Sales Data Analysis and Reporting for a Retail Business

Final Internship Project Submission

Name: *Bhavana Reddymachu*

Email: Bhavanasrihari2003@gmail.com

Platform: Internship Studio

Tools: Python, SQL, Excel

INTRODUCTION

Sales data analysis helps organizations understand revenue trends and business performance.
This project focuses on analyzing retail transaction data and generating meaningful reports using Python, SQL, and Excel to support data-driven decision-making.

PROJECT OBJECTIVES

- Analyze retail sales transaction data
- Clean and prepare raw data for analysis
- Apply SQL queries to extract insights
- Generate reports and dashboards using Excel

DATASET DESCRIPTION

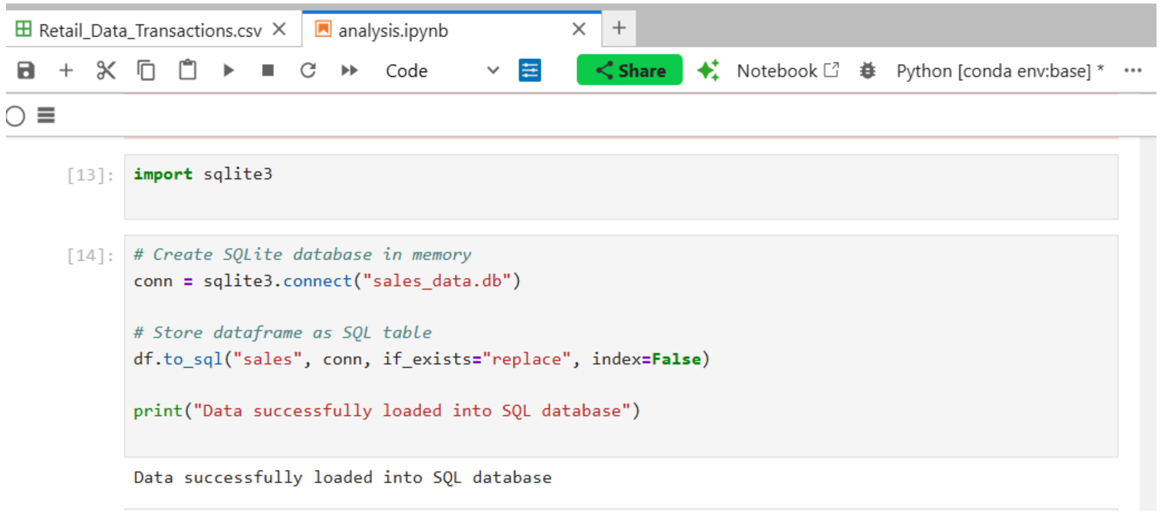
- **Source:** Kaggle – Retail Transaction Data
- **Type:** Transaction-level sales data

Attributes Used:

- Customer ID
- Transaction Date
- Transaction Amount

TOOLS & TECHNOLOGIES

- **Python:** Data cleaning, preparation, and analysis
- **SQL (SQLite):** Querying and aggregation of sales data (SQLite database used for executing SQL queries)



```
[13]: import sqlite3

[14]: # Create SQLite database in memory
      conn = sqlite3.connect("sales_data.db")

      # Store dataframe as SQL table
      df.to_sql("sales", conn, if_exists="replace", index=False)

      print("Data successfully loaded into SQL database")

      Data successfully loaded into SQL database
```

