salesanalysis

February 2, 2024

1 Sales Analysis

Import necessary libraries

```
[1]: import os
     import pandas as pd
```

```
C:\Users\Hxtreme\AppData\Local\Temp\ipykernel_19012\3571106454.py:2:
DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of
pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better
interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd
```

Merge data from each month into one CSV

```
[2]: df = pd.read_csv("D:\\data analytics\\Python for data analytics\\Sales␣
      ↪Analysis\\SalesAnalysis\\Sales_Data\\Sales_April_2019.csv")

     files = [file for file in os.listdir("D:\\data analytics\\Python for data␣
      ↪analytics\\Sales Analysis\\SalesAnalysis\\Sales_Data")]
     all_months_data = pd.DataFrame()
     for file in files:
         df = pd.read_csv("D:\\data analytics\\Python for data analytics\\Sales␣
      ↪Analysis\\SalesAnalysis\\Sales_Data\\"+file)
         all_months_data = pd.concat([all_months_data,df])

     all_months_data.to_csv("all_data.csv",index=False)
```

Read in updated dataframe

```
[3]: all_data = pd.read_csv("all_data.csv")
     all_data.head()
```

```
[3]:   Order ID                     Product Quantity Ordered Price Each  \
     0   176558        USB-C Charging Cable                2      11.95
     1   176559  Bose SoundSport Headphones                1      99.99
     2   176560                Google Phone                1        600
     3   176560            Wired Headphones                1      11.99
     4   176561            Wired Headphones                1      11.99

             Order Date                      Purchase Address
     0   4/19/2019 8:46          917 1st St, Dallas, TX 75001
     1   4/7/2019 22:30     682 Chestnut St, Boston, MA 02215
     2  4/12/2019 14:38  669 Spruce St, Los Angeles, CA 90001
     3  4/12/2019 14:38  669 Spruce St, Los Angeles, CA 90001
     4   4/30/2019 9:27     333 8th St, Los Angeles, CA 90001
```

1.0.1 Clean up the data!

The first step in this is figuring out what we need to clean. I have found in practice, that you find things you need to clean as you perform operations and get errors. Based on the error, you decide how you should go about cleaning the data

Drop rows of NAN

```
[4]: # Find NAN
     nan_df = all_data[all_data.isna().any(axis=1)]
     display(nan_df.head())

     all_data = all_data.dropna(how='all')
     all_data.head()
```

```
         Order ID Product Quantity Ordered Price Each Order Date Purchase Address
     355       NaN     NaN              NaN        NaN        NaN              NaN
     734       NaN     NaN              NaN        NaN        NaN              NaN
     1432      NaN     NaN              NaN        NaN        NaN              NaN
     1552      NaN     NaN              NaN        NaN        NaN              NaN
     1570      NaN     NaN              NaN        NaN        NaN              NaN
```

```
[4]:   Order ID                     Product Quantity Ordered Price Each  \
     0   176558        USB-C Charging Cable                2      11.95
     1   176559  Bose SoundSport Headphones                1      99.99
     2   176560                Google Phone                1        600
     3   176560            Wired Headphones                1      11.99
     4   176561            Wired Headphones                1      11.99

             Order Date                      Purchase Address
     0   4/19/2019 8:46          917 1st St, Dallas, TX 75001
     1   4/7/2019 22:30     682 Chestnut St, Boston, MA 02215
     2  4/12/2019 14:38  669 Spruce St, Los Angeles, CA 90001
     3  4/12/2019 14:38  669 Spruce St, Los Angeles, CA 90001
```

```
4   4/30/2019 9:27     333 8th St, Los Angeles, CA 90001
```

Get rid of text in order date column

```
[5]: all_data = all_data[all_data['Order Date'].str[0:2]!='Or']
```

Make columns correct type

```
[6]: all_data['Quantity Ordered'] = pd.to_numeric(all_data['Quantity Ordered'])
     all_data['Price Each'] = pd.to_numeric(all_data['Price Each'])
```

1.0.2 Augment data with additional columns

Add month column

```
[7]: all_data['Month'] = all_data['Order Date'].str[0:1]
     all_data['Month'] = all_data['Month'].astype('int32')
     all_data.head()
```

```
[7]:   Order ID                     Product  Quantity Ordered  Price Each  \
     0   176558        USB-C Charging Cable                 2       11.95
     1   176559  Bose SoundSport Headphones                 1       99.99
     2   176560                Google Phone                 1      600.00
     3   176560            Wired Headphones                 1       11.99
     4   176561            Wired Headphones                 1       11.99

              Order Date                      Purchase Address  Month
     0   4/19/2019 8:46          917 1st St, Dallas, TX 75001      4
     1   4/7/2019 22:30     682 Chestnut St, Boston, MA 02215      4
     2  4/12/2019 14:38  669 Spruce St, Los Angeles, CA 90001      4
     3  4/12/2019 14:38  669 Spruce St, Los Angeles, CA 90001      4
     4   4/30/2019 9:27     333 8th St, Los Angeles, CA 90001      4
```

Add month column (alternative method)

```
[8]: # all_data['Month 2'] = pd.to_datetime(all_data['Order Date']).dt.month
     # all_data.head()
```

Add city column

```
[9]: def get_city(address):
         return address.split(",")[1].strip(" ")

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

     all_data['City'] = all_data['Purchase Address'].apply(lambda x: f"{get_city(x)}␣
      ↪ ({get_state(x)})")
     all_data.head()
```

```
[9]:   Order ID                     Product  Quantity Ordered  Price Each  \
    0    176558        USB-C Charging Cable                 2       11.95
    1    176559  Bose SoundSport Headphones                 1       99.99
    2    176560                Google Phone                 1      600.00
    3    176560            Wired Headphones                 1       11.99
    4    176561            Wired Headphones                 1       11.99

             Order Date                      Purchase Address  Month  \
    0   4/19/2019 8:46          917 1st St, Dallas, TX 75001      4
    1   4/7/2019 22:30     682 Chestnut St, Boston, MA 02215      4
    2  4/12/2019 14:38  669 Spruce St, Los Angeles, CA 90001      4
    3  4/12/2019 14:38  669 Spruce St, Los Angeles, CA 90001      4
    4   4/30/2019 9:27     333 8th St, Los Angeles, CA 90001      4

                    City
    0        Dallas  (TX)
    1        Boston  (MA)
    2   Los Angeles  (CA)
    3   Los Angeles  (CA)
    4   Los Angeles  (CA)
```

1.1 Data Exploration!

Question 1: What was the best month for sales? How much was earned that month?

```
[10]: all_data['Sales'] = all_data['Quantity Ordered'].astype('int') *␣
       ↪all_data['Price Each'].astype('float')
```

```
[11]: all_data.groupby(['Month']).sum()
```

```
[11]:                                                  Order ID  \
    Month
    0       2366702366712366722366732366742366752366762366…
    1       2956652956662956672956682956692956702956712956…
    4       1765581765591765601765601765611765621765631765…
    5       1769781775511777781777781790761790761791341800…

                                                     Product  Quantity Ordered  \
    Month
    0       Wired HeadphonesBose SoundSport HeadphonesiPho…            117896
    1       Macbook Pro LaptopLG Washing MachineUSB-C Char…             70615
    4       USB-C Charging CableBose SoundSport Headphones…             20539
    5       Apple Airpods Headphones27in FHD MonitoriPhone…                29

             Price Each                                       Order Date  \
    Month
    0       19431500.72  08/31/19 22:2108/15/19 15:1108/06/19 14:4008/2…
    1       11484570.92  12/30/19 00:0112/29/19 07:0312/12/19 18:2112/2…
```

```
4       3362503.59  4/19/2019 8:464/7/2019 22:304/12/2019 14:384/1…
5         10555.45  5/1/2019 3:295/1/2019 0:135/1/2019 0:485/1/201…

                                       Purchase Address  \
Month
0      359 Spruce St, Seattle, WA 98101492 Ridge St, …
1      136 Church St, New York City, NY 10001562 2nd …
4      917 1st St, Dallas, TX 75001682 Chestnut St, B…
5      589 Lake St, Portland, OR 97035615 Lincoln St,…

                                                   City        Sales
Month
0      Seattle  (WA)Dallas  (TX)Portland  (OR)Los Ang…  19546203.44
1      New York City  (NY)New York City  (NY)New York…  11549773.42
4      Dallas  (TX)Boston  (MA)Los Angeles  (CA)Los A…   3385499.82
5      Portland  (OR)San Francisco  (CA)Boston  (MA)B…     10559.29
```

[ ]:
```python
import matplotlib.pyplot as plt

months = range(1, 13)
sales_by_month = all_data.groupby(['Month']).sum()['Sales']

plt.bar(months, sales_by_month)
plt.xticks(months)
plt.ylabel('Sales in USD ($)')
plt.xlabel('Month')
plt.show()
```

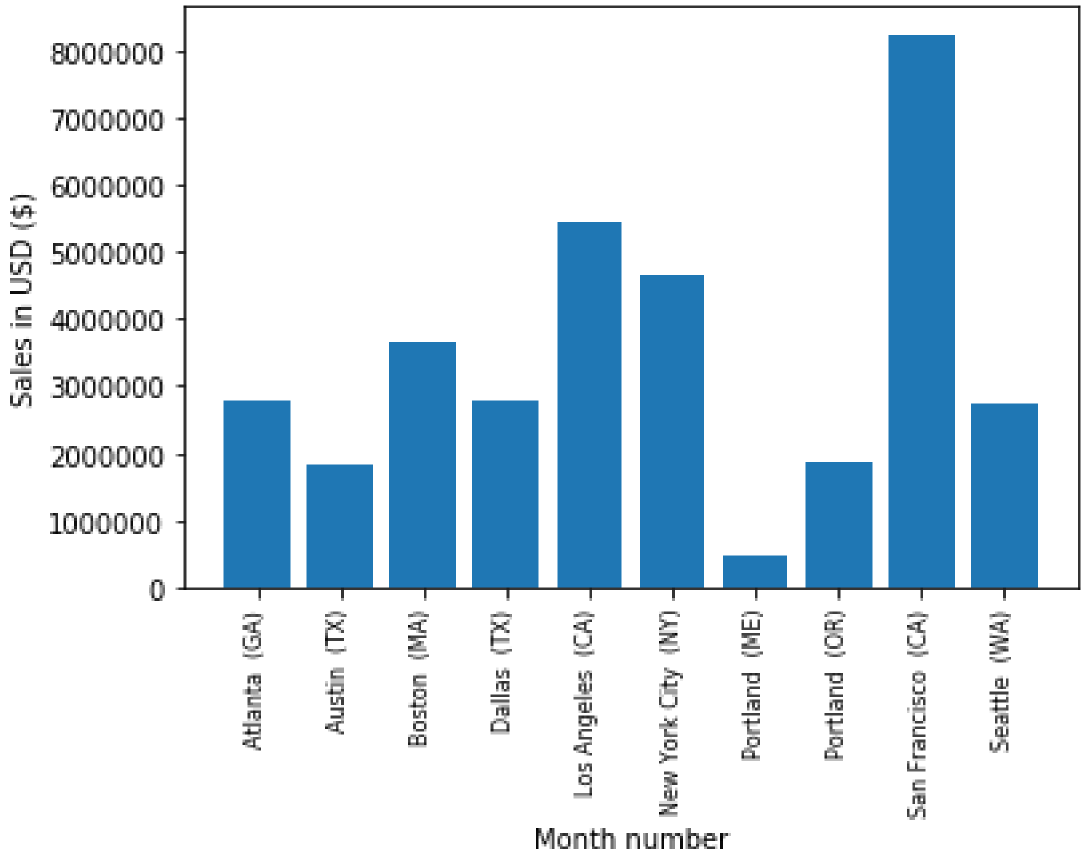
Question 2: What city sold the most product?

[ ]:
```python
all_data.groupby(['City']).sum()
```

[ ]:
```
                     Quantity Ordered    Price Each   Month         Sales
City
Atlanta  (GA)                   16602  2.779908e+06  104794  2.795499e+06
Austin  (TX)                    11153  1.809874e+06   69829  1.819582e+06
Boston  (MA)                    22528  3.637410e+06  141112  3.661642e+06
Dallas  (TX)                    16730  2.752628e+06  104620  2.767975e+06
Los Angeles  (CA)               33289  5.421435e+06  208325  5.452571e+06
New York City  (NY)             27932  4.635371e+06  175741  4.664317e+06
Portland  (ME)                   2750  4.471893e+05   17144  4.497583e+05
Portland  (OR)                  11303  1.860558e+06   70621  1.870732e+06
San Francisco  (CA)             50239  8.211462e+06  315520  8.262204e+06
Seattle  (WA)                   16553  2.733296e+06  104941  2.747755e+06
```

[ ]:
```python
import matplotlib.pyplot as plt
```

```
keys = [city for city, df in all_data.groupby(['City'])]

plt.bar(keys,all_data.groupby(['City']).sum()['Sales'])
plt.ylabel('Sales in USD ($)')
plt.xlabel('Month number')
plt.xticks(keys, rotation='vertical', size=8)
plt.show()
```



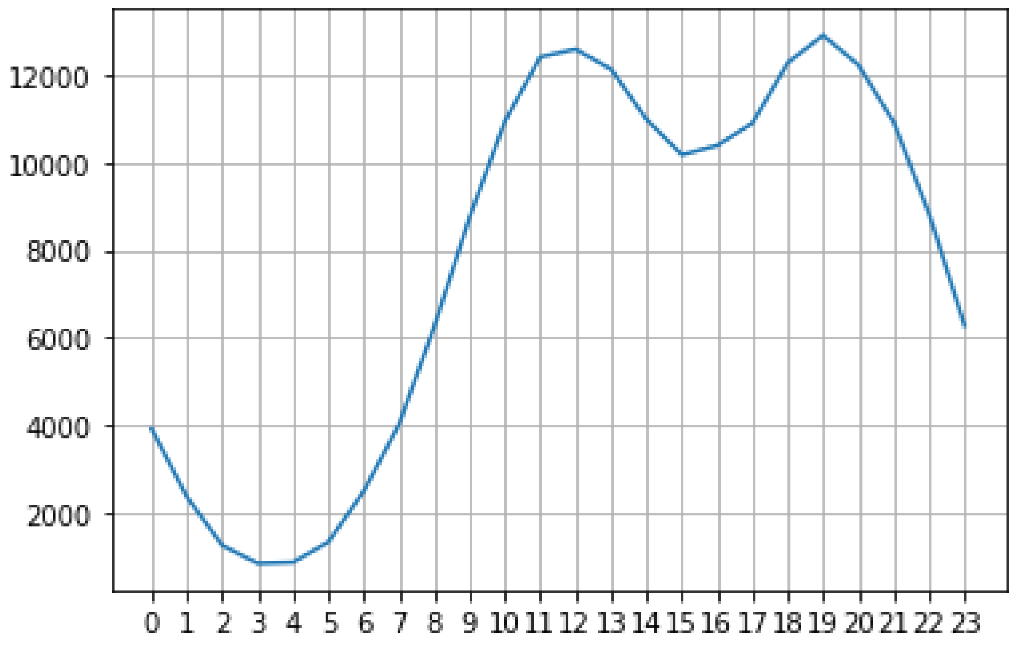
Question 3: What time should we display advertisements to maximize likelihood of customer's buying product?

[ ]:
```
# Add hour column
all_data['Hour'] = pd.to_datetime(all_data['Order Date']).dt.hour
all_data['Minute'] = pd.to_datetime(all_data['Order Date']).dt.minute
all_data['Count'] = 1
all_data.head()
```

[ ]:
```
  Order ID                     Product  Quantity Ordered  Price Each  \
0   176558        USB-C Charging Cable                 2       11.95
2   176559  Bose SoundSport Headphones                 1       99.99
```

```
3   176560            Google Phone                 1      600.00
4   176560        Wired Headphones                 1       11.99
5   176561        Wired Headphones                 1       11.99

       Order Date                      Purchase Address  Month  \
0  04/19/19 08:46          917 1st St, Dallas, TX 75001      4
2  04/07/19 22:30     682 Chestnut St, Boston, MA 02215      4
3  04/12/19 14:38  669 Spruce St, Los Angeles, CA 90001      4
4  04/12/19 14:38  669 Spruce St, Los Angeles, CA 90001      4
5  04/30/19 09:27     333 8th St, Los Angeles, CA 90001      4

                City   Sales  Hour  Minute  Count
0        Dallas (TX)   23.90     8      46      1
2        Boston (MA)   99.99    22      30      1
3   Los Angeles (CA)  600.00    14      38      1
4   Los Angeles (CA)   11.99    14      38      1
5   Los Angeles (CA)   11.99     9      27      1
```

[ ]:
```python
keys = [pair for pair, df in all_data.groupby(['Hour'])]

plt.plot(keys, all_data.groupby(['Hour']).count()['Count'])
plt.xticks(keys)
plt.grid()
plt.show()

# My recommendation is slightly before 11am or 7pm
```



Question 4: What products are most often sold together?

```
[ ]: # https://stackoverflow.com/questions/43348194/
      ↪pandas-select-rows-if-id-appear-several-time
     df = all_data[all_data['Order ID'].duplicated(keep=False)]

     # Referenced: https://stackoverflow.com/questions/27298178/
      ↪concatenate-strings-from-several-rows-using-pandas-groupby
     df['Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x: ','.
      ↪join(x))
     df2 = df[['Order ID', 'Grouped']].drop_duplicates()
```

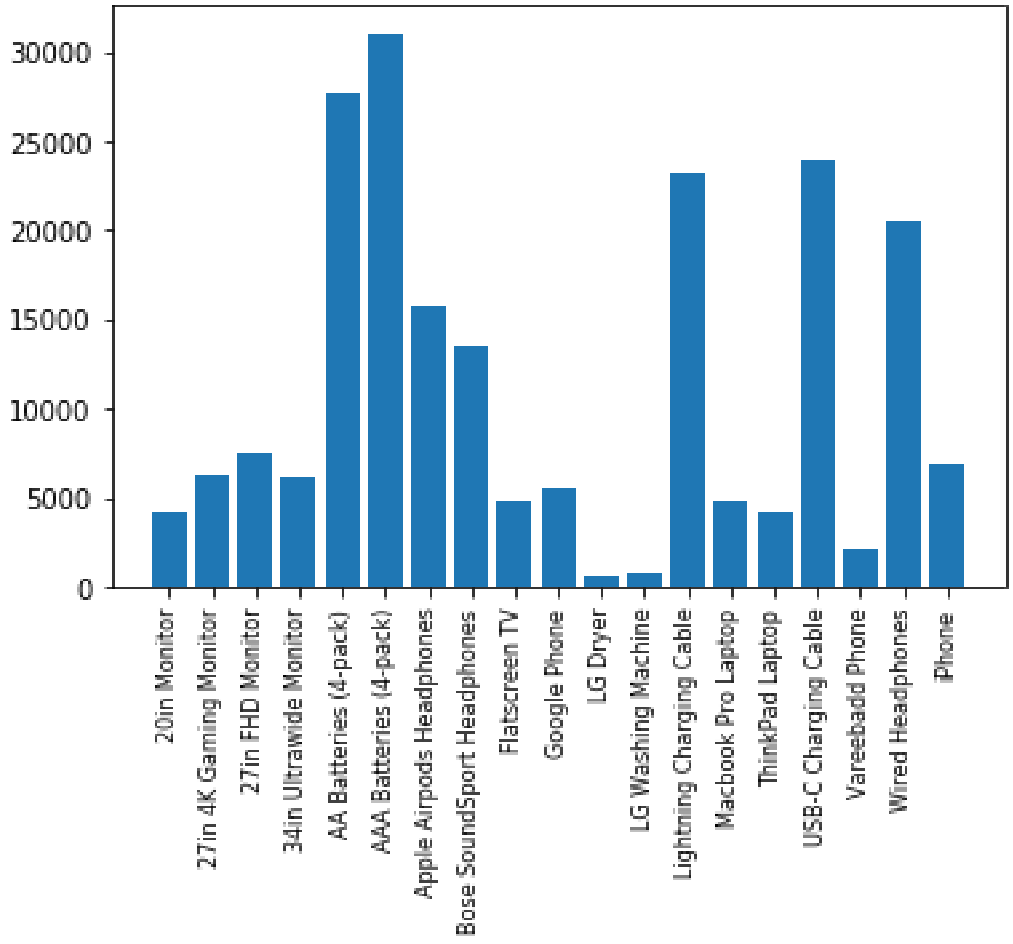
```
C:\Users\keith\Anaconda3\lib\site-packages\ipykernel_launcher.py:5:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  """
```

```
[ ]: # Referenced: https://stackoverflow.com/questions/52195887/
      ↪counting-unique-pairs-of-numbers-into-a-python-dictionary
     from itertools import combinations
     from collections import Counter

     count = Counter()

     for row in df2['Grouped']:
         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
```

What product sold the most? Why do you think it sold the most?

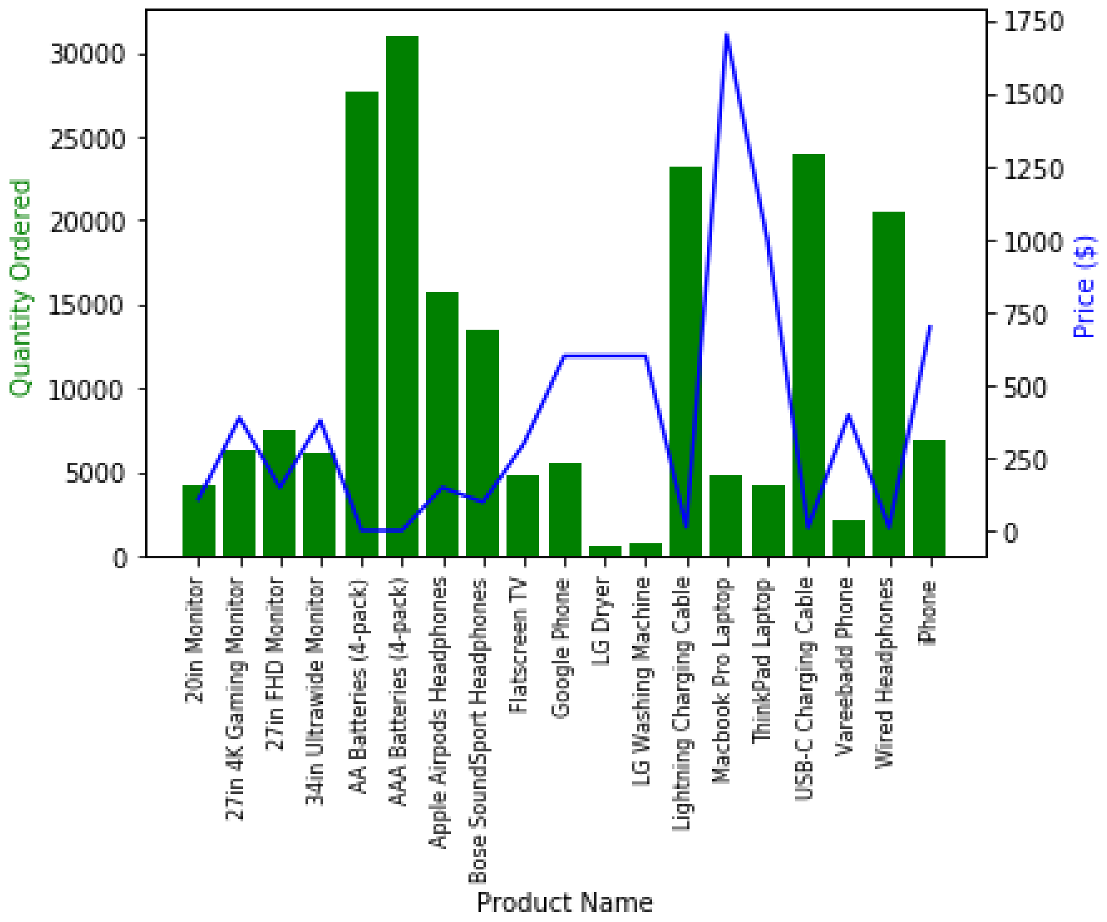
```
[ ]: product_group = all_data.groupby('Product')
     quantity_ordered = product_group.sum()['Quantity Ordered']

     keys = [pair for pair, df in product_group]
     plt.bar(keys, quantity_ordered)
     plt.xticks(keys, rotation='vertical', size=8)
     plt.show()
```



```
[ ]: prices = all_data.groupby('Product').mean()['Price Each']

     fig, ax1 = plt.subplots()

     ax2 = ax1.twinx()
     ax1.bar(keys, quantity_ordered, color='g')
     ax2.plot(keys, prices, color='b')
```

```
ax1.set_xlabel('Product Name')
ax1.set_ylabel('Quantity Ordered', color='g')
ax2.set_ylabel('Price ($)', color='b')
ax1.set_xticklabels(keys, rotation='vertical', size=8)

fig.show()
```

```
C:\Users\keith\Anaconda3\lib\site-packages\ipykernel_launcher.py:16:
UserWarning: Matplotlib is currently using
module://ipykernel.pylab.backend_inline, which is a non-GUI backend, so cannot
show the figure.
  app.launch_new_instance()
```


Quantity Ordered
Price ($)
Product Name
20in Monitor
27in 4K Gaming Monitor
27in FHD Monitor
34in Ultrawide Monitor
AA Batteries (4-pack)
AAA Batteries (4-pack)
Apple Airpods Headphones
Bose SoundSport Headphones
Flatscreen TV
Google Phone
LG Dryer
LG Washing Machine
Lightning Charging Cable
Macbook Pro Laptop
ThinkPad Laptop
USB-C Charging Cable
Vareebadd Phone
Wired Headphones
iPhone