

# **SYNOPSIS**

**Report on**

**Project Title**

## **DATA SALES VISUALIZATION**

**by**

SATYJEET KUMAR 202410116100187

SAURABH KUMAR 202410116100188

**Session:2024-2025 (II Semester)**

Under the supervision of

**KOMAL SALGOTRA**  
**(Assistant Professor)**

**KIET Group of Institutions, Delhi-NCR, Ghaziabad**



**DEPARTMENT OF COMPUTER APPLICATIONS**  
**KIET GROUP OF INSTITUTIONS, DELHI-NCR,**  
**GHAZIABAD-201206**  
**( 2024- 2025)**

# ABSTRACT

In the current competitive business landscape, knowing sales performance is vital in making sound decisions. Sales Data Visualization converts intricate datasets into significant visual information, enabling enterprises to analyze and interpret prominent trends in an effortless manner. This project aims at visualizing the sales data using different methods including pie charts, bar charts, line charts, and heatmaps. Employing Python libraries such as Pandas, Matplotlib, and Seaborn, this project presents an overall perspective of sales performance, allowing companies to comprehend product trends, seasonality, and important correlations among variables such as price, quantity, and total sales.

The key objective is to reveal actionable insights through the visualization of sales data, enabling businesses to maximize inventory, pricing plans, and marketing campaigns. Through the representation of data in an easy-to-understand format, decision-makers can easily detect performance trends, enhance customer interactions, and maximize profitability. The project not only streamlines the interpretation of sales data but also equips businesses with the necessary tools for data-driven decision-making.

Finally, the project will establish a user-friendly front end for depicting sales trends underpinning long-term growth and strategic planning on an informed basis.

# TABLE OF CONTENTS

	Page Number
1. Introduction	4
2. Literature Review	5
3. Project / Research Objective	6
4. Project Flow/ Research Methodology	7-8
5. Sales Data Visualization	9-16
6. Project / Research Outcome	17
7. Result	18
8. Conclusion	19

## INTRODUCTION

In the modern business world, data plays a crucial role in shaping strategies and decisions. One area where data can be especially powerful is in understanding sales performance. Sales Data Visualization is a technique used to translate complex sales data into easy-to-understand visual formats like charts, graphs, and heatmaps. By presenting the data visually, businesses can gain quick insights, identify trends, and make informed decisions that lead to better outcomes.

This project focuses on visualizing sales data to help companies understand sales trends, track performance over time, and identify areas of improvement. Using tools like Python, Pandas, Matplotlib, and Seaborn, this project aims to create clear and meaningful visual representations of sales data. These visuals will include pie charts, bar graphs, line charts, and heatmaps, each designed to highlight different aspects of the sales data, such as product performance, seasonal trends, and correlations between various factors.

## 2. Literature Review

Sales data visualization plays a crucial role in converting complex data into easy-to-understand insights for businesses. According to **Few (2009)**, effective visualization simplifies decision-making by making trends and patterns clearer. Python libraries like **Pandas**, **Matplotlib**, and **Seaborn** are widely used for data manipulation and visualization, enabling businesses to identify sales trends, product performance, and seasonality (Hunter, 2007).

Tools like **Tableau** and **Power BI** are also popular for creating interactive dashboards that allow real-time analysis of sales data. These tools help businesses track performance and adjust strategies dynamically (Chaudhuri et al., 2001).

Visualization types such as **pie charts** and **bar graphs** are ideal for showing sales distribution, while **line charts** are effective for analyzing trends over time (Klein & Sarkar, 2012). **Heatmaps** provide a quick overview of performance across multiple factors (Cleveland & McGill, 1984). However, **Tufte (2001)** emphasizes the importance of selecting the right chart types and ensuring accuracy to avoid misleading interpretations.

In summary, sales data visualization enhances business decision-making by offering clear, actionable insights. Despite challenges in design and data accuracy, when used correctly, these tools help businesses optimize strategies and improve performance.

.

### 3. Project / Research Objective

The main goals of this project are:

- To create clear visualizations that help businesses understand their sales data.
- To use charts and graphs like **pie charts**, **bar charts**, and **line graphs** to display trends, product performance, and other key metrics.
- To highlight the performance of individual products, identify seasonal patterns, and explore relationships between price, quantity sold, and total sales.
- To make the data more accessible and actionable, so that business decisions can be made based on real insights.
- To enable businesses to see their data from different perspectives and make more informed decisions about sales strategies.