Fig.1

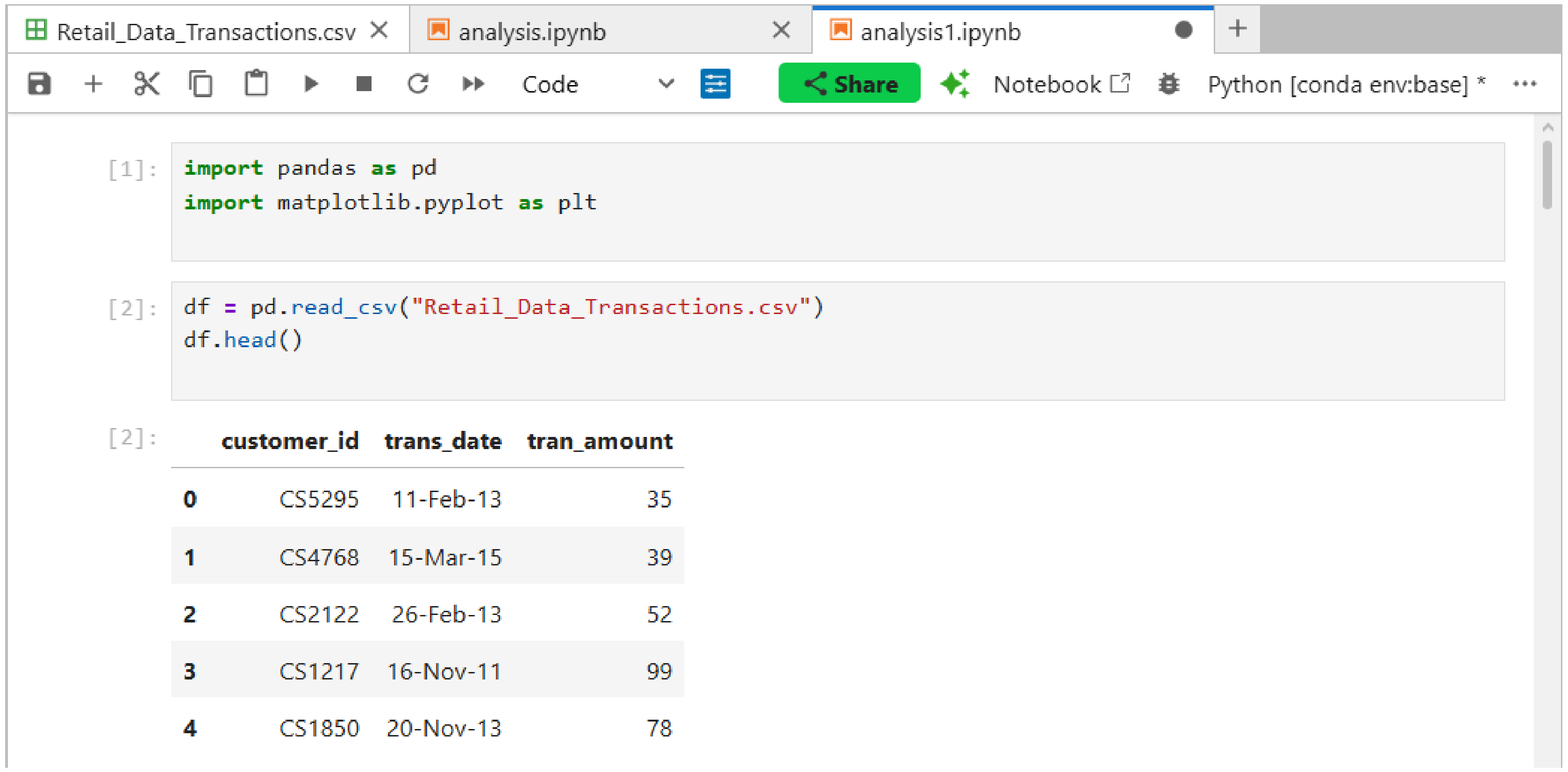
- **Excel:** Reporting and dashboard creation

PROJECT METHODOLOGY

- Data Collection
- Data Cleaning & Preparation (Python)
- Data Analysis (Python + SQL)
- Reporting & Dashboard Creation (Excel)

PHASE 1 – DATA COLLECTION & SETUP

- Retail transaction dataset collected from Kaggle
- Dataset loaded into Python environment
- Cleaned data stored into SQLite database for SQL analysis



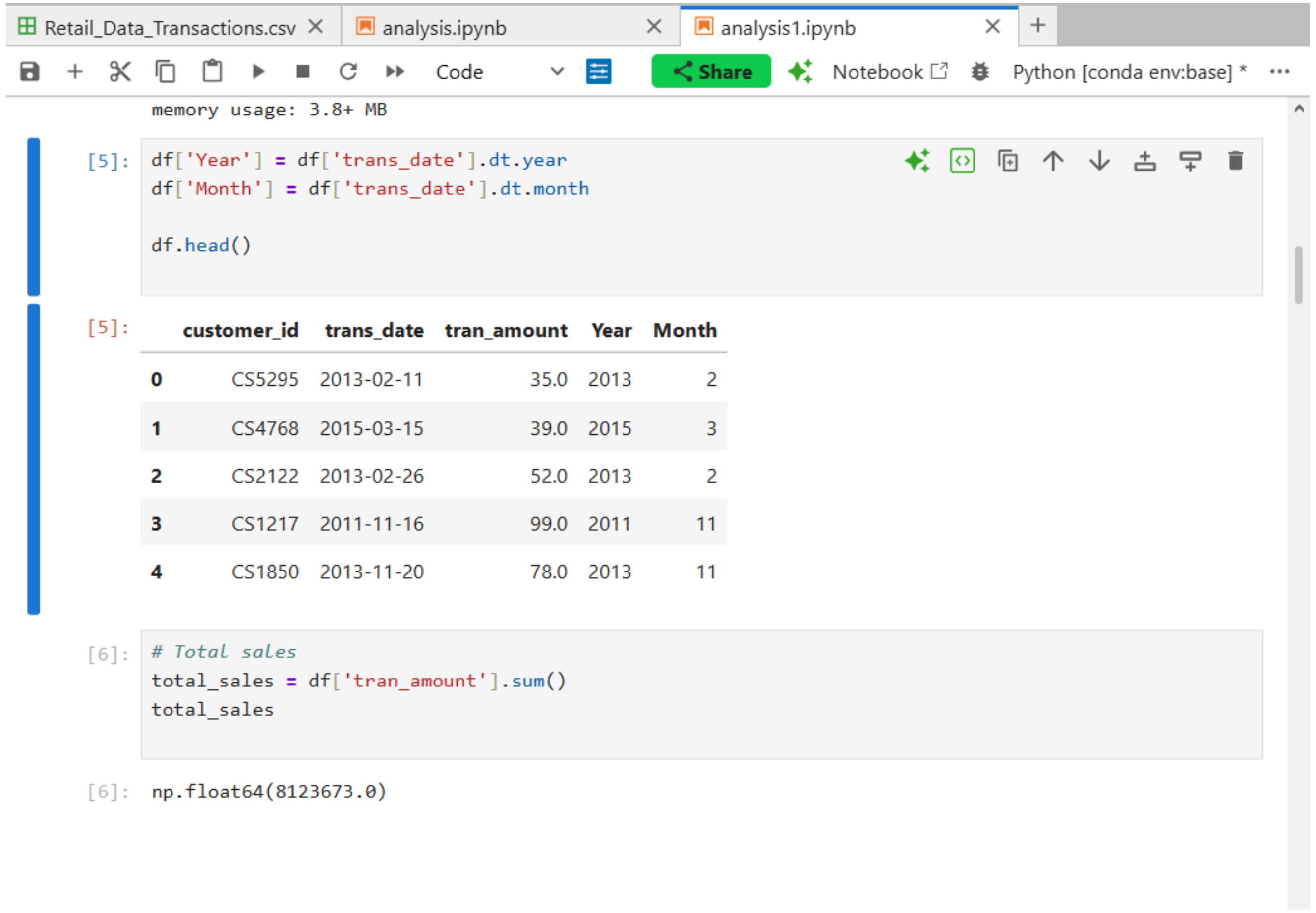
```
[1]: import pandas as pd
     import matplotlib.pyplot as plt

[2]: df = pd.read_csv("Retail_Data_Transactions.csv")
     df.head()
```

```
[2]:   customer_id  trans_date  tran_amount
    0       CS5295   11-Feb-13           35
    1       CS4768   15-Mar-15           39
    2       CS2122   26-Feb-13           52
    3       CS1217   16-Nov-11           99
    4       CS1850   20-Nov-13           78
```

