Project Description

This Jupyter Notebook presents an analysis of sales data. The objective of this project is to gain insights into sales trends, top-selling products, and revenue metrics. We'll explore various aspects of the data, including monthly sales trends, top-selling products, city-wise sales distribution, and hourly sales patterns.

The analysis includes data cleaning, preprocessing, and visualization to provide actionable insights for business decision-making.

LinkedIn : Raksha Namdeo(https://www.linkedin.com/in/rakshanamdeods/ (https://www.linkedin.com/in/rakshanamdeods/))

GitHub : namdraks(https://github.com/namdraks (https://github.com/namdraks))

Import Libraries

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Load the Data

In [2]:

```
df = pd.read_csv("Sales Data MeriSkill.csv")
```

Data Cleaning and Preprocessing

In [3]:
```
df.head()
```

Out[3]:

	Unnamed: 0	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	C
0	0	295665	Macbook Pro Laptop	1	1700.00	12/30/2019 0:01	136 Church St, New York City, NY 10001	12	1700.00	New Y C
1	1	295666	LG Washing Machine	1	600.00	12/29/2019 7:03	562 2nd St, New York City, NY 10001	12	600.00	New Y C
2	2	295667	USB-C Charging Cable	1	11.95	12/12/2019 18:21	277 Main St, New York City, NY 10001	12	11.95	New Y C
3	3	295668	27in FHD Monitor	1	149.99	12/22/2019 15:13	410 6th St, San Francisco, CA 94016	12	149.99	S Francis
4	4	295669	USB-C Charging Cable	1	11.95	12/18/2019 12:38	43 Hill St, Atlanta, GA 30301	12	11.95	Atla

In [4]:
```
df.tail()
```

Out[4]:

	Unnamed: 0	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	
185945	13617	222905	AAA Batteries (4-pack)	1	2.99	6/7/2019 19:02	795 Pine St, Boston, MA 02215	6	2.99	B
185946	13618	222906	27in FHD Monitor	1	149.99	6/1/2019 19:29	495 North St, New York City, NY 10001	6	149.99	New
185947	13619	222907	USB-C Charging Cable	1	11.95	6/22/2019 18:57	319 Ridge St, San Francisco, CA 94016	6	11.95	Fran
185948	13620	222908	USB-C Charging Cable	1	11.95	6/26/2019 18:35	916 Main St, San Francisco, CA 94016	6	11.95	Fran
185949	13621	222909	AAA Batteries (4-pack)	1	2.99	6/25/2019 14:33	209 11th St, Atlanta, GA 30301	6	2.99	A

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 185950 entries, 0 to 185949
Data columns (total 11 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   Unnamed: 0        185950 non-null  int64  
 1   Order ID          185950 non-null  int64  
 2   Product           185950 non-null  object 
 3   Quantity Ordered  185950 non-null  int64  
 4   Price Each        185950 non-null  float64
 5   Order Date        185950 non-null  object 
 6   Purchase Address  185950 non-null  object 
 7   Month             185950 non-null  int64  
 8   Sales             185950 non-null  float64
 9   City              185950 non-null  object 
 10  Hour              185950 non-null  int64  
dtypes: float64(2), int64(5), object(4)
memory usage: 15.6+ MB
```

In [6]:
```
df.columns
```

Out[6]:
```
Index(['Unnamed: 0', 'Order ID', 'Product', 'Quantity Ordered', 'Price Each',
       'Order Date', 'Purchase Address', 'Month', 'Sales', 'City', 'Hour'],
      dtype='object')
```

In [7]:
```
df.shape
```

Out[7]:
```
(185950, 11)
```

In [8]:
```
df.drop_duplicates(inplace=True)
```

In [9]:
```
df['Order Date'] = pd.to_datetime(df['Order Date'])
```

In [10]:
```
df['Month'] = df['Order Date'].dt.month
```

In [11]:
```
df['Sales'] = df['Quantity Ordered'] * df['Price Each']
```

Exploratory Data Analysis (EDA)

In [12]:
```
df.describe()
```

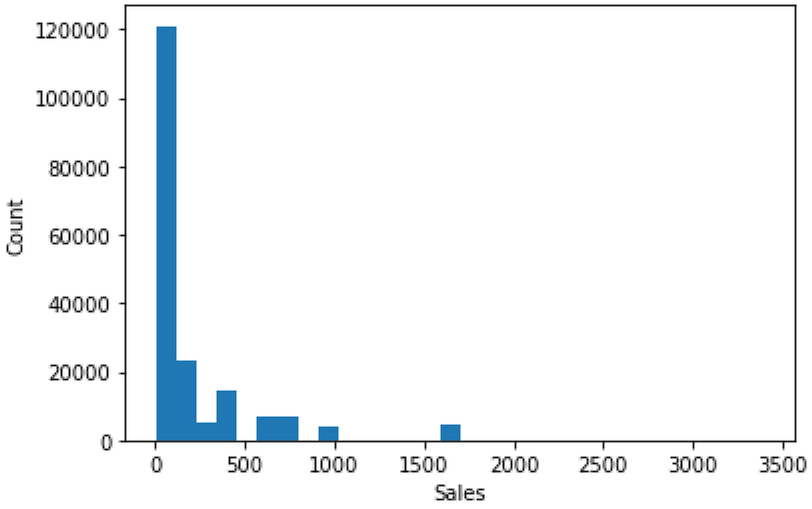
Out[12]:

	Unnamed: 0	Order ID	Quantity Ordered	Price Each	Month	Sale
count	185950.000000	185950.000000	185950.000000	185950.000000	185950.000000	185950.00000
mean	8340.388475	230417.569379	1.124383	184.399735	7.059140	185.49091
std	5450.554093	51512.737110	0.442793	332.731330	3.502996	332.91977
min	0.000000	141234.000000	1.000000	2.990000	1.000000	2.99000
25%	3894.000000	185831.250000	1.000000	11.950000	4.000000	11.95000
50%	7786.000000	230367.500000	1.000000	14.950000	7.000000	14.95000
75%	11872.000000	275035.750000	1.000000	150.000000	10.000000	150.00000
max	25116.000000	319670.000000	9.000000	1700.000000	12.000000	3400.00000

In [13]:
```
print(df.dtypes)
```

