Sales Data Analysis

October 28, 2023

1 Sales Data Analysis Project

1.1 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: Chandramani Kumar Chandravnsi

GitHub: ckrazaditya

1.2 Import Libraries

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

1.3 Load the Data

[2]:
```
df = pd.read_csv("C:/Users/Lenovo/Desktop/Meri skill Project/
 ↪drive-download-20231003T163556Z-001/Project 1 - Sales Data Analysis/Dataset␣
 ↪MeriSKILL/Sales Data.csv")
```

1.4 Data Cleaning and Preprocessing

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

[3]:
```
   Unnamed: 0  Order ID               Product  Quantity Ordered  Price Each  \
0           0    295665    Macbook Pro Laptop                 1     1700.00
1           1    295666    LG Washing Machine                 1      600.00
2           2    295667  USB-C Charging Cable                 1       11.95
3           3    295668      27in FHD Monitor                 1      149.99
4           4    295669  USB-C Charging Cable                 1       11.95

            Order Date                        Purchase Address  Month  \
0  2019-12-30 00:01:00  136 Church St, New York City, NY 10001     12
1  2019-12-29 07:03:00     562 2nd St, New York City, NY 10001     12
2  2019-12-12 18:21:00    277 Main St, New York City, NY 10001     12
3  2019-12-22 15:13:00     410 6th St, San Francisco, CA 94016     12
4  2019-12-18 12:38:00           43 Hill St, Atlanta, GA 30301     12

     Sales           City  Hour
0  1700.00  New York City     0
1   600.00  New York City     7
2    11.95  New York City    18
3   149.99  San Francisco    15
4    11.95        Atlanta    12
```

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

[4]:
```
        Unnamed: 0  Order ID                 Product  Quantity Ordered  \
185945       13617    222905  AAA Batteries (4-pack)                 1
185946       13618    222906         27in FHD Monitor                 1
185947       13619    222907    USB-C Charging Cable                 1
185948       13620    222908    USB-C Charging Cable                 1
185949       13621    222909  AAA Batteries (4-pack)                 1

        Price Each           Order Date  \
185945        2.99  2019-06-07 19:02:00
185946      149.99  2019-06-01 19:29:00
185947       11.95  2019-06-22 18:57:00
185948       11.95  2019-06-26 18:35:00
185949        2.99  2019-06-25 14:33:00

                               Purchase Address  Month   Sales           City  \
185945          795 Pine St, Boston, MA 02215      6    2.99         Boston
185946  495 North St, New York City, NY 10001      6  149.99  New York City
185947  319 Ridge St, San Francisco, CA 94016      6   11.95  San Francisco
185948   916 Main St, San Francisco, CA 94016      6   11.95  San Francisco
185949         209 11th St, Atlanta, GA 30301      6    2.99        Atlanta

        Hour
185945    19
185946    19
185947    18
185948    18
185949    14
```

[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
```

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

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

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

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

1.5 Data Cleaning and Preprocessing

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

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

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

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

1.6 Exploratory Data Analysis (EDA)

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

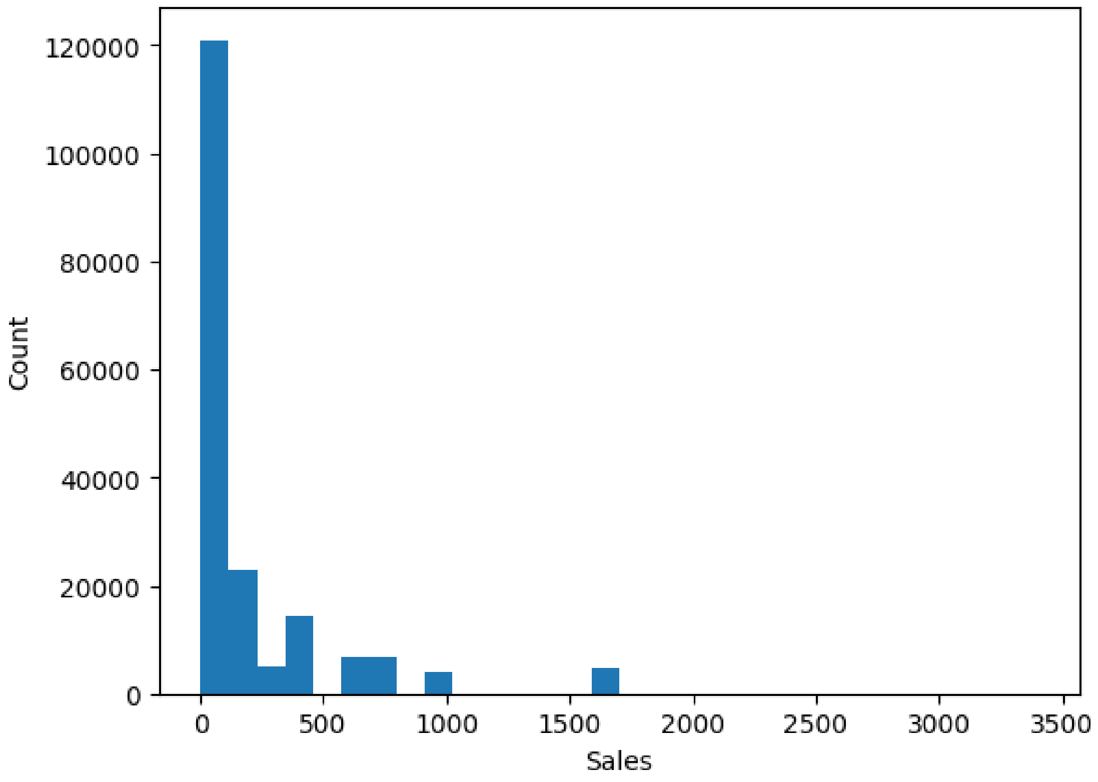
[12]:
```
          Unnamed: 0       Order ID  Quantity Ordered     Price Each  \
count  185950.000000  185950.000000     185950.000000  185950.000000
mean     8340.388475  230417.569379          1.124383     184.399735
min         0.000000  141234.000000          1.000000       2.990000
25%      3894.000000  185831.250000          1.000000      11.950000
50%      7786.000000  230367.500000          1.000000      14.950000
75%     11872.000000  275035.750000          1.000000     150.000000
max     25116.000000  319670.000000          9.000000    1700.000000
```