Fig.2

PHASE 2 – DATA CLEANING & PREPARATION

- Removed duplicate and missing records
- Converted transaction amount to numeric format
- Converted transaction date to datetime format
- Extracted Year and Month for time-based analysis



```
memory usage: 3.8+ MB
```

```
[5]: df['Year'] = df['trans_date'].dt.year
     df['Month'] = df['trans_date'].dt.month

     df.head()
```

[5]:

	customer_id	trans_date	tran_amount	Year	Month
0	CS5295	2013-02-11	35.0	2013	2
1	CS4768	2015-03-15	39.0	2015	3
2	CS2122	2013-02-26	52.0	2013	2
3	CS1217	2011-11-16	99.0	2011	11
4	CS1850	2013-11-20	78.0	2013	11

```
[6]: # Total sales
     total_sales = df['tran_amount'].sum()
     total_sales
```

```
[6]: np.float64(8123673.0)
```

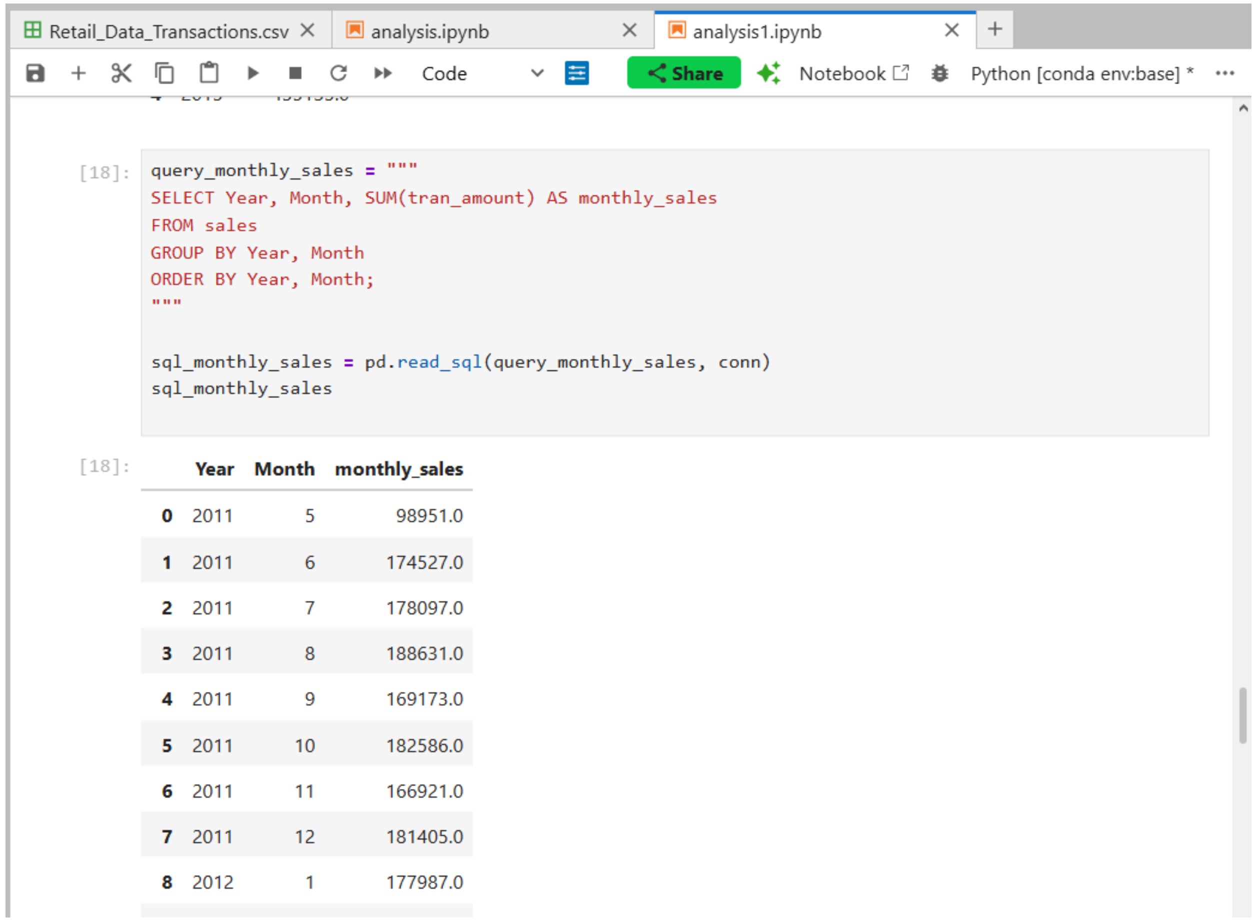
Fig.3

PHASE 3 – DATA ANALYSIS (PYTHON & SQL)

- Total sales calculation
- Year-wise sales analysis
- Monthly sales trend analysis

SQL Concepts Applied:

- SELECT
- GROUP BY
- Aggregate functions (SUM)
- “Sales data was stored in a SQLite database and analyzed using SQL queries for aggregation and trend analysis.”



```
[18]: query_monthly_sales = """
      SELECT Year, Month, SUM(tran_amount) AS monthly_sales
      FROM sales
      GROUP BY Year, Month
      ORDER BY Year, Month;
      """

      sql_monthly_sales = pd.read_sql(query_monthly_sales, conn)
      sql_monthly_sales
```

[18]:

	Year	Month	monthly_sales
0	2011	5	98951.0
1	2011	6	174527.0
2	2011	7	178097.0
3	2011	8	188631.0
4	2011	9	169173.0
5	2011	10	182586.0
6	2011	11	166921.0
7	2011	12	181405.0
8	2012	1	177987.0

Fig.4

PHASE 4 – REPORTING

- Generated tabular sales reports using Python and SQL
- Exported processed data to Excel
- Created visual reports using charts and graphs
- Developed an Excel dashboard summarizing sales performance

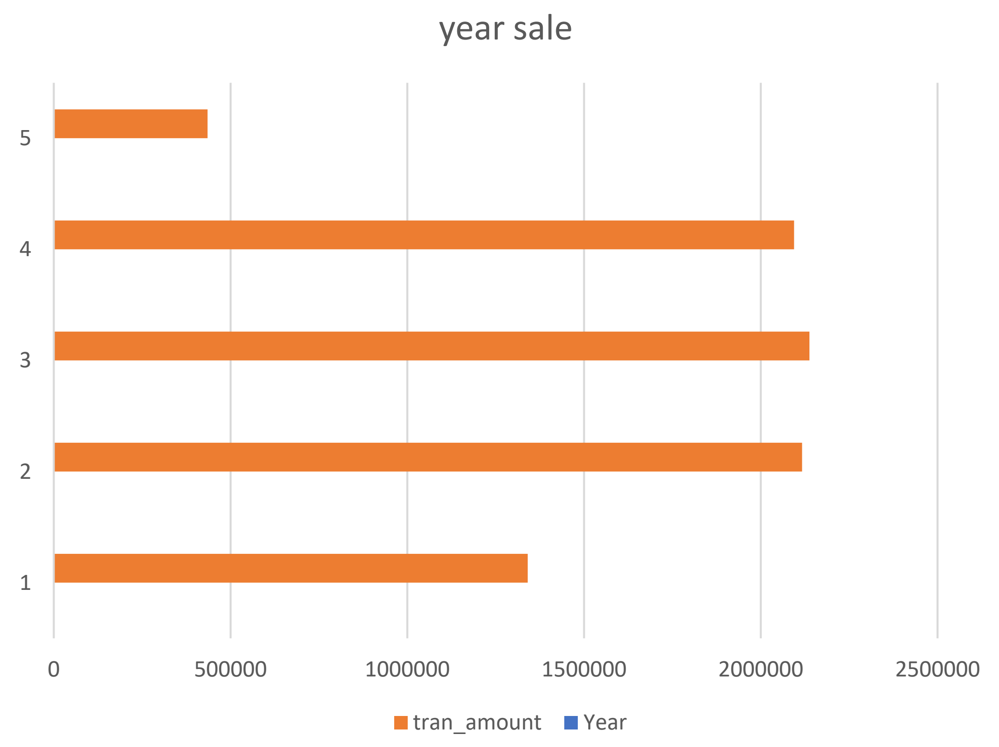

year sale
5
4
3
2
1
0
500000
1000000
1500000
2000000
2500000
tran_amount
Year

Fig.5

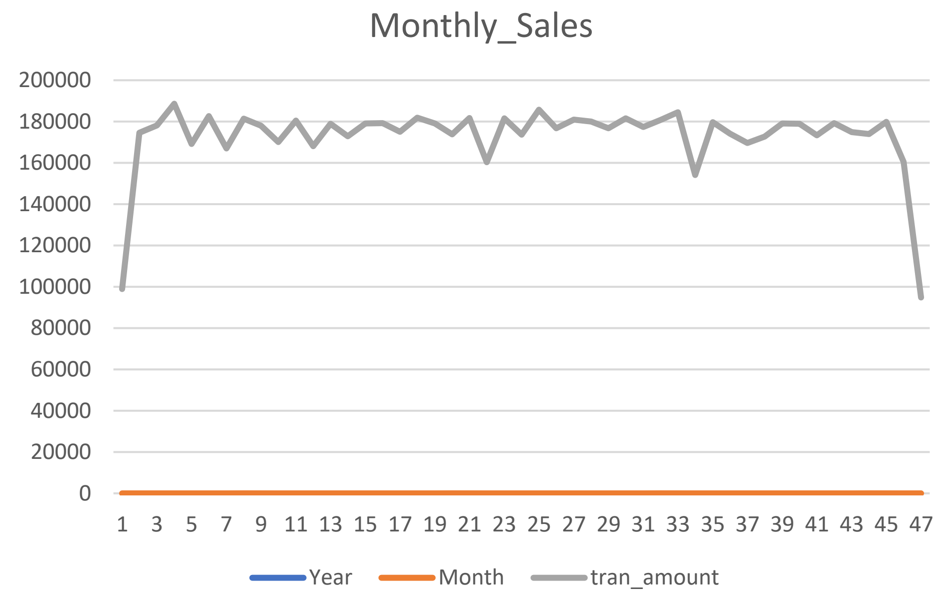

Monthly_Sales
200000
180000
160000
140000
120000
100000
80000
60000
40000
20000
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47
Year
Month
tran_amount