## 4. Project Flow / Research Methodology

This project will follow a step-by-step approach to visualize sales data:

### 1. Data Collection:

- A dataset will be used that includes essential sales data, such as product names, quantities sold, unit prices, and transaction dates.
- The data will be in a **CSV** format, which will be loaded using **Python's Pandas** library.

### 2. Data Preparation:

- The data will be cleaned to remove any errors or inconsistencies. Missing values will be handled, and irrelevant columns will be removed.
- Dates will be converted into a **datetime** format to allow easy extraction of year, month, and day for time-based analysis.

### 3. Data Visualization:

- The data will be visualized using various types of charts:
  - **Pie Chart:** To show the distribution of sales across different products.
  - **Bar Chart:** To show monthly sales performance and track trends.

- **Line Graph:** To observe the sales trend over time.
- **Heatmap:** To visualize sales performance by product and month, showing seasonality.
- **Correlation Heatmap:** To understand how different variables like price, quantity, and total sales are related.

#### 4. Tools and Libraries:

- **Pandas:** For handling and preparing the data.
- **Matplotlib** and **Seaborn:** For creating the visualizations.
- **Python** will be the main programming language used for this project, and **Jupyter Notebook** or any Python IDE will be used for execution.






## 5. SALES DATA VISUALIZATION

```
import pandas as pd
from prophet import Prophet
import matplotlib.pyplot as plt
import seaborn as sns

# enable jupyter Notebook to display matplotlib graphs, use plt.show() for non-notebook users
%matplotlib inline

data=pd.read_csv('/content/sales_data.csv')

df = pd.DataFrame(data)
data
```

	Date	Product	UnitsSold	Revenue	
0	2024-01-01	Phone	26	43188	
1	2024-01-02	Phone	85	32755	
2	2024-01-03	Laptop	11	5579	
3	2024-01-04	Monitor	61	28188	
4	2024-01-05	Monitor	64	15223	
5	2024-01-06	Laptop	18	15893	
6	2024-01-07	Phone	92	34084	
7	2024-01-08	Tablet	84	32583	
8	2024-01-09	Monitor	57	40668	
9	2024-01-10	Phone	74	39400	
10	2024-01-11	Laptop	79	23234	
11	2024-01-12	Phone	12	17356	
12	2024-01-13	Monitor	51	34242	
13	2024-01-14	Monitor	73	7582	
14	2024-01-15	Monitor	72	39434	
15	2024-01-16	Tablet	69	32656	
16	2024-01-17	Phone	38	44135	
17	2024-01-18	Tablet	44	28193	
18	2024-01-19	Laptop	86	12652	
19	2024-01-20	Tablet	69	31307	

```
data.info()
data.describe()
data['Revenue'].describe()
sample_for_plotting=data[['Product','Revenue' ]]
sample_for_plotting.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Date        20 non-null    object
1   Product     20 non-null    object
2   UnitsSold   20 non-null    int64
3   Revenue     20 non-null    int64
dtypes: int64(2), object(2)
memory usage: 772.0+ bytes
```

	Product	Revenue
0	Phone	43188
1	Phone	32755
2	Laptop	5579
3	Monitor	28188
4	Monitor	15223



```
[50] plotting_group=sample_for_plotting.groupby('Product')  
      plotting_group.size().head()
```



0

Product

Laptop 4

Monitor 6

Phone 6

Tablet 4

dtype: int64



```
# Total sale by Product Type  
total_sales=plotting_group.sum()  
total_sales.head()
```



Revenue



Product

Laptop 57358

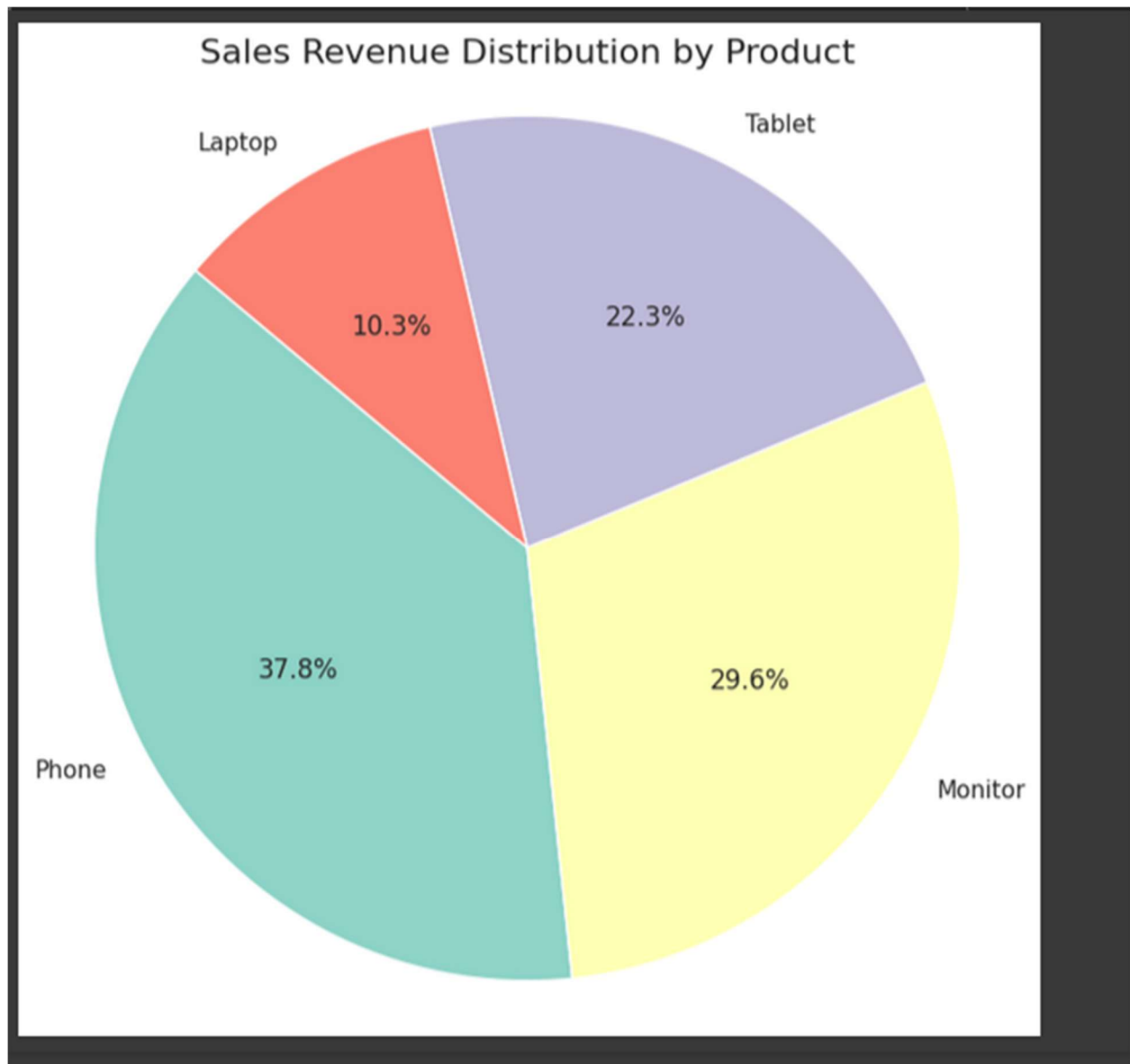
Monitor 165337

Phone 210918

Tablet 124739



```
# Plotting the pie chart
plt.figure(figsize=(8, 8))
plt.pie(product_revenue, labels=product_revenue.index, autopct='%1.1f%%', startangle=140,
        colors=sns.color_palette('Set3', len(product_revenue)))
plt.title('Sales Revenue Distribution by Product', fontsize=16)
plt.axis('equal') # Equal aspect ratio ensures the pie chart is drawn as a circle
plt.show()
```



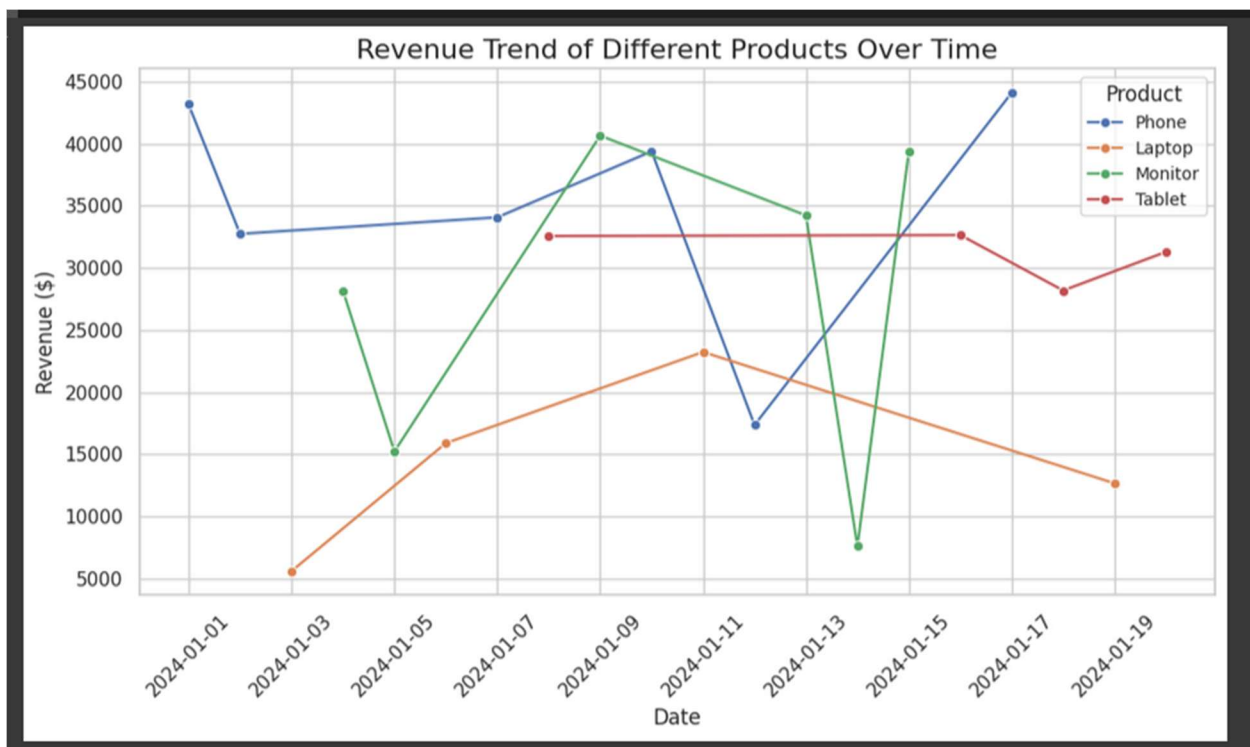
```

# Convert Date column to datetime type
df['Date'] = pd.to_datetime(df['Date'])

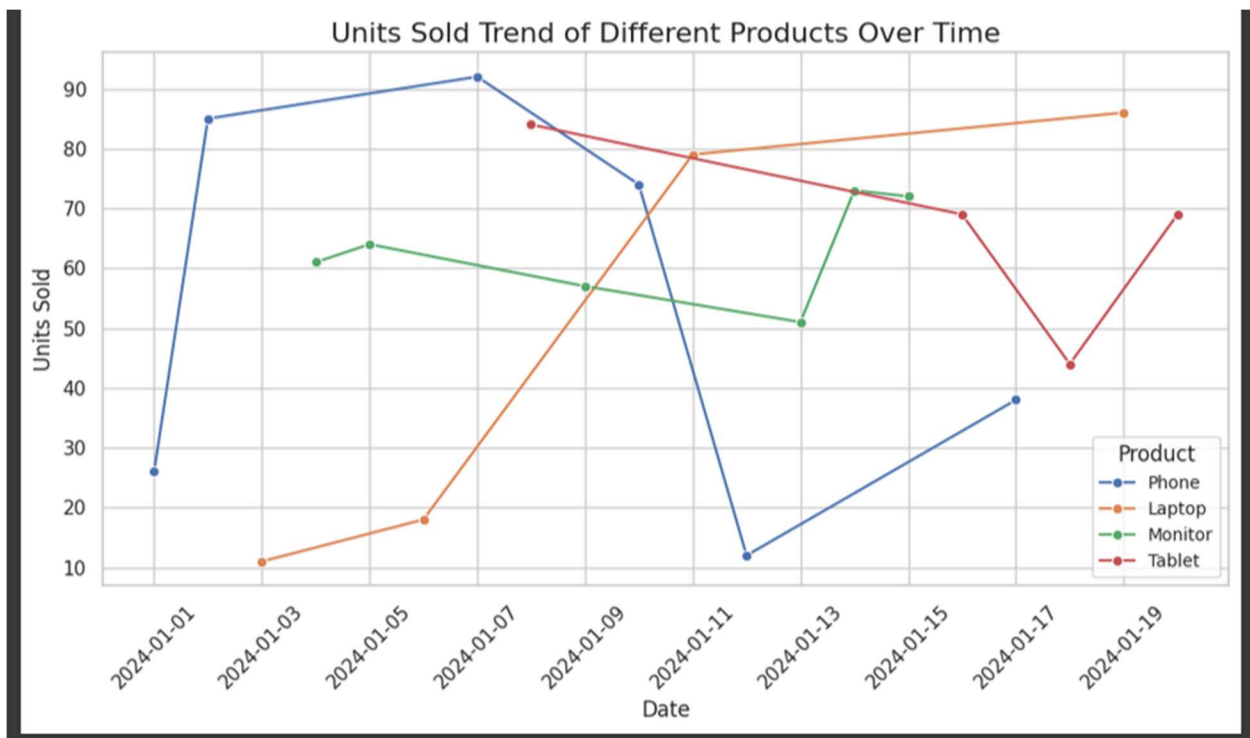
# Set the style of seaborn for better visualization
sns.set(style="whitegrid")

# Plot the sales trends of different products over time
plt.figure(figsize=(10, 6))
sns.lineplot(data=df, x='Date', y='Revenue', hue='Product', marker='o')
plt.title('Revenue Trend of Different Products Over Time', fontsize=16)
plt.xlabel('Date', fontsize=12)
plt.ylabel('Revenue ($)', fontsize=12)
plt.xticks(rotation=45)
plt.legend(title='Product', fontsize=10)
plt.tight_layout()
plt.show()

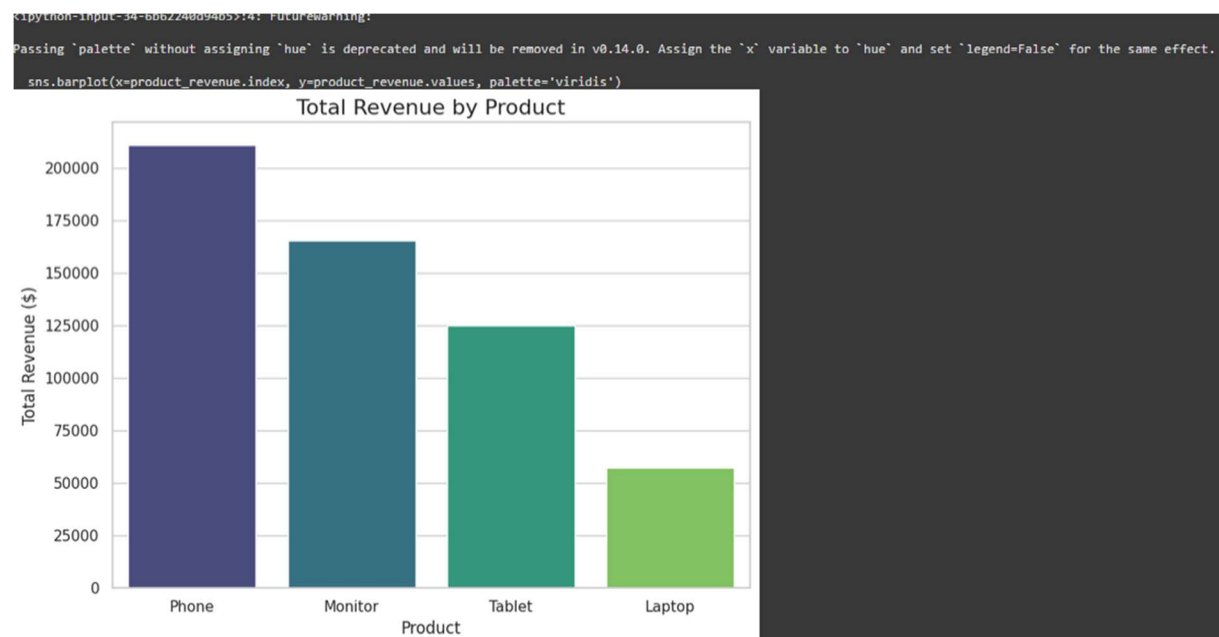
```



```
# Plot Units Sold for each product over time
plt.figure(figsize=(10, 6))
sns.lineplot(data=df, x='Date', y='UnitsSold', hue='Product', marker='o')
plt.title('Units Sold Trend of Different Products Over Time', fontsize=16)
plt.xlabel('Date', fontsize=12)
plt.ylabel('Units Sold', fontsize=12)
plt.xticks(rotation=45)
plt.legend(title='Product', fontsize=10)
plt.tight_layout()
plt.show()
```