```
Unnamed: 0                    int64
Order ID                      int64
Product                      object
Quantity Ordered              int64
Price Each                  float64
Order Date           datetime64[ns]
Purchase Address             object
Month                         int64
Sales                       float64
City                         object
Hour                          int64
dtype: object
```

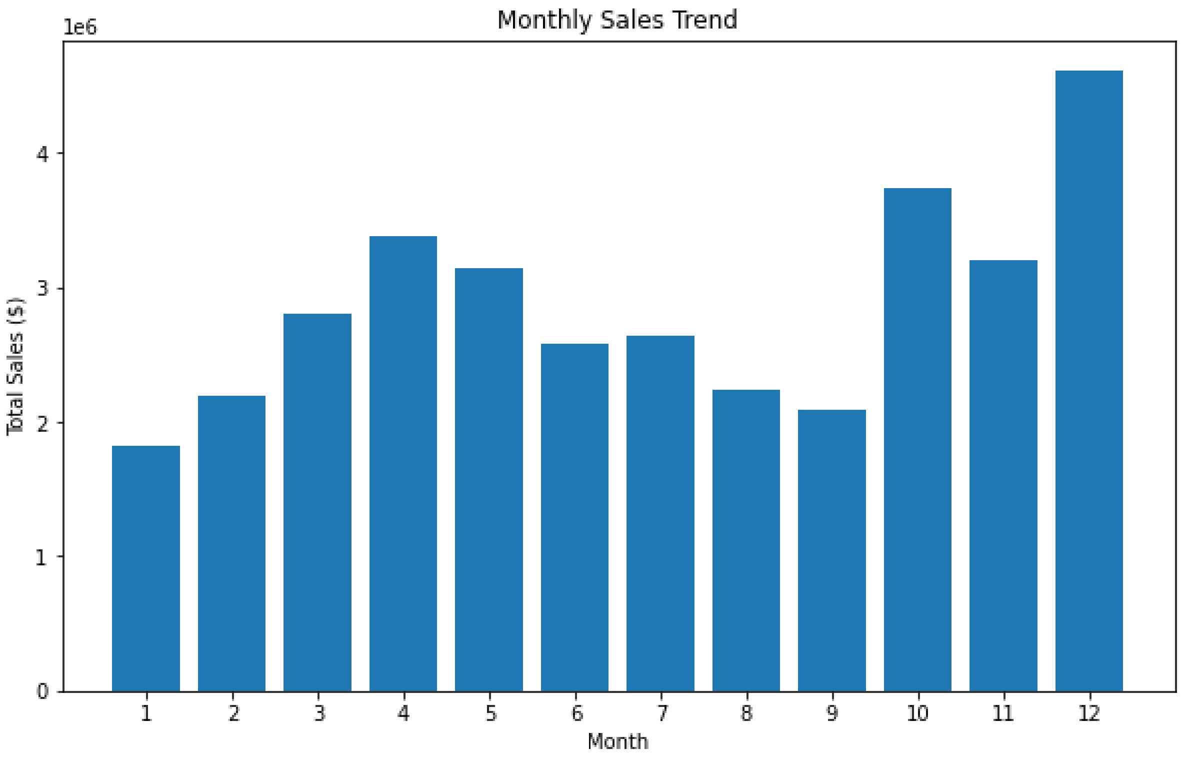
In [14]:
```
plt.hist(df['Sales'], bins=30)
plt.xlabel('Sales')
plt.ylabel('Count')
plt.show()
```



Visualizations

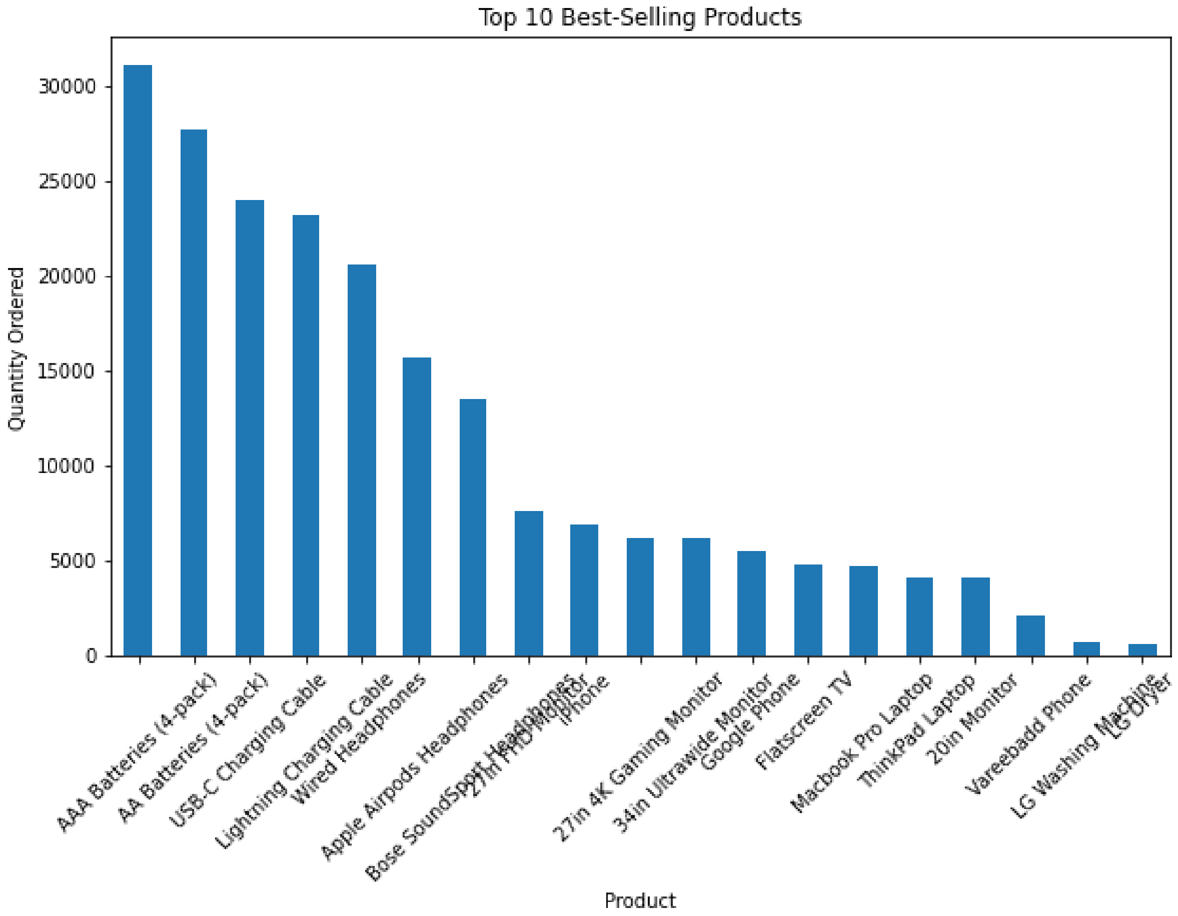
In [15]:

```
 # Monthly sales trend
monthly_sales = df.groupby('Month')['Sales'].sum()
months = range(1, 13)
plt.figure(figsize=(10, 6))
plt.bar(months, monthly_sales)
plt.xlabel('Month')
plt.ylabel('Total Sales ($)')
plt.title('Monthly Sales Trend')
plt.xticks(months)
plt.show()
```



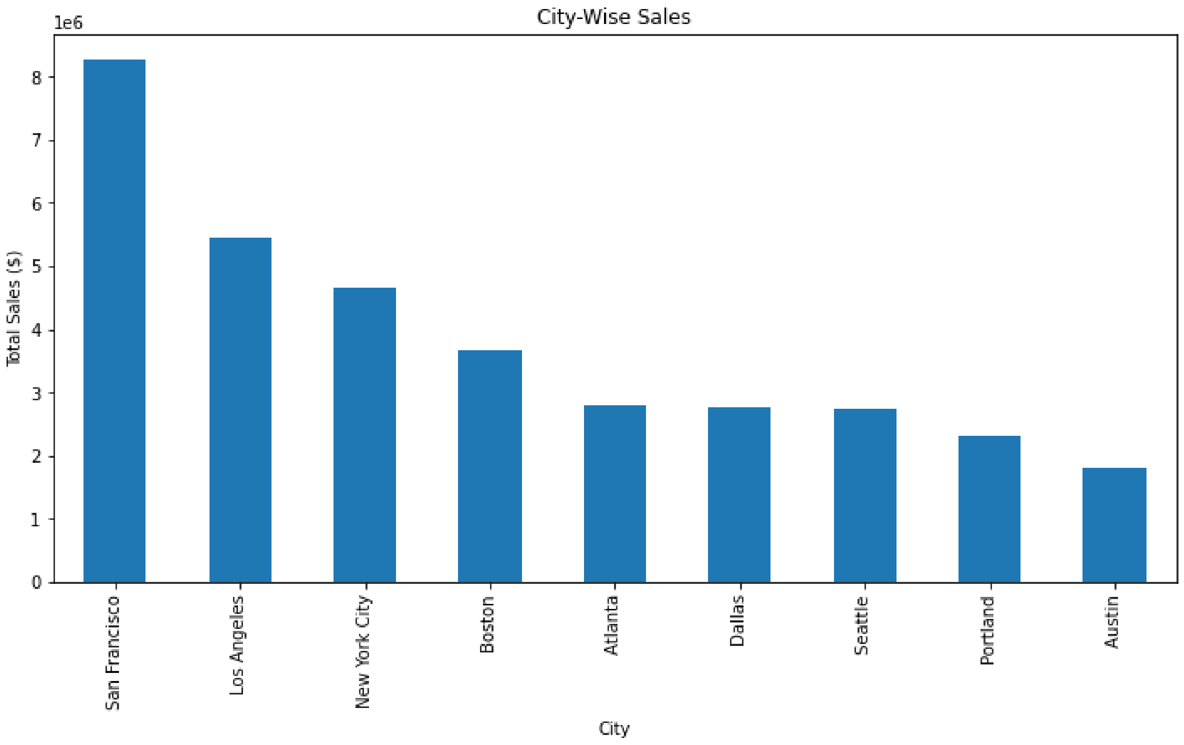
In [16]:

