Title:
Simple Sales Data Visualization – Analyzing Revenue, Product Demand, and Seasonal Sales Trends.

Introduction

In this report, we analyze and visualize sales data for a company over the course of a year. The data contains information about products sold, the number of units sold, and the price per unit for each day in the year. Using this data, we aim to:

- Analyze **revenue trends** over time.
- Track **product demand** on a daily basis.
- Identify **seasonal sales trends** to understand how revenue fluctuates throughout the months.

By visualizing this data, we can uncover insights into sales performance, product popularity, and seasonal patterns, which can help in better decision-making for business strategies

Methodology

Data Collection

We have a dataset containing the following columns:

- **Date**: The date of the sale (365 days).
- **Product**: Different products sold during the year (Product A, Product B, Product C).
- **Units Sold**: The quantity of each product sold each day.
- **Price per Unit**: The selling price for each product.

Data Processing

We use Python, specifically the pandas library, to process and structure the data. For each product, we calculate the revenue by multiplying the number of units sold by the price per unit.

Data Visualization

To visualize the data, we use matplotlib and seaborn:

- **Revenue Over Time**: A line plot showing the total revenue generated each day.
- **Product Demand Over Time**: A line plot showing the demand (units sold) for each product over time.
- **Seasonal Sales Trends**: A bar chart to show how revenue changes throughout the months of the year.