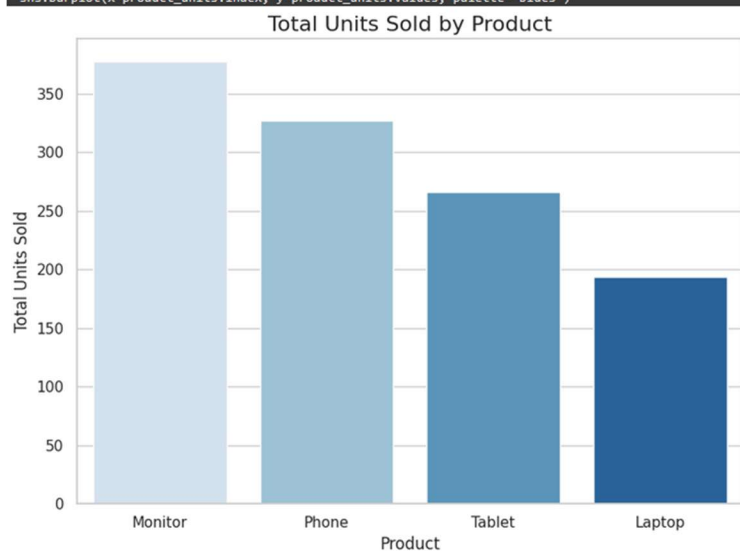


```
# Bar plot for total revenue by product
plt.figure(figsize=(8, 6))
product_revenue = df.groupby('Product')['Revenue'].sum().sort_values(ascending=False)
sns.barplot(x=product_revenue.index, y=product_revenue.values, palette='viridis')
plt.title('Total Revenue by Product', fontsize=16)
plt.xlabel('Product', fontsize=12)
plt.ylabel('Total Revenue ($)', fontsize=12)
plt.tight_layout()
plt.show()
```



```
# Bar plot for total units sold by product
plt.figure(figsize=(8, 6))
product_units = df.groupby('Product')['UnitsSold'].sum().sort_values(ascending=False)
sns.barplot(x=product_units.index, y=product_units.values, palette='Blues')
plt.title('Total Units Sold by Product', fontsize=16)
plt.xlabel('Product', fontsize=12)
plt.ylabel('Total Units Sold', fontsize=12)
plt.tight_layout()
plt.show()
```

```
<ipython-input-35-3d9726962492>:4: FutureWarning:
Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'x' variable to 'hue' and set 'legend=False' for the same effect.
sns.barplot(x=product_units.index, y=product_units.values, palette='Blues')
```





