sales-data-analysis

November 22, 2023

[1]:
```
import os
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objs as go
from plotly.offline import iplot
```

[2]:
```
all_data=pd.read_csv('Sales_Data.csv')
all_data.head()
```

[2]:
```
   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
```

1 Data cleaning and formatting

[3]:
```
all_data.dtypes
```

[3]:
```
Unnamed: 0            int64
Order ID              int64
Product              object
Quantity Ordered      int64
Price Each          float64
Order Date           object
Purchase Address     object
Month                 int64
Sales               float64
City                 object
Hour                  int64
dtype: object
```

[4]:
```
all_data.isnull().sum()
```

[4]:
```
Unnamed: 0          0
Order ID            0
Product             0
Quantity Ordered    0
Price Each          0
Order Date          0
Purchase Address    0
Month               0
Sales               0
City                0
Hour                0
dtype: int64
```

[5]:
```
all_data = all_data.dropna(how='all')
all_data.shape
```

[5]:
```
(185950, 11)
```

2 What is the best month for sale?

[6]:
```
'04/19/19 08:46'.split('/')[0]
```

[6]:
```
'04'
```

[7]:
```
def month(x):
    return x.split('/')[0]
```

3 Add month column

```
[8]: all_data['Month']=all_data['Order Date'].apply(month)
```

```
[9]: all_data.dtypes
```

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

```
[10]: all_data['Month'].unique()
```

```
[10]: array(['2019-12-30 00:01:00', '2019-12-29 07:03:00',
             '2019-12-12 18:21:00', …, '2019-06-09 22:07:00',
             '2019-06-26 18:35:00', '2019-06-25 14:33:00'], dtype=object)
```

```
[11]: filter=all_data['Month']=='Order Date'
      len(all_data[~filter])
```

```
[11]: 185950
```

```
[12]: all_data=all_data[~filter]
```

```
[13]: all_data.shape
```

```
[13]: (185950, 11)
```

```
[14]: all_data.head()
```

```
[14]:    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  \
      0  2019-12-30 00:01:00  136 Church St, New York City, NY 10001
```

```
1  2019-12-29 07:03:00     562 2nd St, New York City, NY 10001
2  2019-12-12 18:21:00    277 Main St, New York City, NY 10001
3  2019-12-22 15:13:00     410 6th St, San Francisco, CA 94016
4  2019-12-18 12:38:00           43 Hill St, Atlanta, GA 30301

                 Month    Sales           City  Hour
0  2019-12-30 00:01:00  1700.00  New York City     0
1  2019-12-29 07:03:00   600.00  New York City     7
2  2019-12-12 18:21:00    11.95  New York City    18
3  2019-12-22 15:13:00   149.99  San Francisco    15
4  2019-12-18 12:38:00    11.95        Atlanta    12
```

[15]:
```
all_data['Month']
```

[15]:
```
0         2019-12-30 00:01:00
1         2019-12-29 07:03:00
2         2019-12-12 18:21:00
3         2019-12-22 15:13:00
4         2019-12-18 12:38:00
                 …
185945    2019-06-07 19:02:00
185946    2019-06-01 19:29:00
185947    2019-06-22 18:57:00
185948    2019-06-26 18:35:00
185949    2019-06-25 14:33:00
Name: Month, Length: 185950, dtype: object
```

[16]:
```
all_data.dtypes
```

[16]:
```
Unnamed: 0            int64
Order ID              int64
Product              object
Quantity Ordered      int64
Price Each          float64
Order Date           object
Purchase Address     object
Month                object
Sales               float64
City                 object
Hour                  int64
dtype: object
```

[17]:
```
all_data['Price Each']=all_data['Price Each'].astype(float)
```

[18]:
```
all_data['Quantity Ordered']=all_data['Quantity Ordered'].astype(int)
```

```
[19]: all_data['sales']=all_data['Quantity Ordered']*all_data['Price Each']
      all_data.head(5)
```

```
[19]:    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  \
      0 2019-12-30 00:01:00  136 Church St, New York City, NY 10001
      1 2019-12-29 07:03:00     562 2nd St, New York City, NY 10001
      2 2019-12-12 18:21:00    277 Main St, New York City, NY 10001
      3 2019-12-22 15:13:00     410 6th St, San Francisco, CA 94016
      4 2019-12-18 12:38:00           43 Hill St, Atlanta, GA 30301

                      Month    Sales           City  Hour    sales
      0 2019-12-30 00:01:00  1700.00  New York City     0  1700.00
      1 2019-12-29 07:03:00   600.00  New York City     7   600.00
      2 2019-12-12 18:21:00    11.95  New York City    18    11.95
      3 2019-12-22 15:13:00   149.99  San Francisco    15   149.99
      4 2019-12-18 12:38:00    11.95        Atlanta    12    11.95
```

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

```
[20]: Month
      2019-01-01 03:07:00     11.99
      2019-01-01 03:40:00     11.95
      2019-01-01 04:56:00    150.00
      2019-01-01 05:53:00      2.99
      2019-01-01 06:03:00     23.90
                              …
      2020-01-01 04:06:00    149.99
      2020-01-01 04:13:00      2.99
      2020-01-01 04:21:00     11.95
      2020-01-01 04:54:00     99.99
      2020-01-01 05:13:00    114.94
      Name: sales, Length: 142395, dtype: float64
```

4 Which city has max order

```
[21]: '917 1st St, Dallas, TX 75001'.split(',')[1]
```

```
[21]: ' Dallas'
```

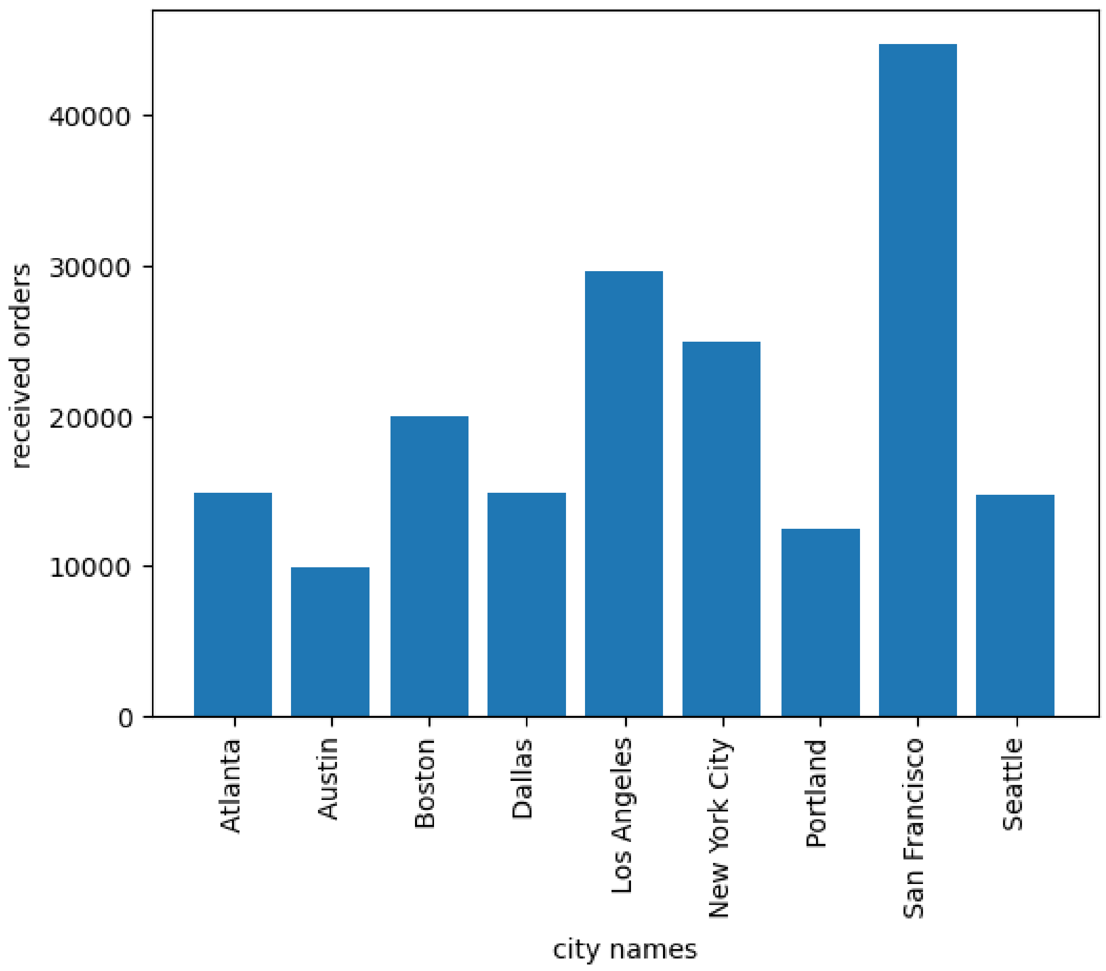
```
[22]: def city(x):
          return x.split(',')[1]
```

```
[23]: all_data['city']=all_data['Purchase Address'].apply(city)
```

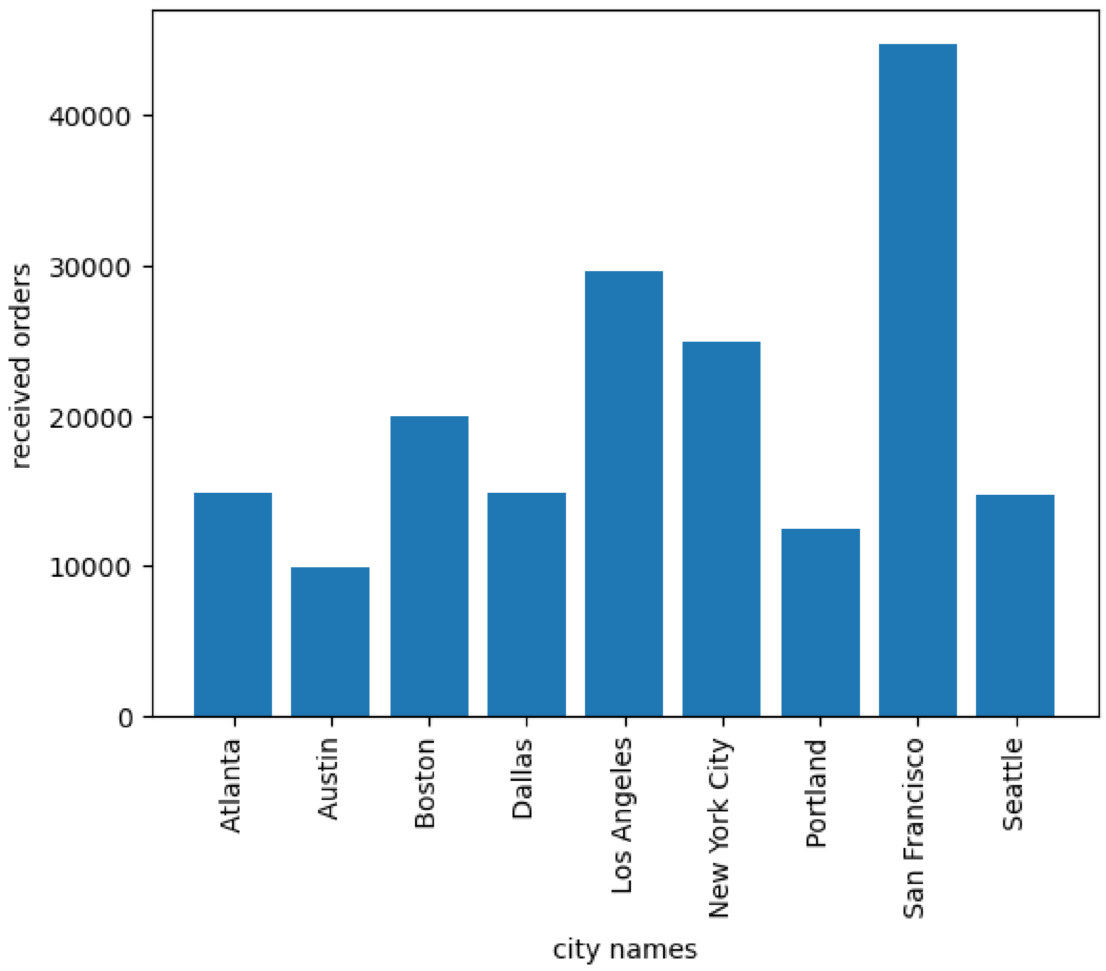
```
[24]: all_data.groupby('city')['city'].count()
```

```
[24]: city
       Atlanta          14881
       Austin            9905
       Boston           19934
       Dallas           14820
       Los Angeles      29605
       New York City    24876
       Portland         12465
       San Francisco    44732
       Seattle          14732
      Name: city, dtype: int64
```

```
[25]: plt.bar(all_data.groupby('city')['city'].count().index,all_data.
        ↪groupby('city')['city'].count())
      plt.xticks(rotation='vertical')
      plt.ylabel('received orders')
      plt.xlabel('city names')
      plt.show()
```



[26]:
```
plt.bar(all_data.groupby('city')['city'].count().index,all_data.
  ↪groupby('city')['city'].count())
plt.xticks(rotation='vertical')
plt.ylabel('received orders')
plt.xlabel('city names')
plt.show()
```



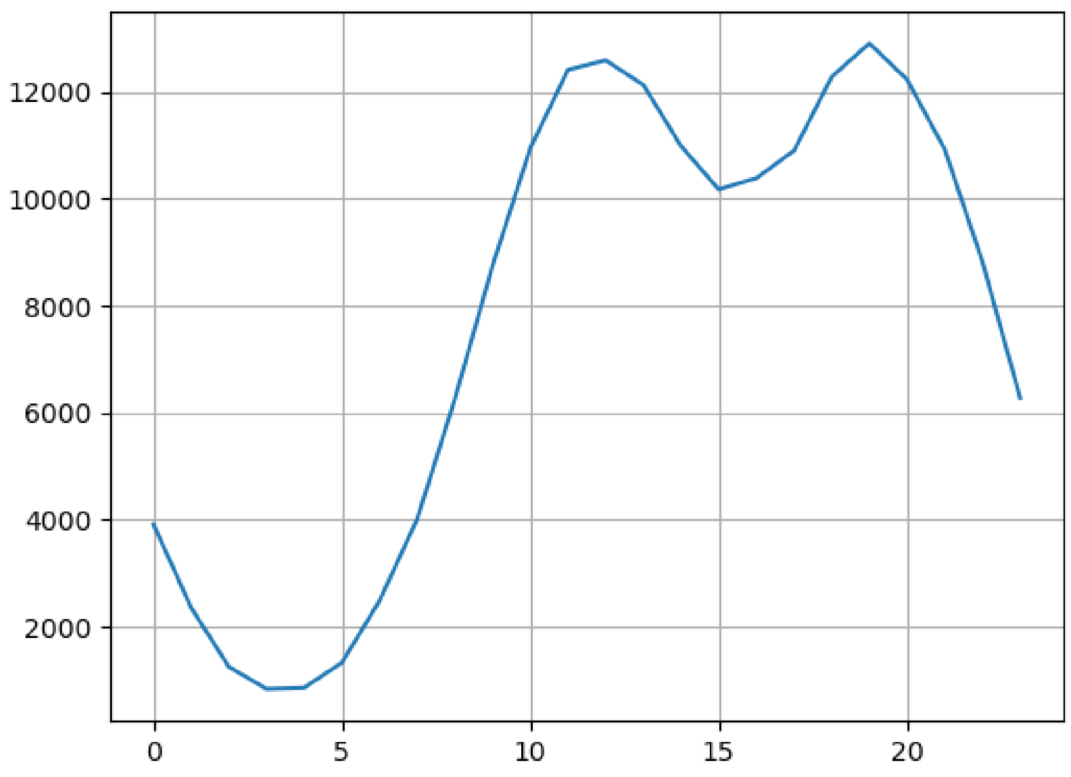
5 What time should we display advertisements to maximise for product purchase?

[27]:
```
all_data['Hour'] = pd.to_datetime(all_data['Order Date']).dt.hour
```

[28]:
```
keys=[]
hour=[]
for key,hour_df in all_data.groupby('Hour'):
    keys.append(key)
    hour.append(len(hour_df))
```

[29]:
```
plt.grid()
plt.plot(keys,hour)
```

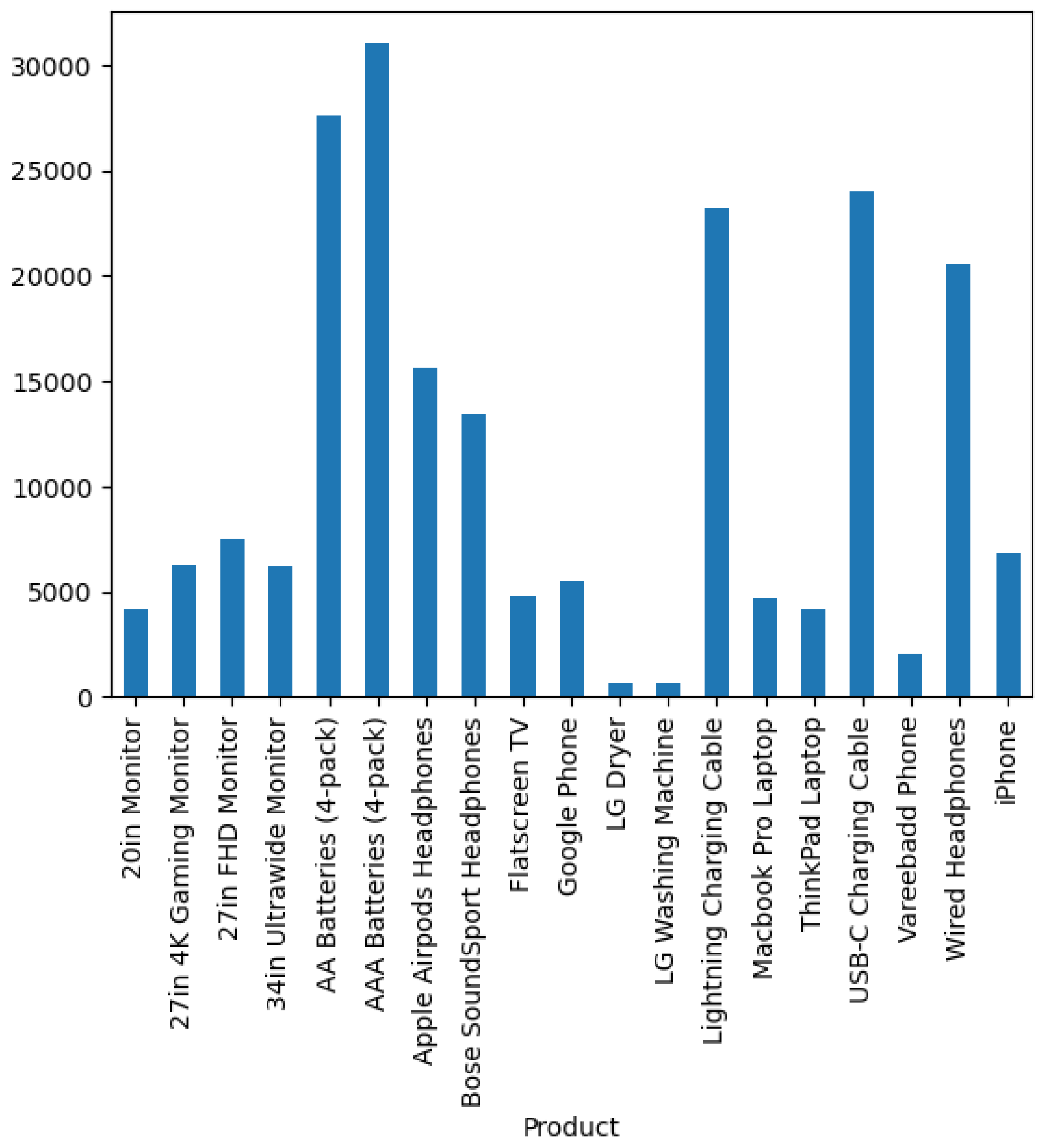
[29]:
```
[<matplotlib.lines.Line2D at 0x20b235bef80>]
```



6 Between 12pm and 7pm is probably the best time to advertise to maximise product purchase, What product sold the most? & Why?

[30]:
```
all_data.groupby('Product')['Quantity Ordered'].sum().plot(kind='bar')
```

[30]:
```
<Axes: xlabel='Product'>
```



[31]:
```
all_data.groupby('Product')['Price Each'].mean()
```

[31]:
```
Product
20in Monitor                  109.99
27in 4K Gaming Monitor        389.99
27in FHD Monitor              149.99
34in Ultrawide Monitor        379.99
AA Batteries (4-pack)           3.84
AAA Batteries (4-pack)          2.99
Apple Airpods Headphones      150.00
```

```
Bose SoundSport Headphones      99.99
Flatscreen TV                  300.00
Google Phone                   600.00
LG Dryer                       600.00
LG Washing Machine             600.00
Lightning Charging Cable        14.95
Macbook Pro Laptop            1700.00
ThinkPad Laptop                999.99
USB-C Charging Cable            11.95
Vareebadd Phone                400.00
Wired Headphones                11.99
iPhone                         700.00
Name: Price Each, dtype: float64
```

[32]:
```
products=all_data.groupby('Product')['Quantity Ordered'].sum().index
quantity=all_data.groupby('Product')['Quantity Ordered'].sum()
prices=all_data.groupby('Product')['Price Each'].mean()
```

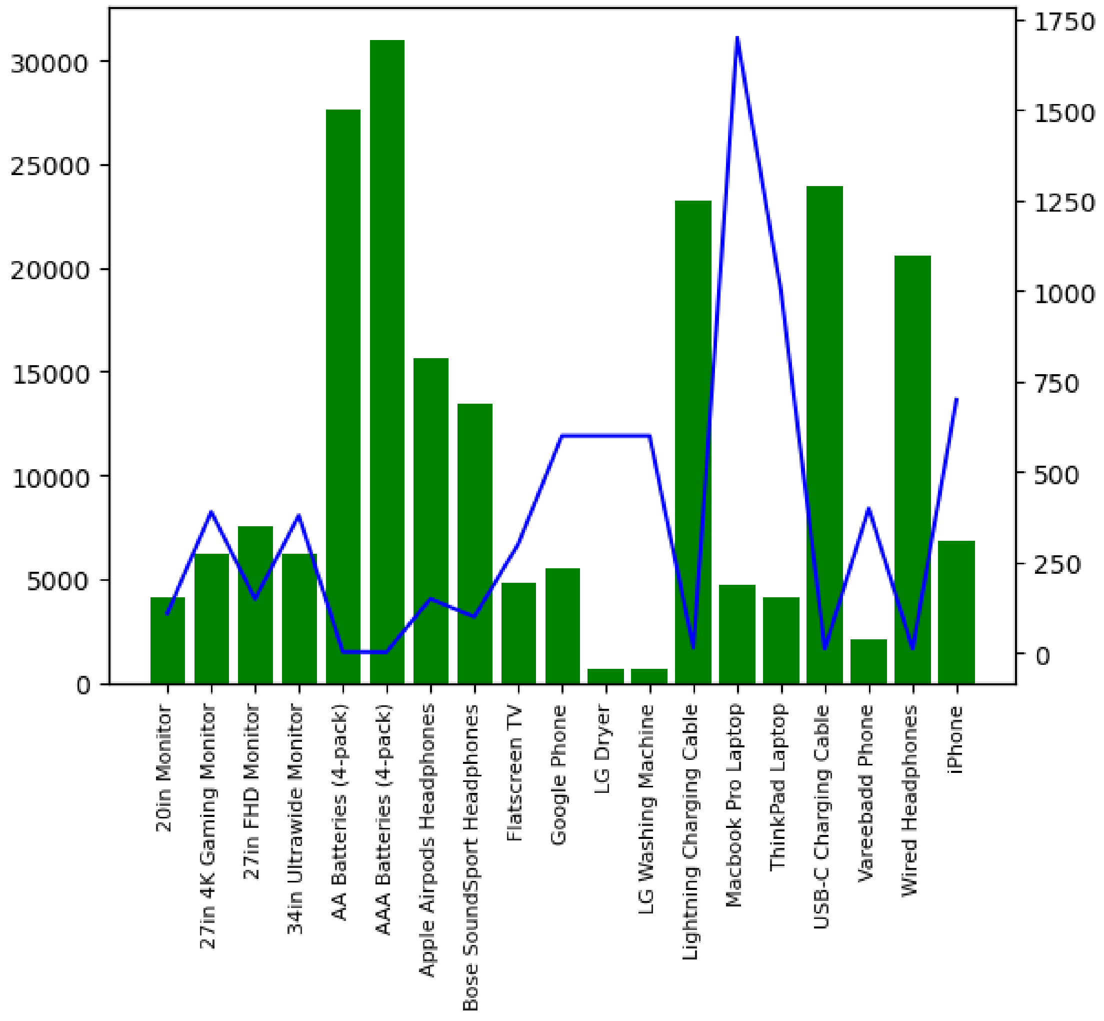
[33]:
```
plt.figure(figsize=(40,24))
fig,ax1=plt.subplots()
ax2=ax1.twinx()
ax1.bar(products, quantity, color='g')
ax2.plot(products, prices, 'b-')
ax1.set_xticklabels(products, rotation = 'vertical',size=8)
```

```
C:\Users\Saimo\AppData\Local\Temp\ipykernel_21364\977601400.py:6: UserWarning:

FixedFormatter should only be used together with FixedLocator
```

[33]:
```
[Text(0, 0, '20in Monitor'),
 Text(1, 0, '27in 4K Gaming Monitor'),
 Text(2, 0, '27in FHD Monitor'),
 Text(3, 0, '34in Ultrawide Monitor'),
 Text(4, 0, 'AA Batteries (4-pack)'),
 Text(5, 0, 'AAA Batteries (4-pack)'),
 Text(6, 0, 'Apple Airpods Headphones'),
 Text(7, 0, 'Bose SoundSport Headphones'),
 Text(8, 0, 'Flatscreen TV'),
 Text(9, 0, 'Google Phone'),
 Text(10, 0, 'LG Dryer'),
 Text(11, 0, 'LG Washing Machine'),
 Text(12, 0, 'Lightning Charging Cable'),
 Text(13, 0, 'Macbook Pro Laptop'),
 Text(14, 0, 'ThinkPad Laptop'),
 Text(15, 0, 'USB-C Charging Cable'),
 Text(16, 0, 'Vareebadd Phone'),
 Text(17, 0, 'Wired Headphones'),
 Text(18, 0, 'iPhone')]
```

```
<Figure size 4000x2400 with 0 Axes>
```



7 The top selling product is 'AAA Batteries'. The top selling products seem to have a correlation with the price of the price of the product. The cheaper the product higher the quantity ordered and vice versa

[34]:
```
all_data.shape
```

[34]:
```
(185950, 13)
```

8 What products are most often sold together? note: keep orders that have same order id, are sold mostly together

[35]:
```
df=all_data[all_data['Order ID'].duplicated(keep=False)]
df.head(20)
```

[35]:
```
     Unnamed: 0  Order ID                     Product  Quantity Ordered  \
16           16    295681                Google Phone                 1
17           17    295681        USB-C Charging Cable                 1
18           18    295681  Bose SoundSport Headphones                 1
19           19    295681            Wired Headphones                 1
36           36    295698            Vareebadd Phone                  1
37           37    295698        USB-C Charging Cable                 2
42           42    295703       AA Batteries (4-pack)                 1
43           43    295703  Bose SoundSport Headphones                 1
66           66    295726                      iPhone                 1
67           67    295726    Lightning Charging Cable                 1
76           76    295735                      iPhone                 1
77           77    295735    Apple Airpods Headphones                 1
78           78    295735            Wired Headphones                 1
80           80    295737                      iPhone                 1
81           81    295737    Lightning Charging Cable                 1
97           97    295753        34in Ultrawide Monitor                 1
98           98    295753    Lightning Charging Cable                 1
104         104    295759  Bose SoundSport Headphones                 1
105         105    295759            Wired Headphones                 1
129         129    295783             Vareebadd Phone                 1

     Price Each           Order Date                     Purchase Address  \
16       600.00  2019-12-25 12:37:00              79 Elm St, Boston, MA 02215
17        11.95  2019-12-25 12:37:00              79 Elm St, Boston, MA 02215
18        99.99  2019-12-25 12:37:00              79 Elm St, Boston, MA 02215
19        11.99  2019-12-25 12:37:00              79 Elm St, Boston, MA 02215
36       400.00  2019-12-13 14:32:00       175 1st St, New York City, NY 10001
37        11.95  2019-12-13 14:32:00       175 1st St, New York City, NY 10001
42         3.84  2019-12-17 12:27:00        502 Jefferson St, Austin, TX 73301
43        99.99  2019-12-17 12:27:00        502 Jefferson St, Austin, TX 73301
66       700.00  2019-12-25 14:49:00         203 Lakeview St, Boston, MA 02215
67        14.95  2019-12-25 14:49:00         203 Lakeview St, Boston, MA 02215
76       700.00  2019-12-22 18:25:00  374 Lincoln St, New York City, NY 10001
77       150.00  2019-12-22 18:25:00  374 Lincoln St, New York City, NY 10001
78        11.99  2019-12-22 18:25:00  374 Lincoln St, New York City, NY 10001
80       700.00  2019-12-19 08:51:00            966 10th St, Atlanta, GA 30301
81        14.95  2019-12-19 08:51:00            966 10th St, Atlanta, GA 30301
97       379.99  2019-12-25 06:26:00       365 Washington St, Dallas, TX 75001
98        14.95  2019-12-25 06:26:00       365 Washington St, Dallas, TX 75001
104       99.99  2019-12-25 06:53:00       15 Pine St, New York City, NY 10001
```

```
105       11.99  2019-12-25 06:53:00      15 Pine St, New York City, NY 10001
129      400.00  2019-12-06 12:41:00       87 5th St, San Francisco, CA 94016

                   Month   Sales           City  Hour   sales           city
16   2019-12-25 12:37:00  600.00         Boston    12  600.00         Boston
17   2019-12-25 12:37:00   11.95         Boston    12   11.95         Boston
18   2019-12-25 12:37:00   99.99         Boston    12   99.99         Boston
19   2019-12-25 12:37:00   11.99         Boston    12   11.99         Boston
36   2019-12-13 14:32:00  400.00  New York City    14  400.00  New York City
37   2019-12-13 14:32:00   23.90  New York City    14   23.90  New York City
42   2019-12-17 12:27:00    3.84         Austin    12    3.84         Austin
43   2019-12-17 12:27:00   99.99         Austin    12   99.99         Austin
66   2019-12-25 14:49:00  700.00         Boston    14  700.00         Boston
67   2019-12-25 14:49:00   14.95         Boston    14   14.95         Boston
76   2019-12-22 18:25:00  700.00  New York City    18  700.00  New York City
77   2019-12-22 18:25:00  150.00  New York City    18  150.00  New York City
78   2019-12-22 18:25:00   11.99  New York City    18   11.99  New York City
80   2019-12-19 08:51:00  700.00        Atlanta     8  700.00        Atlanta
81   2019-12-19 08:51:00   14.95        Atlanta     8   14.95        Atlanta
97   2019-12-25 06:26:00  379.99         Dallas     6  379.99         Dallas
98   2019-12-25 06:26:00   14.95         Dallas     6   14.95         Dallas
104  2019-12-25 06:53:00   99.99  New York City     6   99.99  New York City
105  2019-12-25 06:53:00   11.99  New York City     6   11.99  New York City
129  2019-12-06 12:41:00  400.00  San Francisco    12  400.00  San Francisco
```

[36]:
```
df['Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x: ','.
 ↪join(x))
```

```
C:\Users\Saimo\AppData\Local\Temp\ipykernel_21364\2345761670.py:1:
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
```

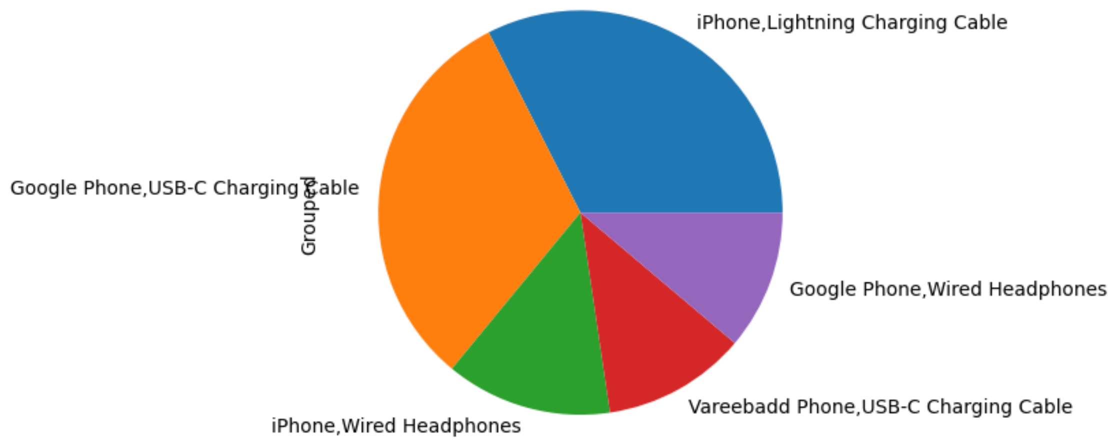
[37]:
```
df.shape
```

[37]:
```
(14649, 14)
```

[38]:
```
df2=df.drop_duplicates(subset=['Order ID'])
```

[39]:
```
df2['Grouped'].value_counts()[0:5].plot.pie()
```

```
[39]: <Axes: ylabel='Grouped'>
```



```
[40]: values=df2['Grouped'].value_counts()[0:5]
      labels=df['Grouped'].value_counts()[0:5].index
```

```
[41]: trace=go.Pie(labels=labels, values=values,
                  hoverinfo='label+percent', textinfo='value',
                  textfont=dict(size=25),
                  pull=[0,0,0,0.2,0]
                  )
      iplot([trace])
```

```
[ ]:
```