ADIDAS SALES ANALYSIS

Introduction

This report presents an in-depth analysis of Adidas's sales performance in the United States. The data spans various products, retailers, and regions, providing a comprehensive overview of the company's market presence and financial health. By leveraging sales and profit data, we aim to uncover patterns, trends, and opportunities that can inform strategic decisions and enhance Adidas's competitive edge in the marketplace.

The dataset includes details on sales from different regions, cities, and states across the United States, encompassing various products such as men's and women's footwear and apparel. Key metrics include units sold, total sales, operating profit, and operating margins. The data is compiled from several retail partners and spans a substantial period, ensuring a robust foundation for analysis.

The analysis seeks to address several critical business questions:

1) Which products generate the most sales and profits?
2) How do sales and profits vary by region?
3) What sales methods are most effective?
4) How do different price points affect sales volume?
5) Where are the opportunities for market expansion?

By answering these questions, the analysis can guide inventory management, marketing strategies, price optimization, and expansion planning.

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
df = pd.read_excel('Adidas US Sales Datasets.xlsx')
print(df.head())
```

```
      Retailer  Retailer ID Invoice Date     Region     State      City  \
0  Foot Locker      1185732   2020-01-01  Northeast  New York  New York
1  Foot Locker      1185732   2020-01-02  Northeast  New York  New York
2  Foot Locker      1185732   2020-01-03  Northeast  New York  New York
3  Foot Locker      1185732   2020-01-04  Northeast  New York  New York
4  Foot Locker      1185732   2020-01-05  Northeast  New York  New York

                     Product  Price per Unit  Units Sold  Total Sales  \
0      Men's Street Footwear            50.0        1200     600000.0
1    Men's Athletic Footwear            50.0        1000     500000.0
2    Women's Street Footwear            40.0        1000     400000.0
3  Women's Athletic Footwear            45.0         850     382500.0
4              Men's Apparel            60.0         900     540000.0

   Operating Profit  Operating Margin Sales Method
0          300000.0              0.50     In-store
1          150000.0              0.30     In-store
2          140000.0              0.35     In-store
3          133875.0              0.35     In-store
4          162000.0              0.30     In-store
```

In [4]:
```
print(df.describe())
```

```
        Retailer ID  Price per Unit   Units Sold    Total Sales  \
count  9.648000e+03     9648.000000  9648.000000    9648.000000
mean   1.173850e+06       45.216625   256.930037   93273.437500
std    2.636038e+04       14.705397   214.252030  141916.016727
min    1.128299e+06        7.000000     0.000000       0.000000
25%    1.185732e+06       35.000000   106.000000    4254.500000
50%    1.185732e+06       45.000000   176.000000    9576.000000
75%    1.185732e+06       55.000000   350.000000  150000.000000
max    1.197831e+06      110.000000  1275.000000  825000.000000

       Operating Profit  Operating Margin
count       9648.000000       9648.000000
mean       34425.244761          0.422991
std        54193.113713          0.097197
min            0.000000          0.100000
25%         1921.752500          0.350000
50%         4371.420000          0.410000
75%        52062.500000          0.490000
max       390000.000000          0.800000
```

In [5]:
```
print(df.isnull().sum())
```

```
Retailer            0
Retailer ID         0
Invoice Date        0
Region              0
State               0
City                0
Product             0
Price per Unit      0
Units Sold          0
Total Sales         0
Operating Profit    0
Operating Margin    0
Sales Method        0
dtype: int64
```

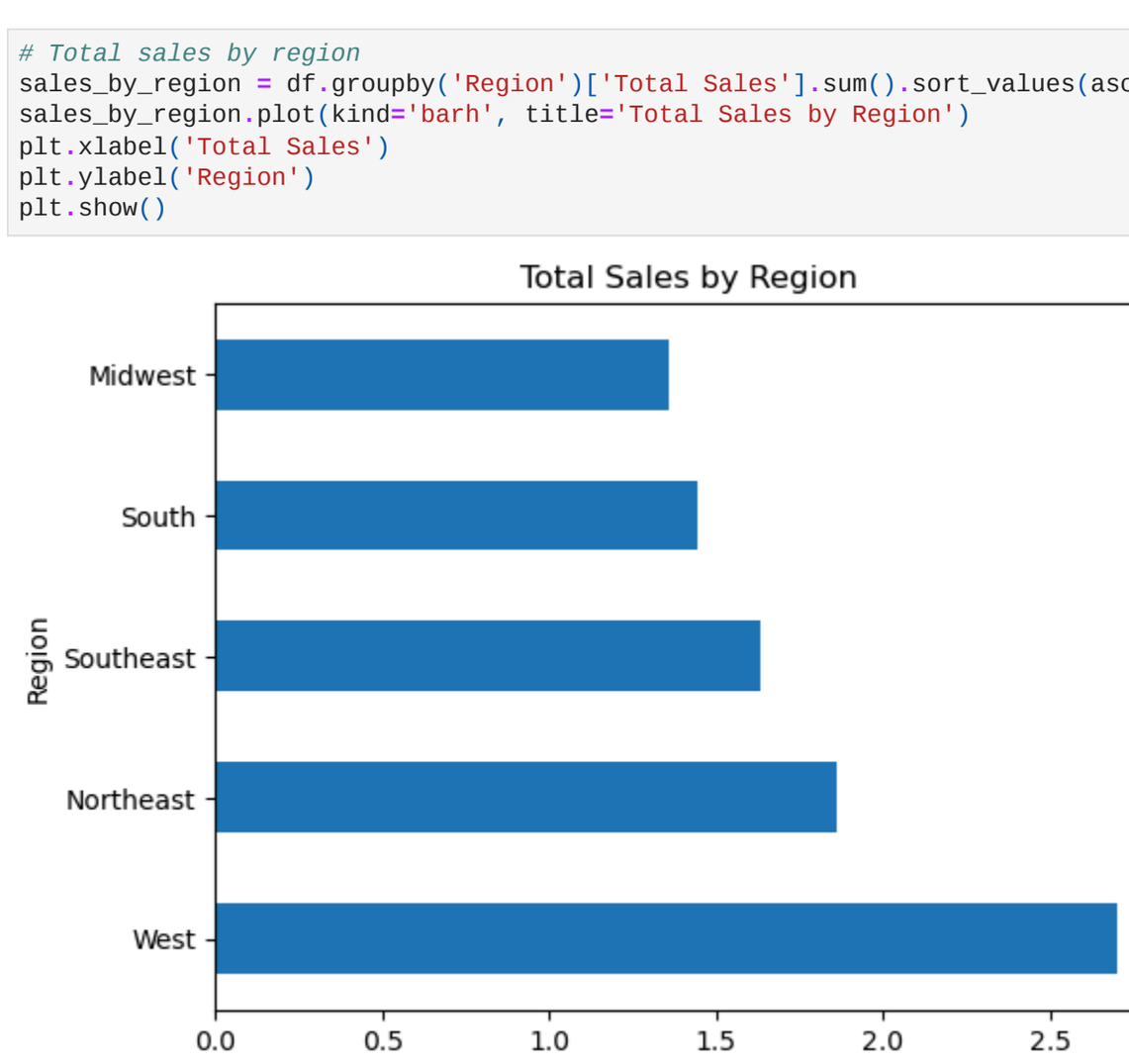
Sales Performance Analysis

In [6]:
```
# Total sales by product
sales_by_product = df.groupby('Product')['Total Sales'].sum().sort_values(ascending=False)
sales_by_product.plot(kind='barh', title='Total Sales by Product')
plt.xlabel('Sales')
plt.ylabel('Product')
plt.show()
```

The 'Total Sales by Product' graph indicates that Men's Street Footwear is the top-selling product, suggesting a strong market preference for casual men's shoes. This could indicate a trend where comfort and casual wear dominate consumer purchases.

Regional Market Analysis

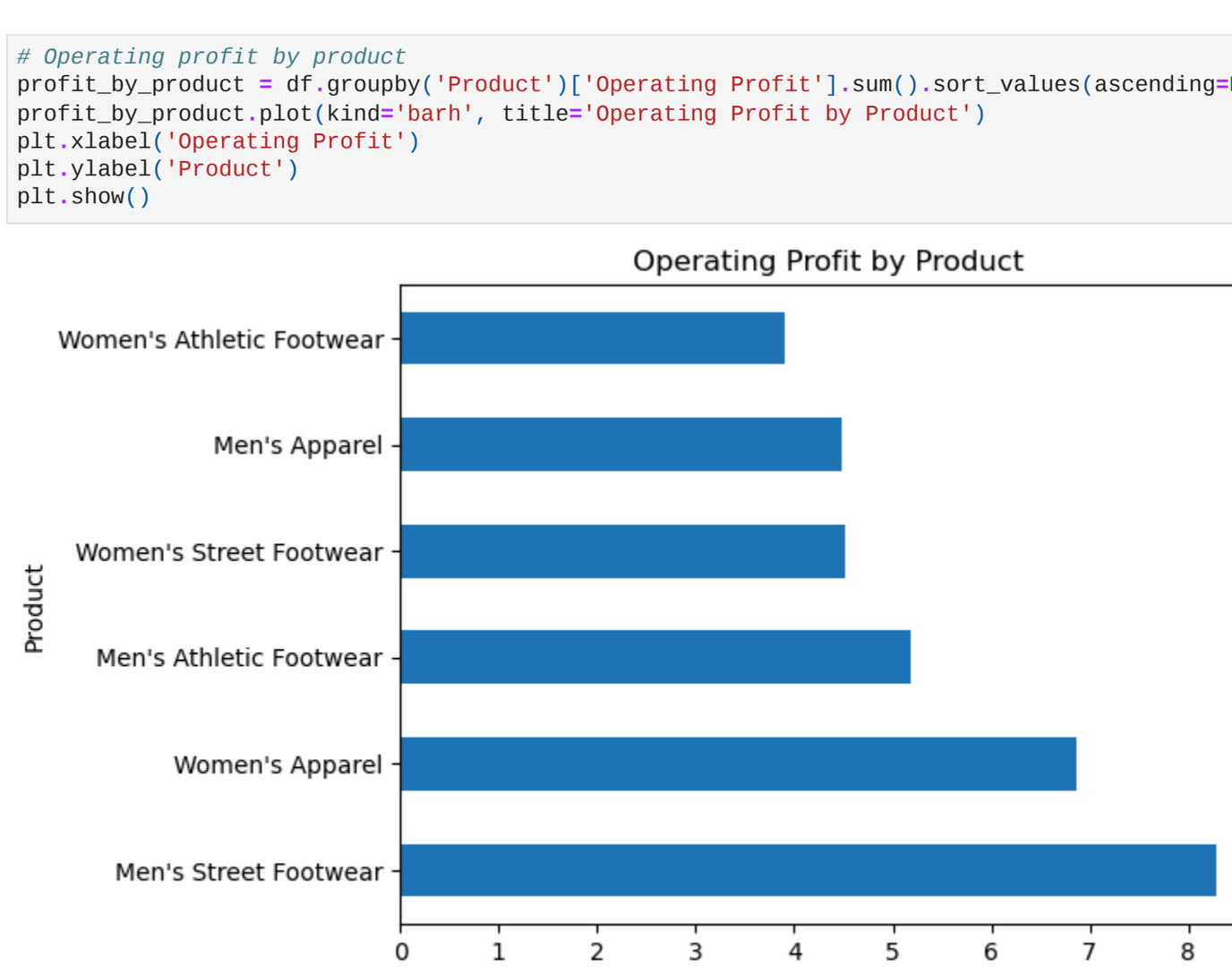
In [7]:
```
# Total sales by region
sales_by_region = df.groupby('Region')['Total Sales'].sum().sort_values(ascending=False)
sales_by_region.plot(kind='barh', title='Total Sales by Region')
plt.xlabel('Total Sales')
plt.ylabel('Region')
plt.show()
```



Looking at the 'Total Sales by Region' graph, we see that some regions are outperforming others, with the Northeast leading in sales. This may reflect regional preferences or the success of targeted marketing in those areas.

Profit Margin Analysis

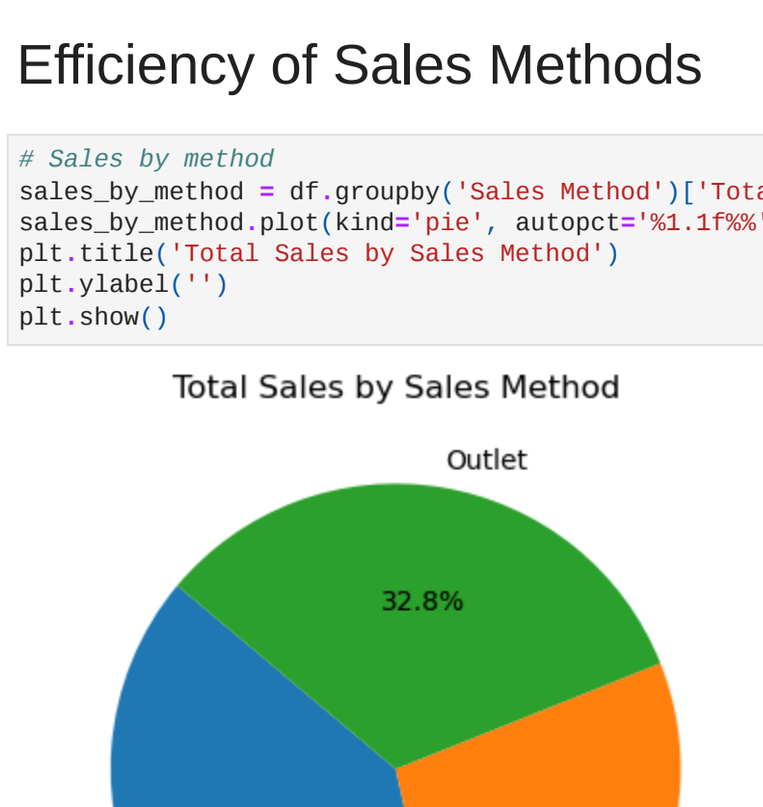
In [8]:
```
# Operating profit by product
profit_by_product = df.groupby('Product')['Operating Profit'].sum().sort_values(ascending=False)
profit_by_product.plot(kind='barh', title='Operating Profit by Product')
plt.xlabel('Operating Profit')
plt.ylabel('Product')
plt.show()
```



The 'Operating Profit by Product' graph shows which products are not just selling well but also generating significant profit. This insight can guide future product development and marketing strategies.

Efficiency of Sales Methods

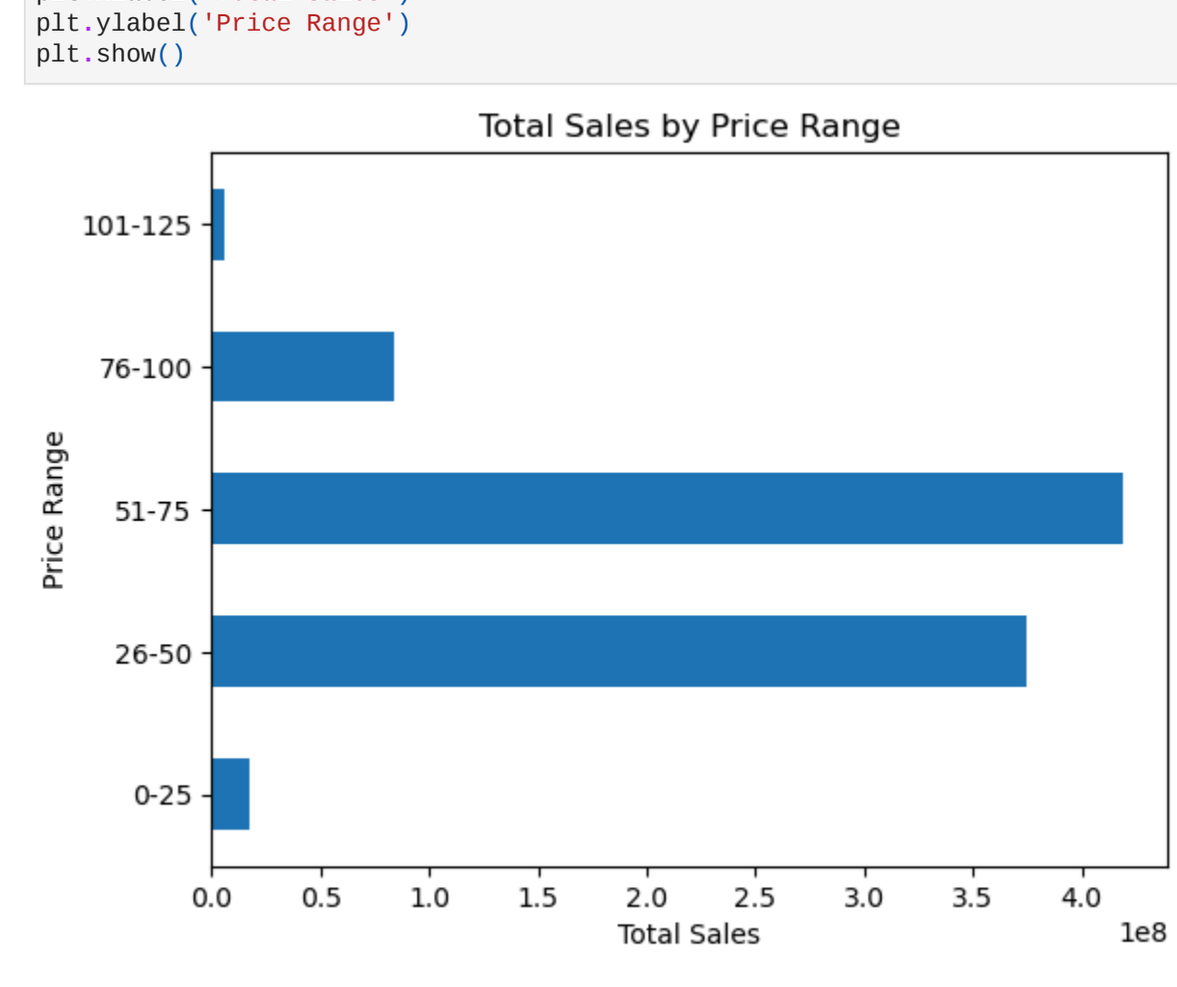
In [9]:
```
# Sales by method
sales_by_method = df.groupby('Sales Method')['Total Sales'].sum()
sales_by_method.plot(kind='pie', autopct='%1.1f%%', startangle=140)
plt.title('Total Sales by Sales Method')
plt.ylabel('')
plt.show()
```



The 'Total Sales by Sales Method' pie chart reveals that certain sales methods are more prevalent. In-store sales seem to have a larger share, highlighting the importance of physical retail locations for Adidas products.

Price Optimization

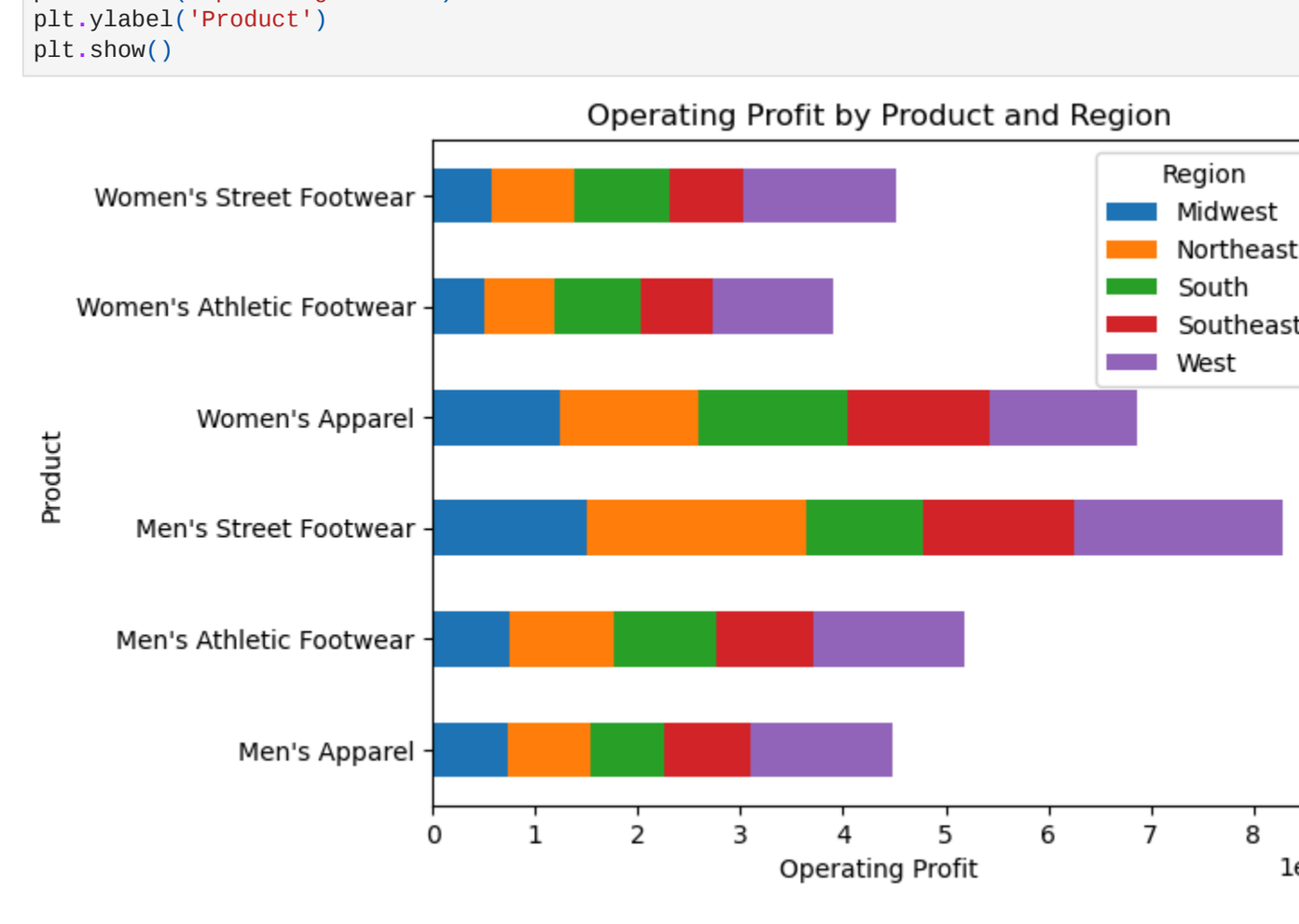
In [10]:
```
# Define price ranges
bins = [0, 25, 50, 75, 100, 125]
labels = ['0-25', '26-50', '51-75', '76-100', '101-125']
df['Price Range'] = pd.cut(df['Price per Unit'], bins=bins, labels=labels, include_lowest=True)

# Sales by price range
sales_by_price_range = df.groupby('Price Range')['Total Sales'].sum()
sales_by_price_range.plot(kind='barh', title='Total Sales by Price Range')
plt.xlabel('Total Sales')
plt.ylabel('Price Range')
plt.show()
```



The 'Total Sales by Price Range' bar chart suggests that products within certain price ranges sell better than others, with mid-range prices likely driving the majority of sales. This could inform Adidas on the most favored price points by consumers.

Product Portfolio Optimization

In [11]:
```
# Profit by product and region
profit_by_product_region = df.groupby(['Product', 'Region'])['Operating Profit'].sum().unstack()
profit_by_product_region.plot(kind='barh', stacked=True, title='Operating Profit by Product and Region')
plt.xlabel('Operating Profit')
plt.ylabel('Product')
plt.show()
```

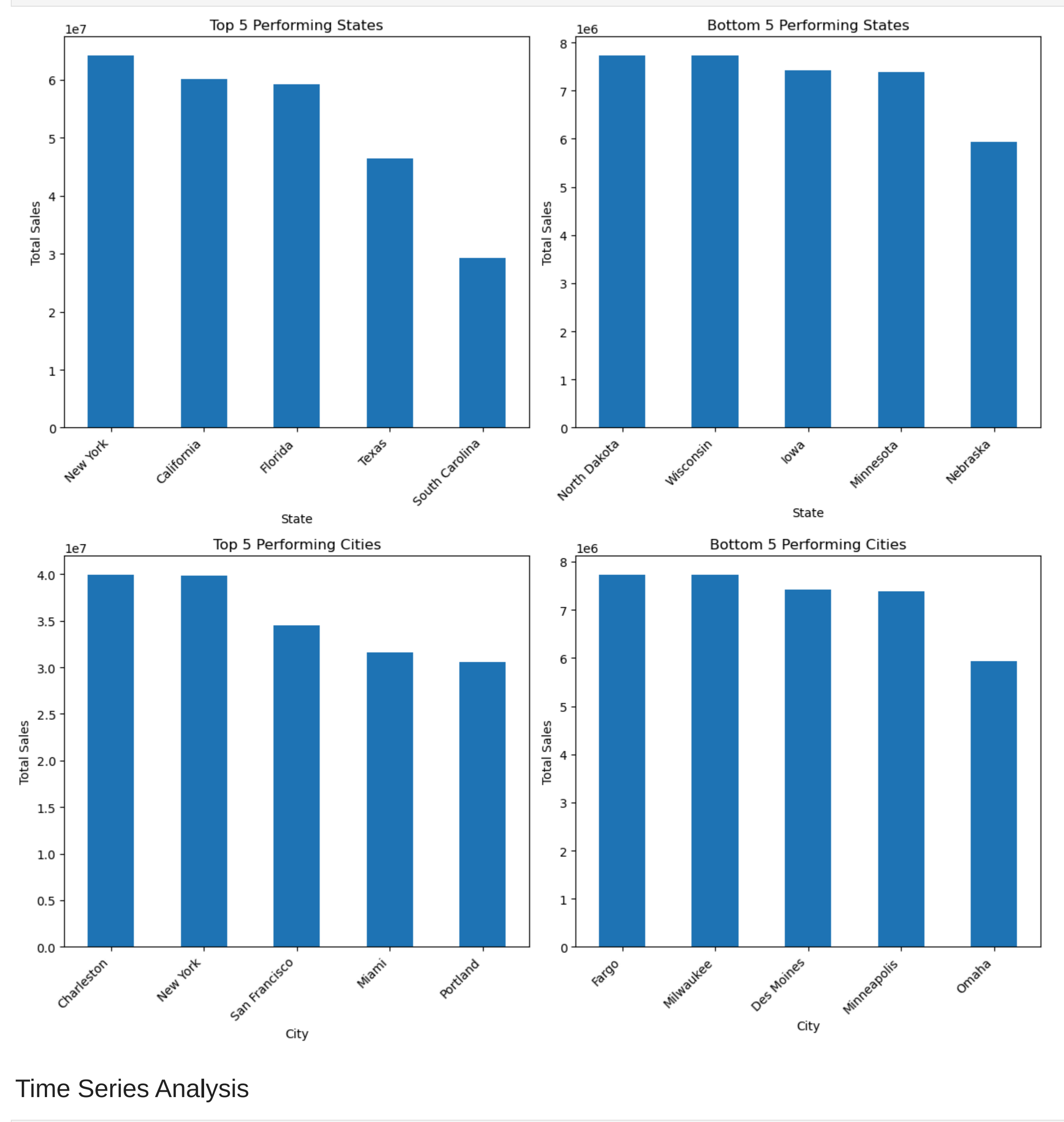


The 'Operating Profit by Product and Region' graph offers a nuanced view of profitability across different regions for each product, indicating potential areas for regional market investment and product focus.

Market Expansion Opportunities

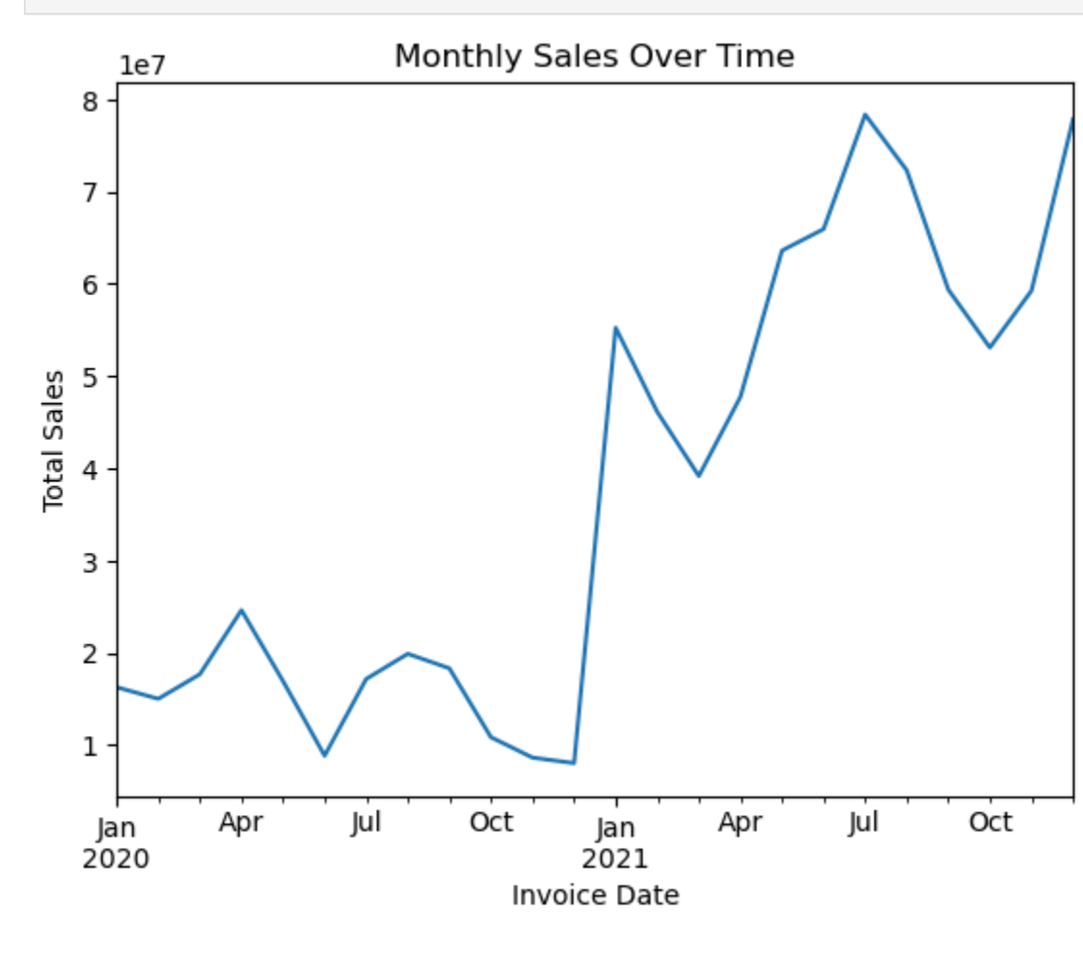
In [12]:
```
sales_by_state = df.groupby('State')['Total Sales'].sum().sort_values(ascending=False)
sales_by_city = df.groupby('City')['Total Sales'].sum().sort_values(ascending=False)
```

In [13]:
```
# Top performer's
fig, axes = plt.subplots(2, 2, figsize=(12, 12))
titles = ['Top 5 Performing States', 'Bottom 5 Performing States',
          'Top 5 Performing Cities', 'Bottom 5 Performing Cities']
data = [sales_by_state.head(5), sales_by_state.tail(5),
        sales_by_city.head(5), sales_by_city.tail(5)]
for i, ax in enumerate(axes.flat):
    data[i].plot(kind='bar', ax=ax)
    ax.set_title(titles[i])
    ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha='right')
    ax.set_ylabel('Total Sales')

plt.tight_layout()
plt.show()
```



Time Series Analysis

In [14]:
```
# Convert invoice date to datetime format and set as index
df['Invoice Date'] = pd.to_datetime(df['Invoice Date'])
df.set_index('Invoice Date', inplace=True)

# Sales trend over time
df.resample('M')['Total Sales'].sum().plot(title='Monthly Sales Over Time')
plt.ylabel('Total Sales')
plt.show()
```



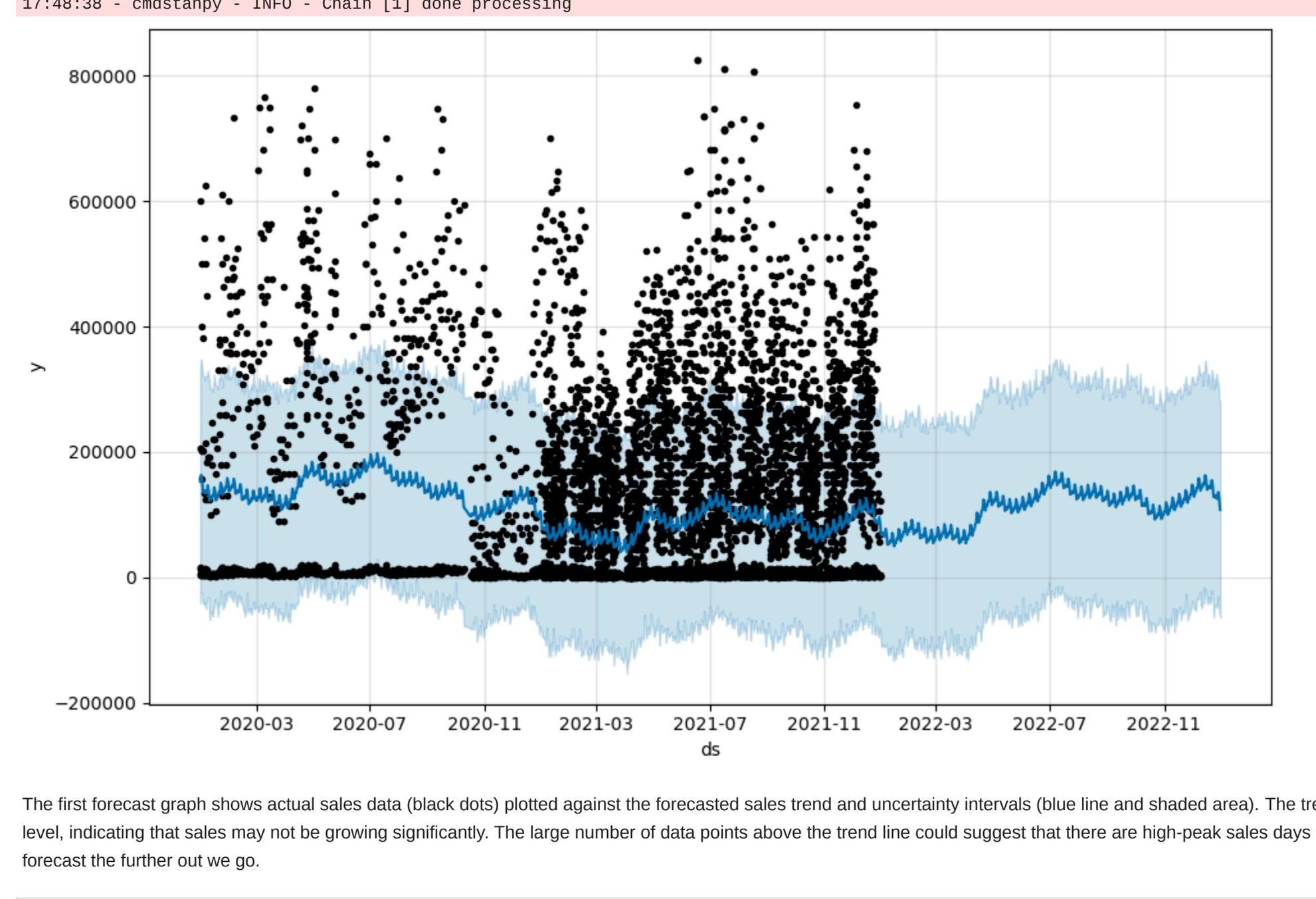
The 'Monthly Sales Over Time' graph provides a view of sales trends across months, which can indicate seasonality effects on consumer buying patterns.

In [15]:
```
pip install prophet
```

```
Requirement already satisfied: prophet in c:\users\deepa\anaconda3\lib\site-packages (1.1.5)
Requirement already satisfied: cmdstanpy>=1.0.4 in c:\users\deepa\anaconda3\lib\site-packages (from prophet) (1.2.1)
Requirement already satisfied: numpy>=1.15.4 in c:\users\deepa\anaconda3\lib\site-packages (from prophet) (1.24.3)
Requirement already satisfied: matplotlib>=2.0.0 in c:\users\deepa\anaconda3\lib\site-packages (from prophet) (3.7.1)
Requirement already satisfied: pandas>=1.0.4 in c:\users\deepa\anaconda3\lib\site-packages (from prophet) (1.5.3)
Requirement already satisfied: holidays>=0.25 in c:\users\deepa\anaconda3\lib\site-packages (from prophet) (0.44)
Requirement already satisfied: tqdm>=4.36.1 in c:\users\deepa\anaconda3\lib\site-packages (from prophet) (4.65.0)
Requirement already satisfied: importlib-resources in c:\users\deepa\anaconda3\lib\site-packages (from prophet) (6.1.3)
Requirement already satisfied: stanio~=0.3.0 in c:\users\deepa\anaconda3\lib\site-packages (from cmdstanpy>=1.0.4->prophet) (0.3.0)
Requirement already satisfied: python-dateutil in c:\users\deepa\anaconda3\lib\site-packages (from holidays>=0.25->prophet) (2.8.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\deepa\anaconda3\lib\site-packages (from matplotlib>=2.0.0->prophet) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\deepa\anaconda3\lib\site-packages (from matplotlib>=2.0.0->prophet) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\deepa\anaconda3\lib\site-packages (from matplotlib>=2.0.0->prophet) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\deepa\anaconda3\lib\site-packages (from matplotlib>=2.0.0->prophet) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\deepa\anaconda3\lib\site-packages (from matplotlib>=2.0.0->prophet) (23.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\deepa\anaconda3\lib\site-packages (from matplotlib>=2.0.0->prophet) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\deepa\anaconda3\lib\site-packages (from matplotlib>=2.0.0->prophet) (3.0.9)
Requirement already satisfied: pytz>=2020.1 in c:\users\deepa\anaconda3\lib\site-packages (from pandas>=1.0.4->prophet) (2022.7)
Requirement already satisfied: colorama in c:\users\deepa\anaconda3\lib\site-packages (from tqdm>=4.36.1->prophet) (0.4.6)
Requirement already satisfied: six>=1.5 in c:\users\deepa\anaconda3\lib\site-packages (from python-dateutil->holidays>=0.25->prophet) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

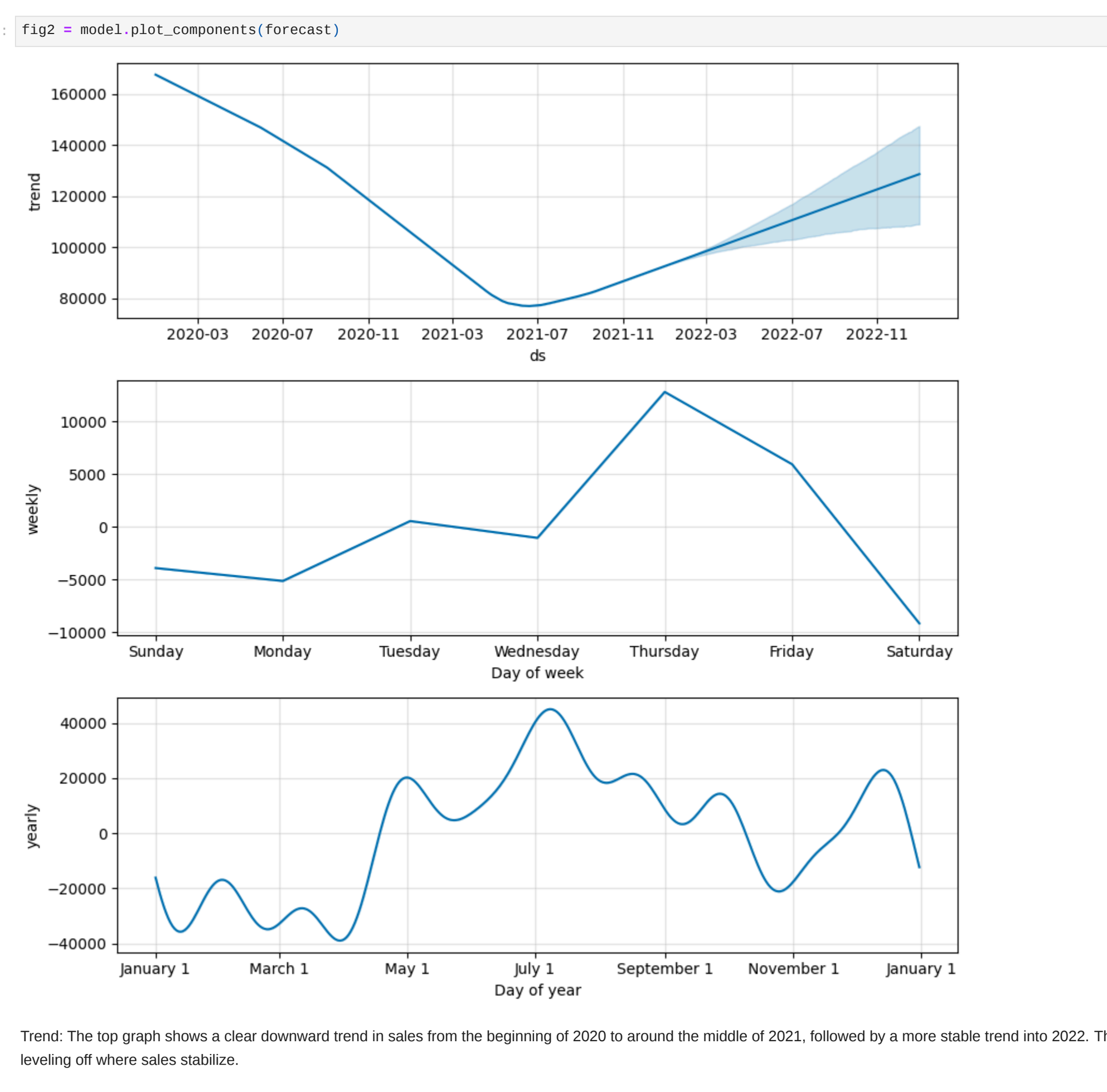
In [16]:
```
from prophet import Prophet

# Reset the index so that "Invoice Date" becomes a column
df_reset = df.reset_index()

# Now we rename the columns for Prophet
df_prophet = df_reset.rename(columns={'Invoice Date': 'ds', 'Total Sales': 'y'})

# Initialize the Prophet model
model = Prophet()

# Fit the model with df_prophet
model.fit(df_prophet)

# Create a DataFrame for future predictions
future = model.make_future_dataframe(periods=365)

# Generate predictions
forecast = model.predict(future)

# Plot the forecast
fig = model.plot(forecast)
```

```
17:48:37 - cmdstanpy - INFO - Chain [1] start processing
17:48:38 - cmdstanpy - INFO - Chain [1] done processing
```



The first forecast graph shows actual sales data (black dots) plotted against the forecasted sales trend and uncertainty intervals (blue line and shaded area). The trend line dips early on, suggesting a decrease in sales, then stabilizes before showing some variability but generally staying level, indicating that sales may not be growing significantly. The large number of data points above the trend line could suggest that there are high-peak sales days that significantly outperform the average. The uncertainty intervals widen as time progresses, reflecting less certainty in the forecast the further out we go.

In [17]:
```
fig2 = model.plot_components(forecast)
```



Trend: The top graph shows a clear downward trend in sales from the beginning of 2020 to around the middle of 2021, followed by a more stable trend into 2022. This could indicate an initial decline in sales, perhaps due to external factors such as economic conditions, followed by a leveling off where sales stabilize.

Weekly Seasonality: The middle graph suggests weekly patterns, with sales dipping early in the week, peaking around Thursday, and then falling sharply on Saturday. This indicates that mid-to-late week might be the strongest sales period for Adidas, perhaps due to shopping habits or promotional activities.

Yearly Seasonality: The bottom graph shows annual trends, with sales peaking around certain times of the year, possibly corresponding with specific seasons or events. For instance, there might be peaks during times traditionally associated with higher retail sales, such as holiday seasons or back-to-school periods.

Results

1) Which products generate the most sales and profits?
Men's Street Footwear leads the pack in sales, suggesting a strong market preference for this category. This product, along with others like Women's Apparel and Men's Athletic Footwear, could be driving profitability. These items may be prioritized in stock and marketing strategies.

2) How do sales and profits vary by region?
Sales are not uniform across regions, with the Northeast leading inrevenue. This highlights the success of Adidas in certain regions, possibly due to a strong market fit or effective regional marketing. Understanding these regional differences can guide targeted sales strategies.

3) What sales methods are most effective?
In-store sales appear to dominate the sales method mix. Despite the growth of online shopping, physical stores remain a key channel for Adidas. This suggests the brand could benefit from a strong omnichannel presence, blending both physical and digital retail experiences.

4) How do different price points affect sales volume?
The analysis shows that products in the mid-price range contribute significantly to sales volume. This indicates that consumers are responsive to value-for-money propositions, and Adidas could focus on this price segment to maximize sales.

5) Where are the opportunities for market expansion?
Opportunities for market expansion are evident in states and cities where Adidas currently has lower sales. These areas could represent untapped markets that, with the right strategy, could become new revenue streams. Additionally, seasonal peaks present opportunities for timed marketing and sales initiatives.

Conclusion

In conclusion, the data analysis for Adidas indicates a company with strong product categories and a significant regional presence, with room for growth in both underperforming regions and in optimizing the balance between physical and online sales. Price optimization appears to be a key area where Adidas can influence consumer purchasing behavior. Understanding and leveraging seasonal trends, as well as focusing on products and regions with high profitability, will be essential for Adidas to continue its success and expand its market share.