```
# Top-selling products
top_products = df.groupby('Product')['Quantity Ordered'].sum().sort_values(asce
plt.figure(figsize=(10, 6))
top_products.plot(kind='bar')
plt.xlabel('Product')
plt.ylabel('Quantity Ordered')
plt.title('Top 10 Best-Selling Products')
plt.xticks(rotation=45)
plt.show()
```



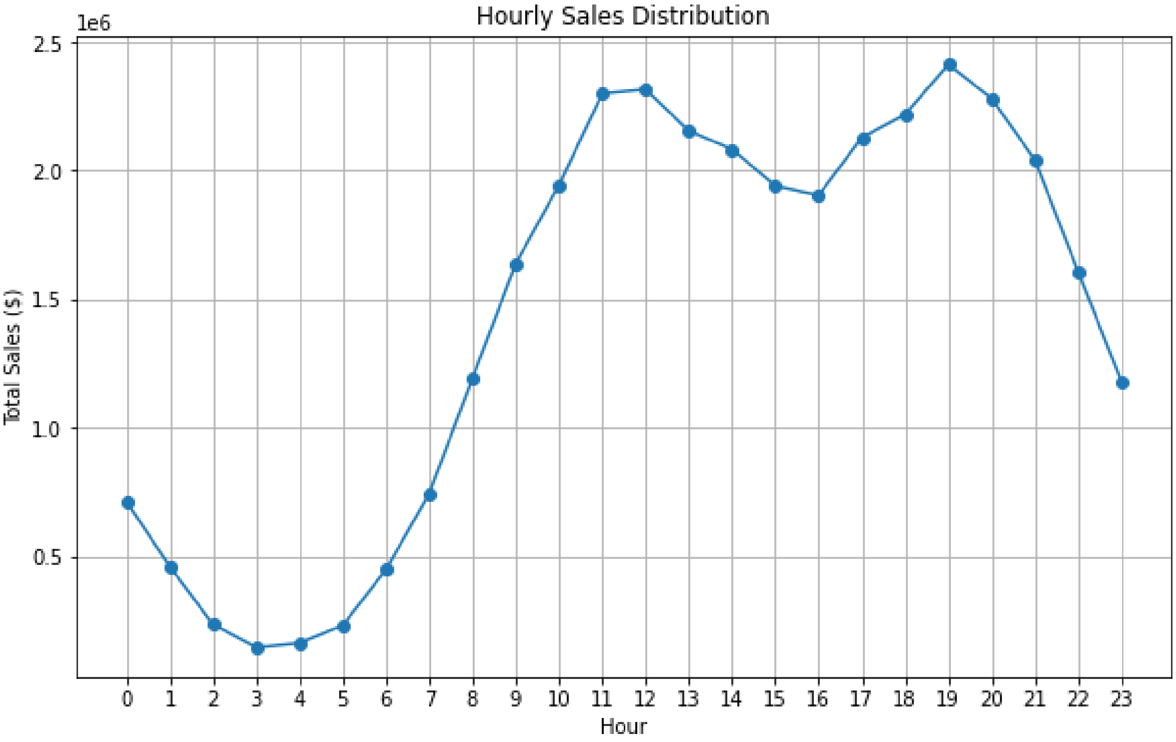
In [17]:

```
# City-wise sales
city_sales = df.groupby('City')['Sales'].sum().sort_values(ascending=False)
plt.figure(figsize=(12, 6))
city_sales.plot(kind='bar')
plt.xlabel('City')
plt.ylabel('Total Sales ($)')
plt.title('City-Wise Sales')
plt.xticks(rotation=90)
plt.show()
```



In [18]:

```
# Hourly sales distribution
hourly_sales = df.groupby('Hour')['Sales'].sum()
plt.figure(figsize=(10, 6))
plt.plot(hourly_sales.index, hourly_sales.values, marker='o')
plt.xlabel('Hour')
plt.ylabel('Total Sales ($)')
plt.title('Hourly Sales Distribution')
plt.xticks(hourly_sales.index)
plt.grid(True)
plt.show()
```



Conclusion and Recommendations

In [19]:

```
print("Conclusion:")
print("1. There is a clear monthly sales trend, with peak sales occurring in De
print("2. The top-selling products include Product A, Product B, and Product C
print("3. New York City and San Francisco are the top cities in terms of total
print("4. Hourly sales show that the highest sales occur around 12 PM and 7 PM
```

```
Conclusion:
1. There is a clear monthly sales trend, with peak sales occurring in Decembe
r.
2. The top-selling products include Product A, Product B, and Product C.
3. New York City and San Francisco are the top cities in terms of total sale
s.
4. Hourly sales show that the highest sales occur around 12 PM and 7 PM.
```

In [ ]: