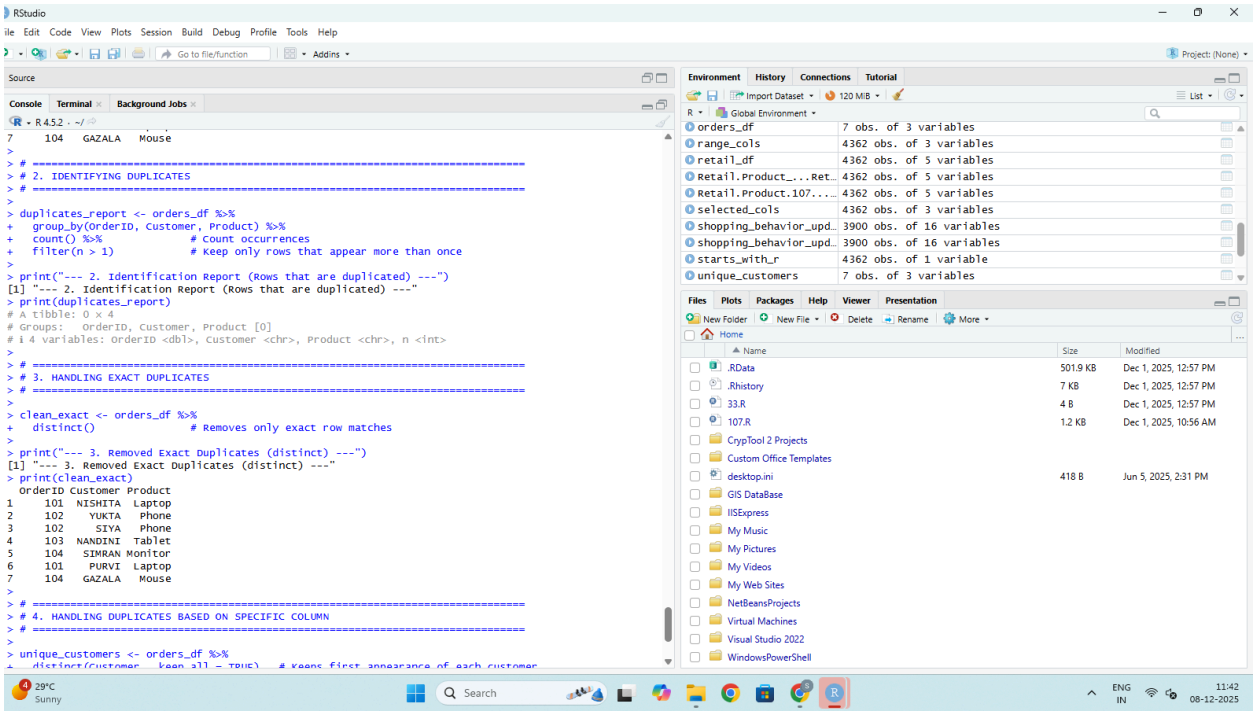
SHETH L.U.J AND SIR M.V COLLEGE



```
2     102      Bebo   Phone
3     103   Charlie  Tablet
4     104    Dinesh Monitor
> # ================================================================================
> # R Script: Identifying and Handling Duplicates
> # Function: distinct() from the dplyr package
> # ================================================================================
>
> # Load library
> library(dplyr)
>
> # ================================================================================
> # 1. SETUP: Create a Dataset with Intentional Duplicates
>
>
> orders_df <- data.frame(
+   OrderID = c(101, 102, 102, 103, 104, 101, 104),
+   Customer = c("NISHITA", "YUKTA", "SIYA", "NANDINI", "SIMRAN", "PURVI", "GAZALA"),
+   Product = c("Laptop", "Phone", "Phone", "Tablet", "Monitor", "Laptop", "Mouse")
+ )
>
> print("--- 1. Original Dataset (Note 7 rows) ---")
[1] "--- 1. Original Dataset (Note 7 rows) ---"
> print(orders_df)
  OrderID Customer Product
1     101  NISHITA  Laptop
2     102    YUKTA   Phone
3     102     SIYA   Phone
4     103  NANDINI  Tablet
5     104   SIMRAN Monitor
6     101    PURVI  Laptop
7     104   GAZALA   Mouse
>
> # ================================================================================
> # 2. IDENTIFYING DUPLICATES
> # ================================================================================
>
> duplicates_report <- orders_df %>%
+   group_by(OrderID, Customer, Product) %>%
+   count() %>%              # Count occurrences
+   filter(n > 1)            # Keep only rows that appear more than once
>
```

orders_df	7 obs. of 3 variables
range_cols	4362 obs. of 3 variables
retail_df	4362 obs. of 5 variables
Retail.Product_...Ret...	4362 obs. of 5 variables
Retail.Product.107....	4362 obs. of 5 variables
selected_cols	4362 obs. of 3 variables
shopping_behavior_upd...	3900 obs. of 16 variables
shopping_behavior_upd...	3900 obs. of 16 variables
starts_with_r	4362 obs. of 1 variable
unique_customers	7 obs. of 3 variables

Name	Size	Modified
.RData	501.9 KB	Dec 1, 2025, 12:57 PM
.Rhistory	7 KB	Dec 1, 2025, 12:57 PM
33.R	4 B	Dec 1, 2025, 12:57 PM
107.R	1.2 KB	Dec 1, 2025, 10:56 AM
CrypTool 2 Projects		
Custom Office Templates		
desktop.ini	418 B	Jun 5, 2025, 2:31 PM
GIS DataBase		
IISExpress		
My Music		
My Pictures		
My Videos		
My Web Sites		
NetBeansProjects		
Virtual Machines		
Visual Studio 2022		
WindowsPowerShell		



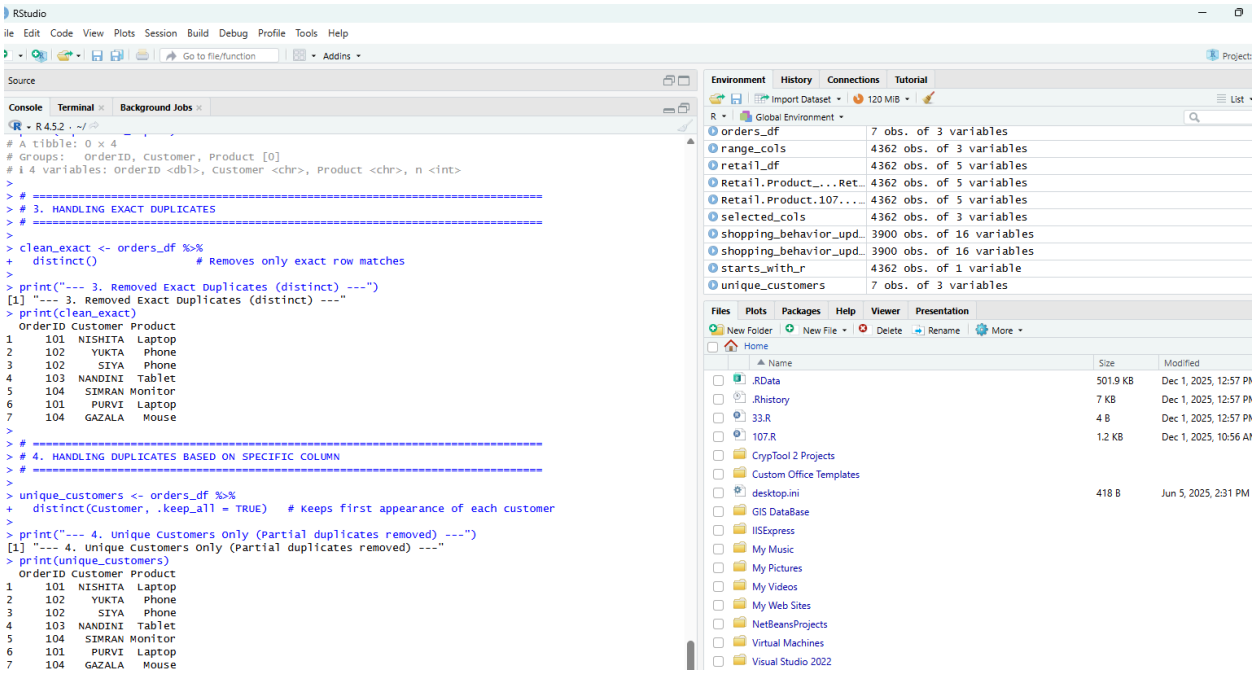
```
7     104   GAZALA   Mouse
>
> # ================================================================================
> # 2. IDENTIFYING DUPLICATES
> # ================================================================================
>
> duplicates_report <- orders_df %>%
+   group_by(OrderID, Customer, Product) %>%
+   count() %>%              # Count occurrences
+   filter(n > 1)            # Keep only rows that appear more than once
>
> print("--- 2. Identification Report (Rows that are duplicated) ---")
[1] "--- 2. Identification Report (Rows that are duplicated) ---"
> print(duplicates_report)
# A tibble: 0 × 4
# Groups:   OrderID, Customer, Product [0]
# ℹ 4 variables: OrderID <dbl>, Customer <chr>, Product <chr>, n <int>
>
> # ================================================================================
> # 3. HANDLING EXACT DUPLICATES
> # ================================================================================
>
> clean_exact <- orders_df %>%
+   distinct()               # Removes only exact row matches
>
> print("--- 3. Removed Exact Duplicates (distinct) ---")
[1] "--- 3. Removed Exact Duplicates (distinct) ---"
> print(clean_exact)
  OrderID Customer Product
1     101  NISHITA  Laptop
