

DATA SCIENCE TOOLBOX: PYTHON PROGRAMMING

PROJECT REPORT

(Project Semester January-April 2025)

SALES AND SHIPPING DATA ANALYSIS EDA

Submitted by

Ishita

Registration No. 12302892

Programme and Section BTECH CSE and K23EU

Course Code INT375

Under the Guidance of

Dr. Tanim Thakur

UID 224252

Discipline of CSE/IT

Lovely School of Computer Science and Engineering

Lovely Professional University, Phagwara

CERTIFICATE

This is to certify that Ishita bearing Registration no. 12302892 has completed INT375 project titled, **“SALES AND SHIPPING DATA ANALYSIS EDA”** under my guidance and supervision. To the best of my knowledge, the present work is the result of his/her original development, effort and study.

Signature and Name of the Supervisor

Designation of the Supervisor

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab.

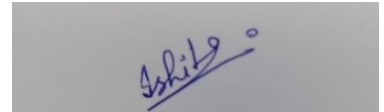
Date: 13-04-2025

DECLARATION

I, Ishita student of Computer Science and Engineering under CSE/IT Discipline at, Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date: 13-04-2025

Signature

A rectangular box containing a handwritten signature in blue ink. The signature appears to be 'Ishita' with a small circle at the end of the stroke.

Registration No. 12302892

Name of the student Ishita

ACKNOWLEDGEMENT

I would like to take this opportunity to express my sincere gratitude to **Dr. Tanima Thakur** for her invaluable support and guidance throughout this project. This Python-based **Exploratory Data Analysis (EDA)** was a part of my coursework, and it gave me the chance to apply my data analysis and visualization skills in a practical setting.

I'm also thankful to **Lovely Professional University** for encouraging hands-on learning and giving us the platform to work on real-world data using tools like **Pandas**, **Matplotlib**, and **Seaborn**.

A big shoutout to the data visualizations and EDA examples that inspired me during the process—they helped shape the direction of my analysis and made the learning experience smoother.

Lastly, heartfelt thanks to my friends and classmates for their constant encouragement and constructive feedback. Your motivation made this journey even more enjoyable!

CONTENTS OF THE REPORT

- Cover page
- Declaration
- Certificate
- Acknowledgement
- Table of Content

1. Introduction
2. Source of dataset
3. EDA process
4. Analysis on dataset (for each analysis)
 - i. Introduction
 - ii. General Description
 - iii. Specific Requirements, functions and formulas
 - iv. Analysis results
 - v. Visualization
5. Conclusion
6. Future scope
7. References

INTRODUCTION

In the era of digital transformation, organizations generate and store vast amounts of data on a daily basis. However, raw data in itself holds limited value unless properly analysed and interpreted. This is where the role of data analysis becomes critical. **Exploratory Data Analysis (EDA)** is one of the most important initial steps in the data science process, as it helps to understand the underlying structure of the data, identify patterns, detect anomalies, and formulate hypotheses for further analysis. This project presents an EDA conducted on a sample dataset of employee information using **Python** and its data analysis libraries.

The dataset used in this project includes various attributes related to employees, such as salary, gender, job category, previous experience, and minority status. The analysis aims to answer specific questions about the dataset, such as how salary is distributed among employees, whether there are any gender-based pay disparities, how job categories influence salary, and how previous work experience correlates with compensation. Additionally, the project explores minority representation across different job categories, providing insights into workforce diversity.

To achieve these objectives, **Python** was chosen as the primary programming language due to its flexibility, readability, and extensive library support. The project primarily utilizes the **Pandas** library for data cleaning and manipulation, and **Matplotlib** and **Seaborn** for creating informative and visually appealing plots. These tools enable efficient data handling and high-quality visualizations, which are essential for deriving meaningful insights from complex datasets.

This project was undertaken as part of my academic coursework at **Lovely Professional University**, under the guidance of **Dr. Tanima Thakur**. The primary goal was not only to gain hands-on experience with real-world data but also to develop a strong foundation in data analysis, which is a crucial skill in today's tech-driven job market. Through this project, I have learned how to approach data analytically, interpret findings, and present results in a coherent and visually engaging manner.

SOURCE OF DATASET

The data employed in this project was sourced from an open site named learning container (<https://www.learningcontainer.com/download/sample-sales-data-excel-xlsx/>). It is one of the generic sales datasets commonly utilized for teaching and practicing methods of data visualization and analysis across tools such as Excel, Power BI, and Tableau.

The dataset includes sales transaction data from the years 2014 to 2017 for various regions and product categories, capturing detailed records of customer orders, sales numbers, profit margins, and shipping information.

