EXERCISE 1:

plt.xticks(rotation=45)

plt.subplot(1,2,1)

plt.plot(df.head(30)['ORDERDATE'], df.head(30)['SALES'], 'r--')

plt.subplot(1,2,2)

plt.plot(df.head(30)['ORDERDATE'], df.head(30)['SALES'], 'g\*-');

EXERCISE 2

plt.figure(figsize=(10, 6))

plt.scatter(df[PRICEEACH], df['SALES'])

plt.xlabel('Quantity Ordered')

plt.ylabel('Sales')

plt.title('Quantity Ordered vs Sales')

plt.grid(True) # Add grid lines

plt.show()

fig, ax = plt.subplots(figsize=(12, 8))

# Scatter plot colored by 'YEAR\_ID'

sc = ax.scatter(df['QUANTITYORDERED'], df['SALES'], label='Sales by Year')

# Adding colorbar

cbar = plt.colorbar(sc)

cbar.set\_label('Year', rotation=90)

# Marking specific data points

motorcycles = df[df['PRODUCTLINE'].str.contains('Motorcycles')]

classic\_cars = df[df['PRODUCTLINE'].str.contains('Classic Cars')]

ax.scatter(motorcycles['QUANTITYORDERED'], motorcycles['SALES'], c='red', marker='^', label='Motorcycles')

ax.scatter(classic\_cars['QUANTITYORDERED'], classic\_cars['SALES'], c='green', marker='o', label='Classic Cars')

# Customize the plot

plt.xlabel('Quantity Ordered')

plt.ylabel('Sales Amount (USD)')

plt.title('Sales vs Quantity Ordered')

plt.grid(True)

plt.legend()

plt.show()

EXERCISE 3A: (Question 1)

plt.bar(df['MONTH\_ID'], df['SALES'], color='green')

plt.xlabel('Month')

plt.ylabel('Total Sales')

plt.title('Total Sales by Month')

unique\_months = df['MONTH\_ID'].unique()

plt.xticks(unique\_months)

plt.show()

EXERCISE 3A: (Question 2)

# Group the data by 'PRODUCTLINE' and calculate the total quantity ordered for each product category

product\_quantity = df.groupby('PRODUCTLINE')['QUANTITYORDERED'].sum()

# Create a bar chart to visualize the total quantity ordered for each product category

plt.figure(figsize=(10, 6)) # Adjust the figure size as needed

plt.bar(product\_quantity.index, product\_quantity, color='blue')

plt.xlabel('Product Category')

plt.ylabel('Total Quantity Ordered')

plt.title('Quantity Ordered by Product Category')

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

plt.show()

EXERCISE 3B:

plt.hist(df['SALES'], bins=50, color='skyblue', edgecolor='black')

plt.xlabel('Sales')

plt.ylabel('Frequency')

plt.title('Sales Distribution')

plt.show()

EXERCISE 4:

yearly\_data = df[df['YEAR\_ID'] == 2003].head(20)

selected\_year = 2003

plt.figure(figsize=(8, 8))

plt.pie(yearly\_data['SALES'], labels=yearly\_data['CITY'], autopct='%1.1f%%')

plt.title(f'Monthly Sales Distribution for Year {selected\_year}')

# Display the pie chart

plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.

plt.show()