Annual Sales Report Analysis
By Seghosime Joshua

In [31]:
```
# import the libraries
import pandas as pd
import numpy as np
import os
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

In [4]:
```
# load directly from folder dir
path = r"C:\Users\user\Downloads\Lasop_ebooks\excel_data_analysis\datasets\Sales_Da
# loop through the folder merge data
files = [file for file in os.listdir(path)]
# print the files in path
files
```

Out[4]:
```
['Sales_April_2019.csv',
 'Sales_August_2019.csv',
 'Sales_December_2019.csv',
 'Sales_February_2019.csv',
 'Sales_January_2019.csv',
 'Sales_July_2019.csv',
 'Sales_June_2019.csv',
 'Sales_March_2019.csv',
 'Sales_May_2019.csv',
 'Sales_November_2019.csv',
 'Sales_October_2019.csv',
 'Sales_September_2019.csv']
```

In [5]:
```
# concatenate all dataset into one dataframe
df = pd.DataFrame()
for file in files[1:]:
 data = pd.read_csv(path + '/' + file)
 df = pd.concat([df, data])
# check
df.shape
```

Out[5]:
```
(168467, 6)
```

In [6]:
```
# view all dataframes
df.head()
```

Out[6]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	236670	Wired Headphones	2	11.99	08/31/19 22:21	359 Spruce St, Seattle, WA 98101
1	236671	Bose SoundSport Headphones	1	99.99	08/15/19 15:11	492 Ridge St, Dallas, TX 75001
2	236672	iPhone	1	700.0	08/06/19 14:40	149 7th St, Portland, OR 97035
3	236673	AA Batteries (4-pack)	2	3.84	08/29/19 20:59	631 2nd St, Los Angeles, CA 90001
4	236674	AA Batteries (4-pack)	2	3.84	08/15/19 19:53	736 14th St, New York City, NY 10001

Data Cleaning

In [7]:
```
# general data info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 168467 entries, 0 to 11685
Data columns (total 6 columns):
 #   Column            Non-Null Count   Dtype 
---  ------            --------------   ----- 
 0   Order ID          167981 non-null  object
 1   Product           167981 non-null  object
 2   Quantity Ordered  167981 non-null  object
 3   Price Each        167981 non-null  object
 4   Order Date        167981 non-null  object
 5   Purchase Address  167981 non-null  object
dtypes: object(6)
memory usage: 9.0+ MB
```

In [8]:
```
# check for misimg values
df.isnull().sum()
```

Out[8]:
```
Order ID            486
Product             486
Quantity Ordered    486
Price Each          486
Order Date          486
Purchase Address    486
dtype: int64
```

In [9]:
```
# drop all NaN or missing values
df.dropna(how = "all", inplace=True)
```

In [11]:
```
# check for duplicates
df.duplicated().sum()
```

Out[11]: 561

In [13]:
```
# drop duplicates immediately
df.drop_duplicates(inplace=True)
```

In [16]:
```
# num of rows and cols
df.shape
```

Out[16]: (167420, 6)

Sales By Month

In [19]:
```
# create a new column and extract month from Order Date
df['month'] = df['Order Date'].apply(lambda date : date.split('/')[0])
# Check the updatated dataframe
df.head()
```

Out[19]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	month
0	236670	Wired Headphones	2	11.99	08/31/19 22:21	359 Spruce St, Seattle, WA 98101	08
1	236671	Bose SoundSport Headphones	1	99.99	08/15/19 15:11	492 Ridge St, Dallas, TX 75001	08
2	236672	iPhone	1	700.0	08/06/19 14:40	149 7th St, Portland, OR 97035	08
3	236673	AA Batteries (4-pack)	2	3.84	08/29/19 20:59	631 2nd St, Los Angeles, CA 90001	08
4	236674	AA Batteries (4-pack)	2	3.84	08/15/19 19:53	736 14th St, New York City, NY 10001	08

In [21]:
```
#check unique values of month feature
df['month'].unique().tolist()
```

Out[21]:
```
['08',
 'Order Date',
 '09',
 '12',
 '01',
 '02',
 '03',
 '07',
 '06',
 '04',
 '05',
 '11',
 '10']
```

In [ ]:
```
# Applying filter to remove invalid entry
filter = df['month']=='Order Date'
df = df[~filter]
```

In [25]:
```
# Convert to month feature to integer
df['month'] = df['month'].astype('int')
df['month'].dtype
```

Out[25]:
```
dtype('int32')
```

In [24]:
```
# converting 'Quantity Ordered' feature to int
df['Quantity Ordered']=df['Quantity Ordered'].astype('int')
df['Quantity Ordered'].dtype
```

Out[24]:
```
dtype('int32')
```

In [26]:
```
# converting 'Price Each' feature to float
df['Price Each']=df['Price Each'].astype('float')
df['Price Each'].dtype
```

Out[26]:
```
dtype('float64')
```

In [27]:
```
# Add a new col to dataframe
df['Total Sales'] = df['Quantity Ordered'] * df['Price Each'].round(2)
df.head()
```

Out[27]:

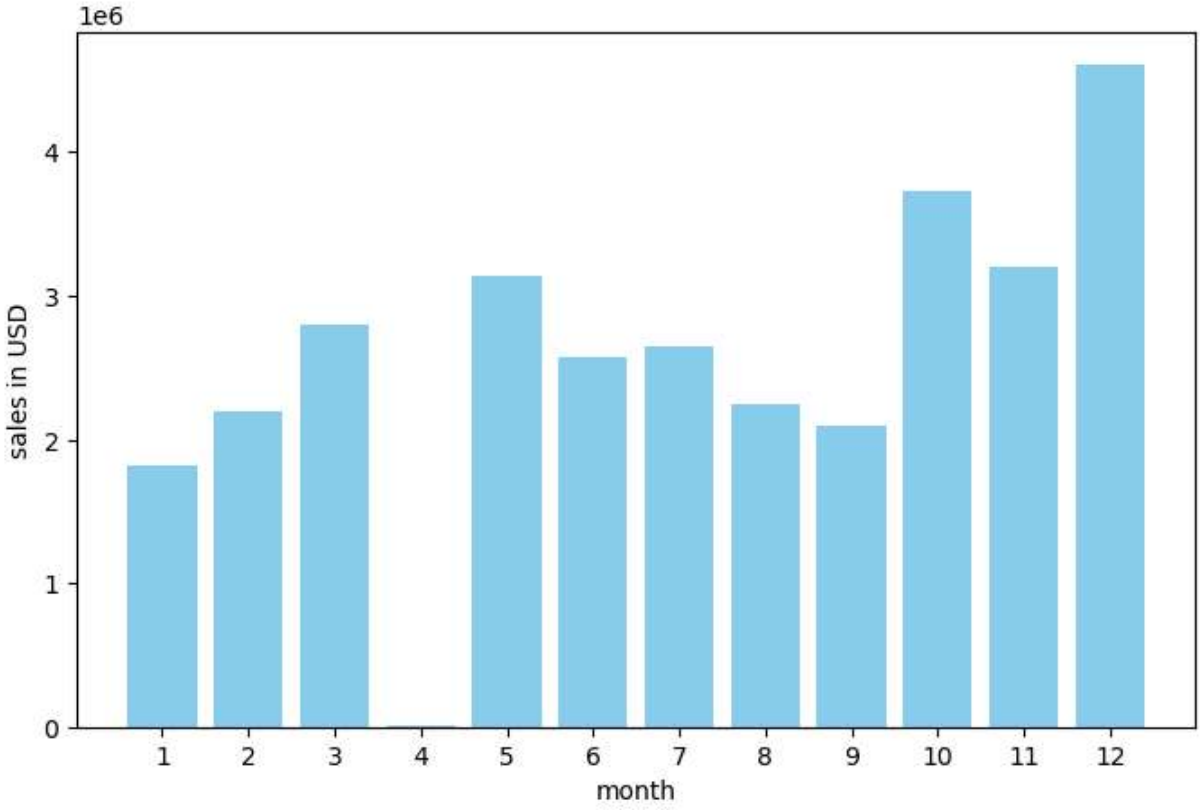
	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	month	Total Sales
0	236670	Wired Headphones	2	11.99	08/31/19 22:21	359 Spruce St, Seattle, WA 98101	8	23.98
1	236671	Bose SoundSport Headphones	1	99.99	08/15/19 15:11	492 Ridge St, Dallas, TX 75001	8	99.99
2	236672	iPhone	1	700.00	08/06/19 14:40	149 7th St, Portland, OR 97035	8	700.00
3	236673	AA Batteries (4-pack)	2	3.84	08/29/19 20:59	631 2nd St, Los Angeles, CA 90001	8	7.68
4	236674	AA Batteries (4-pack)	2	3.84	08/15/19 19:53	736 14th St, New York City, NY 10001	8	7.68

Sales By Month Chart

In [29]:
```
#Group by data on month feature
dat =df.groupby('month')['Total Sales'].sum().sort_values(ascending=False)
dat
```

Out[29]:
```
month
12    4608295.70
10    3734777.86
11    3197875.05
5     3140056.94
3     2804973.35
7     2646461.32
6     2576280.15
8     2241083.37
2     2200078.08
9     2094465.69
1     1821413.16
4        5170.42
Name: Total Sales, dtype: float64
```

In [33]:
```
# Creating a bar chart
plt.figure(figsize=(8,5))
plt.bar(dat.index, dat, color="skyblue")
plt.xticks(dat.index)
plt.xlabel('month')
plt.ylabel('sales in USD')
plt.show()
# Conclusion:- December month has the best sales
```



Sales By City

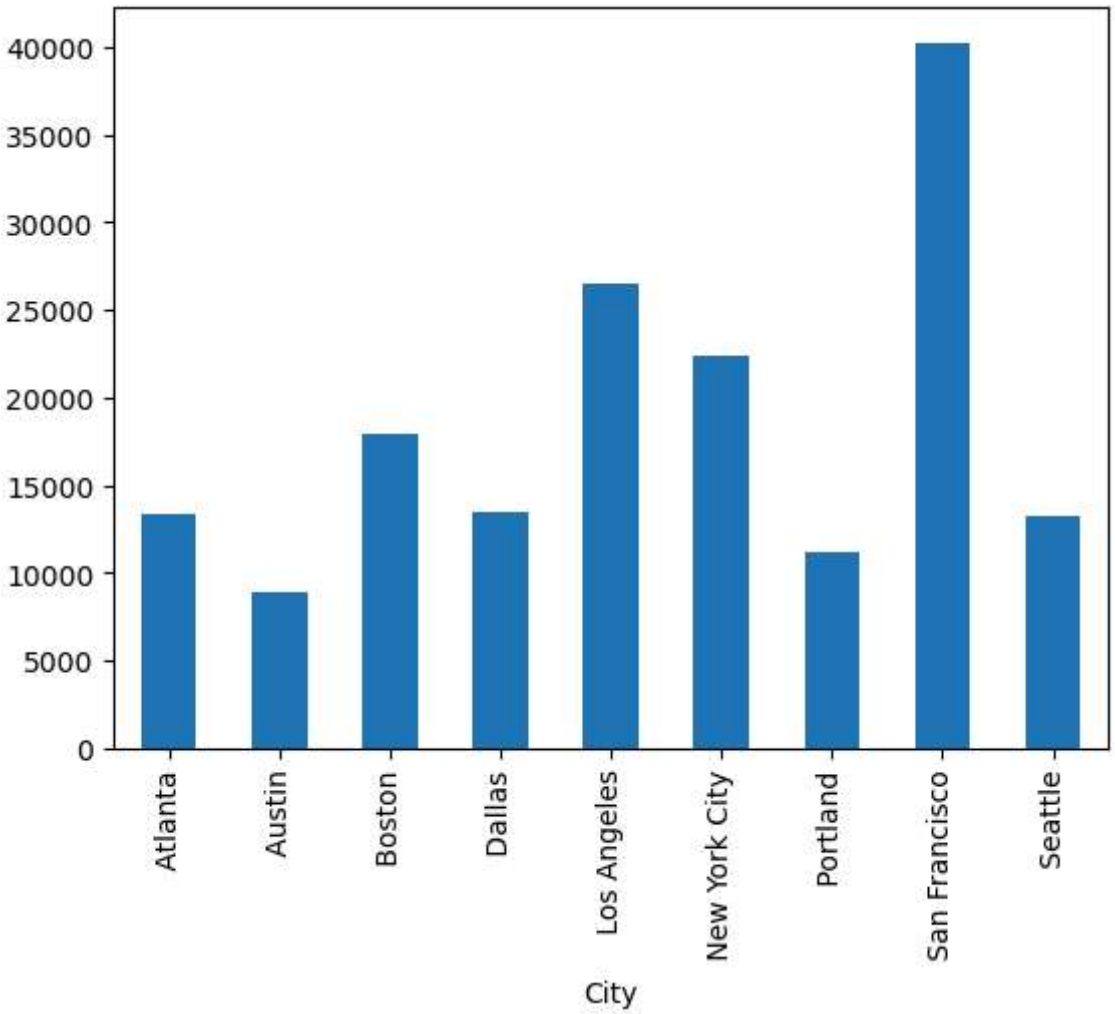
In [34]:
```
# Extract city form the purchase address
df['City'] = df['Purchase Address'].apply(lambda city : city.split(',')[-2])
df.head()
```

Out[34]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	month	Total Sales	City
0	236670	Wired Headphones	2	11.99	08/31/19 22:21	359 Spruce St, Seattle, WA 98101	8	23.98	Seattle
1	236671	Bose SoundSport Headphones	1	99.99	08/15/19 15:11	492 Ridge St, Dallas, TX 75001	8	99.99	Dallas
2	236672	iPhone	1	700.00	08/06/19 14:40	149 7th St, Portland, OR 97035	8	700.00	Portland
3	236673	AA Batteries (4-pack)	2	3.84	08/29/19 20:59	631 2nd St, Los Angeles, CA 90001	8	7.68	Los Angeles
4	236674	AA Batteries (4-pack)	2	3.84	08/15/19 19:53	736 14th St, New York City, NY 10001	8	7.68	New York City

In [ ]:
```
# Count and plot by city
df.groupby('City')['City'].count().plot.bar()
# Conclusion:- San Francisco has maximum orders
```

Out[ ]:
```
<Axes: xlabel='City'>
```


40000
35000
30000
25000
20000
15000
10000
5000
0
Atlanta
Austin
Boston
Dallas
Los Angeles
New York City
Portland
San Francisco
Seattle
City

In [ ]: