Report Writting

Activity_2

Activity Title: Technology-Enhanced Learning (TEL)

Problem Statement: Apply basic data wrangling, exploratory data analysis (EDA), summary statistics, and visualization using R-programming.

Theoretical Concept:

Technology-Enhanced Learning (TEL) combines the power of analytical tools and digital technologies to support data-driven learning.
Using **R and RStudio**, learners can interact with raw data, clean and process it, visualize trends, and interpret insights.
This hands-on approach helps understand how technology supports real-world business decision-making.
In this activity, R was used to analyze an **online retail dataset**, helping identify sales patterns, best-selling products, and high-traffic hours — turning raw data into actionable insights.

Dataset description:

The dataset used is the **Online Retail** dataset from the **UCI Machine Learning Repository**
https://archive.ics.uci.edu/ml/machine-learning-databases/00352/Online%20Retail.xlsx

It includes **transactions from December 2010 to December 2011** for a UK-based online store.
Each record represents a product purchased, including attributes such as:

- **InvoiceNo:** Unique transaction ID
- **StockCode:** Product code
- **Description:** Product name
- **Quantity:** Units sold
- **InvoiceDate:** Date and time of purchase
- **UnitPrice:** Price per item
- **CustomerID:** Customer identifier
- **Country:** Country of the customer

This dataset is widely used for transactional data analysis and sales forecasting exercises

Implementation Concept:

The analysis was done using **R programming** with libraries like tidyverse, lubridate, readxl, and ggplot2. Steps followed:

1. **Load Data:** Imported the .xlsx file using read_excel().
2. **Clean Data:** Removed duplicates, missing CustomerIDs, and negative/zero quantities.
3. **Feature Engineering:**
    - Created a **Revenue** column (Quantity * UnitPrice).
    - Extracted **YearMonth**, **Hour**, and **Weekday** from InvoiceDate.
4. **Compute Metrics:**
    - Total monthly revenue
    - Average sales per transaction
    - Top 10 products by total revenue
5. **Visualizations:**
    - Line chart of **monthly revenue**
    - Bar chart of **top products**
    - Heatmap of **weekday vs hour sales**

Program Execution:

Input:

The input is the "Online Retail" dataset in .xlsx format.

R Code :

```
# --- Load Libraries ---
library(tidyverse)
library(lubridate)
library(readxl)
library(scales)


# --- Load Dataset ---
retail <- read_excel("C:/Users/lenovo/Downloads/Online Retail.xlsx")


# --- Clean Data ---
retail_clean <- retail %>%
  filter(!is.na(CustomerID)) %>%        # Remove missing Customer IDs
  distinct() %>%                        # Remove duplicates
  filter(Quantity > 0) %>%              # Remove cancellations
  mutate(InvoiceDate = as.POSIXct(InvoiceDate, format="%Y-%m-%d %H:%M:%S"))

# --- Feature Engineering ---
retail_clean <- retail_clean %>%
  mutate(
    Revenue = Quantity * UnitPrice,     # Total sale per line
    YearMonth = floor_date(InvoiceDate, "month"),
    Hour = hour(InvoiceDate),
    Weekday = wday(InvoiceDate, label = TRUE)
  )



# --- Summary Metrics ---
monthly_sales <- retail_clean %>%
  group_by(YearMonth) %>%
  summarise(TotalRevenue = sum(Revenue))

avg_sales <- retail_clean %>%
  summarise(AverageSalePerTxn = mean(Revenue))

top_products <- retail_clean %>%
  gro     escription) %>%
```

```
  summarise(TotalRevenue = sum(Revenue)) %>%
  arrange(desc(TotalRevenue)) %>%
  slice_head(n = 10)
#  Time Series: Monthly Revenue
ggplot(monthly_sales, aes(x = YearMonth, y = TotalRevenue)) +
  geom_line(color = "steelblue", size = 1) +
  geom_point(color = "steelblue") +
  scale_y_continuous(labels = comma) +
  labs(title = "Monthly Sales Revenue",
       x = "Month",
       y = "Total Revenue") +
  theme_minimal()


#  Bar Chart: Top 10 Products
ggplot(top_products, aes(x = reorder(Description, TotalRevenue), y = TotalRevenue)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  coord_flip() +
  scale_y_continuous(labels = comma) +
  labs(title = "Top 10 Products by Revenue",
       x = "Product",
       y = "Total Revenue") +
  theme_minimal()

# Heatmap: Weekday vs Hour
heatmap_data <- retail_clean %>%
  group_by(Weekday, Hour) %>%
  summarise(TotalSales = sum(Revenue))

ggplot(heatmap_data, aes(x = Hour, y = Weekday, fill = TotalSales)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "red", labels = comma) +
  labs(title = "Heatmap of Sales by Weekday and Hour",
       x = "Hour of Day",
       y = "Day of Week") +
  theme_minimal()

# --- Insights (Print in Console) ---
cat("Average sale per transaction:", avg_sales$AverageSalePerTxn, "\n")
cat("Top 3 products:\n")
print(head(top_products, 3))


library(tidyverse)
library(lubridate)
library(readxl)
```

```
library(scales)
library(patchwork)   # To combine plots

# Load and clean data
retail <- read_excel("C:/Users/lenovo/Downloads/Online Retail.xlsx") %>%
  filter(!is.na(CustomerID), Quantity > 0) %>%
  distinct() %>%
  mutate(
    InvoiceDate = as.POSIXct(InvoiceDate, format="%Y-%m-%d %H:%M:%S"),
    Revenue = Quantity * UnitPrice,
    YearMonth = floor_date(InvoiceDate, "month"),
    Hour = hour(InvoiceDate),
    Weekday = wday(InvoiceDate, label = TRUE)
  )

# --- Monthly Revenue ---
monthly_sales <- retail %>%
  group_by(YearMonth) %>%
  summarise(TotalRevenue = sum(Revenue))

p1 <- ggplot(monthly_sales, aes(x = YearMonth, y = TotalRevenue)) +
  geom_line(color = "steelblue", size = 1) +
  geom_point(color = "steelblue") +
  labs(title = " 📈 Monthly Sales Revenue", x = "Month", y = "Total Revenue") +
  theme_minimal()

# --- Top 10 Products ---
top_products <- retail %>%
  group_by(Description) %>%
  summarise(TotalRevenue = sum(Revenue)) %>%
  arrange(desc(TotalRevenue)) %>%
  slice_head(n = 10)

p2 <- ggplot(top_products, aes(x = reorder(Description, TotalRevenue), y = TotalRevenue)) +
  geom_bar(stat = "identity", fill = "orange") +
  coord_flip() +
  labs(title = "Top 10 Products by Revenue", x = "Product", y = "Total Revenue") +
  theme_minimal()

# --- Weekday vs Hour Heatmap ---
heatmap_data <- retail %>%
  group_by(Weekday, Hour) %>%
  summarise(TotalSales = sum(Revenue))

p3 <- ggplot(heatmap_data, aes(x = Hour, y = Weekday, fill = TotalSales)) +
  geom_tile() +
```

```
  scale_fill_gradient(low = "white", high = "red", labels = comma) +
  labs(title = "Sales Heatmap (Weekday vs Hour)", x = "Hour of Day", y = "Day of Week") +
  theme_minimal()

# --- Combine All (Dashboard Layout) ---
dashboard <- (p1 / (p2 | p3)) +
  plot_annotation(title = "🛒 Online Retail Dashboard",
                  subtitle = "Summary of Sales Trends, Top Products, and Purchase Patterns",
                  theme = theme(plot.title = element_text(size = 16, face = "bold")))

# --- Save as Image ---
ggsave("C:/Users/lenovo/Downloads/retail_dashboard.jpg", dashboard, width = 14, height = 9, dpi = 3
```

Output:

The outputs are summary tables and visualizations that provide insights into the retail data.

```
> # ------------------------------------------------------------------
> # Step 1: Load Libraries
> # ------------------------------------------------ .... [TRUNCATED]

> library(lubridate) # Makes it easier to work with dates and times

> library(readxl)    # For reading Excel files (.xls and .xlsx)

> # ------------------------------------------------------------------
> # Step 2: Data Loading and Inspection
> # ---------------------------------- .... [TRUNCATED]

> # Inspect the first few rows of the data
> print("--- First 6 rows of the dataset ---")
[1] "--- First 6 rows of the dataset ---"

> print(head(retail_data))
# A tibble: 6 × 8
  InvoiceNo StockCode Description     Quantity InvoiceDate         UnitPrice CustomerID Country
  <chr>     <chr>     <chr>              <dbl> <dttm>                  <dbl>      <dbl> <chr>
1 536365    85123A    WHITE HANGING …        6 2010-12-01 08:26:00      2.55      17850 United…
2 536365    71053     WHITE METAL LA…        6 2010-12-01 08:26:00      3.39      17850 United…
3 536365    84406B    CREAM CUPID HE…        8 2010-12-01 08:26:00      2.75      17850 United…
4 536365    84029G    KNITTED UNION …        6 2010-12-01 08:26:00      3.39      17850 United…
5 536365    84029E    RED WOOLLY HOT…        6 2010-12-01 08:26:00      3.39      17850 United…
6 536365    22752     SET 7 BABUSHKA…        2 2010-12-01 08:26:00      7.65      17850 United…

> # Inspect the structure of the dataset (column names, data types, etc.)
> print("--- Structure of the dataset ---")
[1] "--- Structure of the dataset ---"

> str(retail_data)
tibble [541,909 × 8] (S3: tbl_df/tbl/data.frame)
 $ InvoiceNo  : chr [1:541909] "536365" "536365" "536365" "536365" ...
 $ StockCode  : chr [1:541909] "85123A" "71053" "84406B" "84029G" ...
 $ Description: chr [1:541909] "WHITE HANGING HEART T-LIGHT HOLDER" "WHITE METAL LANTERN" "CREAM CUPID HEARTS
COAT HANGER" "KNITTED UNION FLAG HOT WATER BOTTLE" ...
 $ Quantity   : num [1:541909] 6 6 8 6 6 2 6 6 6 32
```

