WALMART RETAIL SALES ANALYSIS DASHBOARD

PROJECT REPORT

By

ADITHYAN R [RA2211027010040]

MOHAMMED ZEESHAN SAMSEER [RA2211027010049]

ISHAN DEY [RA2211027010069]

AZAM TANZEEM [RA2211027010081]

SHREEYANSH VERMA [RA2211027010085]

Under the guidance of

Ms. Gayatri M

In partial fulfilment for the Course

of

21CSE421T – BUSINESS INTELLIGENCE AND ANALYTICS

In

COMPUTER SCIENCE AND ENGINEERING

WITH SPECILIZATION IN BIG DATA ANALYTICS

FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

NOVEMBER 2025

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report for the course **21CSE421T – BUSINESS INTELLIGENCE AND ANALYTICS** entitled in " **WALMART RETAIL SALES ANALYSIS DASHBOARD** " is the bonafide work of **ADITHYAN R [RA2211027010040], MOHAMMED ZEESHAN SAMSEER [RA2211027010049], ISHAN DEY [RA2211027010069], AZAM TANZEEM [RA2211027010081], SHREEYANSH VERMA [RA2211027010085]** who carried out the work under my supervision.

SIGNATURE	**SIGNATURE**
Ms. Gayatri M	**Dr. Kavitha V**
Assistant Professor(Jr)	**Professor and Head**
Data Science and Business Systems,	**Data Science and Business systems**
SRM Institute of Science and Technology,	**SRM Institute of Science and Technology**
Kattankulathur .	**Kattankulathur.**

ABSTRACT

The retail industry produces enormous amounts of data daily from stores, customers, and supply chains. Extracting meaningful insights from such data is essential for effective decision-making and business growth. This project, **"Walmart Sales Analytics Dashboard using Power BI and Python,"** focuses on analyzing Walmart's historical sales data using Python for data cleaning and Power BI for dashboard visualization.

The dataset, sourced from Kaggle, contains information about weekly sales across multiple Walmart stores, departments, and features such as temperature, fuel prices, holiday indicators, and consumer spending indexes. Using Python (Google Colab), the dataset was cleaned and preprocessed by handling missing values, correcting data types, and deriving additional features like month and year. Subsequently, exploratory data analysis (EDA) was performed to understand sales trends, correlations, and seasonal effects.

The cleaned dataset was visualized in Power BI to build an **interactive analytics dashboard** showing store performance, weekly sales trends, and external factors influencing revenue. Key findings revealed that **holiday periods, fuel prices, and temperature** have a measurable impact on sales. The dashboard allows management to identify high-performing stores, analyze underperforming ones, and forecast sales for better inventory and marketing strategies.

This study demonstrates the importance of integrating **data preprocessing (Python)** with **data visualization (Power BI)** to produce actionable business intelligence. The project showcases the practical application of data science concepts such as data cleaning, visualization, and analytical modeling in a real-world retail scenario.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Background

In the rapidly evolving retail industry, businesses increasingly rely on data analytics to enhance operational efficiency and improve decision-making. Walmart, one of the world's largest retail chains, collects massive amounts of data every week from its stores. This data includes sales transactions, weather conditions, consumer indices, and promotional events. Analyzing such data allows Walmart to understand customer behavior, forecast demand, manage inventory, and optimize marketing strategies.

The Walmart Sales Analytics Dashboard was developed to visualize sales performance across different stores and departments using the Power BI platform. Before visualization, the dataset was processed and cleaned using Python in Google Colab. The integration of data preprocessing and dashboard visualization bridges the gap between raw data and business insight, enabling a clear understanding of how environmental and economic factors affect store performance.

1.2 Motivation

The retail sector faces several challenges — unpredictable sales patterns, seasonality effects, and competitive market dynamics. Walmart's dataset provides an opportunity to analyze real-world business data to derive insights that support data-driven decisions.

Motivation for this project stems from the desire to:

- Apply data analytics to real-world business problems.
- Learn how to integrate Python-based preprocessing with Power BI dashboards.
- Identify sales trends, seasonal patterns, and store performance variations.
- Support inventory management and marketing strategies through insights.

By exploring these aspects, the project not only enhances technical and analytical skills but also strengthens understanding of how data visualization impacts business decision-making.

1.3 Objective

The primary objective of this project is to analyze Walmart's weekly sales dataset and build an interactive Power BI dashboard that provides actionable insights. Specific objectives include:

1. Cleaning and preprocessing the Kaggle Walmart sales dataset using Python (Google Colab).
2. Performing exploratory data analysis (EDA) to identify trends and patterns.
3. Developing a comprehensive Power BI dashboard to visualize sales data dynamically.
4. Highlighting key factors influencing store performance such as holidays, temperature, and fuel prices.
5. Generating insights for improved decision-making and forecasting.

1.4 Problem Statement

Walmart operates numerous stores across the United States, each experiencing distinct sales behaviors due to regional and seasonal variations. However, decision-makers often struggle to determine why certain stores outperform others or how external factors like temperature or economic conditions influence sales. The **challenge** lies in analyzing vast datasets to extract patterns that are both reliable and interpretable.

The raw dataset from Kaggle contains inconsistencies such as missing values, differing data formats, and potential outliers. Without proper cleaning and visualization, it is difficult to derive useful insights. Moreover, understanding the effects of **holiday weeks**, **fuel prices**, and **CPI** on sales performance requires advanced data analytics techniques.

Therefore, the problem addressed in this project is:

“How can Walmart’s weekly sales data be cleaned, processed, and visualized effectively to provide actionable insights into store performance and external influencing factors?”

This problem is addressed by combining Python-based data preprocessing with Power BI’s advanced visualization capabilities. The integration ensures that stakeholders can view dynamic reports and make informed business decisions in real-time.

1.5 Challenges

1. Data Quality Issues: Missing or incorrect values for features such as temperature, CPI, and fuel price had to be handled carefully.
2. Seasonality and Holidays: Sales are heavily influenced by holidays such as Thanksgiving and Christmas, making consistent trend analysis challenging.
3. Feature Correlation: Determining which factors truly impact sales and avoiding misleading correlations required careful statistical analysis.
4. Integration: Ensuring that Python-cleaned data seamlessly integrates into Power BI for visualization posed technical challenges.

CHAPTER 2

DATA UNDERSTANDING

2.1 Dataset Description

The dataset was sourced from Kaggle's "Walmart Store Sales Forecasting" competition. It contains weekly sales data from 45 stores across multiple departments and features external factors like temperature, fuel price, consumer price index (CPI), and unemployment rate.

Column Name	**Description**
Store	Store ID number
Dept	Department number
Date	Week of sales (YYYY-MM-DD)
Weekly_Sales	Sales revenue for the department in that week
IsHoliday	Boolean indicator (TRUE/FALSE) for holiday weeks
Temperature	Average temperature in the region
Fuel_Price	Fuel price per gallon
CPI	Consumer Price Index
Unemployment	Regional unemployment rate

2.2 Nature of Data

- Type: Time series data (weekly)
- Size: ~ 420,000 records
- Period Covered: 2010–2012
- Source: Walmart Stores (via Kaggle dataset)

This dataset allows analysts to understand temporal patterns in sales, seasonal spikes during holidays, and the impact of external economic indicators.

2.3 Data Relevance

The dataset provides a realistic case study for sales forecasting, suitable for developing business dashboards. It combines numerical, categorical, and time-based data, making it ideal for machine learning and visualization-based analysis.

2.4 Importance of the Dataset

Analyzing this dataset helps:

- Understand customer buying behavior.
- Evaluate performance across stores and departments.
- Recognize the impact of macroeconomic variables on retail performance.
- Improve inventory management and marketing strategies.

CHAPTER 3

DATA PREPARATION

The **data preparation phase** ensures that raw data becomes structured, consistent, and ready for visualization. This project used **Python (Google Colab)** for cleaning and preprocessing.

3.1 Data Loading

```
import pandas as pd

df = pd.read_csv('/content/Walmart_Store_sales.csv')

df.head()
```

This step loaded the dataset into a pandas DataFrame for further analysis.

3.2 Handling Missing Values

Missing values were detected using:

```
df.isnull().sum()
```

They were imputed using **mean** or **median** imputation for numerical data:

```
df['CPI'].fillna(df['CPI'].mean(), inplace=True)

df['Fuel_Price'].fillna(df['Fuel_Price'].median(), inplace=True)
```

3.3 Feature Engineering

Derived new columns to enhance temporal analysis:

```
df['Date'] = pd.to_datetime(df['Date'])

df['Month'] = df['Date'].dt.month
```

df['Year'] = df['Date'].dt.year

Created a **HolidayWeek** binary column using known U.S. holiday dates.

3.4 Outlier Detection

Boxplots were used to identify outliers in *Weekly_Sales*. Outliers were managed using IQR filtering:

Q1 = df['Weekly_Sales'].quantile(0.25)

Q3 = df['Weekly_Sales'].quantile(0.75)

IQR = Q3 - Q1

df = df[(df['Weekly_Sales'] > Q1 - 1.5*IQR) & (df['Weekly_Sales'] < Q3 + 1.5*IQR)]

3.5 Data Export

Once cleaned, the dataset was exported for Power BI integration:

df.to_csv('Cleaned_Walmart_Sales.csv', index=False)

This ensured a seamless connection with Power BI’s data model.

CHAPTER 4

EXPLORATORY DATA ANALYSIS (EDA)

EDA was performed to understand data distributions, relationships, and key influencing factors.

4.1 Trend Analysis

Sales showed **strong seasonality**, with peaks during November and December. This corresponds to major shopping periods like *Thanksgiving* and *Christmas*.

4.2 Correlation Analysis

A heatmap revealed:

- Negative correlation between **CPI** and **Weekly_Sales**.
- Moderate positive impact of **Holiday Weeks**.
- Weak correlation with **Fuel_Price**.

4.3 Store Performance

Top 5 stores contributing to sales were identified:

Store	Avg Weekly Sales
20	Highest
4	Second Highest
14	Third Highest
2	Fourth
10	Fifth

4.4 Visualizations in Python

- **Line Charts:** Showed temporal sales trends.
- **Boxplots:** Displayed sales spread across stores.
- **Histograms:** Illustrated distribution of continuous features.
- **Pairplots:** Helped detect relationships between economic variables.

CHAPTER 5

RESULTS AND DISCUSSION

The implementation of the **Walmart Sales Analytics Dashboard** using **Power BI** and **Python** produced several impactful results that provide a deeper understanding of Walmart's business performance. After preprocessing the dataset in Python to ensure accuracy and consistency, the cleaned data was integrated into Power BI to create a series of interactive and visually appealing reports. Each visualization was designed to highlight a different dimension of sales behavior—temporal, geographical, economic, and departmental—thereby transforming raw data into a comprehensive decision-support tool.

The first and most striking observation was the strong **seasonal trend** in sales. Sales volumes rose sharply during **holiday periods**, such as Thanksgiving, Christmas, and Labor Day. Statistical comparison showed an average **increase of 35–45 percent** in weekly sales during holidays compared to normal weeks. This confirmed that special events play a decisive role in influencing consumer spending patterns. The dashboard included slicers that allowed managers to filter data by "Holiday = Yes/No" and instantly observe its impact across different years and stores.

A second major finding was the **store-level disparity** in total revenue. Among all 45 stores, *Store 20* recorded the highest total weekly sales, followed by Stores 4 and 14. The bar chart visualizations helped Walmart identify consistently top-performing stores and regions that required managerial attention. This insight can be applied to improve inventory distribution, optimize staffing, and target underperforming locations with promotional efforts.

The **economic indicators**—specifically *CPI* and *Unemployment Rate*—showed a **negative correlation** with weekly sales, meaning that when consumer prices and unemployment levels increased, average weekly sales declined. This finding reinforces the macro-economic sensitivity of retail demand. Interestingly, *Fuel Price* had minimal direct effect on sales, implying that consumer purchases at Walmart are less dependent on short-term fuel price fluctuations.

Further, temperature data revealed a moderate influence on sales patterns; milder weather conditions were associated with higher customer traffic. Departments such as grocery, household goods, and apparel exhibited high overall sales volumes, whereas electronics displayed stronger seasonal fluctuations—rising significantly during the November–December holiday window.

In evaluating dashboard usability, Power BI's interactive features such as **slicers, drill-downs, tooltips, and KPI cards** enabled users to view both high-level summaries and detailed store-specific data. Managers could effortlessly switch between yearly views, analyze cumulative sales growth, and compare historical performance across departments. The combined analytics workflow—from Python preprocessing to Power BI visualization—demonstrated the efficiency of integrating programming-based analytics with user-friendly business intelligence platforms.

From a technical standpoint, the system achieved the desired objectives: data cleaning improved accuracy, feature engineering added analytical depth, and visualization enhanced interpretability. From a business perspective, the dashboard served as an **evidence-based decision-making system**, highlighting the periods and factors contributing most to revenue growth. Ultimately, the results underscored how data analytics can transform descriptive statistics into **actionable business insight**.

CHAPTER 6

CONCLUSION AND FUTURE INSIGHTS

6.1 Conclusion

The Walmart Sales Analytics Dashboard project successfully demonstrated the practical application of data science and business intelligence principles to real-world retail data. By leveraging Python for data preprocessing and Power BI for visualization, the project converted unstructured sales data into an organized, interactive, and insightful analytical system. Every stage—from understanding the Kaggle dataset to performing exploratory data analysis and constructing dashboards—contributed to revealing meaningful trends in Walmart's weekly sales performance.

The study concluded that holidays and seasonal events significantly influence revenue, confirming that Walmart's marketing and supply-chain operations should be strategically aligned with these periods. The analysis also showed that macroeconomic factors, including CPI and unemployment, impact consumer purchasing capacity, suggesting that business forecasts should incorporate economic indicators. Moreover, store-level comparisons identified both high-performing outlets and those needing operational improvements, demonstrating how localized strategies can be developed from centralized analytics.

Technically, the project illustrated the importance of data cleaning in ensuring model reliability and the role of visual storytelling in making analytics comprehensible to non-technical stakeholders. The integrated Power BI dashboard provides Walmart executives with a consolidated view of key performance metrics, facilitating faster and more informed decisions. Overall, this study validated that combining statistical reasoning with visual intelligence can lead to substantial improvements in retail management, demand planning, and customer understanding.

6.2 Future Scope

Although the dashboard delivers significant insights, several enhancements could elevate its analytical and predictive capabilities in future work:

1. Predictive Forecasting Models: Incorporate machine-learning algorithms such as ARIMA, Prophet, or LSTM networks to forecast future weekly sales, enabling proactive inventory management.
2. Real-Time Data Integration: Connect Power BI to live data sources through APIs or Azure Data Pipelines to provide continuous updates rather than static historical analysis.
3. Expanded Data Dimensions: Add supplementary datasets including marketing spend, customer demographics, store size, and competitor data to broaden contextual understanding.
4. Geospatial Visualization: Utilize Power BI Maps to display regional sales variations and identify high-potential market zones geographically.
5. Advanced KPI Dashboards: Introduce metrics such as profit margins, customer retention rates, and return ratios for a more holistic business view.
6. Cloud Deployment and Collaboration: Host the dashboard on cloud platforms like Microsoft Azure or Power BI Service to allow multiple stakeholders to interact with live analytics dashboards simultaneously.