## 6. Project / Research Outcome

By the end of this project, the following outcomes are expected:

- **Multiple Visualizations:** Including pie charts, bar charts, line graphs, and heatmaps that make the data more understandable.
- **Business Insights:**
  - Identifying the **top-selling products** and their contribution to total sales.
  - Understanding **monthly sales trends** to pinpoint peak sales periods.
  - Analyzing how **price and quantity sold** affect total sales.
  - Observing **seasonal patterns** and identifying months with higher or lower sales.
- **Data-Driven Decision Making:** The visualizations will help businesses make more informed decisions about inventory, pricing strategies, and marketing campaigns.

## 7. Results

The results of this project will present actionable insights from the sales data, such as:

- **Sales Distribution:** A pie chart that shows which products are the most popular and which contribute the most to sales.
- **Monthly Trends:** A bar chart that highlights which months had the highest or lowest sales.
- **Sales Performance Over Time:** A line graph showing how sales changed from month to month or day to day.
- **Product-Performance Heatmap:** A heatmap that helps businesses understand how different products performed each month.
- **Correlation Insights:** A correlation heatmap that shows how price, quantity, and sales are interconnected.

## 8. Conclusion

This project demonstrates how **data visualization** can turn raw sales data into valuable business insights. By using graphs and charts to represent data, it becomes much easier to spot trends, monitor performance, and make decisions. **Sales Data Visualization** not only helps track the past performance of products but also provides a roadmap for future strategies.

Through this project, businesses will gain a deeper understanding of their sales patterns and will be able to make better-informed decisions that drive growth and improve profitability. The visualizations will serve as a powerful tool for both **sales managers** and **executives**, helping them understand the story behind the numbers and take proactive actions.