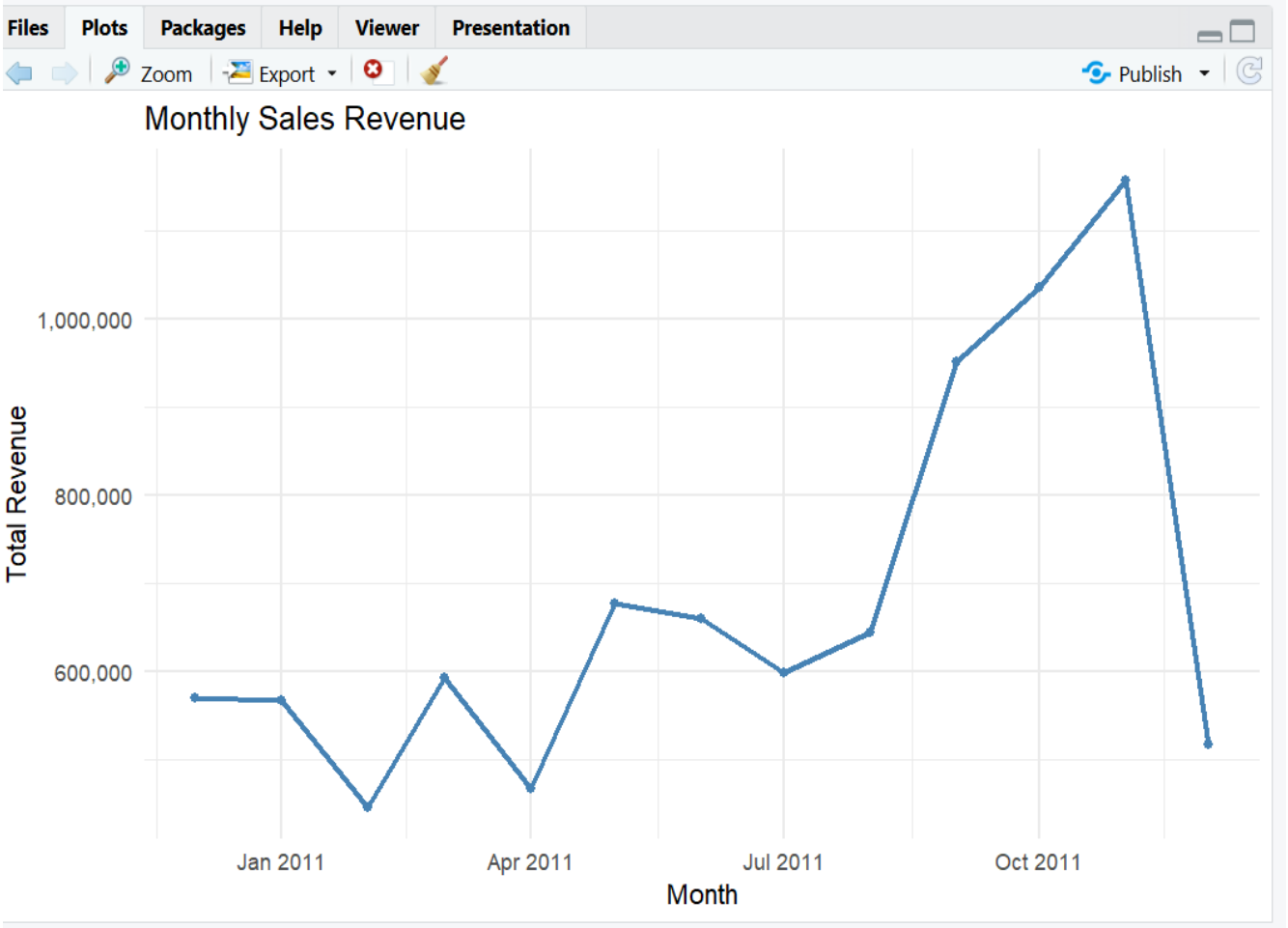
```
> library(lubridate) # Makes it easier to work with dates and times

> library(readxl)    # For reading Excel files (.xls and .xlsx)

> # ================================================================
> # Step 2: Data Loading and Inspection
> # ================================== .... [TRUNCATED]

> # Inspect the first few rows of the data
> print("--- First 6 rows of the dataset ---")
[1] "--- First 6 rows of the dataset ---"

> print(head(retail_data))
# A tibble: 6 × 8
  InvoiceNo StockCode Description     Quantity InvoiceDate         UnitPrice CustomerID Country
  <chr>     <chr>     <chr>              <dbl> <dttm>                  <dbl>      <dbl> <chr>
1 536365    85123A    WHITE HANGING …        6 2010-12-01 08:26:00      2.55      17850 United…
2 536365    71053     WHITE METAL LA…        6 2010-12-01 08:26:00      3.39      17850 United…
3 536365    84406B    CREAM CUPID HE…        8 2010-12-01 08:26:00      2.75      17850 United…
4 536365    84029G    KNITTED UNION …        6 2010-12-01 08:26:00      3.39      17850 United…
5 536365    84029E    RED WOOLLY HOT…        6 2010-12-01 08:26:00      3.39      17850 United…
6 536365    22752     SET 7 BABUSHKA…        2 2010-12-01 08:26:00      7.65      17850 United…

> # Inspect the structure of the dataset (column names, data types, etc.)
> print("--- Structure of the dataset ---")
[1] "--- Structure of the dataset ---"

> str(retail_data)
tibble [541,909 × 8] (S3: tbl_df/tbl/data.frame)
 $ InvoiceNo  : chr [1:541909] "536365" "536365" "536365" "536365" ...
 $ StockCode  : chr [1:541909] "85123A" "71053" "84406B" "84029G" ...
 $ Description: chr [1:541909] "WHITE HANGING HEART T-LIGHT HOLDER" "WHITE METAL LANTERN" "CREAM CUPID HEARTS
COAT HANGER" "KNITTED UNION FLAG HOT WATER BOTTLE" ...
 $ Quantity   : num [1:541909] 6 6 8 6 6 2 6 6 6 32 ...
 $ InvoiceDate: POSIXct[1:541909], format: "2010-12-01 08:26:00" "2010-12-01 08:26:00" "2010-12-01 08:26:00"
...
 $ UnitPrice  : num [1:541909] 2.55 3.39 2.75 3.39 3.39 7.65 4.25 1.85 1.85 1.69 ...
 $ CustomerID : num [1:541909] 17850 17850 17850 17850 17850 ...
 $ Country    : chr [1:541909] "United Kingdom" "United Kingdom" "United Kingdom" "United Kingdom" ...
```

```
> # Step 3: Data Cleaning (Wrangling)
> # ======================================== .... [TRUNCATED]

> # Remove any complete duplicate rows
> retail_data_cleaned <- retail_data_cleaned %>%
+   distinct()

> # Filter out returns/cancellations (where Quantity is zero or negative)
> retail_data_cleaned <- retail_data_cleaned %>%
+   filter(Quantity > 0)

> # Convert the 'InvoiceDate' column to a proper date-time format
> retail_data_cleaned$InvoiceDate <- as.POSIXct(retail_data_cleaned$InvoiceDate, for .... [TRUNCATED]

> print("--- Data cleaning complete. Displaying first few rows of cleaned data ---")
[1] "--- Data cleaning complete. Displaying first few rows of cleaned data ---"

> print(head(retail_data_cleaned))
# A tibble: 6 × 8
  InvoiceNo StockCode Description     Quantity InvoiceDate         UnitPrice CustomerID Country
  <chr>     <chr>     <chr>              <dbl> <dttm>                  <dbl>      <dbl> <chr>
1 536365    85123A    WHITE HANGING …        6 2010-12-01 08:26:00      2.55      17850 United…
2 536365    71053     WHITE METAL LA…        6 2010-12-01 08:26:00      3.39      17850 United…
3 536365    84406B    CREAM CUPID HE…        8 2010-12-01 08:26:00      2.75      17850 United…
4 536365    84029G    KNITTED UNION …        6 2010-12-01 08:26:00      3.39      17850 United…
5 536365    84029E    RED WOOLLY HOT…        6 2010-12-01 08:26:00      3.39      17850 United…
6 536365    22752     SET 7 BABUSHKA…        2 2010-12-01 08:26:00      7.65      17850 United…

> # ============================================================
> # Step 4: Feature Engineering
> # ========================================== .... [TRUNCATED]

> print("--- Feature engineering complete. Displaying first few rows of final data ---")
[1] "--- Feature engineering complete. Displaying first few rows of final data ---"

> print(head(retail_data_final))
# A tibble: 6 × 12
  InvoiceNo StockCode Description     Quantity InvoiceDate         UnitPrice CustomerID Country
  <chr>     <chr>     <chr>              <dbl> <dttm>                  <dbl>      <dbl> <chr>
1 536365    85123A    WHITE HANGING …        6 2010-12-01 08:26:00      2.55      17850 United…
```

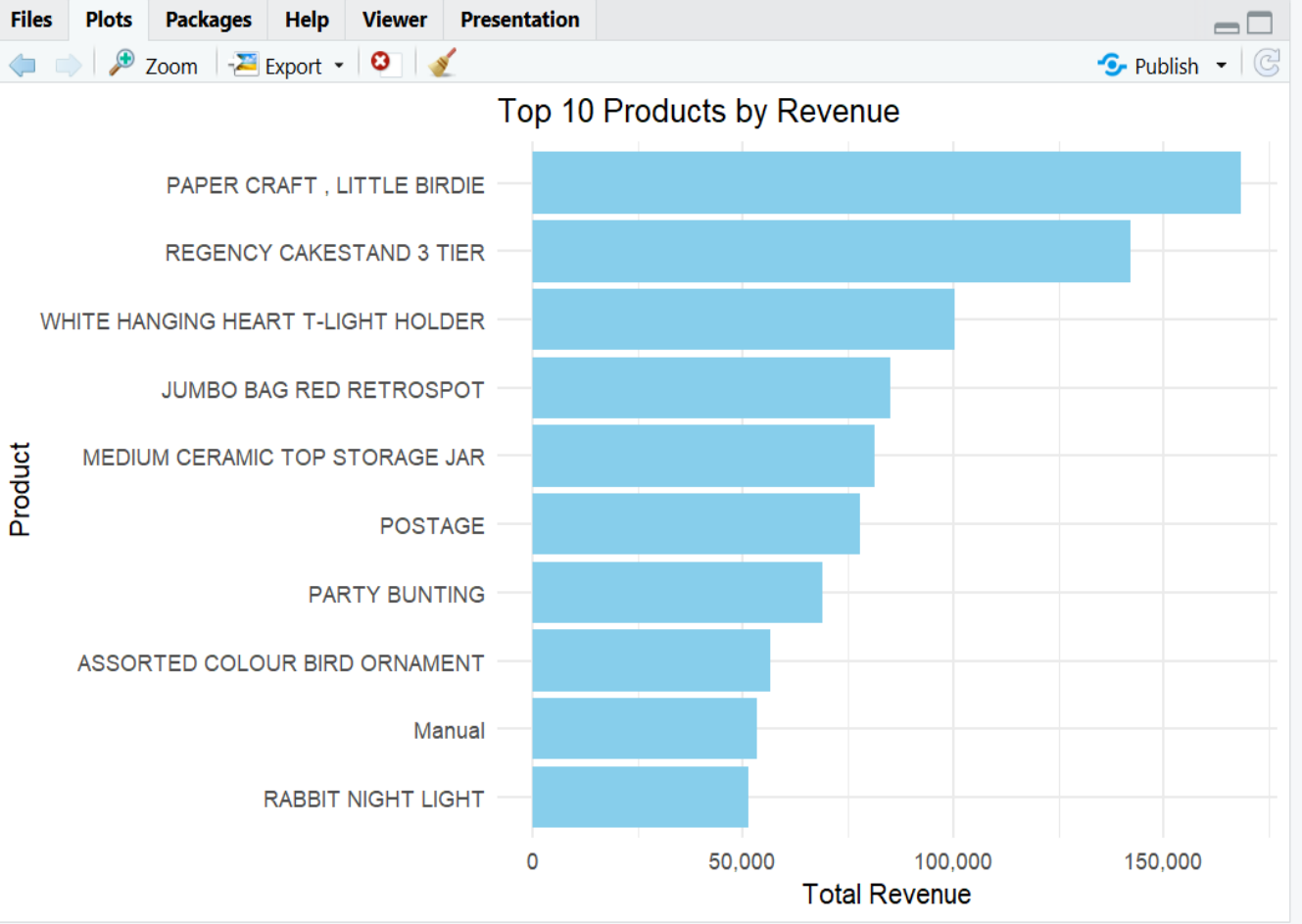
```
2 536365    71053     WHITE METAL LA…        6 2010-12-01 08:26:00       3.39      17850 United…
3 536365    84406B    CREAM CUPID HE…        8 2010-12-01 08:26:00       2.75      17850 United…
4 536365    84029G    KNITTED UNION …        6 2010-12-01 08:26:00       3.39      17850 United…
5 536365    84029E    RED WOOLLY HOT…        6 2010-12-01 08:26:00       3.39      17850 United…
6 536365    22752     SET 7 BABUSHKA…        2 2010-12-01 08:26:00       7.65      17850 United…
# i 4 more variables: Revenue <dbl>, YearMonth <dttm>, Hour <int>, Weekday <ord>

> # ======================================================================
> # Step 5: Exploratory Data Analysis (Summary Metrics)
> # ==================== .... [TRUNCATED]

> print("--- Total Monthly Revenue ---")
[1] "--- Total Monthly Revenue ---"

> print(monthly_revenue)
# A tibble: 13 × 2
   YearMonth           TotalRevenue
   <dttm>                     <dbl>
 1 2010-12-01 00:00:00      570423.
 2 2011-01-01 00:00:00      568101.
 3 2011-02-01 00:00:00      446085.
 4 2011-03-01 00:00:00      594082.
 5 2011-04-01 00:00:00      468374.
 6 2011-05-01 00:00:00      677355.
 7 2011-06-01 00:00:00      660046.
 8 2011-07-01 00:00:00      598963.
 9 2011-08-01 00:00:00      644051.
10 2011-09-01 00:00:00      950690.
11 2011-10-01 00:00:00     1035642.
12 2011-11-01 00:00:00     1156206.
13 2011-12-01 00:00:00      517190.

> # Calculate the top 10 products by revenue
> top_10_products <- retail_data_final %>%
+   group_by(Description) %>%
+   summarise(TotalRevenue = sum .... [TRUNCATED]

> print("--- Top 10 Products by Revenue ---")
[1] "--- Top 10 Products by Revenue ---"
```

```
> print(top_10_products)
# A tibble: 10 × 2
   Description                         TotalRevenue
   <chr>                                      <dbl>
 1 PAPER CRAFT , LITTLE BIRDIE             168470.
 2 REGENCY CAKESTAND 3 TIER                142265.
 3 WHITE HANGING HEART T-LIGHT HOLDER      100392.
 4 JUMBO BAG RED RETROSPOT                  85041.
 5 MEDIUM CERAMIC TOP STORAGE JAR           81417.
 6 POSTAGE                                  77804.
 7 PARTY BUNTING                            68785.
 8 ASSORTED COLOUR BIRD ORNAMENT            56413.
 9 Manual                                   53420.
10 RABBIT NIGHT LIGHT                       51251.

> # ============================================================
> # Step 6: Data Visualization
> # ============================================ .... [TRUNCATED]

> print(monthly_revenue_plot) # This will display the plot

> # 2. Top 10 Products by Revenue
> top_products_plot <- ggplot(top_10_products, aes(x = reorder(Description, TotalRevenue), y = TotalRevenue))
+
+    .... [TRUNCATED]

> print(top_products_plot) # This will display the plot

> # 3. Sales Heatmap (Weekday vs. Hour)
> sales_heatmap_data <- retail_data_final %>%
+   group_by(Weekday, Hour) %>%
+   summarise(TotalSales = sum(R .... [TRUNCATED]
`summarise()` has grouped output by 'Weekday'. You can override using the `.groups` argument.

> sales_heatmap <- ggplot(sales_heatmap_data, aes(x = Hour, y = Weekday, fill = TotalSales)) +
+   geom_tile() +
+   scale_fill_gradient(low = "white" .... [TRUNCATED]

> print(sales_heatmap) # This will display the plot
```

