Sales Analysis

In [91]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
```

Merging Sales Data into a single CSV file

In [36]:
```
files = [file for file in os.listdir(r'C:\Users\sandh\OneDrive\Desktop\Projects\Python_Project\Sales_Analysis\Sales_Data')]

all_months_data = pd.DataFrame()

for file in files:
    df = pd.read_csv('C:/Users/sandh/OneDrive/Desktop/Projects/Python_Project/Sales_Analysis/Sales_Data/'+ file)
    all_months_data = pd.concat([all_months_data, df])

all_months_data.dropna(how='all', inplace=True)
all_months_data.to_csv(r'C:\Users\sandh\OneDrive\Desktop\Projects\Python_Project\Sales_Analysis\all_data.csv', index=False)
```

Reading updated dataframe

In [40]:
```
all_data = pd.read_csv(r'C:\Users\sandh\OneDrive\Desktop\Projects\Python_Project\Sales_Analysis\all_data.csv')
all_data.head()
```

Out[40]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001
1	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215
2	176560	Google Phone	1	600	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
3	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
4	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001

Augmented data with additinal columns

In [58]:
```
# Adding Month column using Order date to a copied data set:

all_data = all_data[all_data['Order Date'].str[0:2].str.isnumeric()].copy()
all_data['Month'] = all_data['Order Date'].str[0:2]
all_data['Month'] = all_data['Month'].astype('int32')

all_data.head()
```

Out[58]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4
1	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4
2	176560	Google Phone	1	600	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4
3	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4
4	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4

In [79]:
```
# Adding Sales column to the table:

all_data['Quantity Ordered'] = pd.to_numeric(all_data['Quantity Ordered'])   # converting string values to int.
all_data['Price Each'] = pd.to_numeric(all_data['Price Each'])               # converting string values to int.

all_data['Sales'] = all_data['Quantity Ordered'] * all_data['Price Each']
```

```
all_data.insert(4, 'Sales', all_data.pop('Sales'))                    # Moving Sales column to index 4 near Price Each
all_data.head()
```

Out[79]:

	Order ID	Product	Quantity Ordered	Price Each	Sales	Order Date	Purchase Address	Month
0	176558	USB-C Charging Cable	2	11.95	23.90	04/19/19 08:46	917 1st St, Dallas, TX 75001	4
1	176559	Bose SoundSport Headphones	1	99.99	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4
2	176560	Google Phone	1	600.00	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4
3	176560	Wired Headphones	1	11.99	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4
4	176561	Wired Headphones	1	11.99	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4

In [149…

```
# Adding City column into the dataset:

def get_city(address):                                    # function is created to split value at index 1 using split().
    return address.split(',')[1]

def get_state(address):
    return address.split(',')[2].split(' ')[1]

all_data['City'] = all_data['Purchase Address'].apply(lambda x : get_city(x) + ' ' + '('+get_state(x)+')')
all_data.head()
```

Out[149...

	Order ID	Product	Quantity Ordered	Price Each	Sales	Order Date	Purchase Address	Month	City
0	176558	USB-C Charging Cable	2	11.95	23.90	04/19/19 08:46	917 1st St, Dallas, TX 75001	4	Dallas (TX)
1	176559	Bose SoundSport Headphones	1	99.99	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4	Boston (MA)
2	176560	Google Phone	1	600.00	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	Los Angeles (CA)
3	176560	Wired Headphones	1	11.99	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	Los Angeles (CA)
4	176561	Wired Headphones	1	11.99	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4	Los Angeles (CA)

Exploratory Data Analysis (EDA)

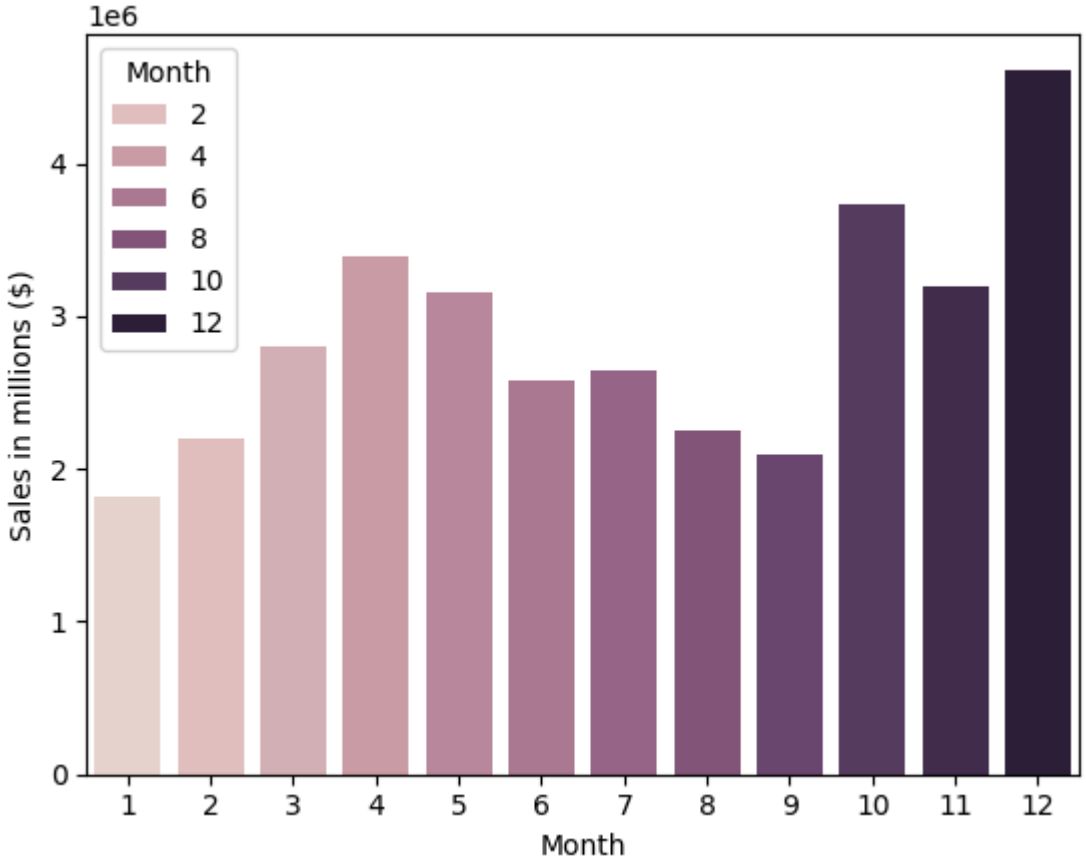
Month with Highest Sales

In [107...

```
monthly_sales = all_data.groupby(['Month'], as_index=False)['Sales'].sum()
sns.barplot(x='Month', y = 'Sales', data = monthly_sales, hue='Month')
plt.ylabel('Sales in millions ($)')
```

Out[107...

```
Text(0, 0.5, 'Sales in millions ($)')
```



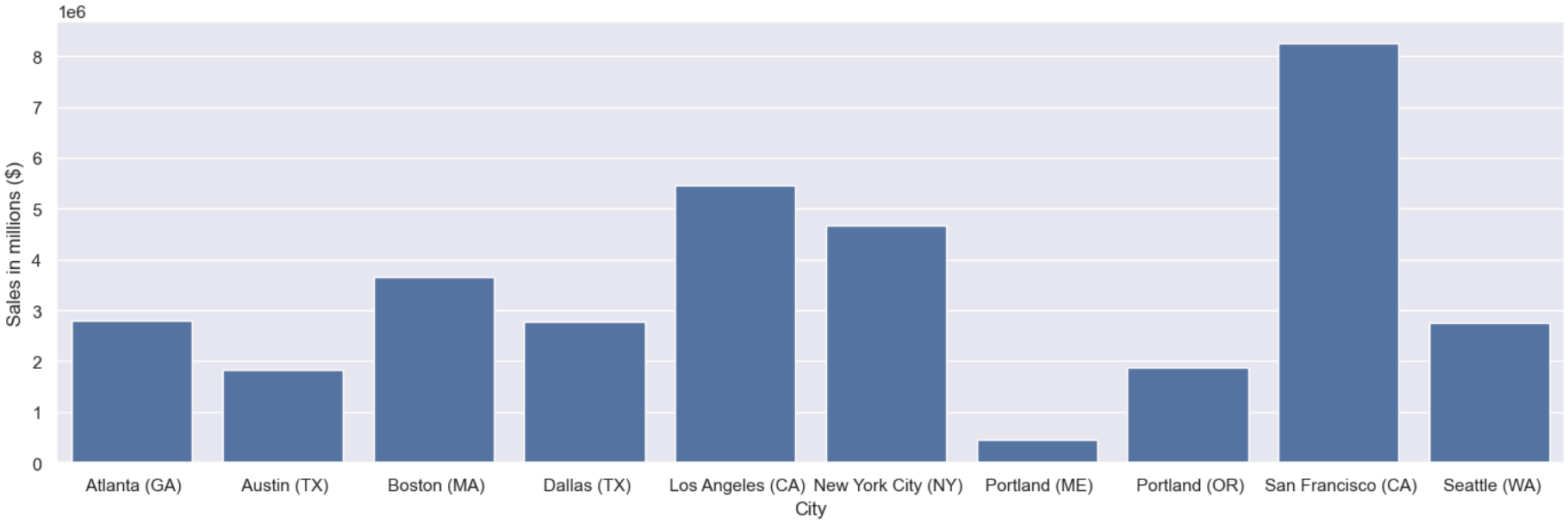
Sales peak in December, October, and April, driven by key market trends such as Christmas gifting, Black Friday sales, new tech product launches, and tax refunds. To maximize revenue, businesses should intensify marketing efforts, offer exclusive discounts, and create appealing bundles during these months. Leveraging strategic pricing and targeted promotions can effectively capitalize on increased consumer spending.

City with Highest number of Sales

In [163…

```
monthly_sales = all_data.groupby(['City'], as_index=False)['Sales'].sum().head(10)
sns.set(rc={'figure.figsize': (17,5)})
an = sns.barplot(x='City', y = 'Sales', data = monthly_sales)

plt.ylabel('Sales in millions ($)')
```

Out[163… Text(0, 0.5, 'Sales in millions ($)')



Sales are highest in San Francisco and Los Angeles due to high population density, strong tech presence, and higher disposable income, making them prime markets for premium electronics. In contrast, Portland and Austin see lower sales, influenced by smaller populations, eco-conscious consumer behavior, and fewer high-end retail options. To maximize growth, businesses should focus premium product launches and aggressive marketing in high-demand cities while introducing eco-friendly, budget-friendly options, and flexible financing in lower-sales regions to drive engagement and conversions.

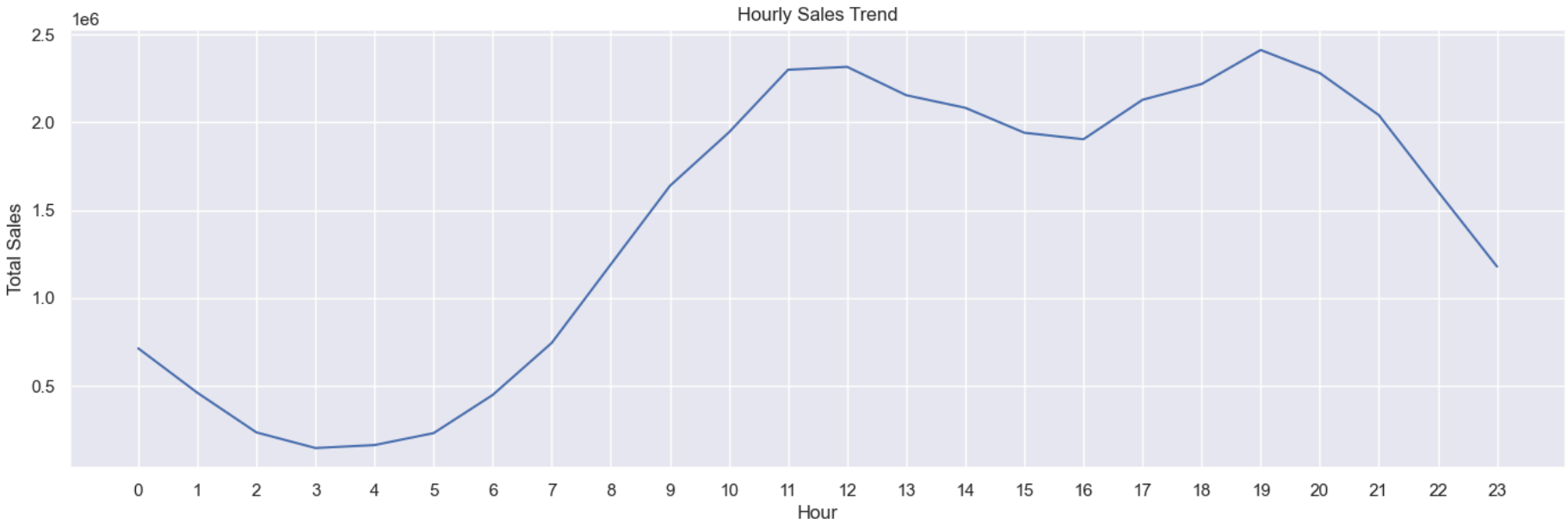
Hour of the day when Sales Maximize

In [178…
```
all_data['Order Date'] = pd.to_datetime(all_data['Order Date'])   # changing dtype to datetime.
```

In [206…
```
all_data['Hour'] = all_data['Order Date'].dt.hour                 # Hour column is added.

Hourly_sales = all_data.groupby(['Hour'], as_index=False)['Sales'].sum()
an = sns.lineplot(x='Hour', y = 'Sales', data = Hourly_sales)

plt.xticks(ticks=range(0, 24))
plt.title('Hourly Sales Trend')
```

```
plt.xlabel('Hour')
plt.ylabel('Total Sales')
```

Out[206… `Text(0, 0.5, 'Total Sales')`



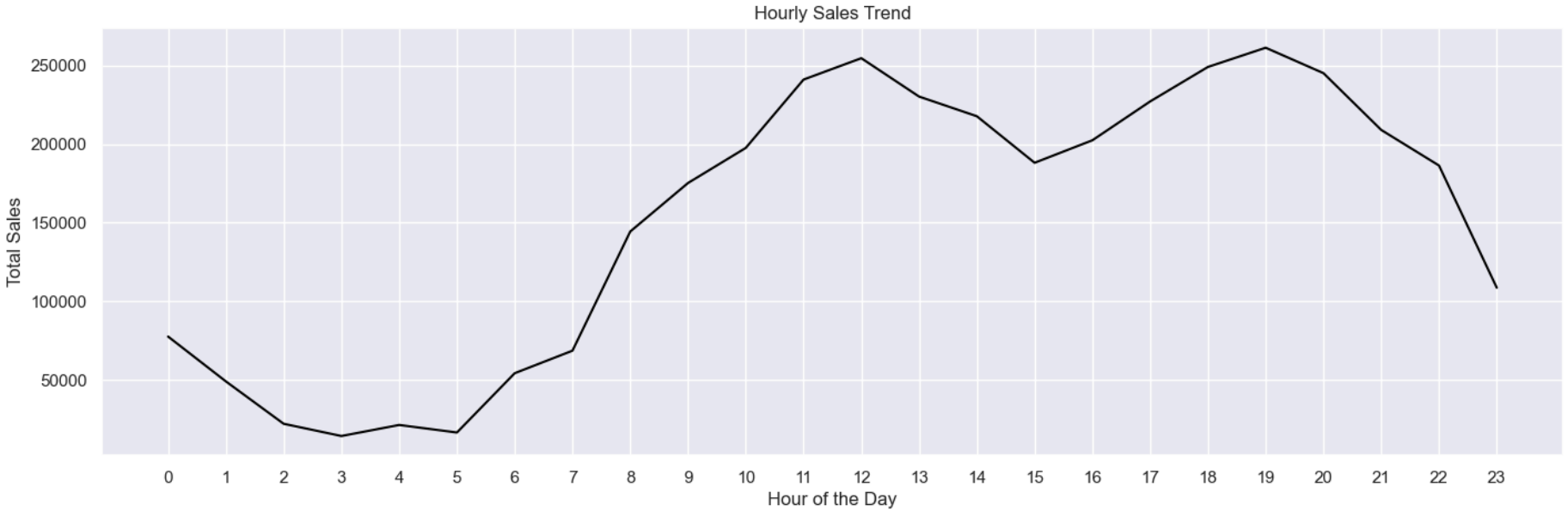
Sales peak at 11 AM, 12 PM, and 7 PM, driven by mid-morning breaks, lunch-hour browsing, and evening relaxation time when consumers are most engaged. To maximize conversions, businesses should schedule targeted ads, promotional emails, and flash sales before these peak hours, ensuring visibility when shoppers are most likely to make purchases. Optimizing marketing strategies around these key time slots can significantly enhance sales performance.

Hourly Sales of Boston(MA) City

In [269…

```
boston_city = all_data[all_data['City']==' Boston (MA)']                    # Boston city is filtered out.
boston_Hourly_sales = boston_city.groupby(['Hour'], as_index=False)['Sales'].sum()
an = sns.lineplot(x='Hour', y = 'Sales', data = boston_Hourly_sales, color='black')

plt.xticks(ticks=range(0, 24))
plt.title('Hourly Sales Trend')
plt.xlabel('Hour of the Day')
plt.ylabel('Total Sales')
```

Out[269… Text(0, 0.5, 'Total Sales')



Products that are most often Sold together

In [306…

```
df = all_data[all_data['Order ID'].duplicated(keep=False)].copy()          # fetching the products that are sold all together.

# products with same Order ID, grouped in a new column using ','.
df['Grouped Products'] = df.groupby('Order ID')['Product'].transform(','.join)
df = df[['Order ID', 'Grouped Products']].drop_duplicates()
df
```

Out[306...

	Order ID	Grouped Products
2	176560	Google Phone,Wired Headphones
17	176574	Google Phone,USB-C Charging Cable
29	176585	Bose SoundSport Headphones,Bose SoundSport Hea...
31	176586	AAA Batteries (4-pack),Google Phone
118	176672	Lightning Charging Cable,USB-C Charging Cable
...	...	...
186237	259296	Apple Airpods Headphones,Apple Airpods Headphones
186239	259297	iPhone,Lightning Charging Cable,Lightning Char...
186247	259303	34in Ultrawide Monitor,AA Batteries (4-pack)
186259	259314	Wired Headphones,AAA Batteries (4-pack)
186296	259350	Google Phone,USB-C Charging Cable

7136 rows × 2 columns

In [425...

```
from itertools import combinations       # It helped to generate all possible pairs (or combinations) of items from a list.
from collections import Counter          # It helped to count occurrences of items.

count = Counter()

for row in df['Grouped Products']:
    row_list = row.split(',')
    count.update(Counter(combinations(row_list,2)))

for key, value in count.most_common(10):
    print(key,value)
```

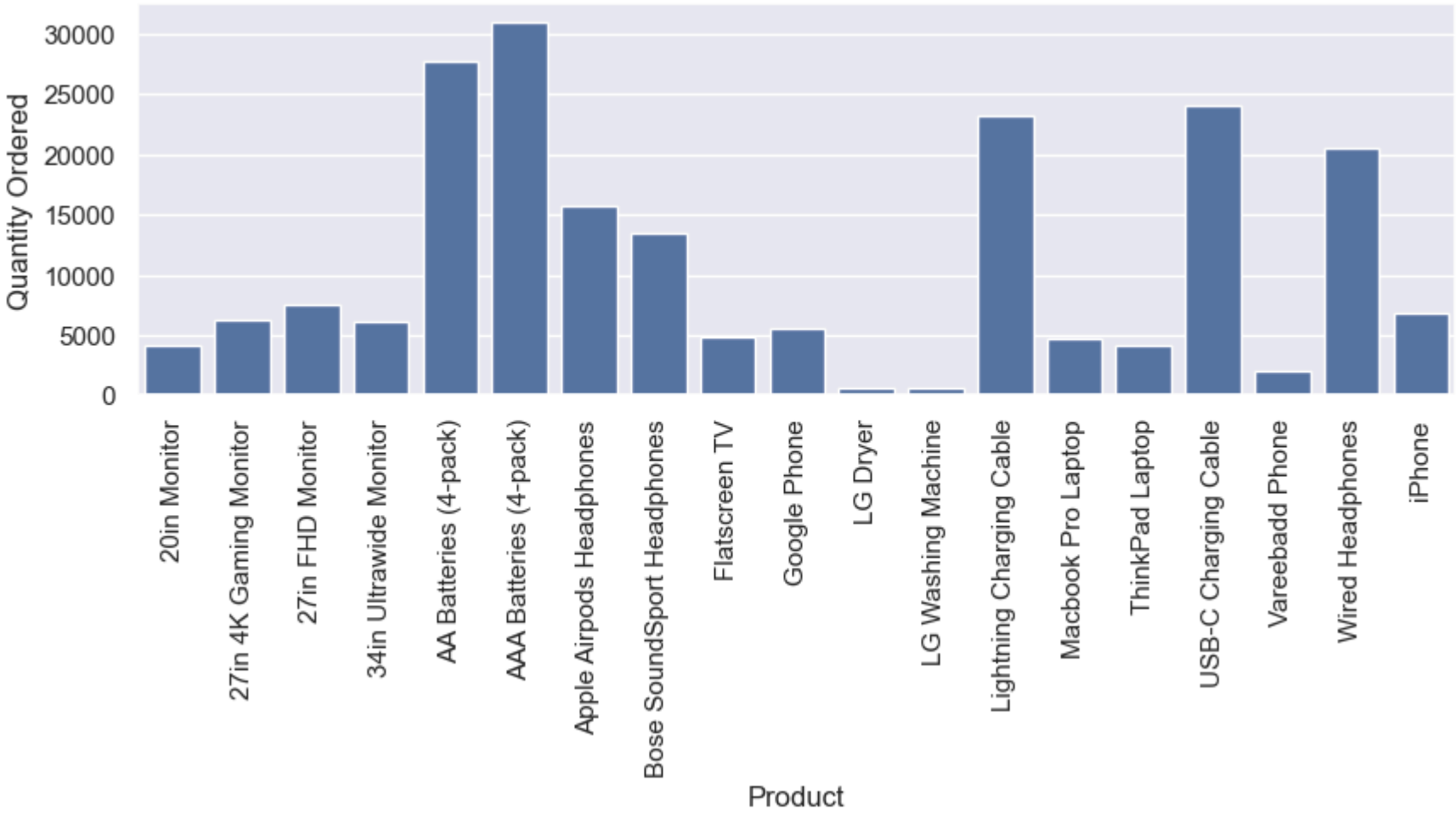
```
('iPhone', 'Lightning Charging Cable') 1005
('Google Phone', 'USB-C Charging Cable') 987
('iPhone', 'Wired Headphones') 447
('Google Phone', 'Wired Headphones') 414
('Vareebadd Phone', 'USB-C Charging Cable') 361
('iPhone', 'Apple Airpods Headphones') 360
('Google Phone', 'Bose SoundSport Headphones') 220
('USB-C Charging Cable', 'Wired Headphones') 160
('Vareebadd Phone', 'Wired Headphones') 143
('Lightning Charging Cable', 'Wired Headphones') 92
```

Most smartphone purchases include essential accessories like charging cables and headphones, with iPhone users preferring Apple accessories (AirPods, Lightning cables) and Google Phone users opting for USB-C cables and Bose headphones, indicating strong brand loyalty. Budget-conscious buyers tend to bundle Vareebadd Phones with wired headphones and USB-C cables. To maximize sales, businesses should offer bundled discounts, cross-sell accessories at checkout, and create brand-specific promotions, ensuring higher average order value and customer satisfaction.

Products that are Sold the most

In [335…

```
product_group = all_data.groupby('Product')['Quantity Ordered'].sum().reset_index()

sns.set(rc={'figure.figsize': (10,3)})
sns.barplot(x='Product', y = 'Quantity Ordered', data= product_group)
plt.xticks(rotation=90)
plt.show()
```

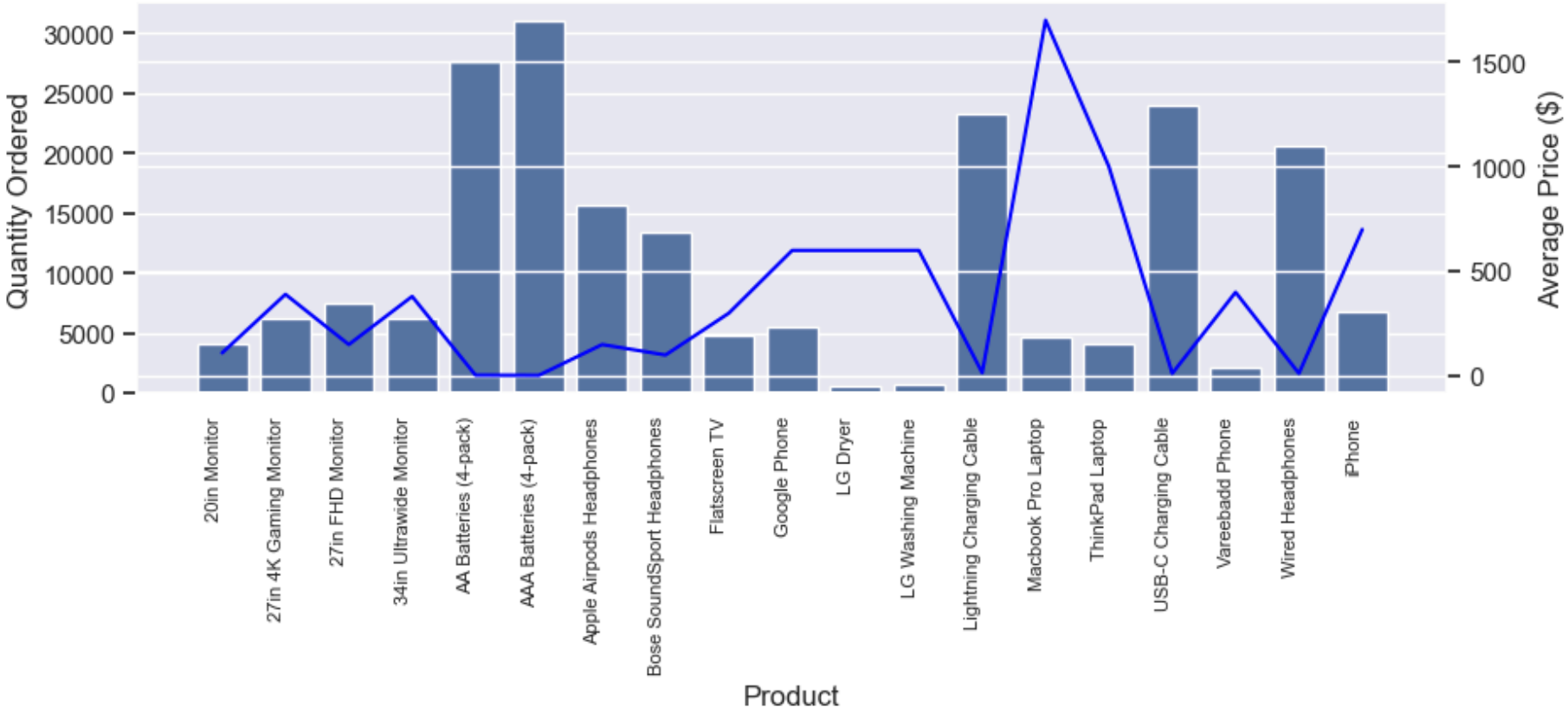


The above figure shows that products like AAA Batteries, AA Batteries, and USB-C Charging Cables were the most ordered. To gain further understanding, a correlation analysis with price has been conducted as follows.

In [418…

```
prices = all_data.groupby('Product')['Price Each'].mean()    # Mean price has been calculated of each product.

merged_data = pd.merge(product_group, prices, on='Product')

fig, ax1 = plt.subplots()

sns.barplot(x='Product', y='Quantity Ordered', data=merged_data)
```

```
ax1.set_ylabel('Quantity Ordered')

ax2 = ax1.twinx()                                          # Adding another Y- Axis.
sns.lineplot(x='Product', y='Price Each', data=merged_data, color='Blue')
ax2.set_ylabel('Average Price ($)')

ax1.set_xticks(range(len(merged_data['Product'])))                    # Setting up fixed ticks.
ax1.set_xticklabels(ax1.get_xticklabels(), rotation=90, ha='right', fontsize=8)
plt.show()
```



Based on the findings, there is a negative correlation between product price and quantity sold. Lower-priced items, such as AA batteries and USB-C cables, tend to have higher sales compared to more expensive products like LG washing machines and MacBooks. However, there are exceptions—for instance, MacBooks are more expensive than washing machines yet sell in higher quantities, possibly due to greater demand among students and professionals