```
std      5450.554093   51512.737110          0.442793     332.731330

                            Order Date          Month          Sales  \
count                           185950  185950.000000  185950.000000
mean   2019-07-18 21:54:38.887550464       7.059140     185.490917
min              2019-01-01 03:07:00       1.000000       2.990000
25%              2019-04-16 21:05:15       4.000000      11.950000
50%              2019-07-17 20:40:30       7.000000      14.950000
75%              2019-10-26 08:14:00      10.000000     150.000000
max              2020-01-01 05:13:00      12.000000    3400.000000
std                              NaN       3.502996     332.919771

                Hour
count  185950.000000
mean       14.413305
min         0.000000
25%        11.000000
50%        15.000000
75%        19.000000
max        23.000000
std         5.423416
```

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

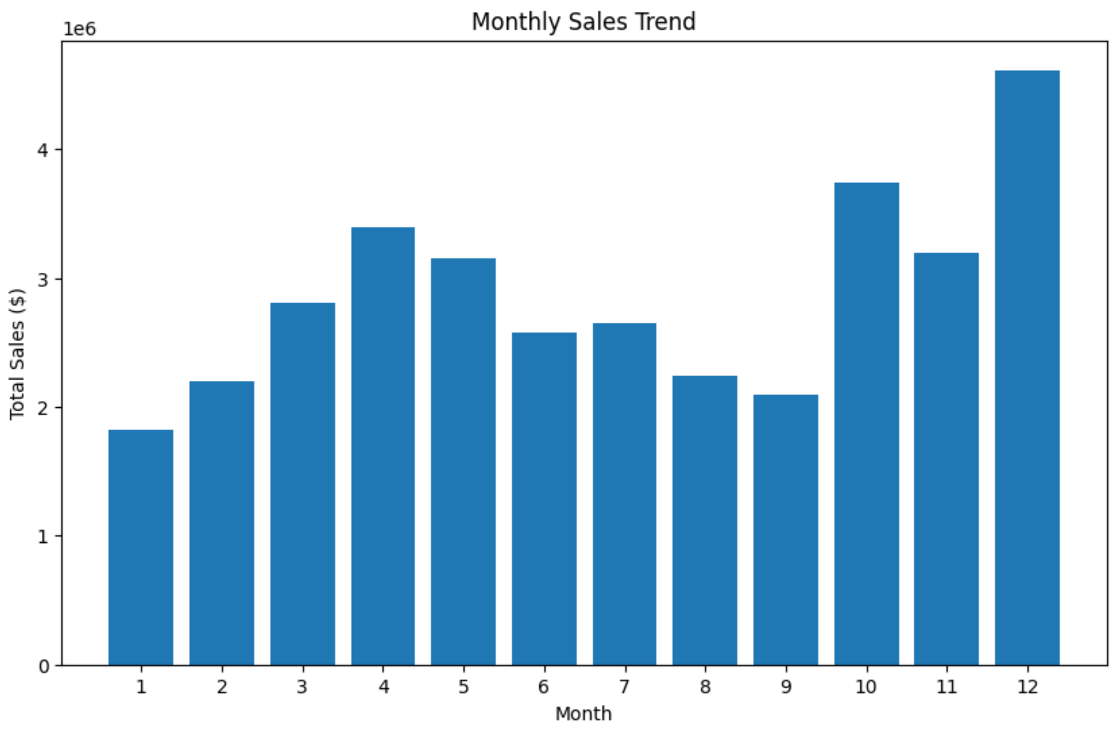
```
Unnamed: 0                   int64
Order ID                     int64
Product                     object
Quantity Ordered             int64
Price Each                 float64
Order Date          datetime64[ns]
Purchase Address            object
Month                        int32
Sales                      float64
City                        object
Hour                         int64
dtype: object
```

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

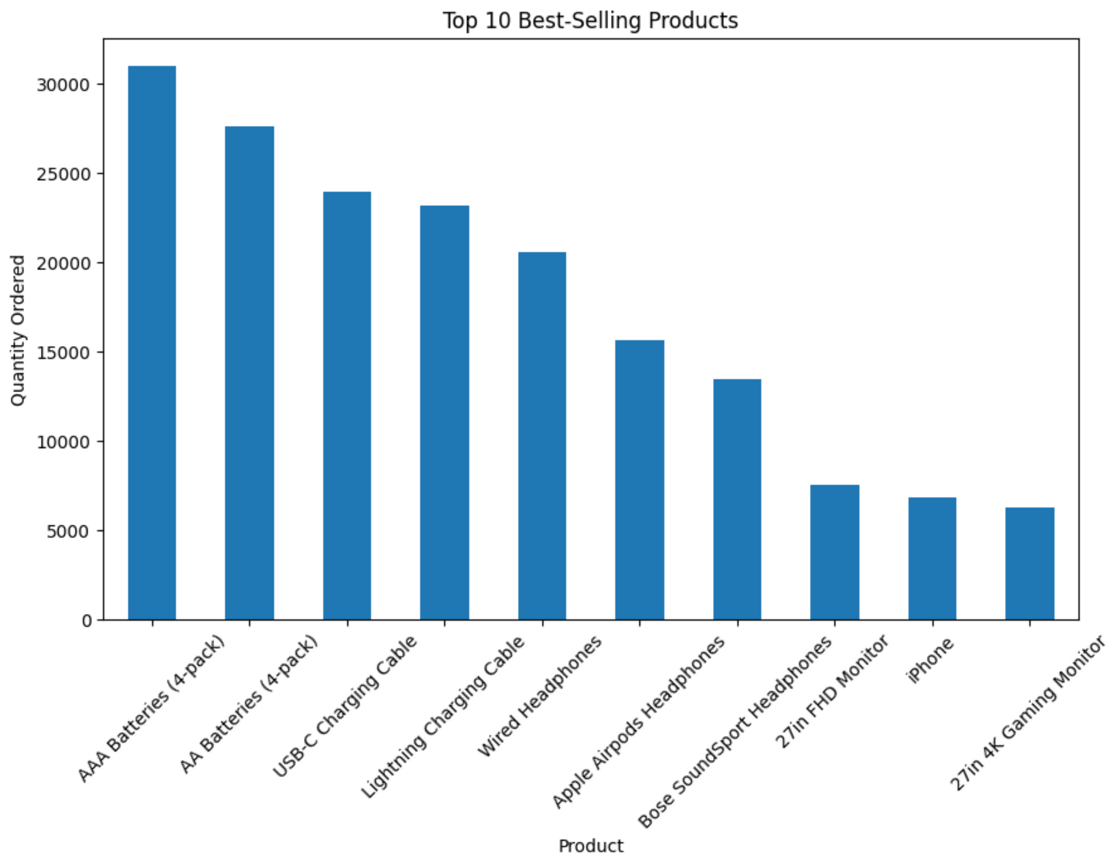


1.7 Visualizations

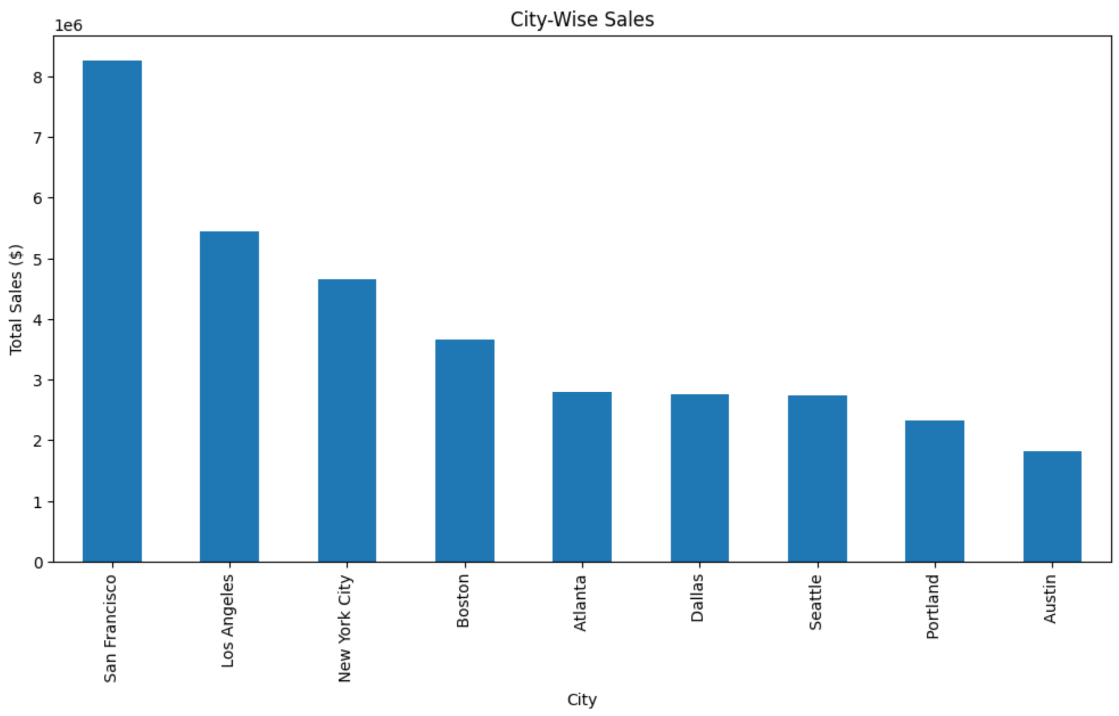
[15]:
```
# Monthly sales trend
monthly_sales = df.groupby('Month')['Sales'].sum()
months = range(1, 13)

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()
```



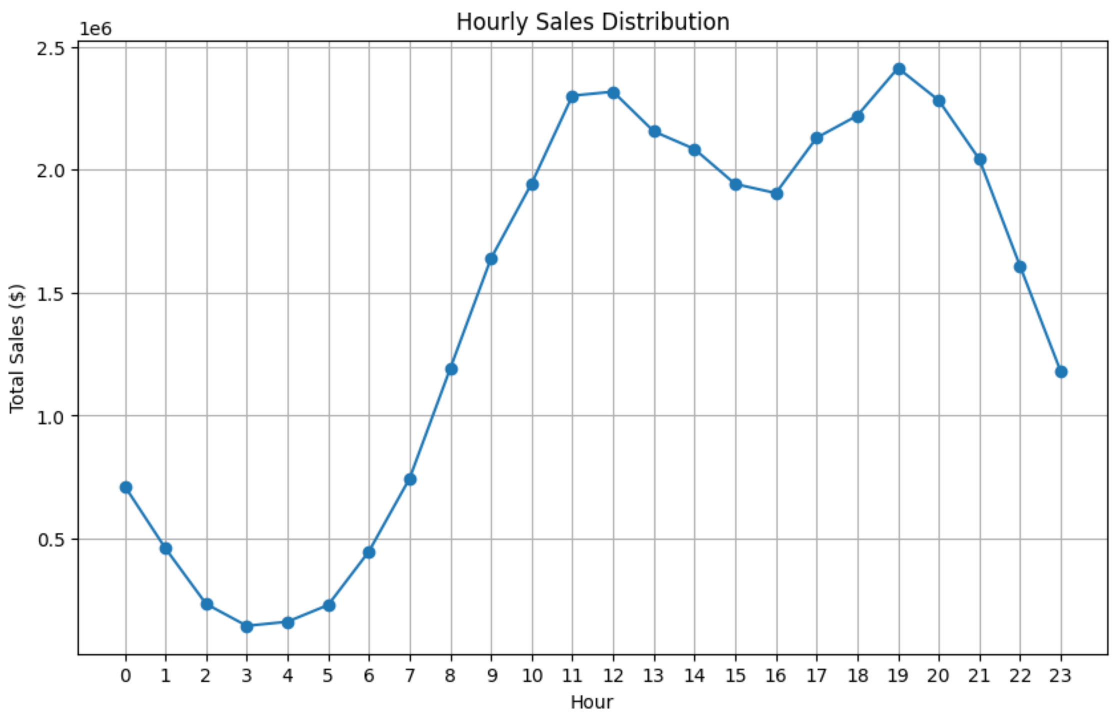
[16]:
```
# Top-selling products
top_products = df.groupby('Product')['Quantity Ordered'].sum().
 ↪sort_values(ascending=False).head(10)

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()
```



[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()
```



[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()
```



1.8 Conclusion and Recommendations

[19]:
```
print("Conclusion:")
print("1. There is a clear monthly sales trend, with peak sales occurring in␣
  ↪December.")
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␣
  ↪sales.")
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 December.
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 sales.
4. Hourly sales show that the highest sales occur around 12 PM and 7 PM.
```

[ ]:
```
```

[ ]:
```
```