Sales Analysis

In [1]:

```
# Import libraries
import pandas as pd
from os import listdir
import matplotlib.pyplot as plt
```

Merge 12 Months' Data into One File

In [2]:

```
# Check which headers will need to be added later
df = pd.read_csv('.\Sales_Data\Sales_April_2019.csv')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18383 entries, 0 to 18382
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Order ID          18324 non-null  object
 1   Product           18324 non-null  object
 2   Quantity Ordered  18324 non-null  object
 3   Price Each        18324 non-null  object
 4   Order Date        18324 non-null  object
 5   Purchase Address  18324 non-null  object
dtypes: object(6)
memory usage: 861.8+ KB
```

In [3]:

```
# Concat files
files = [file for file in listdir('./Sales_Data')]

sales_2019 = pd.DataFrame()

for file in files:
    df = pd.read_csv('./Sales_Data/'+file, header=None, skiprows=1)
    sales_2019 = pd.concat([sales_2019, df])

sales_2019.to_csv('Sales_2019.csv', index=False)
```

In [4]:

```
# Read updated DataFrame
sales_2019 = pd.read_csv('Sales_2019.csv')

# Add column names
sales_2019.columns = ['Order ID', 'Product', 'Quantity Ordered', 'Price Each', 'Ord
sales_2019.head()
```

Out[4]:

	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	NaN	NaN	NaN	NaN	NaN	NaN
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215
3	176560	Google Phone	1	600	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001

Clean the Data

In [5]:

```
# Check nulls
sales_2019.isnull().sum()
```

Out[5]:

```
Order ID            545
Product             545
Quantity Ordered    545
Price Each          545
Order Date          545
Purchase Address    545
dtype: int64
```

In [6]:

```
sales_2019[sales_2019.isnull().any(axis=1)]
```

Out[6]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
1	NaN	NaN	NaN	NaN	NaN	NaN
356	NaN	NaN	NaN	NaN	NaN	NaN
735	NaN	NaN	NaN	NaN	NaN	NaN
1433	NaN	NaN	NaN	NaN	NaN	NaN
1553	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...
185176	NaN	NaN	NaN	NaN	NaN	NaN
185438	NaN	NaN	NaN	NaN	NaN	NaN
186042	NaN	NaN	NaN	NaN	NaN	NaN
186548	NaN	NaN	NaN	NaN	NaN	NaN
186826	NaN	NaN	NaN	NaN	NaN	NaN

545 rows × 6 columns

In [7]:

```
# Drop nulls
sales_2019 = sales_2019.dropna()
sales_2019.isnull().sum()
```

Out[7]:
```
Order ID            0
Product             0
Quantity Ordered    0
Price Each          0
Order Date          0
Purchase Address    0
dtype: int64
```

In [8]:
```
# Check if rows have header names as values
sales_2019['Order ID'].str.contains('Order ID').sum()
```

Out[8]:
```
355
```

In [9]:
```
sales_2019[sales_2019['Order ID'].str.contains('Order ID')]
```

Out[9]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
519	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
1149	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
1155	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
2878	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
2893	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
...	...	...	...	...	...	...
185164	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
185551	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
186563	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
186632	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
186738	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address

355 rows × 6 columns

In [10]:
```
# Remove rows with header names as values
sales_2019 = sales_2019[~sales_2019['Order ID'].str.contains('Order ID')]
sales_2019.head()
```

Out[10]:

	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
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215
3	176560	Google Phone	1	600	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001

In [11]:
```
sales_2019['Order ID'].str.contains('Order ID').sum()
```

Out[11]:

```
0
```

Data Exploration

Which month was the best for sales? How much revenue was made that month?

In [12]:

```
# Change 'Order Date' column to datetime
sales_2019.dtypes
```

Out[12]:

```
Order ID            object
Product             object
Quantity Ordered    object
Price Each          object
Order Date          object
Purchase Address    object
dtype: object
```

In [13]:

```
sales_2019['Order Date'] = pd.to_datetime(sales_2019['Order Date'], format='%m/%d/%
sales_2019.head()
```

Out[13]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	176558	USB-C Charging Cable	2	11.95	2019-04-19 08:46:00	917 1st St, Dallas, TX 75001
2	176559	Bose SoundSport Headphones	1	99.99	2019-04-07 22:30:00	682 Chestnut St, Boston, MA 02215
3	176560	Google Phone	1	600	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001
4	176560	Wired Headphones	1	11.99	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001
5	176561	Wired Headphones	1	11.99	2019-04-30 09:27:00	333 8th St, Los Angeles, CA 90001

In [14]:

```
sales_2019.dtypes
```

Out[14]:

```
Order ID                    object
Product                     object
Quantity Ordered            object
Price Each                  object
Order Date          datetime64[ns]
Purchase Address            object
dtype: object
```

In [15]:

```
# Add 'Month' column
sales_2019['Month'] = sales_2019['Order Date'].dt.strftime('%B')
sales_2019.head()
```

Out[15]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month
0	176558	USB-C Charging Cable	2	11.95	2019-04-19 08:46:00	917 1st St, Dallas, TX 75001	April
2	176559	Bose SoundSport Headphones	1	99.99	2019-04-07 22:30:00	682 Chestnut St, Boston, MA 02215	April
3	176560	Google Phone	1	600	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	April
4	176560	Wired Headphones	1	11.99	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	April
5	176561	Wired Headphones	1	11.99	2019-04-30 09:27:00	333 8th St, Los Angeles, CA 90001	April

In [16]:

```
# Change 'Price Each' and 'Quantity Ordered' columns to numeric
sales_2019['Price Each'] = pd.to_numeric(sales_2019['Price Each'])
sales_2019['Quantity Ordered'] = pd.to_numeric(sales_2019['Quantity Ordered'])

sales_2019.dtypes
```

Out[16]:

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

In [17]:

```
# Add 'Revenue' column
sales_2019['Revenue'] = sales_2019['Quantity Ordered'] * sales_2019['Price Each']
sales_2019.head()
```

Out[17]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Revenue
0	176558	USB-C Charging Cable	2	11.95	2019-04-19 08:46:00	917 1st St, Dallas, TX 75001	April	23.90
2	176559	Bose SoundSport Headphones	1	99.99	2019-04-07 22:30:00	682 Chestnut St, Boston, MA 02215	April	99.99
3	176560	Google Phone	1	600.00	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	April	600.00
4	176560	Wired Headphones	1	11.99	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	April	11.99
5	176561	Wired Headphones	1	11.99	2019-04-30 09:27:00	333 8th St, Los Angeles, CA 90001	April	11.99

In [18]:

```
revenue_per_month = pd.DataFrame(sales_2019.groupby('Month')['Revenue'].sum())
revenue_per_month = revenue_per_month.reset_index(drop=False)

revenue_per_month
```

Out[18]:

	Month	Revenue
0	April	3390670.24
1	August	2244467.88
2	December	4613443.34
3	February	2202022.42
4	January	1822256.73
5	July	2647775.76
6	June	2577802.26
7	March	2807100.38
8	May	3152606.75
9	November	3199603.20
10	October	3736726.88
11	September	2097560.13

In [19]:

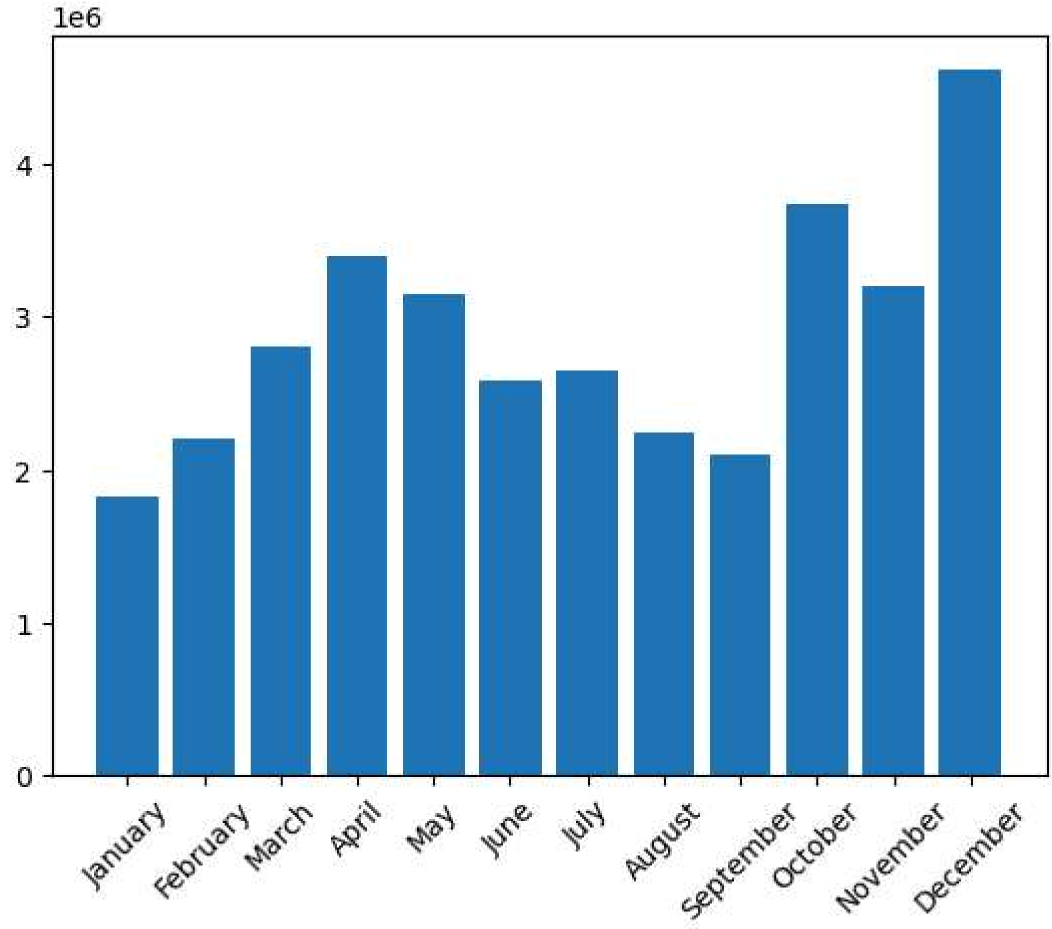
```
# Order by month
dates_in_order = pd.date_range(start='2022-01-01', end='2022-12-01', freq='MS')
months_in_order = dates_in_order.map(lambda x: x.month_name()).to_list()

revenue_per_month['Month'] = pd.Categorical(
    revenue_per_month['Month'],
    categories=months_in_order,
    ordered=True
)

revenue_per_month = revenue_per_month.sort_values(by=['Month'])

revenue_per_month
```

Out[19]:

	Month	Revenue
4	January	1822256.73
3	February	2202022.42
7	March	2807100.38
0	April	3390670.24
8	May	3152606.75
6	June	2577802.26
5	July	2647775.76
1	August	2244467.88
11	September	2097560.13
10	October	3736726.88
9	November	3199603.20
2	December	4613443.34

In [21]:

```
plt.bar(revenue_per_month['Month'], revenue_per_month['Revenue'])
plt.xticks(rotation=45)
plt.show()
```


1e6
0
1
2
3
4
January
February
March
April
May
June
July
August
September
October
November
December