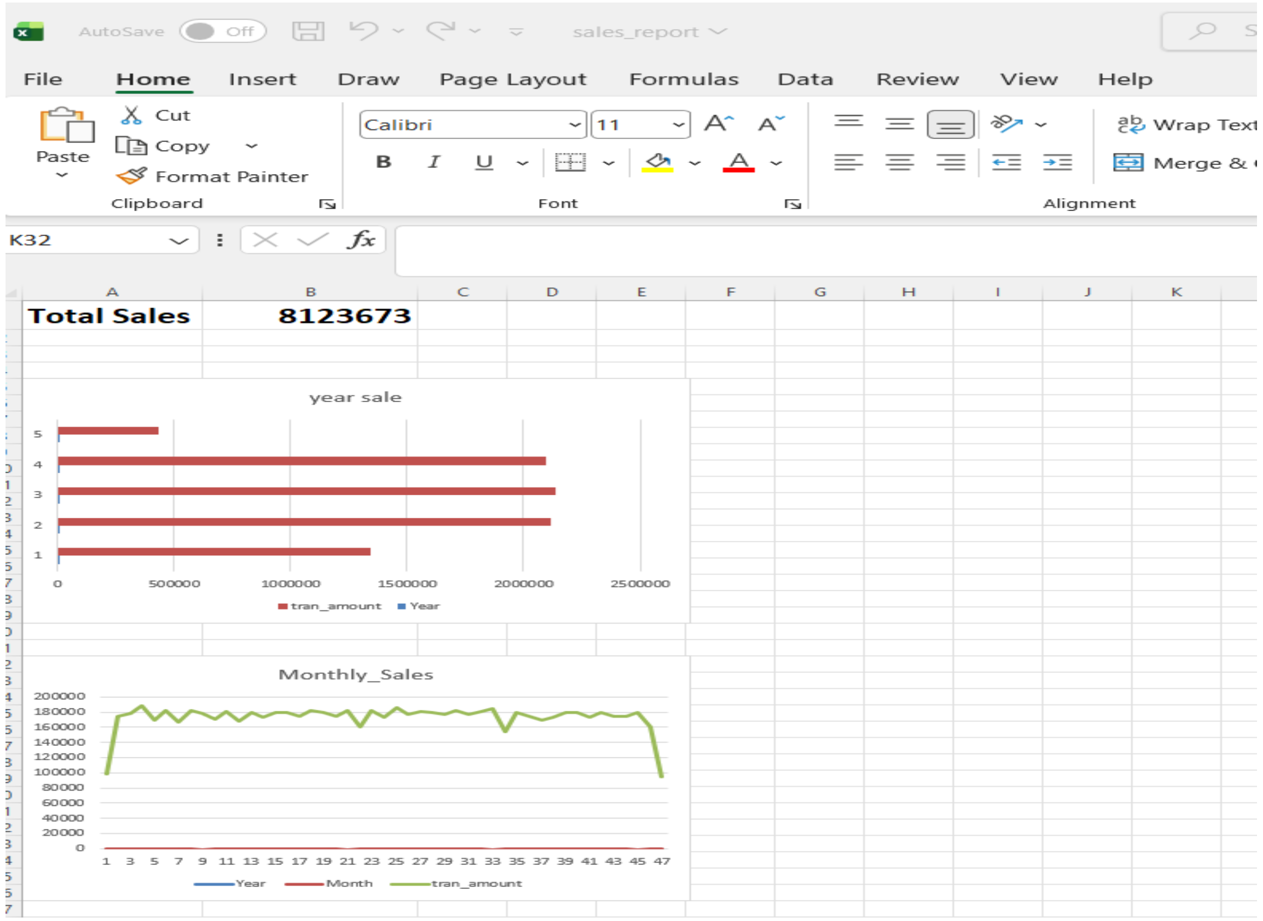
Fig.6

EXCEL REPORTS & DASHBOARD

- Cleaned sales data sheet
- Aggregated sales reports (Year-wise and Month-wise)

Dashboard displaying:

- Total Sales
- Yearly Sales Chart
- Monthly Sales Trend



Total Sales	8123673

Fig.7

RESULTS & INSIGHTS

- Clear visibility of yearly sales performance
- Identification of monthly sales trends
- Dashboard enables quick understanding of business performance
- Sales trends indicate seasonal variations across months.

CONCLUSION

The project successfully demonstrates how Python, SQL, and Excel can be used together for sales data analysis and reporting.
The generated reports and dashboard provide valuable insights that support informed business decisions.

THANK YOU

Thank you for the opportunity to work on this project.