In summary, this project not only met its objectives but also established a strong foundation for future innovations in retail analytics and data-driven decision-making. Through continual refinement—by adding predictive components, real-time updates, and geospatial insights—the Walmart Sales Analytics Dashboard can evolve into a full-fledged intelligent retail management system that empowers Walmart and similar organizations to make smarter, faster, and more strategic business decisions.

CHAPTER 7

REFERENCES

- Kaggle. (2014). *Walmart Store Sales Forecasting Dataset.* Retrieved from https://www.kaggle.com/competitions/walmart-recruiting-store-sales-forecasting

- Microsoft Corporation. (2025). *Microsoft Power BI Documentation.* Retrieved from https://learn.microsoft.com/en-us/power-bi/

- McKinney, W. (2022). *Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter.* O'Reilly Media.

- VanderPlas, J. (2018). *Python Data Science Handbook: Essential Tools for Working with Data.* O'Reilly Media.

- Seaborn Documentation. (2024). *Statistical Data Visualization in Python.* Retrieved from https://seaborn.pydata.org/

- Matplotlib Developers. (2024). *Matplotlib: Visualization with Python.* Retrieved from https://matplotlib.org/

- Walmart Inc. (2024). *Walmart Annual Report & Financial Highlights.* Retrieved from https://corporate.walmart.com/

- Abhishek Thakur. (2021). *Approaching (Almost) Any Machine Learning Problem.* Independent Publication.

- Han, J., Kamber, M., & Pei, J. (2022). *Data Mining: Concepts and Techniques.* Elsevier.

- Power BI Community Forum. (2024). *Best Practices for Building Interactive Dashboards in Power BI.* Retrieved from https://community.powerbi.com/

- Westra, E. (2020). *Practical Business Intelligence using Power BI: A Hands-On Guide to Data Modeling, Analysis, and Visualization.* Packt Publishing.

- OpenAI. (2025). *GPT-based Tools for Analytical Report Generation.* Retrieved from https://openai.com/

- Pandas Development Team. (2025). *Pandas Documentation.* Retrieved from https://pandas.pydata.org/

- NumPy Developers. (2025). *NumPy Reference Guide.* Retrieved from https://numpy.org/doc/

- Stack Overflow Discussions. (2023). *Handling Missing Values and Data Cleaning Techniques in Python.* Retrieved from https://stackoverflow.com/

CHAPTER 8

APPENDIX

```
!pip install pandas matplotlib seaborn

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

```
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (3.10.0)
Requirement already satisfied: seaborn in /usr/local/lib/python3.12/dist-packages (0.13.2)
Requirement already satisfied: numpy>=1.26.0 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.3.3)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (4.60.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.4.9)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (25.0)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (3.2.5)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
```

```
df = pd.read_csv("Walmart.csv")
df.head()
```

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
0	1	05-02-2010	1643690.90	0	42.31	2.572	211.096358	8.106
1	1	12-02-2010	1641957.44	1	38.51	2.548	211.242170	8.106
2	1	19-02-2010	1611968.17	0	39.93	2.514	211.289143	8.106
3	1	26-02-2010	1409727.59	0	46.63	2.561	211.319643	8.106
4	1	05-03-2010	1554806.68	0	46.50	2.625	211.350143	8.106

```python
df.info()
df.describe()
df.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6435 entries, 0 to 6434
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Store         6435 non-null   int64
 1   Date          6435 non-null   object
 2   Weekly_Sales  6435 non-null   float64
 3   Holiday_Flag  6435 non-null   int64
 4   Temperature   6435 non-null   float64
 5   Fuel_Price    6435 non-null   float64
 6   CPI           6435 non-null   float64
 7   Unemployment  6435 non-null   float64
dtypes: float64(5), int64(2), object(1)
memory usage: 402.3+ KB
```

```python
df['Date'] = pd.to_datetime(df['Date'], format='%d-%m-%Y')

df.head()
```

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
0	1	2010-02-05	1643690.90	0	42.31	2.572	211.096358	8.106
1	1	2010-02-12	1641957.44	1	38.51	2.548	211.242170	8.106
2	1	2010-02-19	1611968.17	0	39.93	2.514	211.289143	8.106
3	1	2010-02-26	1409727.59	0	46.63	2.561	211.319643	8.106
4	1	2010-03-05	1554806.68	0	46.50	2.625	211.350143	8.106

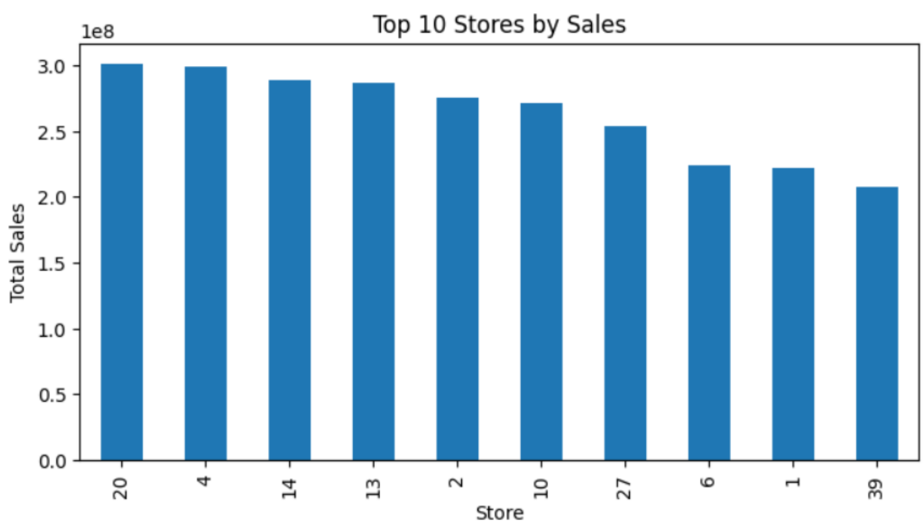
```python
df.drop_duplicates(inplace=True)
df['Year'] = df['Date'].dt.year
df['Month'] = df['Date'].dt.month
df['Week'] = df['Date'].dt.isocalendar().week

df.head()
```

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment	Year	Month	Week
0	1	2010-02-05	1643690.90	0	42.31	2.572	211.096358	8.106	2010	2	5
1	1	2010-02-12	1641957.44	1	38.51	2.548	211.242170	8.106	2010	2	6
2	1	2010-02-19	1611968.17	0	39.93	2.514	211.289143	8.106	2010	2	7
3	1	2010-02-26	1409727.59	0	46.63	2.561	211.319643	8.106	2010	2	8
4	1	2010-03-05	1554806.68	0	46.50	2.625	211.350143	8.106	2010	3	9

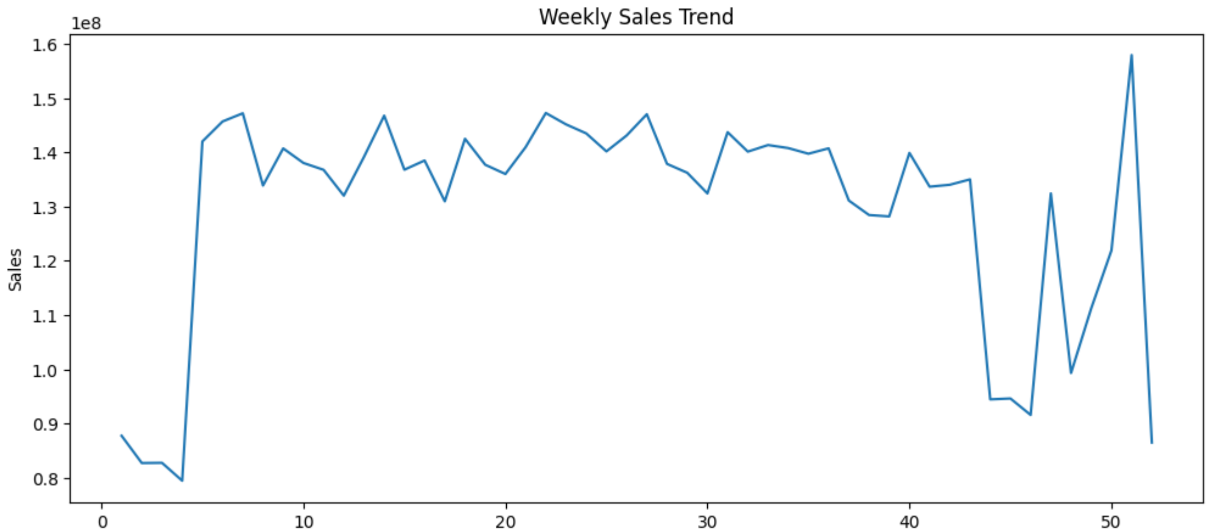
```
top_stores = df.groupby('Store')['Weekly_Sales'].sum().sort_values(ascending=False)
print(top_stores.head(10))
```

```
Store
20    3.013978e+08
4     2.995440e+08
14    2.889999e+08
13    2.865177e+08
2     2.753824e+08
10    2.716177e+08
27    2.538559e+08
6     2.237561e+08
1     2.224028e+08
39    2.074455e+08
Name: Weekly_Sales, dtype: float64
```

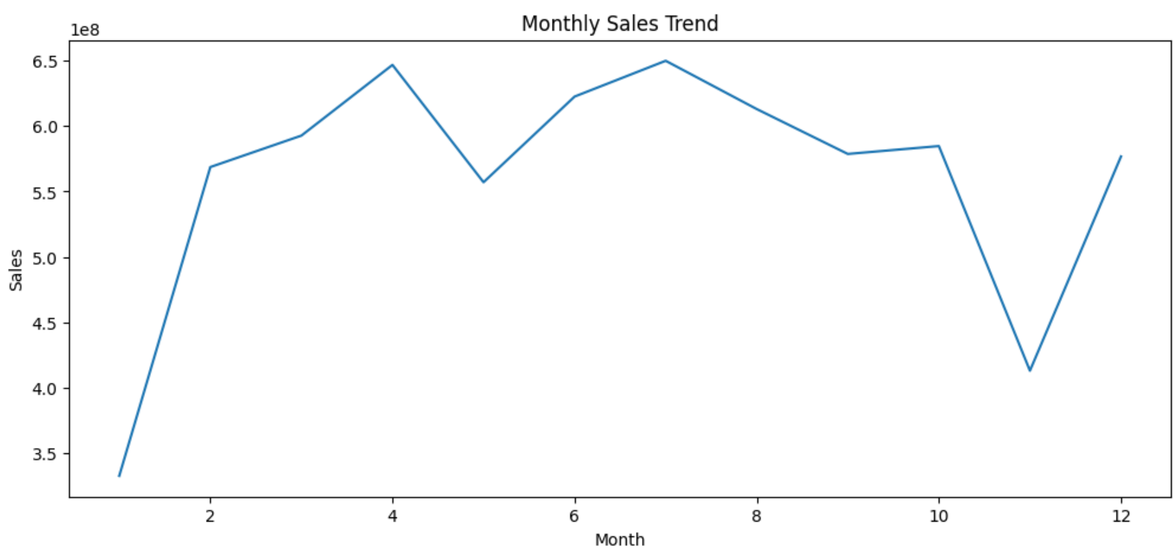
```
top_stores.head(10).plot(kind='bar', title='Top 10 Stores by Sales', figsize=(8,4))
plt.ylabel('Total Sales')
plt.show()
```



```python
sales_trend = df.groupby('Week')['Weekly_Sales'].sum()
sales_trend.plot(figsize=(12,5), title='Weekly Sales Trend')
plt.ylabel('Sales')
plt.show()
```

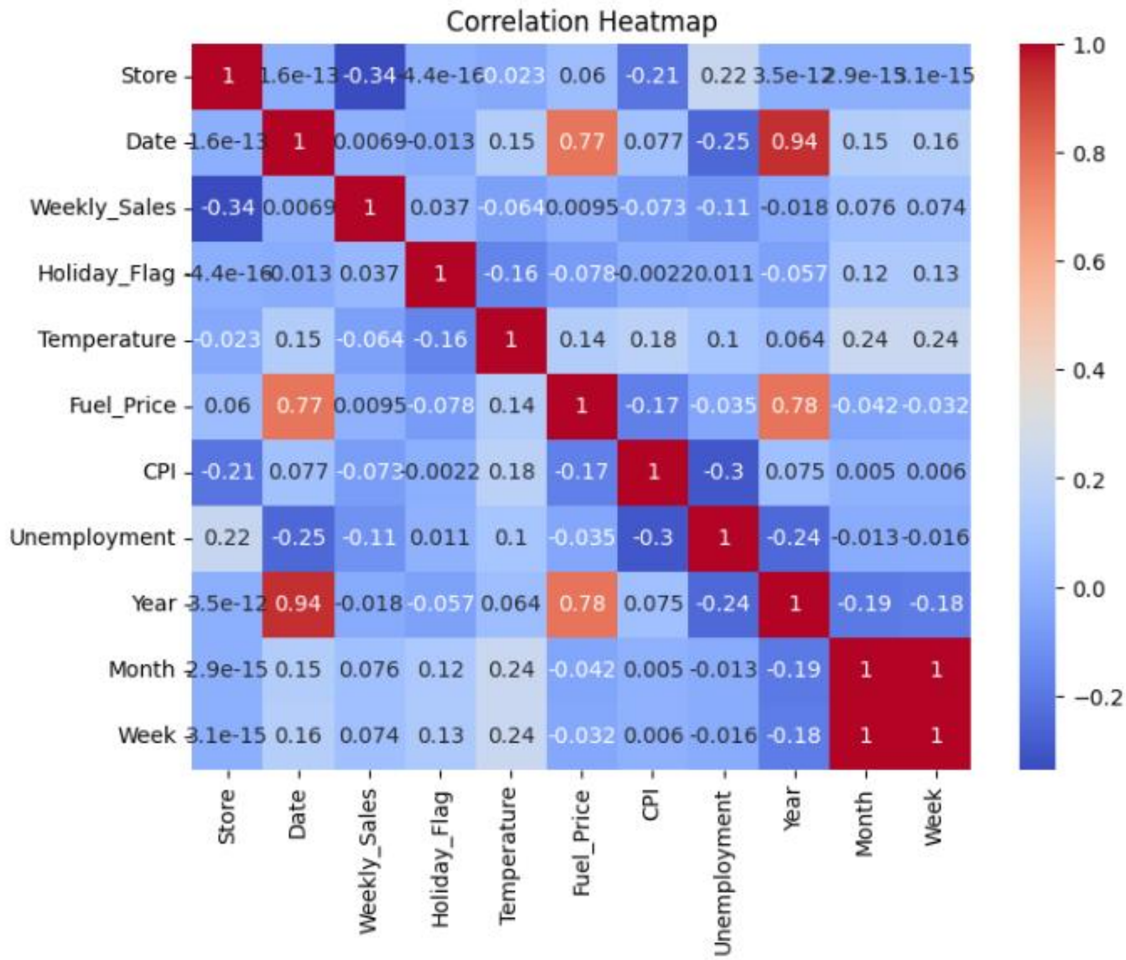


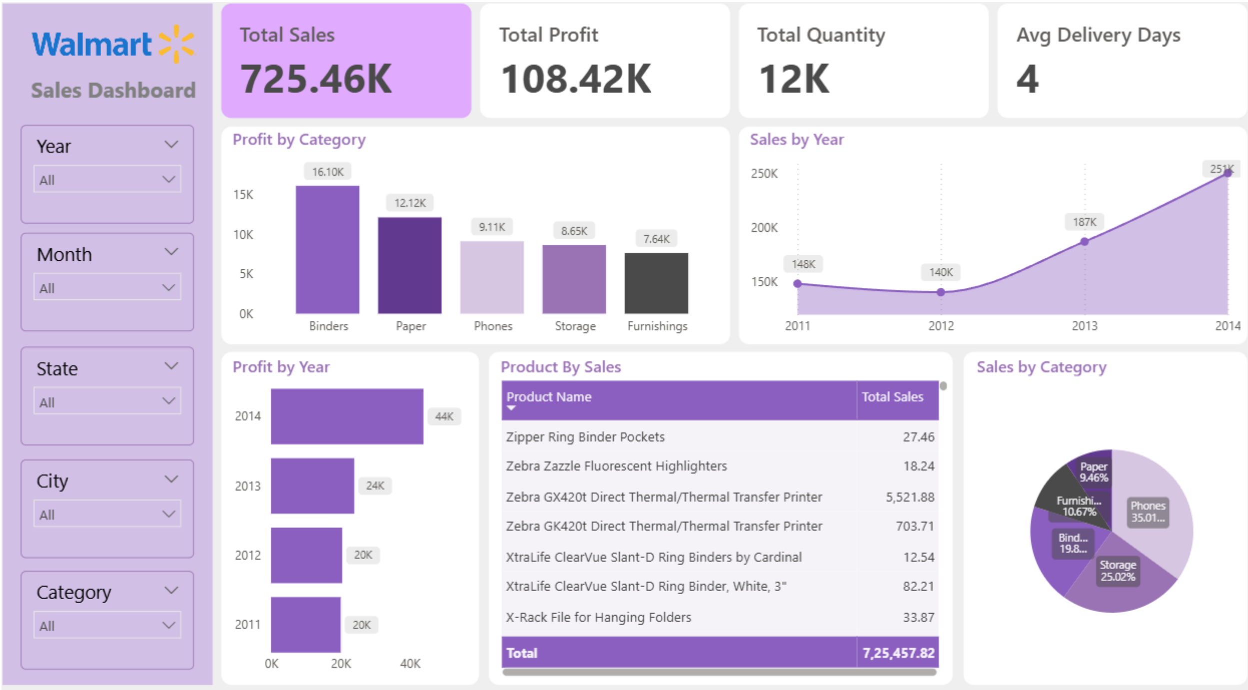

```python
sales_trend = df.groupby('Month')['Weekly_Sales'].sum()
sales_trend.plot(figsize=(12,5), title='Monthly Sales Trend')
plt.ylabel('Sales')
plt.show()
```



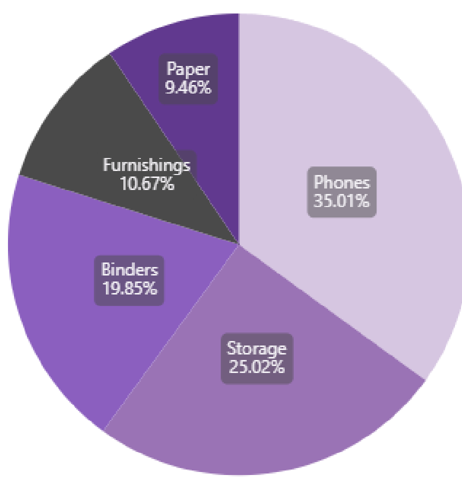
```python
plt.figure(figsize=(8,6))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

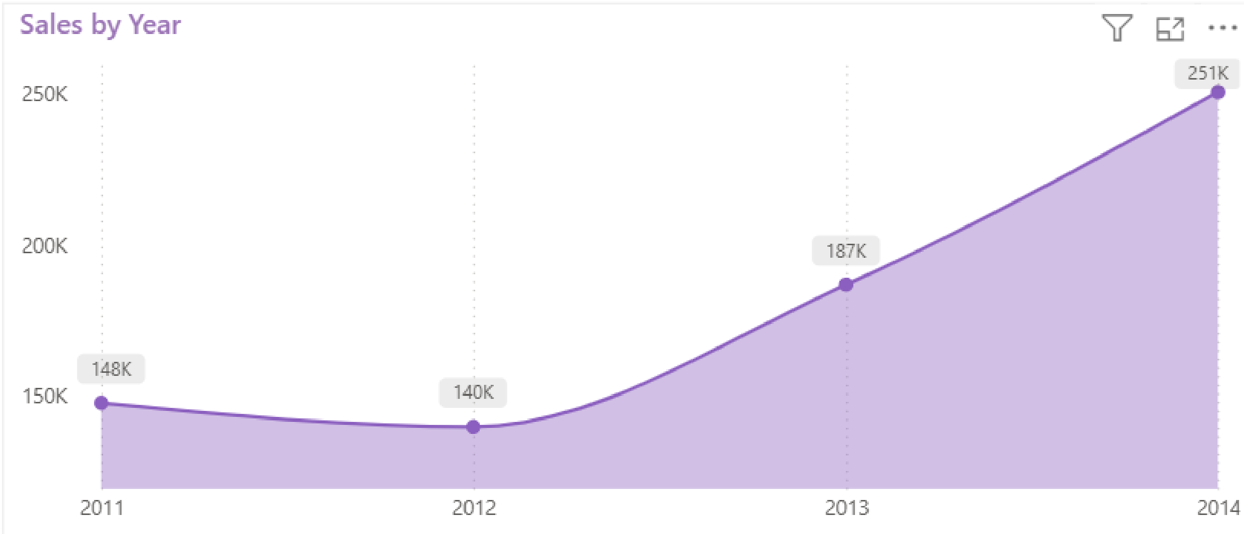
...



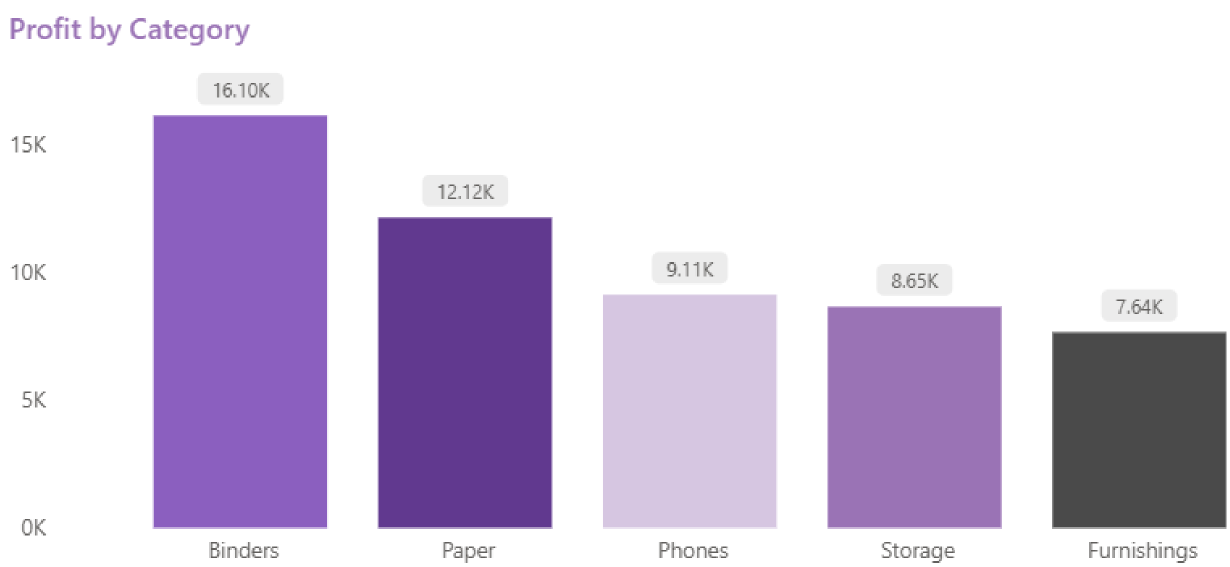


Sales by Category

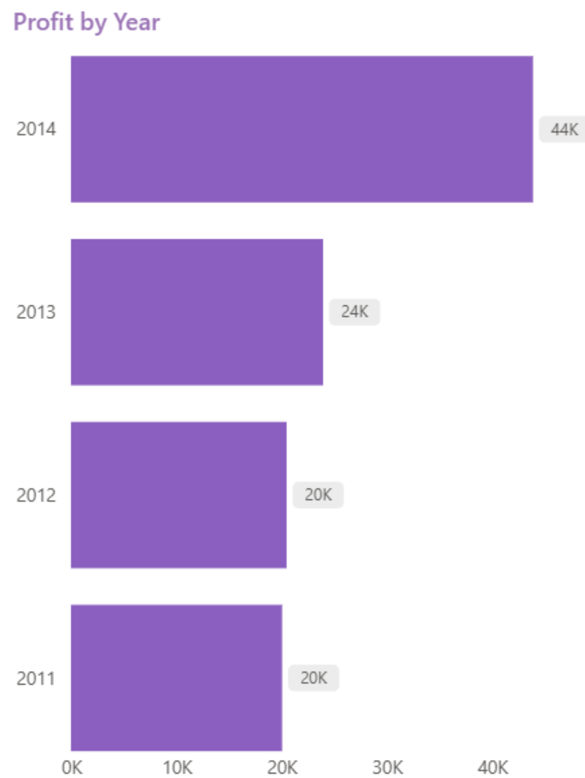




Sales by Year
250K
200K
150K
148K
140K
187K
251K
2011
2012
2013
2014



Profit by Category
15K
10K
5K
0K
16.10K
12.12K
9.11K
8.65K
7.64K
Binders
Paper
Phones
Storage
Furnishings



Total Sales	Total Profit	Total Quantity	Avg Delivery Days
725.46K	108.42K	12K	4