Some of the most important fields in the dataset are:

Order Date – Transaction date

Region – Geographical area where the sale was made

Segment – Customer segment (for example, Consumer, Corporate, Home Office)

Category/Sub-Category – Product type and category

Sales – Revenue from the sale

Profit – Profit made on the transaction

Ship Mode – Shipping mode used

Quantity – Quantity of products sold

Discount – Discount offered on the product

City/Country/State – Customer location details

This dataset gave a good base for examining various facets of sales performance, customer behavior, and shipping efficiency using Excel dashboards.

EDA PROCESS

This project involved performing Exploratory Data Analysis (EDA) on an employee dataset using Python. The main goal was to uncover insights regarding employee salaries, job roles, gender-based differences, experience, and minority representation. The following steps outline the EDA process followed:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the Excel file
file_path = r'C:\Users\ishit\OneDrive\Python Dataset.xls'
df = pd.read_excel(file_path)

# Clean column names (remove extra spaces)
df.columns = df.columns.str.strip()

# Show columns to verify
print("Available columns:", df.columns.tolist())

# Convert dates
df['Order Date'] = pd.to_datetime(df['Order Date'], dayfirst=True)
df['Ship Date'] = pd.to_datetime(df['Ship Date'], dayfirst=True)

# Create new column for monthly analysis
df['Order Month'] = df['Order Date'].dt.to_period('M').astype(str)
```

Figure 1 Cleaning

PROCESS

1. Data Importing and Loading

The first and most essential step in any data analysis project is to import the necessary libraries and load the dataset.

Libraries Used:

- **pandas**: Used for data handling and manipulation such as loading Excel data, cleaning columns, and grouping.
- **matplotlib.pyplot**: Utilized for creating static visualizations like bar plots and line charts.
- **seaborn**: Built on top of matplotlib, it provides a high-level interface for drawing attractive and informative statistical graphics.

Loading the Dataset:

python

CopyEdit

```
df = pd.read_excel(file_path)
```

- `file_path` contains the path to the dataset in Excel format.
- `pd.read_excel()` reads the Excel file and converts it into a DataFrame.
- After this step, the entire dataset is accessible through the variable `df` for further analysis.

2. Initial Data Check

Before beginning the core analysis, it's essential to verify that the dataset has loaded correctly.

Code Used:

python

CopyEdit

```
df.head()
```

What It Does:

- Displays the first five rows of the DataFrame.
- Helps ensure the dataset is correctly loaded, the column names are accurate, and no critical data issues (e.g., nulls or strange formats) are immediately visible.

3. Objective-Wise Visual Analysis

After loading and checking the dataset, Exploratory Data Analysis (EDA) is conducted to derive meaningful insights. Each of the following objectives answers a specific business question.

Objective 1: Sales and Profit Trends Over Time

- **Goal:** Identify how sales and profit vary monthly over the given time frame.
- **Tool:** Line plot
- **Insight:** Reveals seasonal trends and performance patterns throughout the year.

python

CopyEdit

```
sns.lineplot(data=monthly, x='Order Month', y='Sales', label='Sales')
```

```
sns.lineplot(data=monthly, x='Order Month', y='Profit', label='Profit')
```

Objective 2: Top 10 States by Sales and Profit

- **Goal:** Determine which states generate the most revenue and profit.
- **Tool:** Vertical bar chart
- **Insight:** Helps focus marketing and logistics on top-performing regions.

python

CopyEdit

```
top_states.plot(kind='bar', figsize=(10, 5), title='Top 10 States by Sales & Profit')
```

Objective 3: Top 10 Cities by Sales

- **Goal:** Identify cities that contribute the most to overall sales.
- **Tool:** Horizontal bar chart
- **Insight:** Pinpoints city-level hotspots for business activity.

python

CopyEdit

```
top_cities.plot(kind='barh', figsize=(8, 5), title='Top 10 Cities by Sales')
```

Objective 4: Segment-wise Sales Performance

- **Goal:** Analyze how different customer segments contribute to sales.
- **Tool:** Bar plot
- **Insight:** Guides segmentation strategy and customer targeting.

python

CopyEdit

```
sns.barplot(data=df, x='Segment', y='Sales', estimator=sum)
```

Objective 5: Impact of Discount on Profit

- **Goal:** Examine how discounts affect profit margins.
- **Tool:** Scatter plot
- **Insight:** High discounts often lead to reduced or negative profits.

python

CopyEdit

```
sns.scatterplot(data=df, x='Discount', y='Profit')
```

4. How It Is Done: Code Explanation

Here's a breakdown of how each visualization was created:

Objective 1: Monthly Sales and Profit Trends

python

CopyEdit

```
monthly = df.groupby('Order Month')[['Sales', 'Profit']].sum().reset_index()
```

```
sns.lineplot(data=monthly, x='Order Month', y='Sales')
```

```
sns.lineplot(data=monthly, x='Order Month', y='Profit')
```

Objective 2: Top 10 States by Sales and Profit

python

CopyEdit

```
top_states = df.groupby('State')[['Sales', 'Profit']].sum().sort_values('Sales', ascending=False).head(10)
```

```
top_states.plot(kind='bar')
```

Objective 3: Top 10 Cities by Sales

python

CopyEdit

```
top_cities = df.groupby('City')['Sales'].sum().sort_values(ascending=False).head(10)
```

```
top_cities.plot(kind='barh')
```

Objective 4: Segment-wise Sales

python

CopyEdit

```
sns.barplot(data=df, x='Segment', y='Sales', estimator=sum)
```

Objective 5: Discount vs Profit

python

CopyEdit

```
sns.scatterplot(data=df, x='Discount', y='Profit')
```

5. Plot Customization

To improve clarity and presentation, several visualization enhancements were applied:

- **Titles:** Added using `plt.title()` to explain each graph.
- **Axis Labels:** Labeled using `plt.xlabel()` and `plt.ylabel()`.
- **Legends:** Enabled to distinguish between variables like sales and profit.
- **Figure Sizes:** Managed using `plt.figure(figsize=(width, height))`.
- **Tight Layout:** Ensured spacing is handled well using `plt.tight_layout()`.

6. Summary and Interpretation

Each visualization provided answers to critical business questions. The findings are summarized below:

- **Sales & Profit Trends:** Monthly sales and profit vary significantly, revealing high-performing months and seasonal dips.
- **Top States and Cities:** Certain states and cities dominate sales performance, indicating strong market presence.
- **Segment Analysis:** Specific customer segments (e.g., Consumer or Corporate) drive more sales, informing targeting efforts.
- **Discount Impact:** Discounts negatively affect profit in many cases, emphasizing the need for strategic discounting.
- **Overall Insight:** This analysis helps identify patterns in sales operations and profitability, providing a solid foundation for decision-making in sales strategy, customer segmentation, and revenue optimization.

So, the EDA part in my project is:

- Time-series analysis
- Geographical sales distribution
- Customer segment profiling
- Correlation inspection
- Distribution analysis

Objective 1: Sales and Profit Trends Over Time

i. Introduction:

To analyze how Sales and Profit have fluctuated over time, a time-series analysis is performed using monthly aggregated data.

ii. General Description:

This analysis helps in understanding sales seasonality, identifying profitable months, and spotting patterns or inconsistencies in business performance over the time period available.

iii. Specific Requirements, Functions, and Formulas:

- `pd.to_datetime()` to convert date columns.
- `dt.to_period('M')` to extract month and year.
- `groupby('Order Month')[['Sales', 'Profit']].sum()` to aggregate monthly totals.
- `sns.lineplot()` from Seaborn to visualize time trends.
- `plt.xticks(rotation=45)` to improve x-axis readability.

iv. Analysis Results:

The line chart clearly shows fluctuations in both sales and profit over different months. Peaks indicate high-performing months, while dips may represent seasonal lulls or operational challenges.

v. Visualization:

```
# Objective 1: Sales and Profit Trends Over Time
monthly = df.groupby('Order Month')[['Sales', 'Profit']].sum().reset_index()

plt.figure(figsize=(12, 5))
sns.lineplot(data=monthly, x='Order Month', y='Sales', label='Sales')
sns.lineplot(data=monthly, x='Order Month', y='Profit', label='Profit')
plt.xticks(rotation=45)
plt.title('Monthly Sales and Profit')
plt.tight_layout()
plt.show()
```

Figure 2 Code

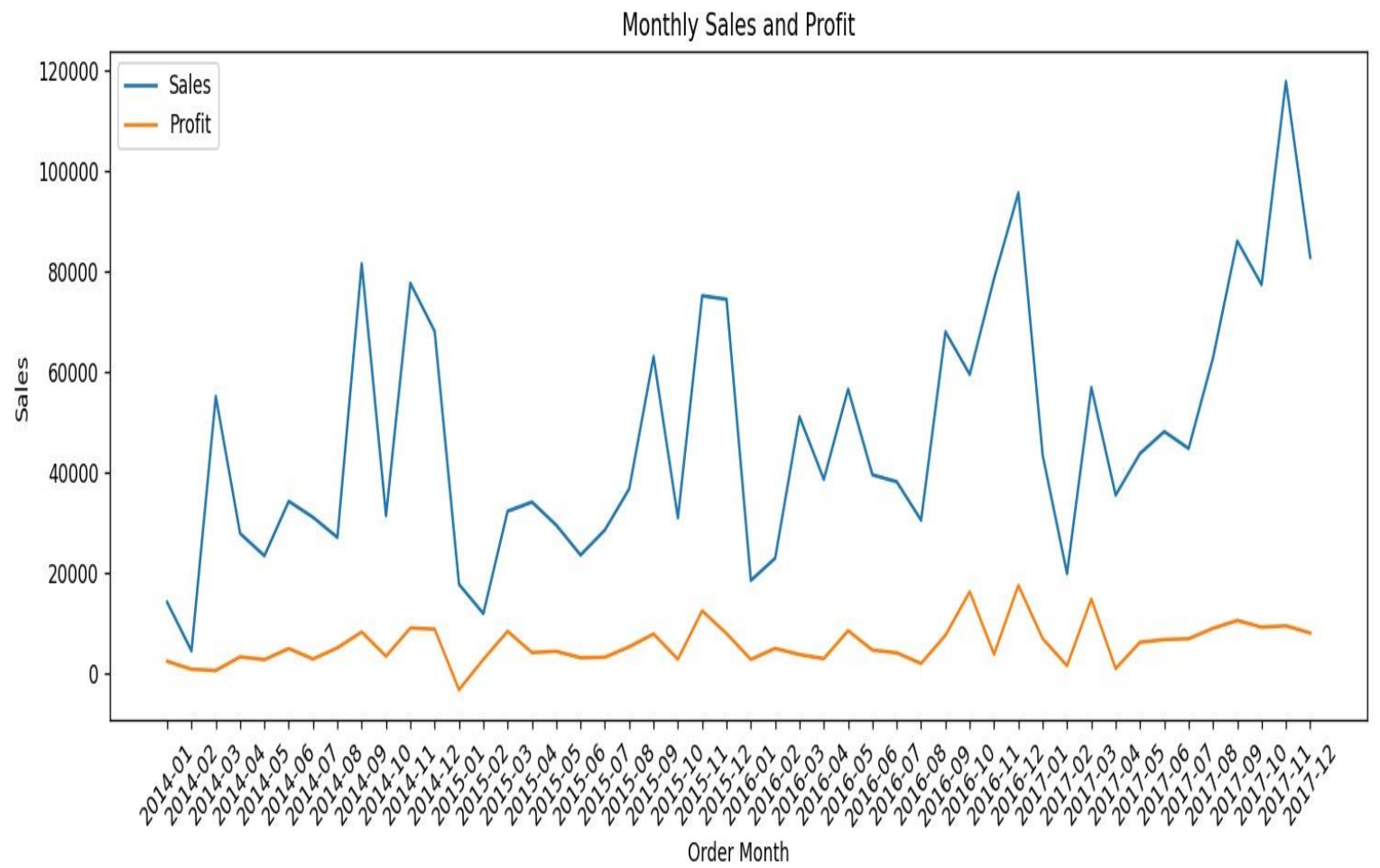


Figure 3 Graph

Objective 2: Top 10 States by Sales and Profit

i. Introduction:

This analysis identifies which states contribute the most to overall sales and profitability.

ii. General Description:

Understanding state-level performance allows businesses to focus marketing and logistics efforts in regions that perform well or need improvement.

iii. Specific Requirements, Functions, and Formulas:

- `groupby('State')[['Sales', 'Profit']].sum()` for state-wise totals.
- `sort_values('Sales', ascending=False).head(10)` to extract the top 10.
- `plot(kind='bar')` to visualize results using a bar graph.

iv. Analysis Results:

The bar plot reveals the top 10 performing states. Some states may show high sales but lower profits, indicating possible high costs or discounts.

v. Visualization:

