project

May 11, 2023

0.1 # Sales Analysis

```
[ ]: import pandas as pd
     from pandas import Series,DataFrame
     import numpy as np
     import matplotlib.pyplot as plt
     import os
     from itertools import combinations
     from collections import Counter
```

```
[ ]: all_file = pd.DataFrame()
     files = [file for file in os.listdir('D:\Sales Analysis\Sales_data')]
     for file in files:
         df = pd.read_csv('D:\Sales Analysis\Sales_data\\'+ file)
         all_file = pd.concat([all_file,df])

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

Read in updated dataframe

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

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

           Order Date                      Purchase Address
     0  04/19/19 08:46          917 1st St, Dallas, TX 75001
     1             NaN                                   NaN
     2  04/07/19 22:30     682 Chestnut St, Boston, MA 02215
     3  04/12/19 14:38  669 Spruce St, Los Angeles, CA 90001
     4  04/12/19 14:38  669 Spruce St, Los Angeles, CA 90001
```

0.1.1 Clean up the data

Drop rows of NaN

```
[ ]: NaN_df = all_data[all_data.isna().any(axis = 1)]
     NaN_df.head()

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

```
[ ]: 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
     4   176560            Wired Headphones                1      11.99
     5   176561            Wired Headphones                1      11.99

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

Find 'Or' and delete it

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

Convert columns to the correct type

```
[ ]: all_data['Quantity Ordered'] = pd.to_numeric(all_data['Quantity Ordered'])
     all_data['Price Each'] = pd.to_numeric(all_data['Price Each'])
     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
     0  04/19/19 08:46          917 1st St, Dallas, TX 75001
     2  04/07/19 22:30     682 Chestnut St, Boston, MA 02215
     3  04/12/19 14:38  669 Spruce St, Los Angeles, CA 90001
     4  04/12/19 14:38  669 Spruce St, Los Angeles, CA 90001
     5  04/30/19 09:27     333 8th St, Los Angeles, CA 90001
```

0.1.2 Augment data with additional columns

0.1.3 Task 2: Add month column

```
[ ]: all_data['Month'] = all_data['Order Date'].str[0:2]
     all_data['Month'] = all_data['Month'].astype('int32')
     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
```

0.1.4 Task 3: Add a sales column

```
[ ]: all_data['Sales'] = all_data['Quantity Ordered'] * all_data['Price Each']
     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   Sales
     0  04/19/19 08:46          917 1st St, Dallas, TX 75001      4   23.90
     2  04/07/19 22:30     682 Chestnut St, Boston, MA 02215      4   99.99
     3  04/12/19 14:38  669 Spruce St, Los Angeles, CA 90001      4  600.00
     4  04/12/19 14:38  669 Spruce St, Los Angeles, CA 90001      4   11.99
     5  04/30/19 09:27     333 8th St, Los Angeles, CA 90001      4   11.99
```

0.1.5 Task 4: Add a city column

```
[ ]: def get_city(address):
         return address.split(',')[1]
     def get_state(address):
         return address.split(',')[2].split(' ')[1]
```

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

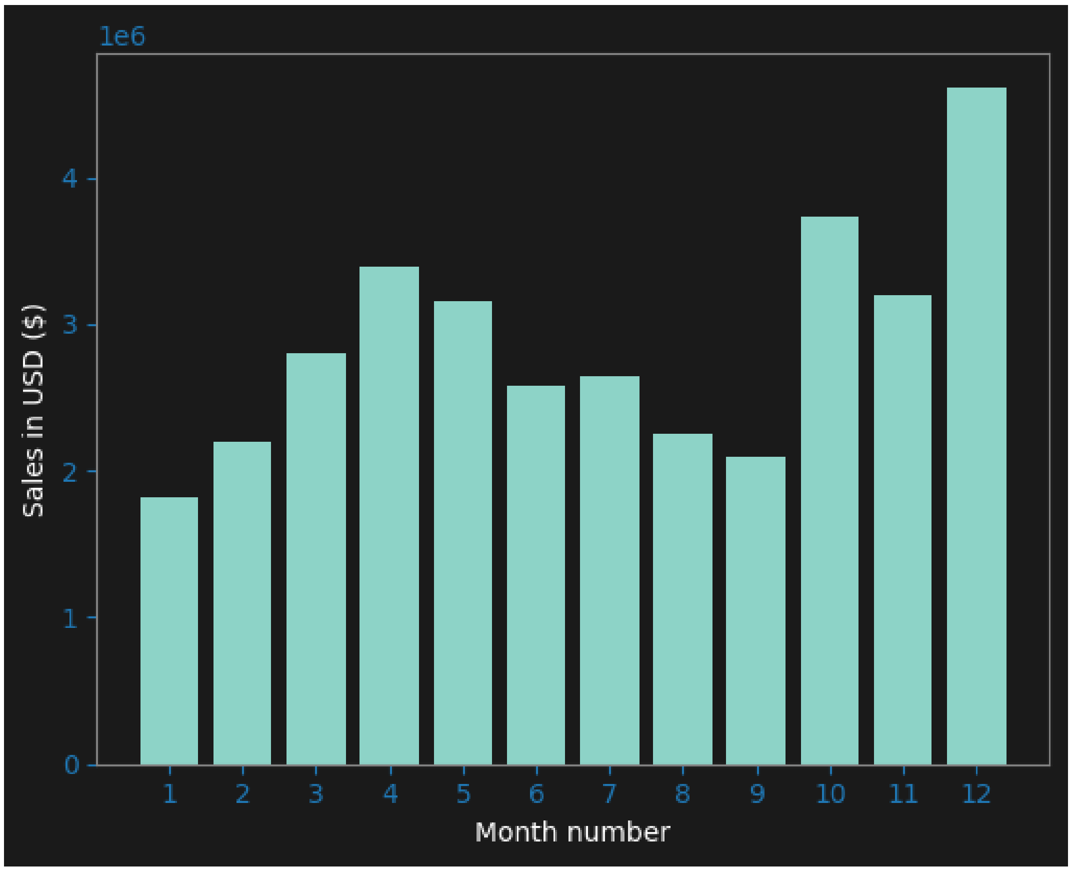
```
[ ]:  Order ID                     Product  Quantity Ordered  Price Each  \
    0   176558        USB-C Charging Cable                 2       11.95
    2   176559  Bose SoundSport Headphones                 1       99.99
    3   176560                Google Phone                 1      600.00
    4   176560            Wired Headphones                 1       11.99
    5   176561            Wired Headphones                 1       11.99

            Order Date                      Purchase Address  Month   Sales  \
    0   04/19/19 08:46            917 1st St, Dallas, TX 75001      4   23.90
    2   04/07/19 22:30       682 Chestnut St, Boston, MA 02215      4   99.99
    3   04/12/19 14:38    669 Spruce St, Los Angeles, CA 90001      4  600.00
    4   04/12/19 14:38    669 Spruce St, Los Angeles, CA 90001      4   11.99
    5   04/30/19 09:27       333 8th St, Los Angeles, CA 90001      4   11.99

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

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

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

```
[ ]:        Quantity Ordered  Price Each       Sales
    Month
    1                  10903  1811768.38  1822256.73
    2                  13449  2188884.72  2202022.42
    3                  17005  2791207.83  2807100.38
    4                  20558  3367671.02  3390670.24
    5                  18667  3135125.13  3152606.75
    6                  15253  2562025.61  2577802.26
    7                  16072  2632539.56  2647775.76
    8                  13448  2230345.42  2244467.88
    9                  13109  2084992.09  2097560.13
    10                 22703  3715554.83  3736726.88
    11                 19798  3180600.68  3199603.20
    12                 28114  4588415.41  4613443.34
```

```
[ ]: months = range(1,13)
     plt.bar(months,results['Sales'])
     plt.xticks(months)
     plt.ylabel('Sales in USD ($)')
     plt.xlabel('Month number')
     plt.show()
```



Question 2: What city has the highest number of sales ?

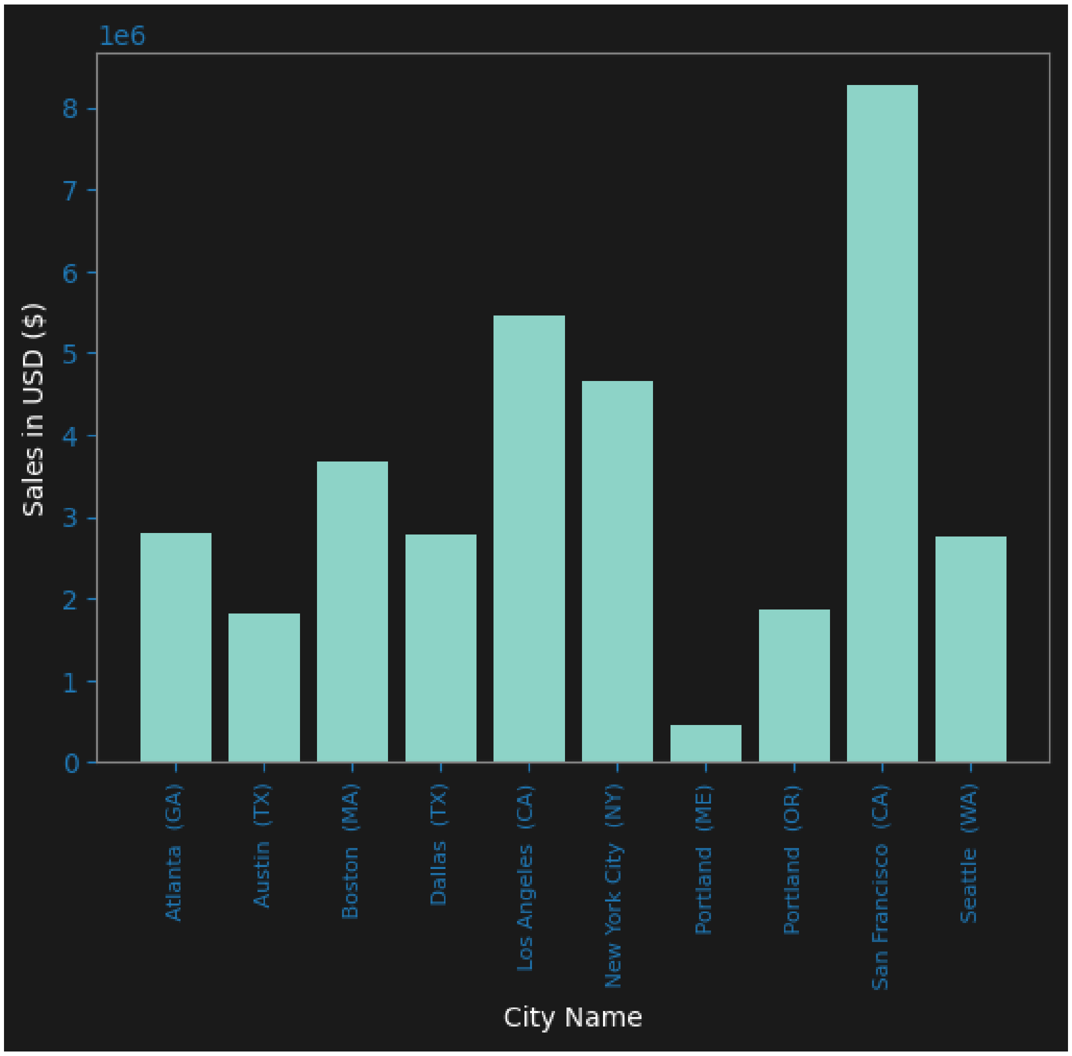
[ ]:
```
results = all_data.groupby('City').sum()
results
```

[ ]:
```
                        Quantity Ordered  Price Each   Month       Sales
City
 Atlanta  (GA)                    16602  2779908.20  104794  2795498.58
 Austin  (TX)                     11153  1809873.61   69829  1819581.75
 Boston  (MA)                     22528  3637409.77  141112  3661642.01
 Dallas  (TX)                     16730  2752627.82  104620  2767975.40
 Los Angeles  (CA)                33289  5421435.23  208325  5452570.80
 New York City  (NY)              27932  4635370.83  175741  4664317.43
 Portland  (ME)                    2750   447189.25   17144   449758.27
 Portland  (OR)                   11303  1860558.22   70621  1870732.34
 San Francisco  (CA)              50239  8211461.74  315520  8262203.91
```

```
Seattle  (WA)                  16553  2733296.01  104941  2747755.48
```

```
[ ]: cities = [city for city, df in all_data.groupby('City')]
     plt.bar(cities,results['Sales'])
     plt.xticks(cities, rotation='vertical', size=8)
     plt.ylabel('Sales in USD ($)')
     plt.xlabel('City Name')
```

```
[ ]: Text(0.5, 0, 'City Name')
```



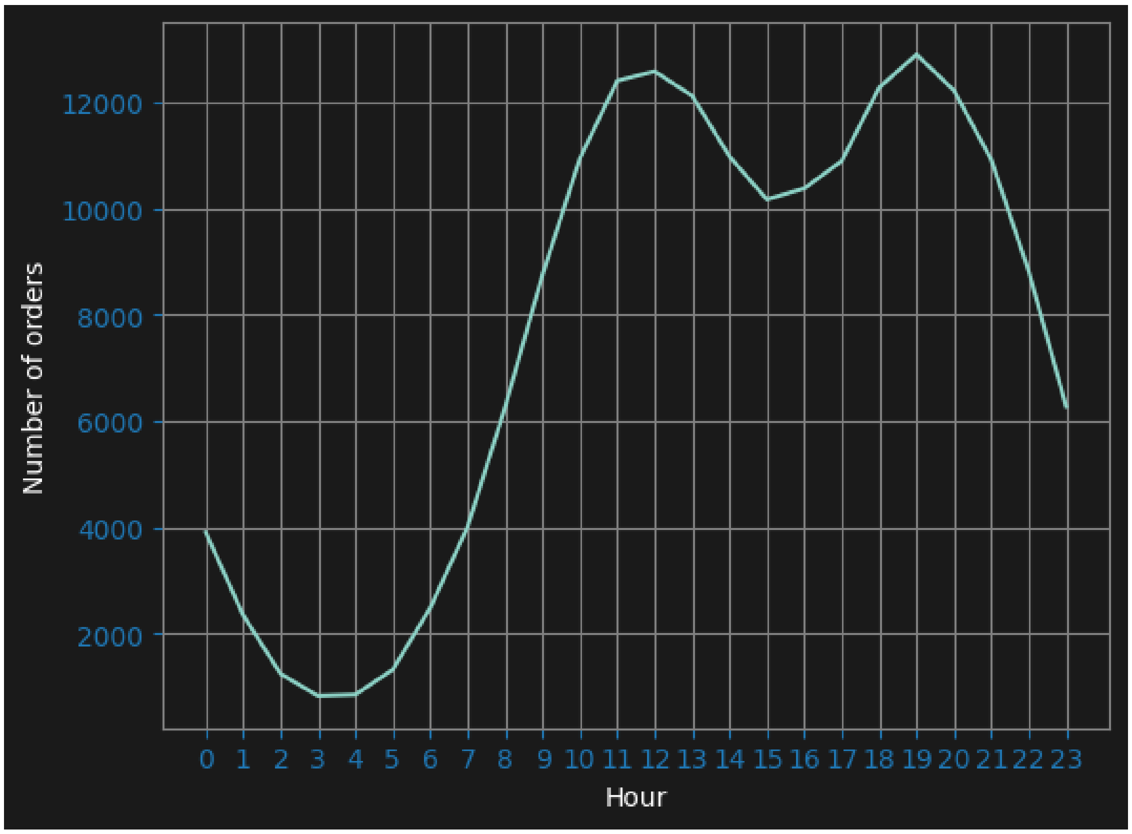
Question 3: What time should we display advertisements to maximize the likelihood of customer's buying product ?

```
[ ]: all_data['Order Date'] = pd.to_datetime(all_data['Order Date'])
     all_data['Hour'] = all_data['Order Date'].dt.hour
     all_data['Minute'] = all_data['Order Date'].dt.minute
```

```
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   Sales  \
    0 2019-04-19 08:46:00          917 1st St, Dallas, TX 75001      4   23.90
    2 2019-04-07 22:30:00     682 Chestnut St, Boston, MA 02215      4   99.99
    3 2019-04-12 14:38:00  669 Spruce St, Los Angeles, CA 90001      4  600.00
    4 2019-04-12 14:38:00  669 Spruce St, Los Angeles, CA 90001      4   11.99
    5 2019-04-30 09:27:00     333 8th St, Los Angeles, CA 90001      4   11.99

                    City  Hour  Minute
    0        Dallas  (TX)     8      46
    2        Boston  (MA)    22      30
    3   Los Angeles  (CA)    14      38
    4   Los Angeles  (CA)    14      38
    5   Los Angeles  (CA)     9      27
```

```
[ ]: hours = [hour for hour, df in all_data.groupby('Hour')]
     hour_counts = all_data.groupby('Hour').count()

     plt.plot(hours, hour_counts['Order ID'])
     plt.xticks(hours)
     plt.xlabel('Hour')
     plt.ylabel('Number of orders')
     plt.grid()
     plt.show()

     # My recommendation is around 11:00 AM or 7:00 PM
```



What products are most often sold together ?

```
[ ]: df = all_data[all_data['Order ID'].duplicated(keep = False)]
     df.loc[:, 'Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x:␣
      ↪','.join(x))
     df = df[['Order ID','Grouped']].drop_duplicates()
     df.head()
```

```
C:\Users\Saswata Paul\AppData\Local\Temp\ipykernel_840\2264370356.py:2:
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: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df.loc[:, 'Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x:
','.join(x))
```

```
[ ]:     Order ID                                      Grouped
     3     176560                   Google Phone,Wired Headphones
     18    176574                 Google Phone,USB-C Charging Cable
     30    176585  Bose SoundSport Headphones,Bose SoundSport Hea…
```

```
32    176586                AAA Batteries (4-pack),Google Phone
119   176672      Lightning Charging Cable,USB-C Charging Cable
```

[ ]:
```
count = Counter()

for row in df['Grouped']:
    row_list = row.split(',')
    count.update(Counter(combinations(row_list, 2)))

for key,value in count.most_common(10):
    print(key,value)
```

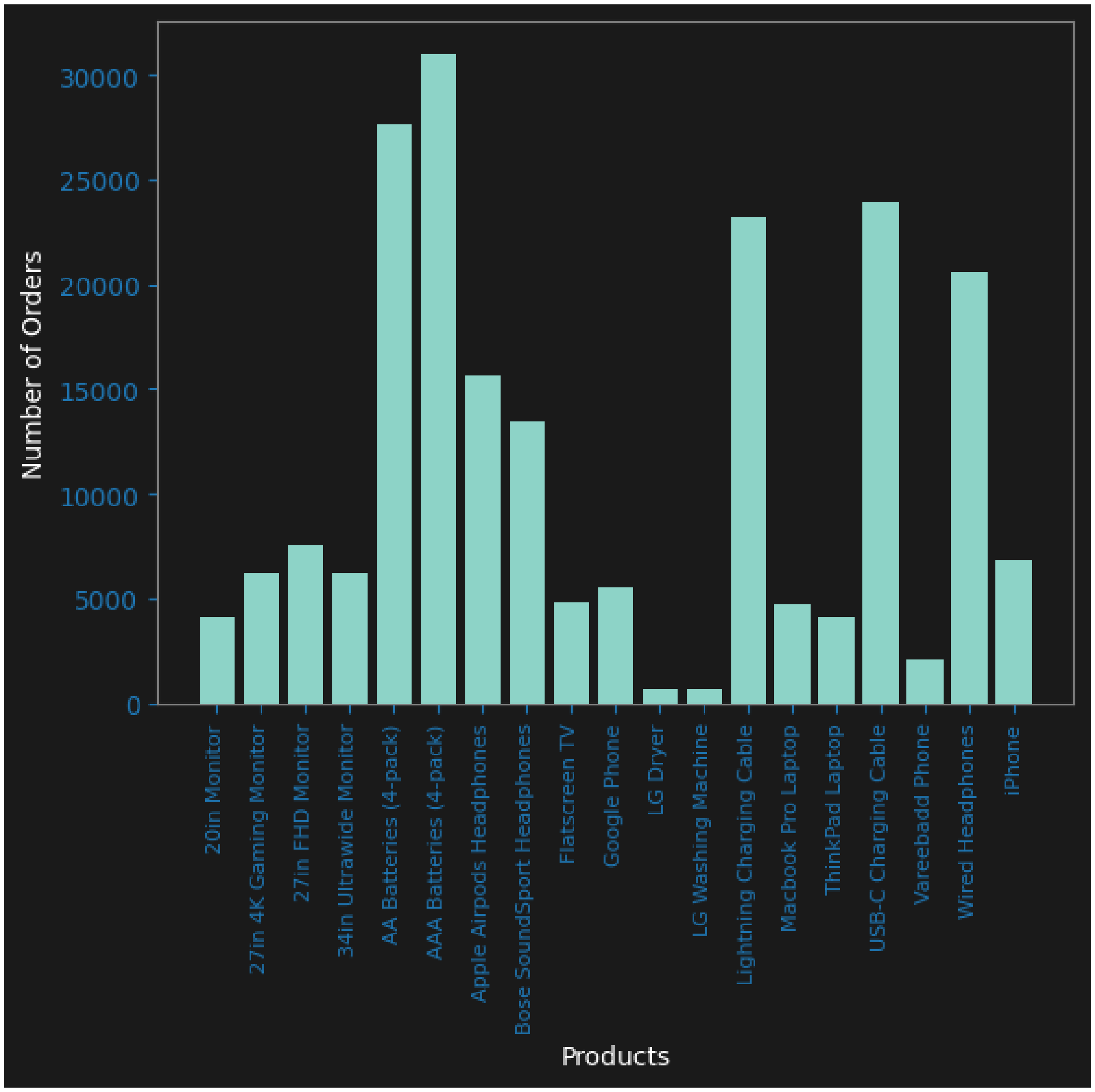
```
('iPhone', 'Lightning Charging Cable') 1005
('Google Phone', 'USB-C Charging Cable') 987
('iPhone', 'Wired Headphones') 447
('Google Phone', 'Wired Headphones') 414
('Vareebadd Phone', 'USB-C Charging Cable') 361
('iPhone', 'Apple Airpods Headphones') 360
('Google Phone', 'Bose SoundSport Headphones') 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?

[ ]:
```
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   Sales  \
0 2019-04-19 08:46:00          917 1st St, Dallas, TX 75001      4   23.90
2 2019-04-07 22:30:00     682 Chestnut St, Boston, MA 02215      4   99.99
3 2019-04-12 14:38:00  669 Spruce St, Los Angeles, CA 90001      4  600.00
4 2019-04-12 14:38:00  669 Spruce St, Los Angeles, CA 90001      4   11.99
5 2019-04-30 09:27:00     333 8th St, Los Angeles, CA 90001      4   11.99

                City  Hour  Minute
0        Dallas  (TX)     8      46
2        Boston  (MA)    22      30
3   Los Angeles  (CA)    14      38
4   Los Angeles  (CA)    14      38
5   Los Angeles  (CA)     9      27
```

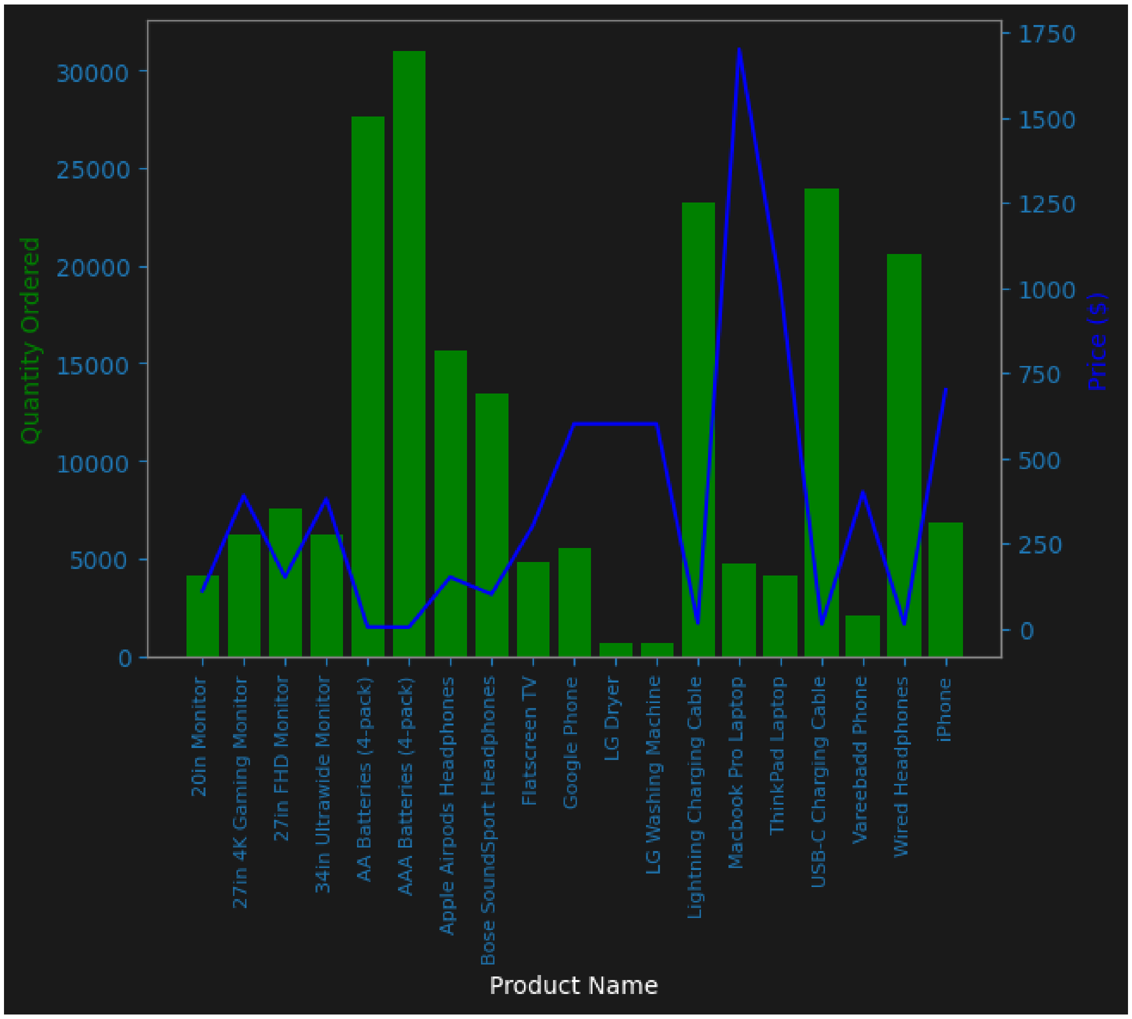
```
[ ]: product_group = all_data.groupby('Product')
     quantity_ordered = product_group.sum()['Quantity Ordered']

     products = [product for product, df in product_group]
     plt.bar(products,quantity_ordered)
     plt.xticks(products, rotation='vertical', size=8)
     plt.ylabel('Number of Orders')
     plt.xlabel('Products')
     plt.show()
```



```
[ ]: prices = all_data.groupby('Product').mean()['Price Each']
     fig, ax1 = plt.subplots()
```

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

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

plt.show()
```

```
C:\Users\Saswata Paul\AppData\Local\Temp\ipykernel_840\3988084817.py:11:
UserWarning: FixedFormatter should only be used together with FixedLocator
  ax1.set_xticklabels(products,rotation = 'vertical', size = 8)
```


Quantity Ordered
Price ($)
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
Product Name