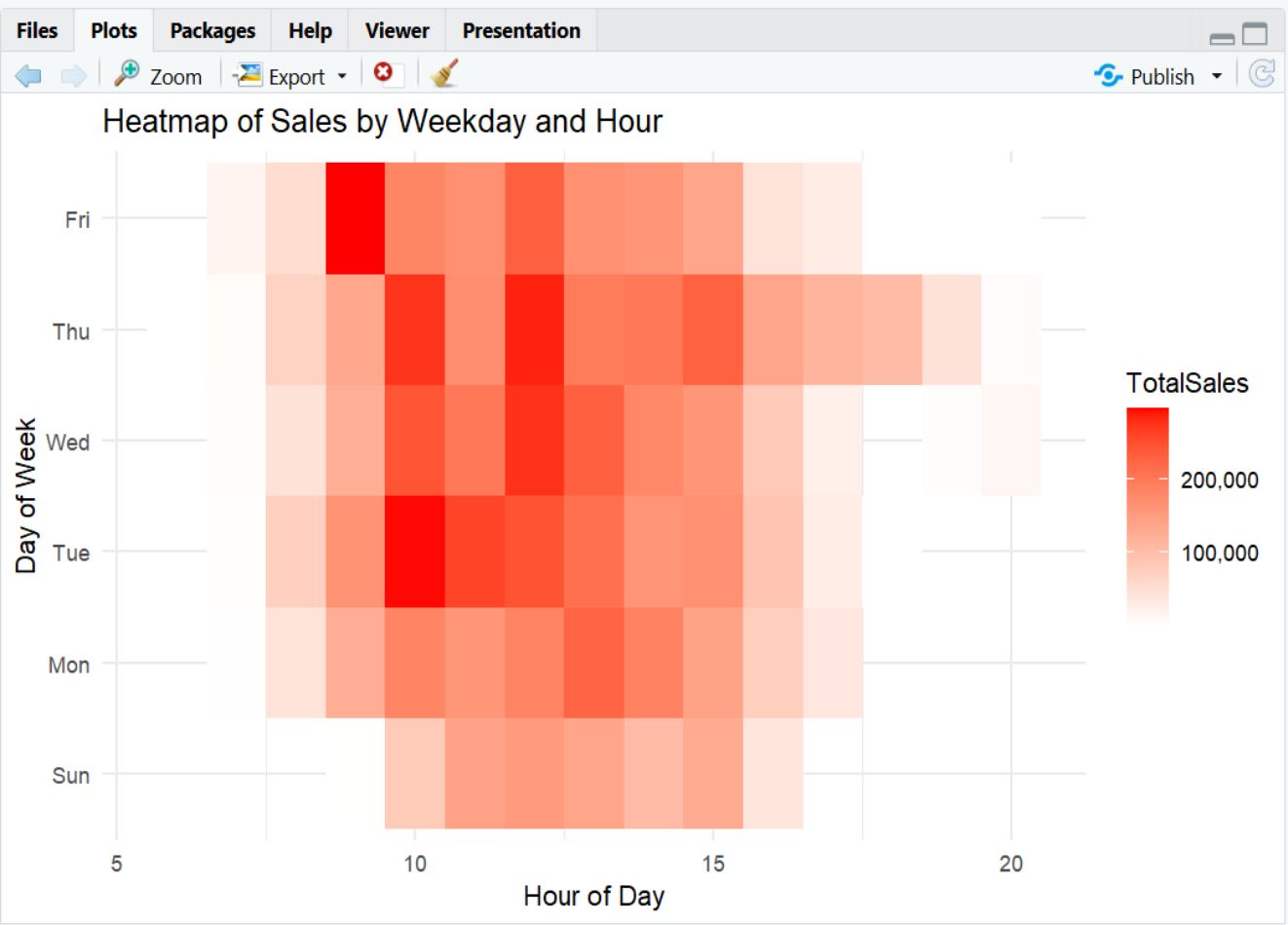
1. Time Series of Monthly Revenue: This plot shows a strong upward trend in sales towards the end of the year, likely due to holiday shopping.



2. Top 10 Products by Revenue: This chart identifies the products that contribute the most to the company's revenue.



3. Sales Heatmap (Weekday vs. Hour): This heatmap reveals that the peak shopping times are during midday on weekdays.



Algorithm:

Algorithm:

1. Start
2. Load required R libraries (tidyverse, lubridate, readxl, patchwork).
3. Read dataset (Online Retail.xlsx).
4. Clean data (remove NAs, duplicates, negative quantities).
5. Create new variables (Revenue, YearMonth, Hour, Weekday).
6. Compute monthly totals, product-level revenue, and time-based patterns.
7. Generate visualizations (line, bar, heatmap).
8. Combine all visuals into one dashboard image.
9. Interpret results and draw conclusions.
10. End.

Conclusion:

This analysis demonstrates how **R** can transform raw retail transaction data into clear business insights. Key findings include:

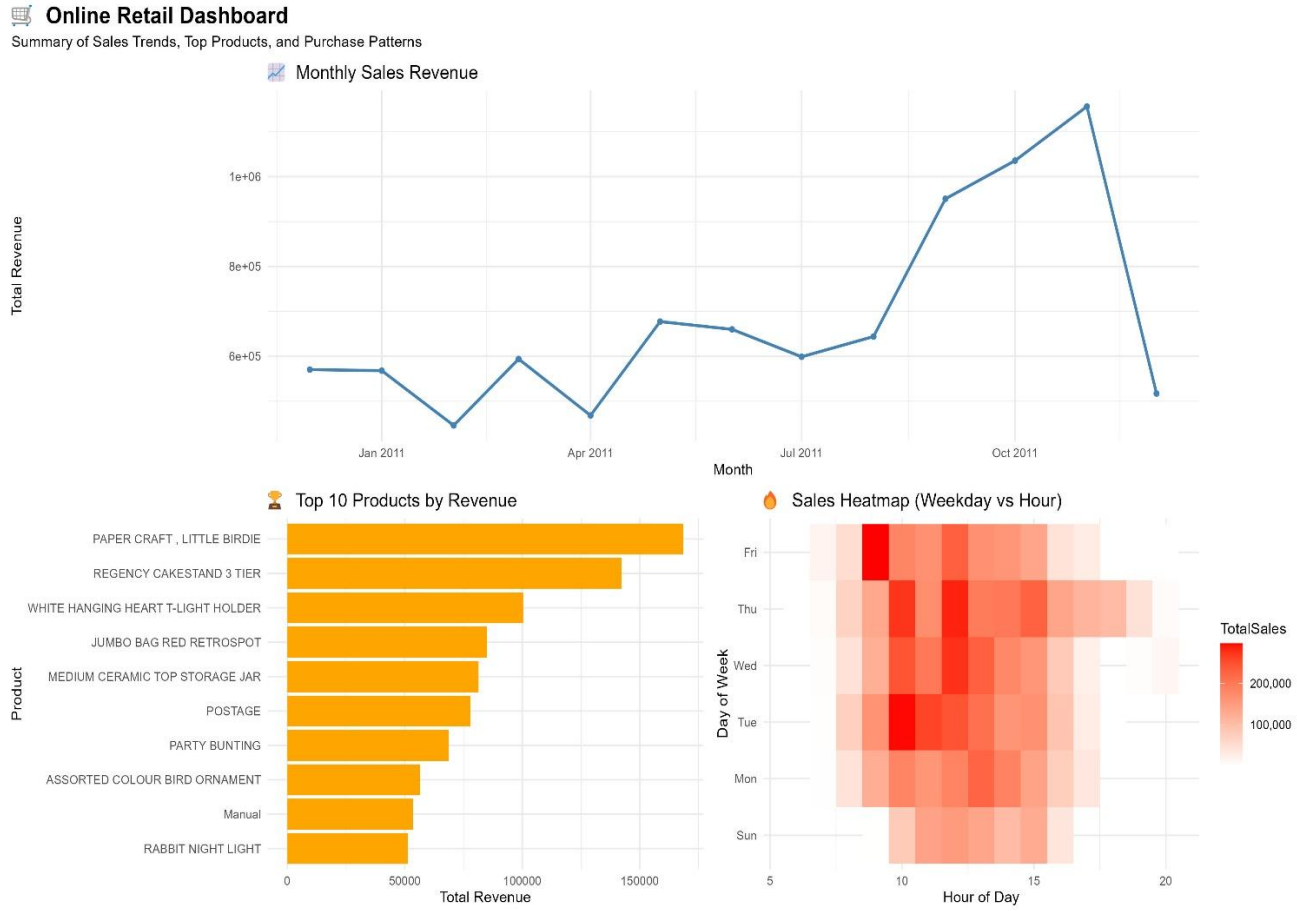
- **Sales spike** during **November–December**, likely due to festive shopping.
- A few **popular products dominate revenue**, such as gift items and decorations.
- **Most purchases occur mid-day on weekdays**, suggesting strong office-hour buying patterns.
-

Recommendations:

- Increase stock and promotions before festive months.
- Focus marketing on top-performing product categories.
- Schedule system/server optimization during peak weekday hours.

Through this, the project highlights **Technology-Enhanced Learning (TEL)** — applying data analytics for real-world decision-making.

Dashboard:


Online Retail Dashboard
Summary of Sales Trends, Top Products, and Purchase Patterns
Monthly Sales Revenue
Total Revenue
1e+06
8e+05
6e+05
Jan 2011
Apr 2011
Jul 2011
Oct 2011
Month
Top 10 Products by Revenue
Product
PAPER CRAFT , LITTLE BIRDIE
REGENCY CAKESTAND 3 TIER
WHITE HANGING HEART T-LIGHT HOLDER
JUMBO BAG RED RETROSPOT
MEDIUM CERAMIC TOP STORAGE JAR
POSTAGE
PARTY BUNTING
ASSORTED COLOUR BIRD ORNAMENT
Manual
RABBIT NIGHT LIGHT
0
50000
100000
150000
Total Revenue
Sales Heatmap (Weekday vs Hour)
Day of Week
Fri
Thu
Wed
Tue
Mon
Sun
5
10
15
20
Hour of Day
TotalSales
200,000
100,000
