Sales Data Visualization & Analysis

Author: UTKARSH KUMAR SINGH

UNIVERSITY ROLL NO: 202401100400202

Introduction

Objective:

The goal of this project is to analyze sales data, identify revenue trends, understand product demand, and observe seasonal sales patterns. The insights derived will help in better decision-making regarding inventory, marketing strategies, and revenue forecasting.

Key Questions Answered:

- How does revenue fluctuate over months and years?
- Which products are in high demand?
- How does sales revenue vary across different seasons?

Methodology

Dataset Used:

- The dataset contains 100 rows of dummy sales data, including columns:
  - Date: The date of the sale.
  - Revenue: The revenue generated on that date.
  - Product: The product sold.
  - Quantity: Number of units sold.

Steps Followed:

1. Data Preprocessing:
   - Loaded the dataset into a Pandas DataFrame.
   - Converted the Date column to datetime format.
   - Extracted Year and Month for trend analysis.

2. **Revenue Trend Analysis:**
   - **Aggregated monthly revenue.**
   - Visualized revenue trends over time using a line chart.
3. Product Demand Analysis:
   - Grouped sales data by Product and calculated the total quantity sold.
   - Displayed a bar chart showing product demand.
4. Seasonal Sales Analysis:
   - Mapped months to seasons (Winter, Spring, Summer, Fall).
   - Aggregated revenue per season and plotted a bar chart.

Code Typed

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

# Load sales data (replace 'sales_data.csv' with your actual file)
df = pd.read_csv("sales_data.csv")

# Ensure date column is in datetime format
df["Date"] = pd.to_datetime(df["Date"])

# Extract year and month for trend analysis
df["Year"] = df["Date"].dt.year
df["Month"] = df["Date"].dt.month

# Aggregate revenue by month
monthly_revenue = df.groupby(["Year",
"Month"])["Revenue"].sum().reset_index()
```

```
# Plot monthly revenue trends
plt.figure(figsize=(10, 5))
sns.lineplot(data=monthly_revenue, x="Month", y="Revenue", hue="Year", marker="o")
plt.title("Monthly Revenue Trends")
plt.xlabel("Month")
plt.ylabel("Revenue")
plt.grid()
plt.show()

# Aggregate product demand
product_demand = df.groupby("Product")["Quantity"].sum().reset_index()

# Plot product demand
plt.figure(figsize=(10, 5))
sns.barplot(data=product_demand, x="Product", y="Quantity", palette="viridis")
plt.xticks(rotation=45)
plt.title("Product Demand")
plt.xlabel("Product")
plt.ylabel("Quantity Sold")
plt.show()
```

```
# Seasonal sales trend (assuming a 'Season' column exists or inferred from months)
df["Season"] = df["Month"].map({12: "Winter", 1: "Winter", 2: "Winter",
                              3: "Spring", 4: "Spring", 5: "Spring",
                              6: "Summer", 7: "Summer", 8: "Summer",
                              9: "Fall", 10: "Fall", 11: "Fall"})

seasonal_sales = df.groupby("Season")["Revenue"].sum().reset_index()

# Plot seasonal sales trends
plt.figure(figsize=(8, 5))
sns.barplot(data=seasonal_sales, x="Season", y="Revenue", palette="coolwarm")
plt.title("Seasonal Sales Trends")
plt.xlabel("Season")
plt.ylabel("Revenue")
plt.show()
```

OUTPUT

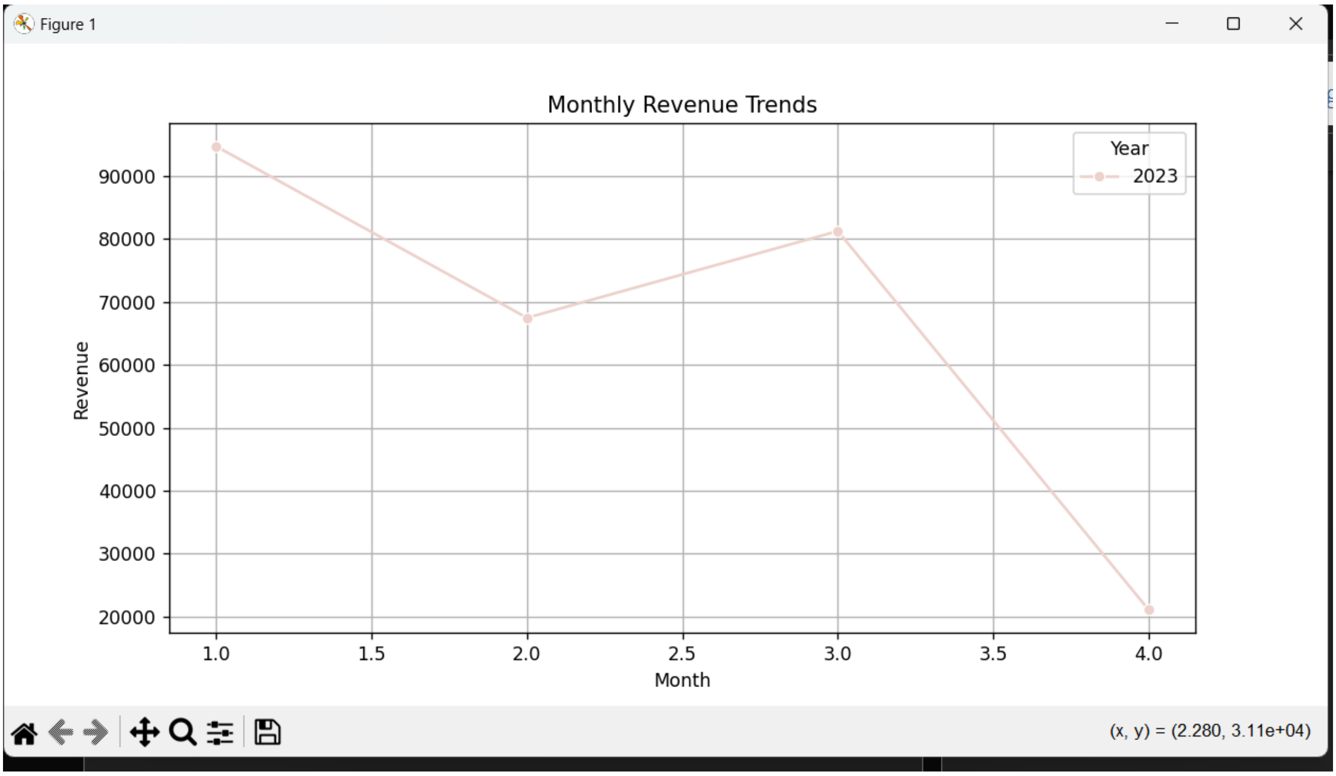
Figure 1
Monthly Revenue Trends
Revenue
90000
80000
70000
60000
50000
40000
30000
20000
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Month
Year
2023
(x, y) = (2.280, 3.11e+04)