2     102    YUKTA   Phone
3     102     SIYA   Phone
4     103  NANDINI  Tablet
5     104   SIMRAN Monitor
6     101    PURVI  Laptop
7     104   GAZALA   Mouse
>
> # ================================================================================
> # 4. HANDLING DUPLICATES BASED ON SPECIFIC COLUMN
> # ================================================================================
>
> unique_customers <- orders_df %>%
+   distinct(Customer, .keep_all = TRUE)   # Keeps first appearance of each customer
```

orders_df	7 obs. of 3 variables
range_cols	4362 obs. of 3 variables
retail_df	4362 obs. of 5 variables
Retail.Product_...Ret...	4362 obs. of 5 variables
Retail.Product.107....	4362 obs. of 5 variables
selected_cols	4362 obs. of 3 variables
shopping_behavior_upd...	3900 obs. of 16 variables
shopping_behavior_upd...	3900 obs. of 16 variables
starts_with_r	4362 obs. of 1 variable
unique_customers	7 obs. of 3 variables

Name	Size	Modified
.RData	501.9 KB	Dec 1, 2025, 12:57 PM
.Rhistory	7 KB	Dec 1, 2025, 12:57 PM
33.R	4 B	Dec 1, 2025, 12:57 PM
107.R	1.2 KB	Dec 1, 2025, 10:56 AM
CrypTool 2 Projects		
Custom Office Templates		
desktop.ini	418 B	Jun 5, 2025, 2:31 PM
GIS DataBase		
IISExpress		
My Music		
My Pictures		
My Videos		
My Web Sites		
NetBeansProjects		
Virtual Machines		
Visual Studio 2022		
WindowsPowerShell		

RIYA RANE
S107 SYCS

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



PRACTICAL NO:6

```
#
================================================================================
======
# R Script: Extracting Date Components using lubridate
#
================================================================================
======

# Install and load necessary libraries
install.packages("lubridate")
library(lubridate)
library(dplyr)

#
================================================================================
======
# 1. SETUP: Create Sample Data
#
================================================================================
======

dates_df <- data.frame(
  Event_ID = 1:4,
  Date_String = c("2023-01-15", "2023-10-31", "2024-02-29", "2024-12-25")
```

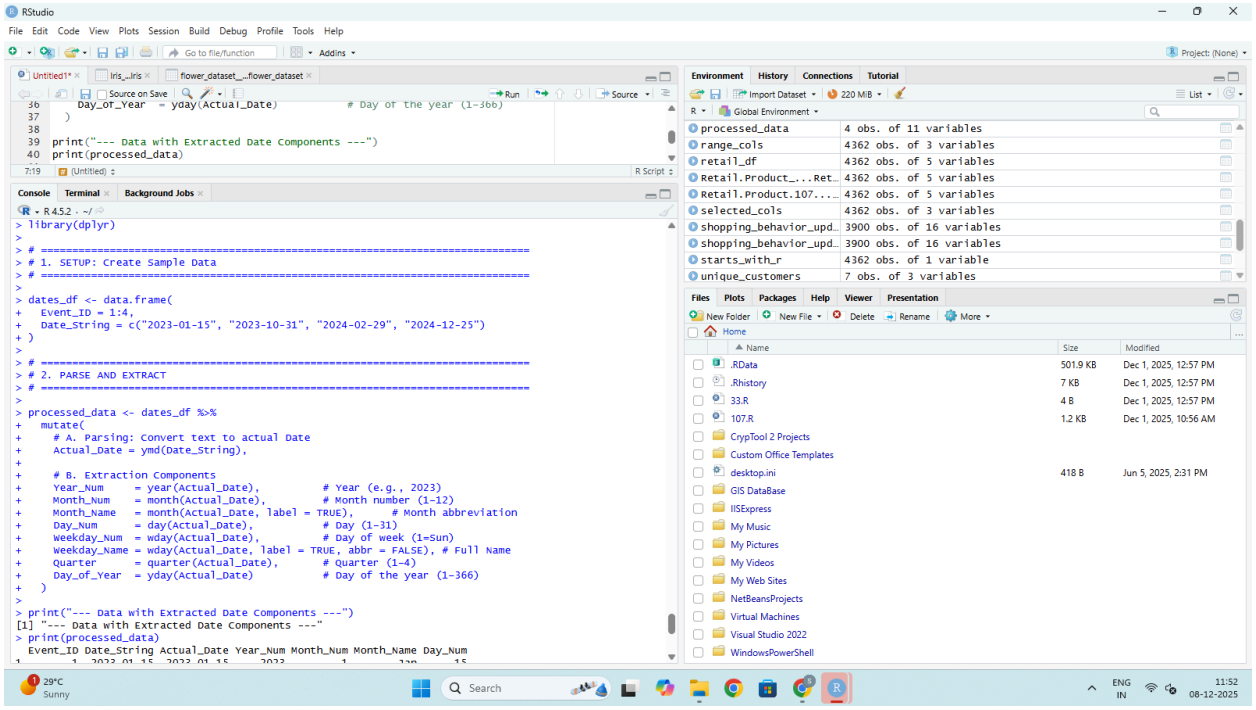
RIYA RANE
S107 SYCS

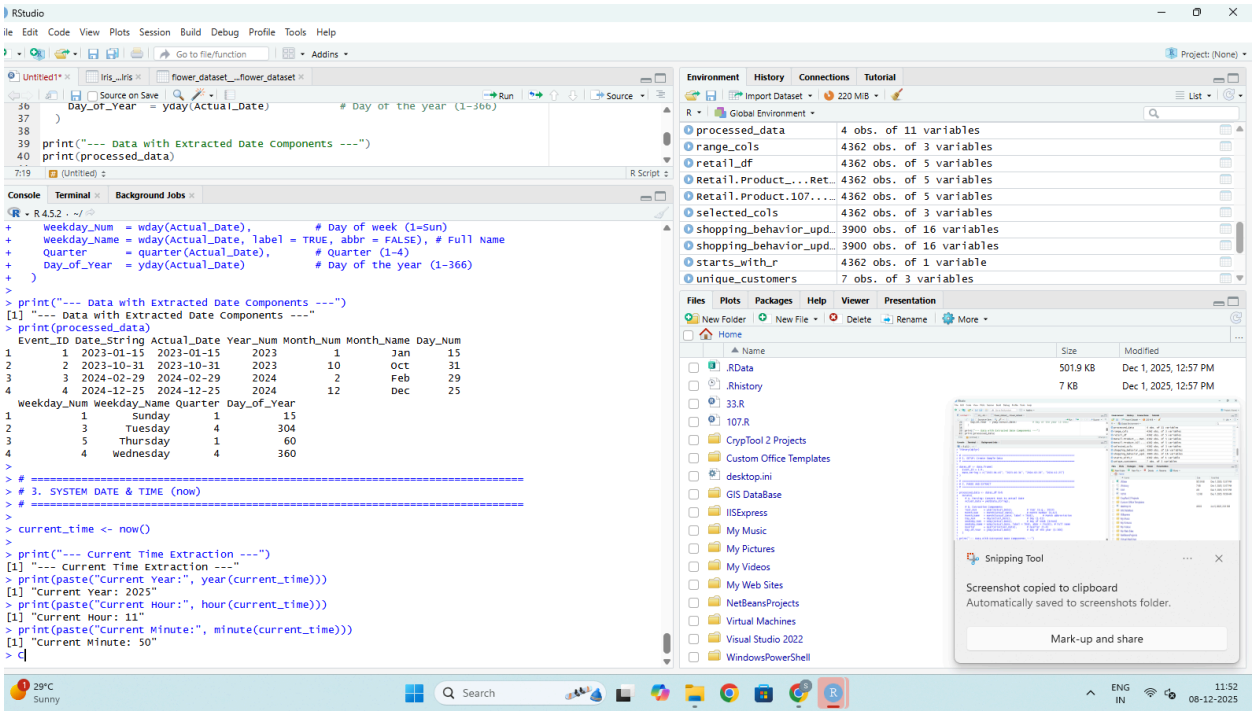
```
)

#
================================================================================
======
# 2. PARSE AND EXTRACT
#
================================================================================
======

processed_data <- dates_df %>%
  mutate(
    # A. Parsing: Convert text to actual Date
    Actual_Date = ymd(Date_String),

    # B. Extraction Components
    Year_Num     = year(Actual_Date),          # Year (e.g., 2023)
    Month_Num    = month(Actual_Date),         # Month number (1–12)
    Month_Name   = month(Actual_Date, label = TRUE),      # Month abbreviation
    Day_Num      = day(Actual_Date),           # Day (1–31)
    Weekday_Num  = wday(Actual_Date),          # Day of week (1=Sun)
    Weekday_Name = wday(Actual_Date, label = TRUE, abbr = FALSE), # Full Name
    Quarter      = quarter(Actual_Date),       # Quarter (1–4)
    Day_of_Year  = yday(Actual_Date)           # Day of the year (1–366)
  )

print("--- Data with Extracted Date Components ---")
print(processed_data)

#
================================================================================
======
# 3. SYSTEM DATE & TIME (now)
#
================================================================================
======

current_time <- now()

print("--- Current Time Extraction ---")
print(paste("Current Year:", year(current_time)))
print(paste("Current Hour:", hour(current_time)))
print(paste("Current Minute:", minute(current_time)))
```

RIYA RANE
S107 SYCS

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

OUTPUT:





PRACTICAL NO:15
CODE:

RIYA RANE
S107 SYCS

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

```
#
================================================================================
======
# R Script: Generating Basic Summaries
# Functions: str() and summary()
# Dataset: Retail Product Data (Simulated)
#
================================================================================
======

#
================================================================================
======
# 1. SETUP: Create Sample Data
#
================================================================================
======

retail_df <- data.frame(
  ID = 1:6,
  Category = c("Electronics", "Home", "Electronics", "Clothing", "Home", "Clothing"),
  Price = c(500.50, 45.00, 900.00, NA, 300.00, 25.00),  # Note NA
  In_Stock = c(TRUE, TRUE, FALSE, TRUE, FALSE, TRUE),
  Rating = c(4.5, 3.8, 4.9, 4.0, 3.5, 4.2)
)

print("--- Data Loaded ---")
print(retail_df)

#
================================================================================
======
# 2. USING str() — Structure of the Data
#
================================================================================
======

print("--- OUTPUT OF str() ---")
str(retail_df)

#
================================================================================
======
# 3. USING summary() — Statistical Summary
```

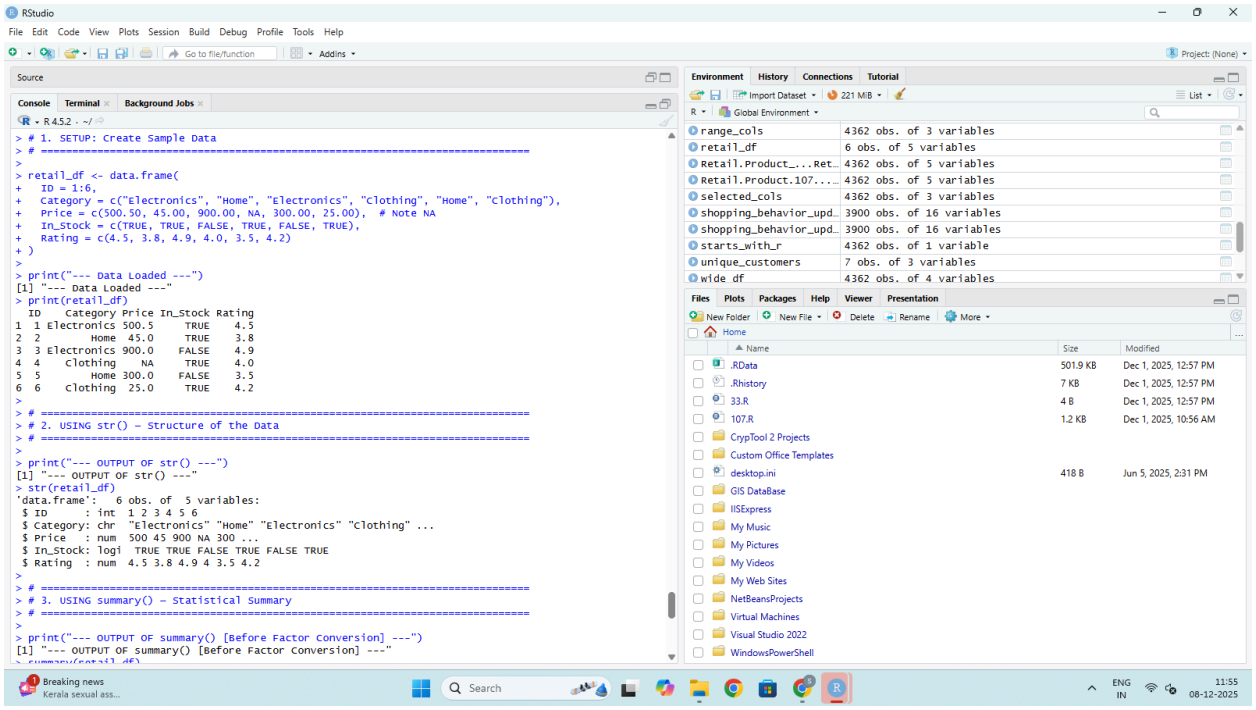
RIYA RANE
S107 SYCS

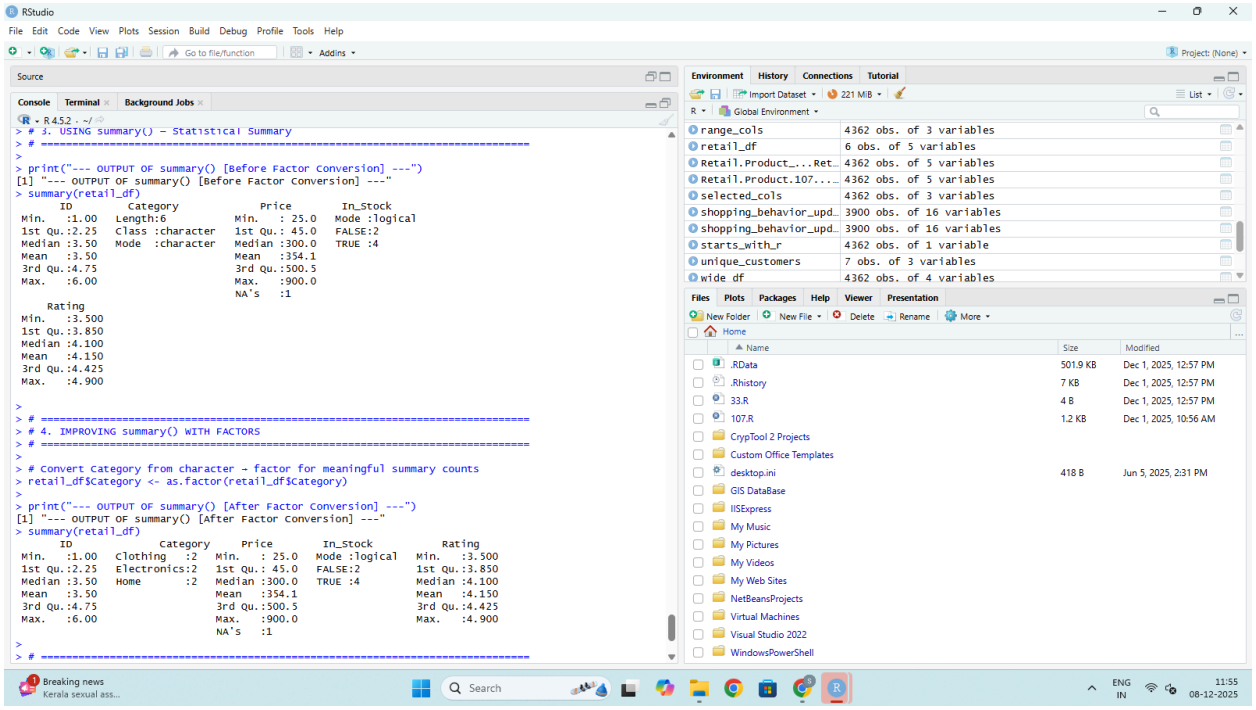
```
#
================================================================================
======

print("--- OUTPUT OF summary() [Before Factor Conversion] ---")
summary(retail_df)

#
================================================================================
======
# 4. IMPROVING summary() WITH FACTORS
#
================================================================================
======

# Convert Category from character → factor for meaningful summary counts
retail_df$Category <- as.factor(retail_df$Category)

print("--- OUTPUT OF summary() [After Factor Conversion] ---")
summary(retail_df)

#
================================================================================
======
# 5. Accessing Specific Summaries
#
================================================================================
======

avg_rating <- mean(retail_df$Rating)                    # Average rating
max_price <- max(retail_df$Price, na.rm = TRUE)         # Ignore NA

print(paste("Average Rating:", avg_rating))
print(paste("Highest Price:", max_price))
```

OUTPUT:

RIYA RANE
S107 SYCS

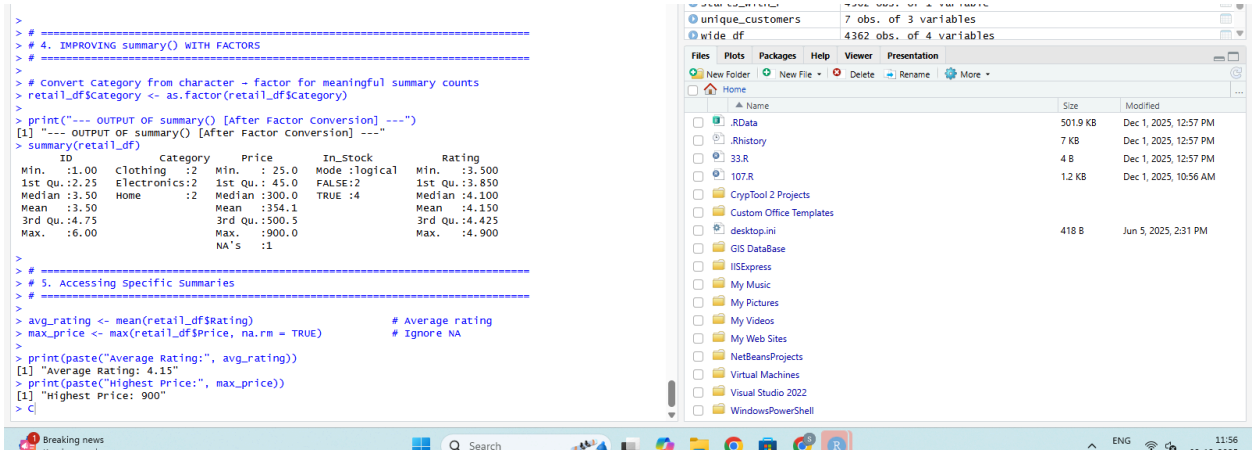
SHETH L.U.J AND SIR M.V COLLEGE





RIYA RANE
S107 SYCS

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



```
> # ============================================================================
> # 4. IMPROVING summary() WITH FACTORS
> # ============================================================================
>
> # Convert Category from character → factor for meaningful summary counts
> retail_df$Category <- as.factor(retail_df$Category)
>
> print("--- OUTPUT OF summary() [After Factor Conversion] ---")
[1] "--- OUTPUT OF summary() [After Factor Conversion] ---"
> summary(retail_df)
       ID          Category      Price         In_Stock          Rating
 Min.   :1.00   Clothing   :2   Min.   : 25.0   Mode :logical   Min.   :3.500
 1st Qu.:2.25   Electronics:2   1st Qu.: 45.0   FALSE:2         1st Qu.:3.850
 Median :3.50   Home       :2   Median :300.0   TRUE :4         Median :4.100
 Mean   :3.50                  Mean   :354.1                   Mean   :4.150
 3rd Qu.:4.75                  3rd Qu.:500.5                   3rd Qu.:4.425
 Max.   :6.00                  Max.   :900.0                   Max.   :4.900
                               NA's   :1
>
> # ============================================================================
> # 5. Accessing Specific Summaries
> # ============================================================================
>
> avg_rating <- mean(retail_df$Rating)                    # Average rating
> max_price <- max(retail_df$Price, na.rm = TRUE)         # Ignore NA
>
> print(paste("Average Rating:", avg_rating))
[1] "Average Rating: 4.15"
> print(paste("Highest Price:", max_price))
[1] "Highest Price: 900"
> C
```

unique_customers 7 obs. of 3 variables
wide_df 4362 obs. of 4 variables

Files | Plots | Packages | Help | Viewer | Presentation

New Folder | New File | Delete | Rename | More

Home

Name	Size	Modified
.RData	501.9 KB	Dec 1, 2025, 12:57 PM
.Rhistory	7 KB	Dec 1, 2025, 12:57 PM
33.R	4 B	Dec 1, 2025, 12:57 PM
107.R	1.2 KB	Dec 1, 2025, 10:56 AM
CrypTool 2 Projects		
Custom Office Templates		
desktop.ini	418 B	Jun 5, 2025, 2:31 PM
GIS DataBase		
IISExpress		
My Music		
My Pictures		
My Videos		
My Web Sites		
NetBeansProjects		
Virtual Machines		
Visual Studio 2022		
WindowsPowerShell		

Breaking news | Search | ENG | 11:56

RIYA RANE
S107 SYCS