```
# Objective 2: Top 10 States by Sales and Profit
top_states = df.groupby('State')[['Sales', 'Profit']].sum().sort_values('Sales', ascending=False).head(10)
top_states.plot(kind='bar', figsize=(10, 5), title='Top 10 States by Sales & Profit')
plt.ylabel('Amount')
plt.tight_layout()
plt.show()
```

Figure 4 Code

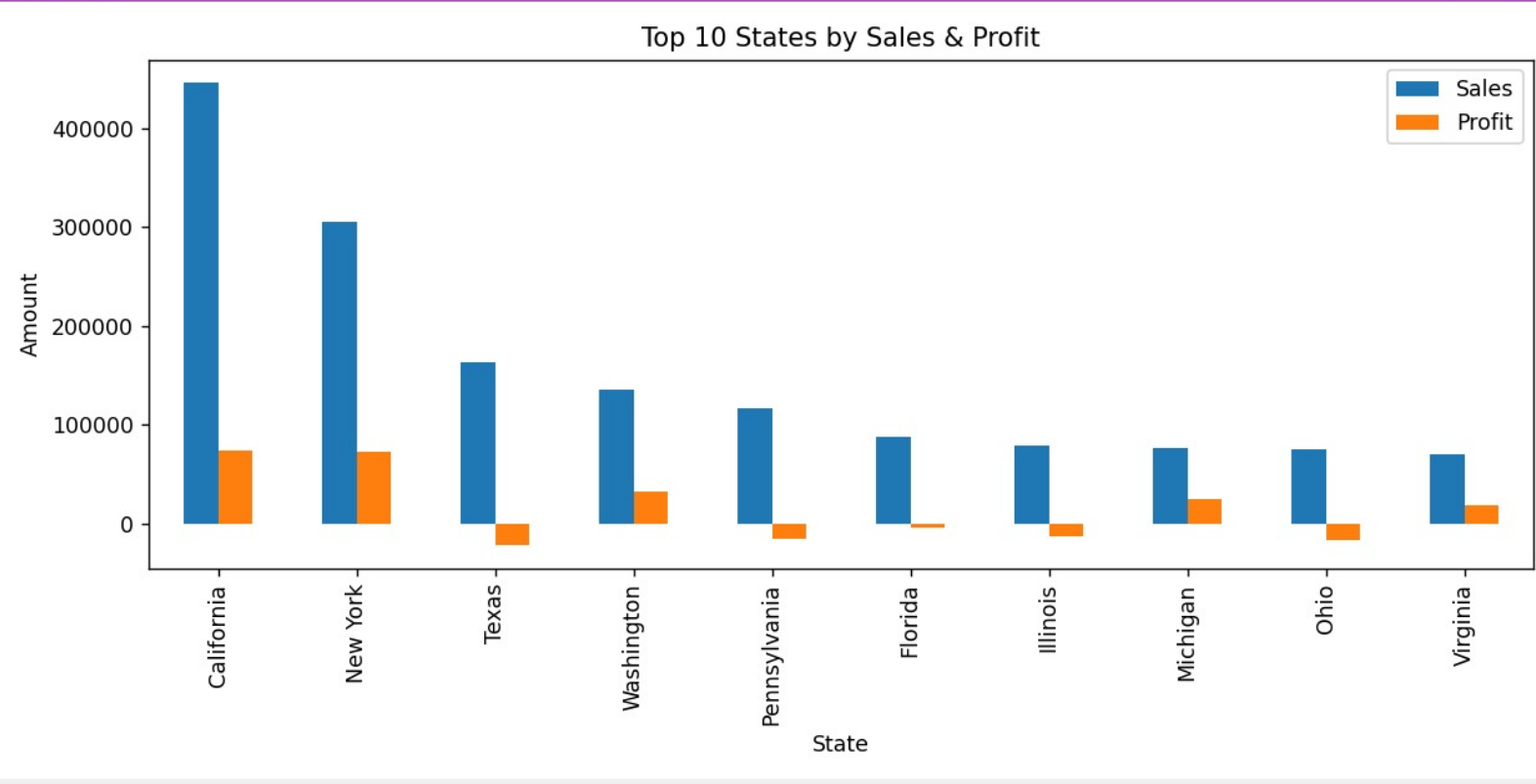


Figure 5 Graph

Objective 3: Top 10 Cities by Sales

i. Introduction:

To determine which cities generate the highest sales, a focused city-level sales analysis is done.

ii. General Description:

Analyzing top cities helps pinpoint specific urban markets that drive revenue and identify potential expansion areas.

iii. Specific Requirements, Functions, and Formulas:

- `groupby('City')['Sales'].sum()` for city-level sales.
- `sort_values(ascending=False).head(10)` to get top cities.
- `plot(kind='barh')` to create a horizontal bar graph.

iv. Analysis Results:

The graph highlights the cities that contribute most significantly to sales. These cities can be targeted for promotional offers or sales incentives.

v. Visualization:

```
# Objective 3: Top 10 Cities by Sales
top_cities = df.groupby('City')['Sales'].sum().sort_values(ascending=False).head(10)
top_cities.plot(kind='barh', figsize=(8, 5), title='Top 10 Cities by Sales')
plt.xlabel('Sales')
plt.tight_layout()
plt.show()
```

Figure 6 Code

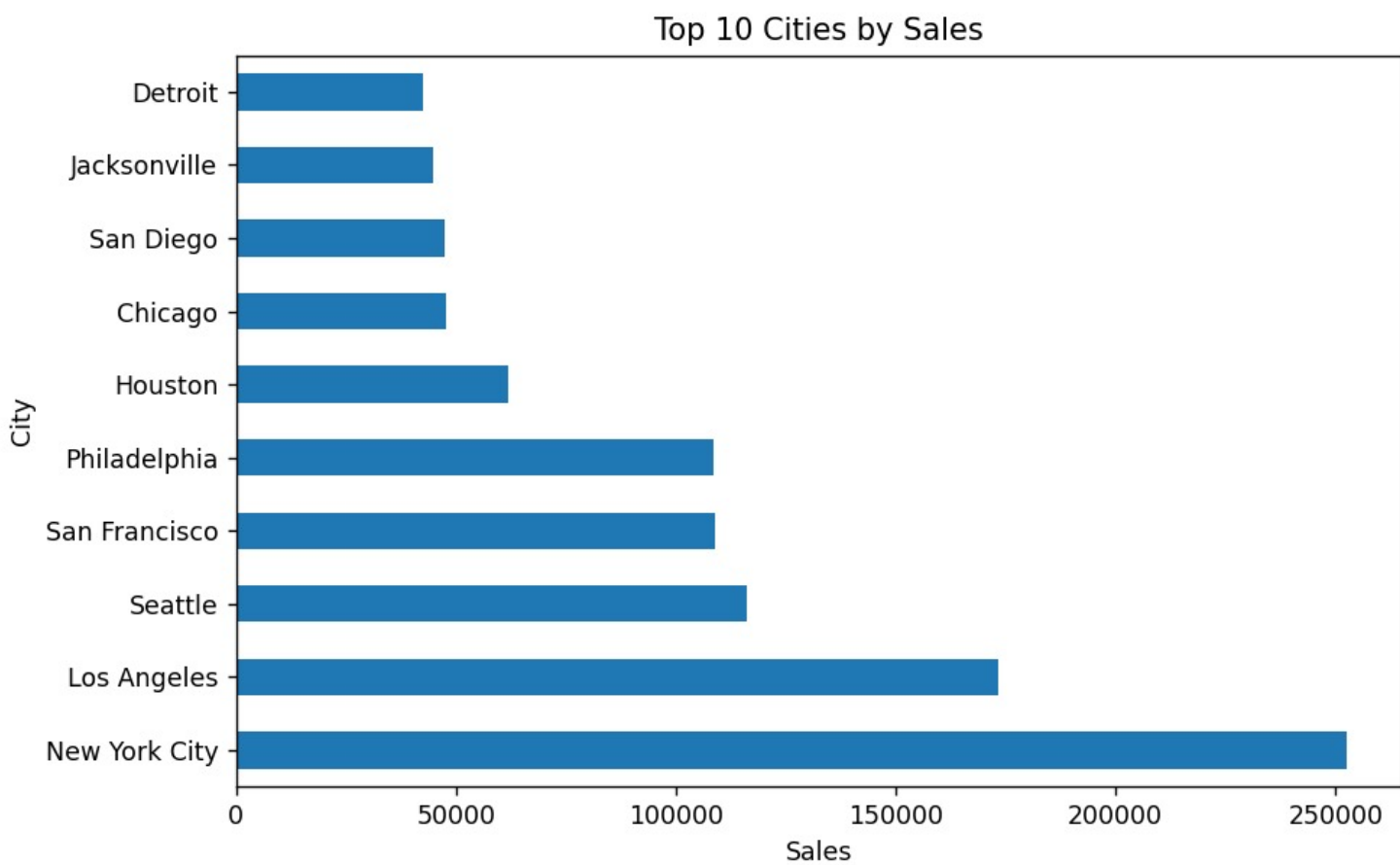


Figure 7 Graph

Objective 4: Segment-wise Sales Performance

i. Introduction:

This analysis evaluates which customer segments bring in the most sales.

ii. General Description:

Customer segmentation includes groups like Consumer, Corporate, and Home Office. Knowing which segment performs best helps in targeted marketing and resource allocation.

iii. Specific Requirements, Functions, and Formulas:

- `sns.barplot()` with `estimator=sum` to calculate total sales by segment.
- Grouped directly by the Segment column.

iv. Analysis Results:

The bar plot shows that one segment (usually Consumer) generates the highest sales, while others may show lower revenue, helping shape customer engagement strategies.

v. Visualization:

```
# Objective 4: Segment-wise Sales Performance
plt.figure(figsize=(8, 5))
sns.barplot(data=df, x='Segment', y='Sales', estimator=sum)
plt.title('Total Sales by Customer Segment')
plt.tight_layout()
plt.show()
```

Figure 8 Code

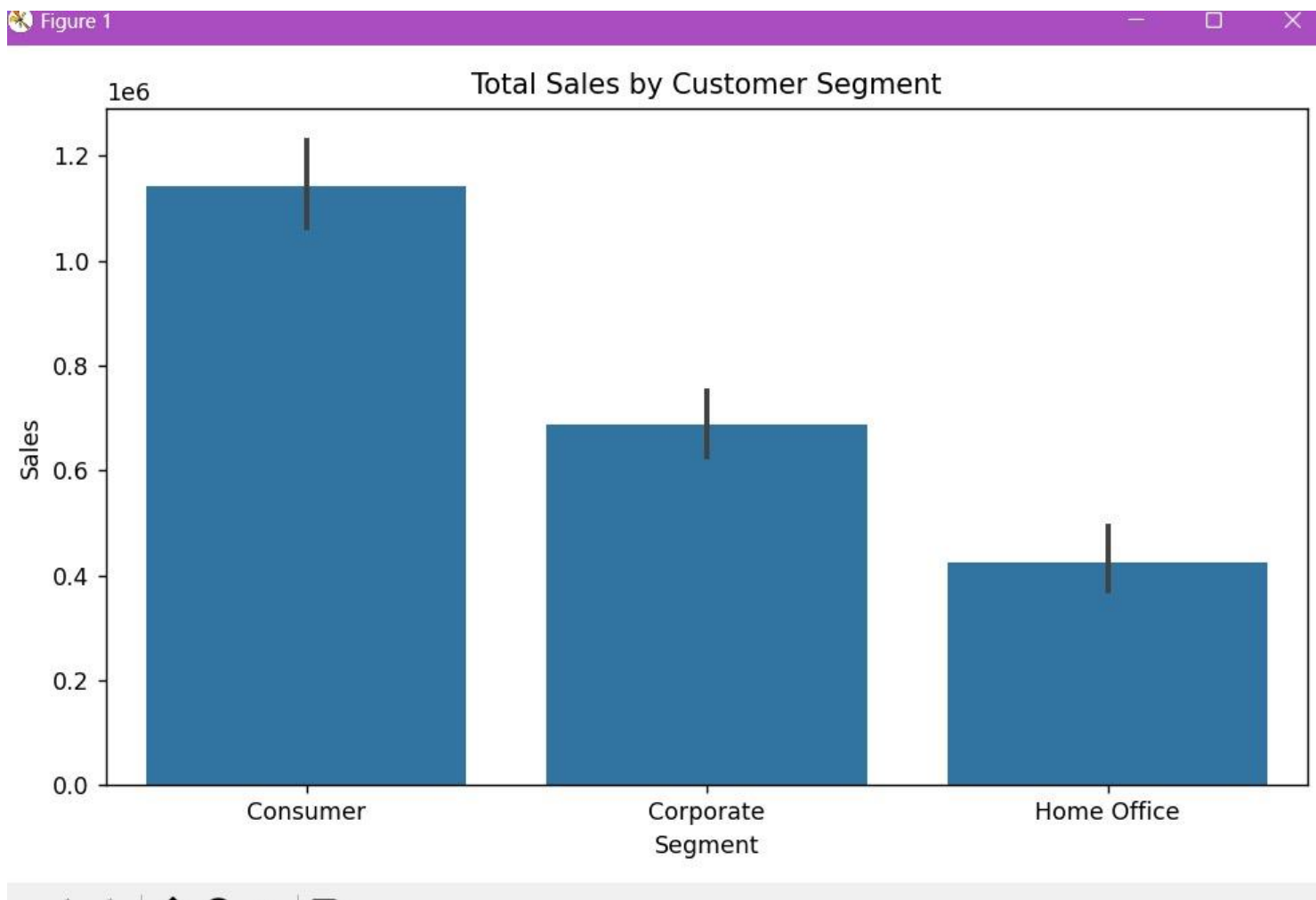


Figure 9 Graph

Objective 5: Impact of Discount on Profit

i. Introduction:

This analysis examines whether discounts positively or negatively affect profit margins.

ii. General Description:

Discounting is a common pricing strategy, but excessive or poorly planned discounts can reduce profitability.

iii. Specific Requirements, Functions, and Formulas:

- `sns.scatterplot(x='Discount', y='Profit')` to create a scatter plot.
- Used to explore the relationship between discounts and profits for all transactions.

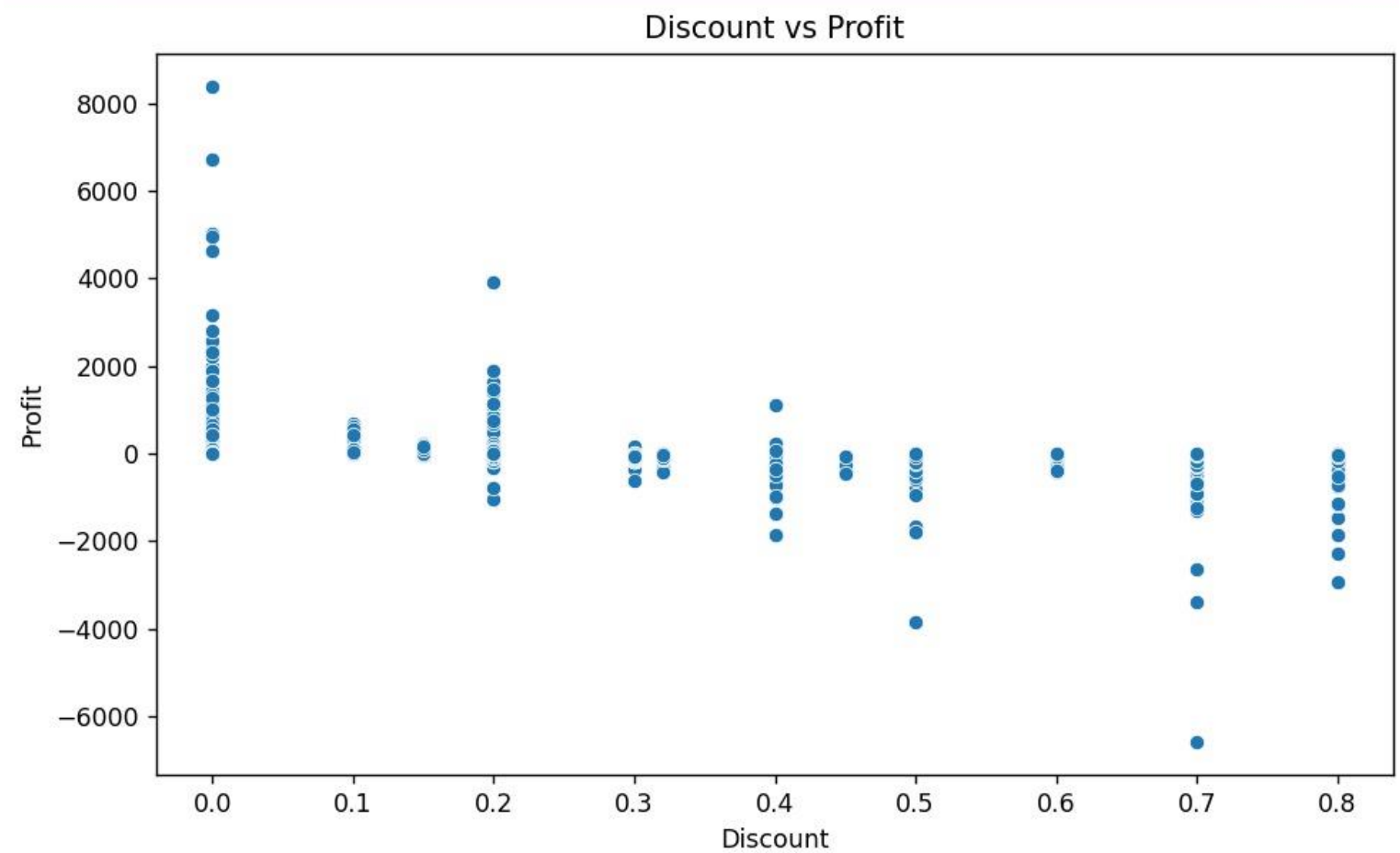
iv. Analysis Results:

The scatter plot generally shows that higher discounts often lead to negative profits, highlighting the need to carefully manage discount strategies.

v. Visualization:

```
# Objective 5: Impact of Discount on Profit
plt.figure(figsize=(8, 5))
sns.scatterplot(data=df, x='Discount', y='Profit')
plt.title('Discount vs Profit')
plt.tight_layout()
plt.show()
```

Figure 11 Code



CONCLUSION

This project aimed to perform Exploratory Data Analysis (EDA) on a sales dataset to extract meaningful insights related to sales performance, regional contributions, customer segmentation, discount impacts, and profit trends. Through a combination of grouping operations and visualizations, several key findings emerged:

- **Sales and profit trends** showed noticeable fluctuations over time, highlighting seasonal patterns and performance peaks.
- **Top-performing states and cities** were identified, providing a clear picture of high-revenue regions that can be targeted for strategic business growth.
- **Customer segment analysis** revealed that certain segments contribute significantly more to sales, helping in focused marketing and customer engagement.
- **Discount analysis** demonstrated that higher discounts often correlated with reduced profit, emphasizing the need for a balanced pricing strategy.
- **Regional and segment-wise analysis** provided clarity on which areas and demographics drive the most value to the business.

Overall, this analysis offers valuable insights into sales operations and profitability. These findings can guide data-driven decision-making in areas such as regional sales strategy, customer targeting, pricing models, and promotional planning.

FUTURE SCOPE

While this project offered valuable insights, there are several directions where this work can be extended:

- **Predictive Modeling:** Implement regression models to predict salary based on experience, job category, and other factors.
- **Interactive Dashboards:** Use tools like Plotly or Power BI to create interactive versions of the visualizations for better exploration.
- **More Features:** Incorporate additional data points such as education level, performance scores, or geographic location to enhance analysis depth.
- **Time-Based Trends:** If longitudinal data is available, analyse trends over time to understand changes in pay structure, diversity, and hiring patterns.
- **Anomaly Detection:** Use machine learning to detect outliers or anomalies in salary or experience data.

REFERENCES

1. McKinney, W. (2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython* (2nd ed.). O'Reilly Media.
2. Hunter, J. D. (2007). *Matplotlib: A 2D graphics environment*. Computing in Science & Engineering, 9(3), 90-95.
3. Waskom, M. L. (2021). *Seaborn: Statistical data visualization*. Journal of Open Source Software, 6(60), 3021.
4. Microsoft. (n.d.). *Sample Data file with shipping and sales data*. Retrieved from <https://www.learningcontainer.com/download/sample-sales-data-excel-xlsx/>