



TITLE

Name – Gaurav Kaushal

Branch - CSE(AIML)

Roll no. – 202401100400089

Instructor's name – Mr. Abhishek Shukla

INTRODUCTION

Problem statement:

Analyzing sales data is crucial for understanding how a business performs and for making informed decisions. In this project, we will use Python to visualize revenue, product demand, and seasonal sales trends. Our goal is to uncover insights into variations in sales and to identify patterns that can help optimize inventory and marketing strategies.

2. Methodology

To analyze and visualize the sales data, we follow these steps:

1. **Data Collection:** Load sales data from a CSV file containing details like date, product category, quantity sold, and revenue.
2. **Data Cleaning:** Handle missing values and format the date column for time-based analysis.
3. **Exploratory Data Analysis (EDA):** Compute summary statistics and identify key trends.
4. **Visualization:**
 - Revenue trends over time
 - Seasonal variations (monthly/quarterly sales)
 - Product-wise demand analysis
5. **Insights & Conclusion:** Interpret the findings for business applications.

CODE:

Solution Approach:

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Sample Sales Data (Replace with actual dataset)

data = {

    'Date': pd.date_range(start='2023-01-01', periods=12, freq='M',

    'Product': ['A', 'B', 'C', 'A', 'B', 'C', 'A', 'B', 'C', 'A', 'B', 'C'],

    'Revenue': [1200, 1500, 1800, 1300, 1700, 1900, 1600, 2000, 2200,

2100, 2500, 2700

    'Units_Sold': [100, 120, 140, 110, 130, 150, 120, 140, 160, 150,

180, 200]

}

# Create a DataFrame from the dictionary

df = pd.DataFrame(data)

# Convert the 'Date' column to datetime format for proper time-
based analysis

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

Plot Revenue Over Time

```
plt.figure(figsize=(10, 5)) # Set figure size (width=10, height=5 inches)
```

```
sns.lineplot(x='Date', y='Revenue', data=df, marker='o', label='Total  
Revenue') # Line plot with markers for revenue trend
```

```
plt.xlabel('Date') # Set x-axis label
```

```
plt.ylabel('Revenue') # Set y-axis label
```

```
plt.title('Revenue Trend Over Time') # Set plot title
```

```
plt.xticks(rotation=45) # Rotate x-axis labels by 45 degrees for better  
readability
```

```
plt.legend() # Display legend
```

```
plt.show() # Show the plot
```

Product Demand Analysis (Bar Chart)

```
plt.figure(figsize=(8, 5)) # Set figure size
```

```
sns.barplot(x='Product', y='Units_Sold', data=df, estimator=sum,  
ci=None, palette='viridis')
```

```
plt.xlabel('Product') # Set x-axis label
```

```
plt.ylabel('Total Units Sold') # Set y-axis label
```

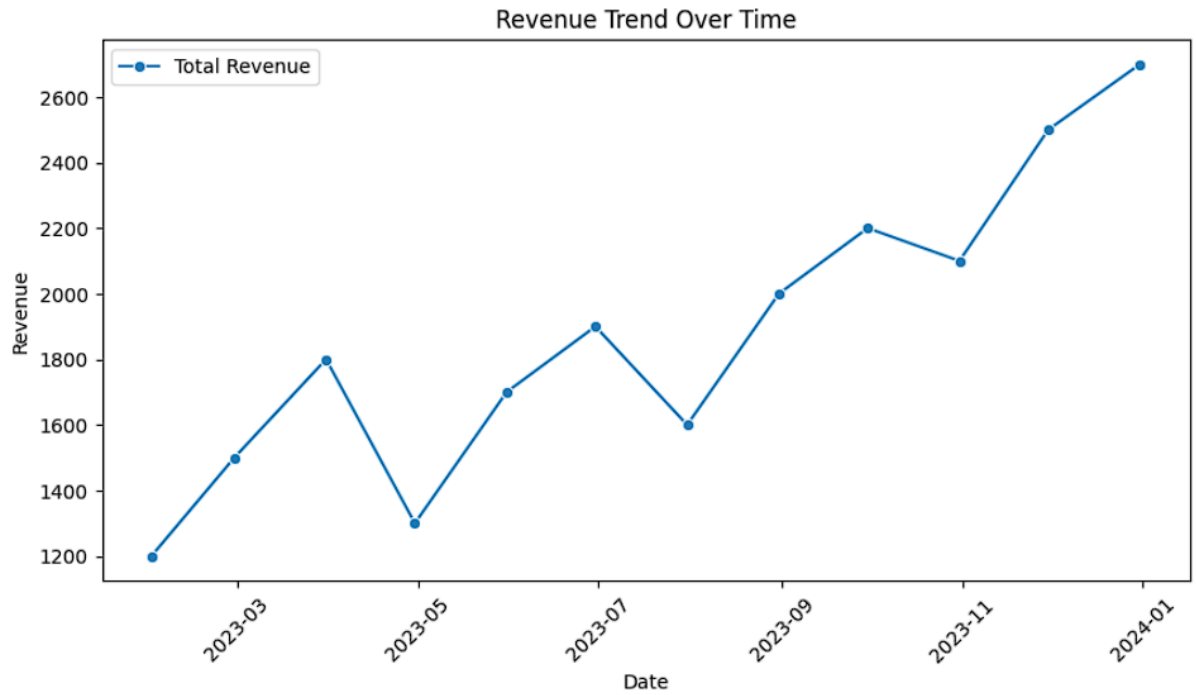
```
plt.title('Product Demand') # Set plot title
```

```
plt.show() # Show the plot
```

Seasonal Sales Trend (Month-wise Revenue)

```
df['Month'] = df['Date'].dt.strftime('%b')  
plt.figure(figsize=(10, 5)) # Set figure size  
sns.boxplot(x='Month', y='Revenue', data=df, palette='coolwarm')  
plt.xlabel('Month') # Set x-axis label  
plt.ylabel('Revenue') # Set y-axis label  
plt.title('Seasonal Sales Trend') # Set plot title  
plt.show() # Show the plot
```

OUTPUT



```
<ipython-input-1-2017cbc76fbc>:30: FutureWarning:
```

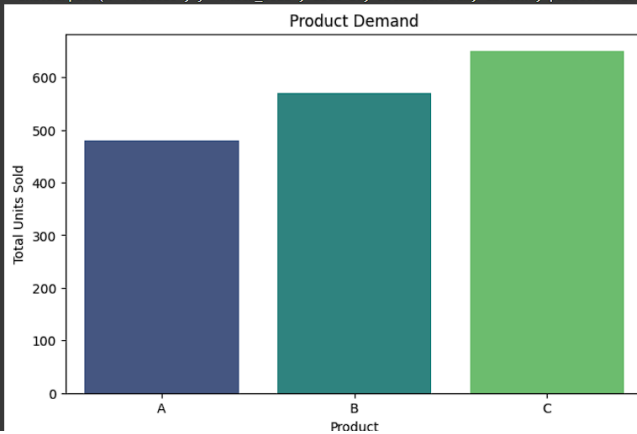
The 'ci' parameter is deprecated. Use 'errorbar=None' for the same effect.

```
sns.barplot(x='Product', y='Units_Sold', data=df, estimator=sum, ci=None, palette='viridis')
```

```
<ipython-input-1-2017cbc76fbc>:30: 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.

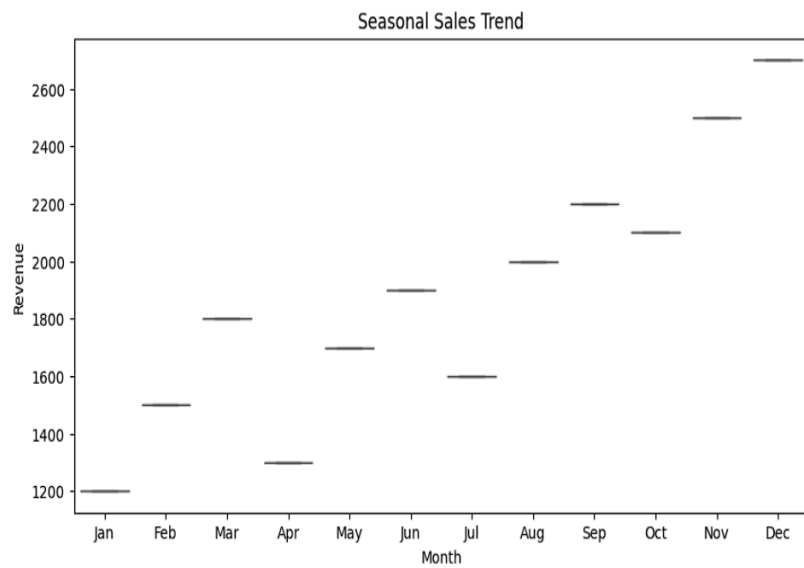
```
sns.barplot(x='Product', y='Units_Sold', data=df, estimator=sum, ci=None, palette='viridis')
```



```
<ipython-input-1-2017cbc76fbc>:39: 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.boxplot(x='Month', y='Revenue', data=df, palette='coolwarm')
```



REFERENCE:

- Chat Gpt 4.o
- Google COllab
- Libraries used: pandas, matplotlib, seaborn