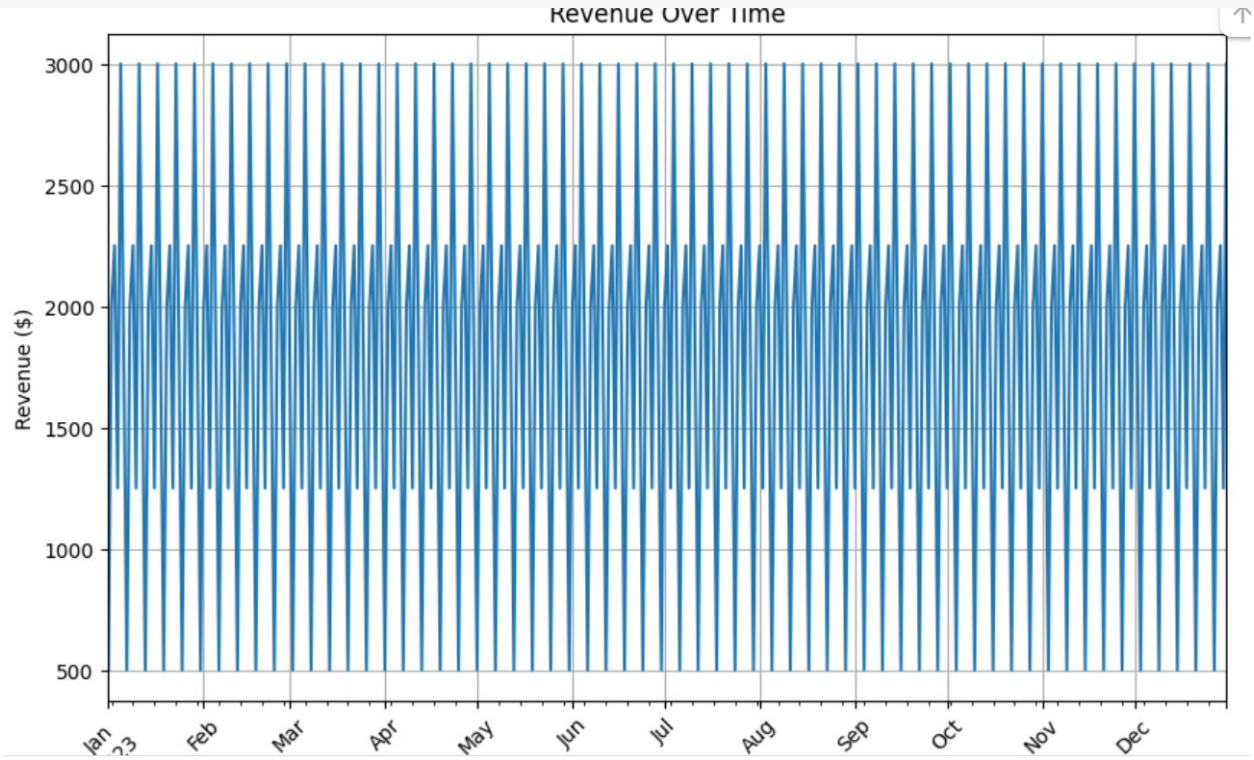
Code

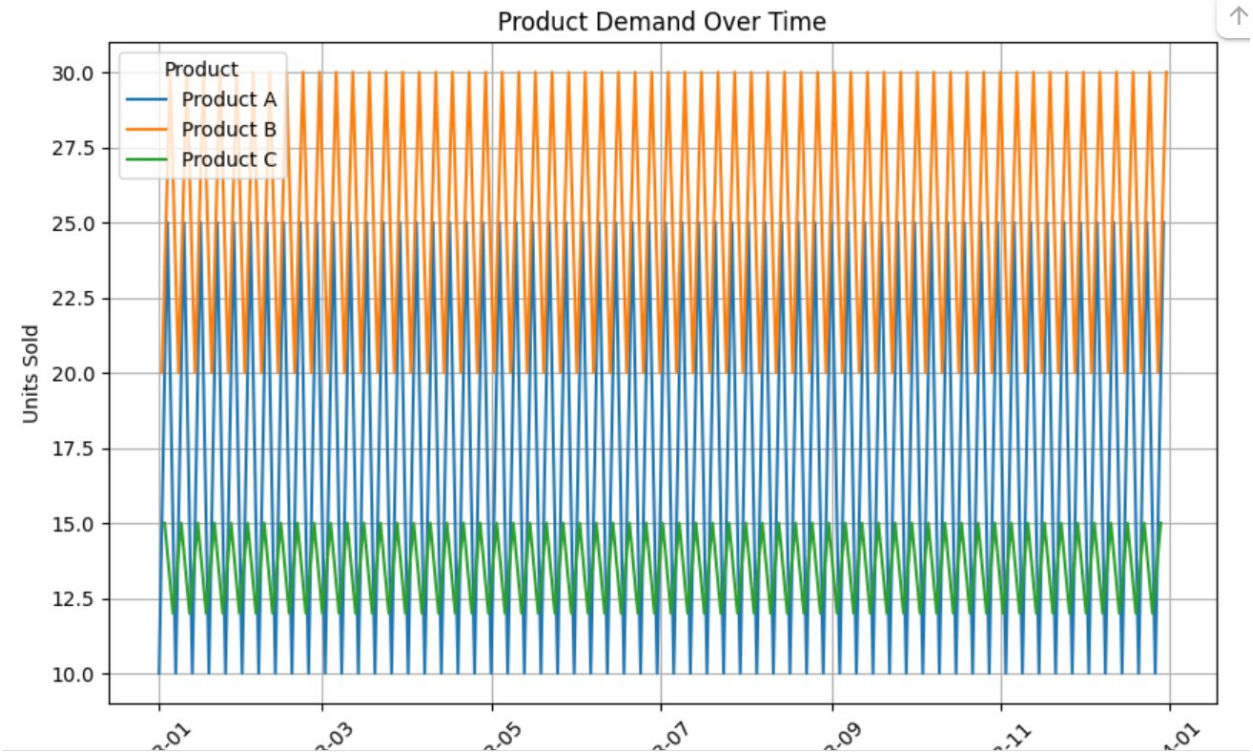
```
# Step 1: Import Libraries
import pandas as pd  # For data handling and analysis
import matplotlib.pyplot as plt  # For creating plots and visualizations
import seaborn as sns  # For creating attractive statistical plots

# Step 2: Create Sales Data (365 days)
num_days = 365  # Total number of days in the dataset (1 year)
# Define the repeating lists for products, units sold, and price per unit
products = ['Product A', 'Product B', 'Product C']
units_sold = [10, 20, 15, 25, 30, 12]
prices_per_unit = [50, 100, 150]

# Function to repeat the lists and ensure they match the length of num_days (365 days)
repeat_data = lambda lst: (lst * (num_days // len(lst))) + lst[:(num_days % len(lst))]

# Prepare the data by repeating the lists to match num_days
data = {
   'Date': pd.date_range(start='2023-01-01', periods=num_days, freq='D'),  # Generate dates for one year (365 days)
   'Product': repeat_data(products),  # Repeat product names to match num_days
   'Units Sold': repeat_data(units_sold),  # Repeat units sold values to match num_days
```

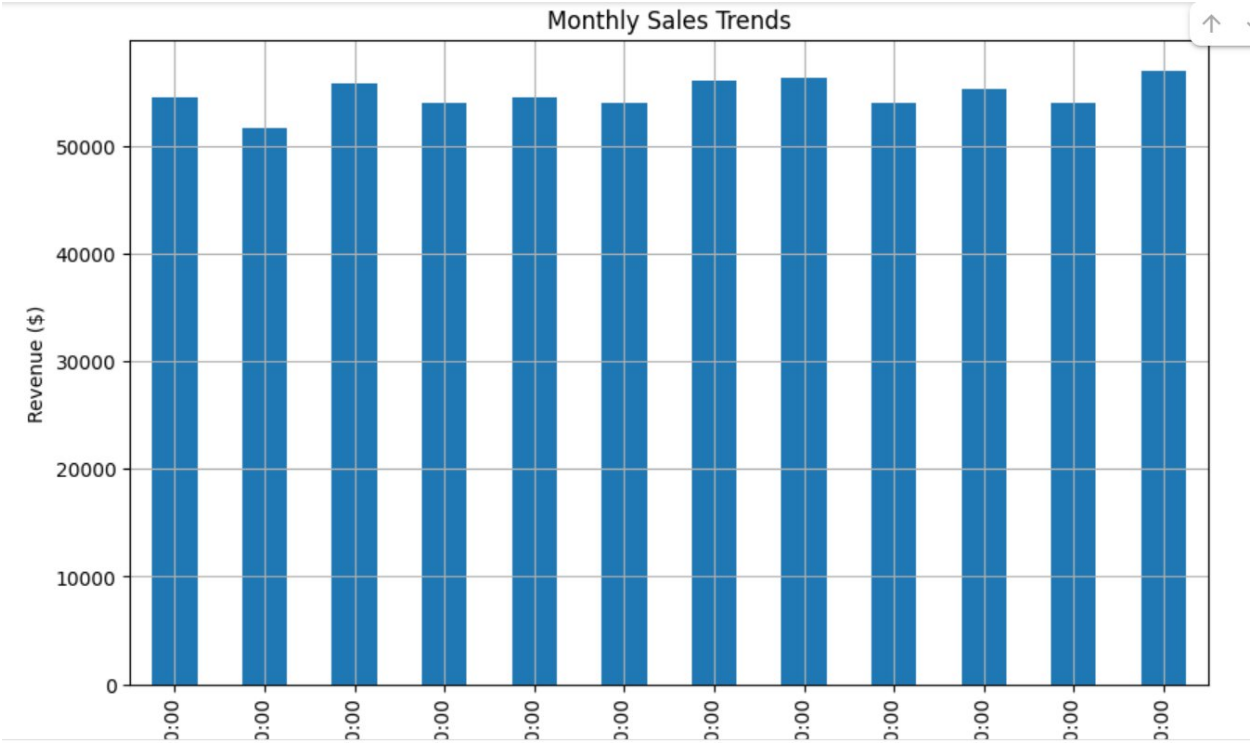
```
    'Price Per Unit': repeat_data(prices_per_unit)  # Repeat price per unit values to match num_days
}

# Step 3: Create DataFrame
df = pd.DataFrame(data)  # Convert the data dictionary into a pandas DataFrame

# Step 4: Calculate Revenue
df['Revenue'] = df['Units Sold'] * df['Price Per Unit']  # Calculate revenue by multiplying units sold with price per unit

# Step 5: Plot Revenue Over Time
plt.figure(figsize=(10, 6))  # Set the figure size for the plot
df.groupby('Date')['Revenue'].sum().plot(  # Group by date and calculate total revenue
    title='Revenue Over Time',  # Title of the plot
    xlabel='Date',  # Label for the X-axis (Date)
    ylabel='Revenue ($)'  # Label for the Y-axis (Revenue)
)
plt.xticks(rotation=45)  # Rotate the X-axis labels (dates) for better visibility
plt.grid(True)  # Display gridlines for easier readability
plt.show()  # Display the plot

# Step 6: Plot Product Demand Over Time
plt.figure(figsize=(10, 6))  # Set the figure size for the plot
sns.lineplot(x='Date', y='Units Sold', hue='Product', data=df)  # Line plot to show product demand over time
plt.title('Product Demand Over Time')  # Title for the plot
plt.xticks(rotation=45)  # Rotate the X-axis labels for better visibility
plt.grid(True)  # Add gridlines to the plot
plt.show()  # Display the plot

# Step 7: Plot Monthly Sales Trends
df.groupby('Date')['Revenue'].sum().resample('M').sum().plot(  # Resample data by month and sum revenue for each month
    kind='bar',  # Use bar chart for monthly trends
    figsize=(10, 6),  # Set the figure size for the plot
    title='Monthly Sales Trends'  # Title of the plot
)
```

```
plt.xlabel('Month')  # Label for the X-axis (Month)
plt.ylabel('Revenue ($)')  # Label for the Y-axis (Revenue)
plt.grid(True)  # Add gridlines to the plot for better readability
plt.show()  # Display the plot
```

Output




Product Demand Over Time
Product
Product A
Product B
Product C
Units Sold
30.0
27.5
25.0
22.5
20.0
17.5
15.0
12.5
10.0


Monthly Sales Trends
Revenue ($)
50000
40000
30000
20000
10000
0