In [79]:
```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objs as go
from plotly.offline import iplot
```

In [15]:
```
data = pd.read_csv("./Sales Data.csv")
data
```

3	3	295668	27in FHD Monitor	1	149.99	2019-12-22 15:13:00	410 6th St, San Francisco, CA 94016	12	149.99	San Francisco	15
4	4	295669	USB-C Charging Cable	1	11.95	2019-12-18 12:38:00	43 Hill St, Atlanta, GA 30301	12	11.95	Atlanta	12
...	...	...	...	...	...	...	...	...	...	...	...
185945	13617	222905	AAA Batteries (4-pack)	1	2.99	2019-06-07 19:02:00	795 Pine St, Boston, MA 02215	6	2.99	Boston	19
185946	13618	222906	27in FHD Monitor	1	149.99	2019-06-01 19:29:00	495 North St, New York City, NY 10001	6	149.99	New York City	19
185947	13619	222907	USB-C Charging Cable	1	11.95	2019-06-22 18:57:00	319 Ridge St, San Francisco, CA 94016	6	11.95	San Francisco	18
185948	13620	222908	USB-C Charging Cable	1	11.95	2019-06-26 18:35:00	916 Main St, San Francisco, CA 94016	6	11.95	San Francisco	18
185949	13621	222909	AAA Batteries (4-pack)	1	2.99	2019-06-25 14:33:00	209 11th St, Atlanta, GA 30301	6	2.99	Atlanta	14

In [14]:
```
data.dtypes
```

Out[14]:
```
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
```

In [16]:
```
data = pd.read_csv("./Sales Data.csv")
data.head(5)
```

Out[16]:

	Unnamed: 0	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour
0	0	295665	Macbook Pro Laptop	1	1700.00	2019-12-30 00:01:00	136 Church St, New York City, NY 10001	12	1700.00	New York City	0
1	1	295666	LG Washing Machine	1	600.00	2019-12-29 07:03:00	562 2nd St, New York City, NY 10001	12	600.00	New York City	7
2	2	295667	USB-C Charging Cable	1	11.95	2019-12-12 18:21:00	277 Main St, New York City, NY 10001	12	11.95	New York City	18
3	3	295668	27in FHD Monitor	1	149.99	2019-12-22 15:13:00	410 6th St, San Francisco, CA 94016	12	149.99	San Francisco	15
4	4	295669	USB-C Charging Cable	1	11.95	2019-12-18 12:38:00	43 Hill St, Atlanta, GA 30301	12	11.95	Atlanta	12

In [17]:
```
data.isnull().sum()
```

Out[17]:
```
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
```

In [18]:
```
data = data.dropna(how = 'all')
data.shape
```

Out[18]:
```
(185950, 11)
```

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

Out[19]:
```
'04'
```

In [21]:
```
    def month(x):
        return x.split('/')[0]
```

In [22]:
```
    data['month']=data['Order Date'].apply(month)
```

In [24]:
```
data.dtypes
```

Out[24]:
```
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
month               object
dtype: object
```

In [26]:
```
data['month'].unique()
```

Out[26]:
```
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)
```

In [27]:
```
filter = data['month'] == 'Order Date'
len(data[~filter])
```

Out[27]:
```
185950
```

In [29]:
```
data = data[~filter]
```

In [32]:
```
data.shape
```

Out[32]:
```
(185950, 12)
```

In [33]:
```
data.head()
```

Out[33]:

	Unnamed: 0	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour	month
0	0	295665	Macbook Pro Laptop	1	1700.00	2019-12-30 00:01:00	136 Church St, New York City, NY 10001	12	1700.00	New York City	0	2019-12-30 00:01:00
1	1	295666	LG Washing Machine	1	600.00	2019-12-29 07:03:00	562 2nd St, New York City, NY 10001	12	600.00	New York City	7	2019-12-29 07:03:00
2	2	295667	USB-C Charging Cable	1	11.95	2019-12-12 18:21:00	277 Main St, New York City, NY 10001	12	11.95	New York City	18	2019-12-12 18:21:00
3	3	295668	27in FHD Monitor	1	149.99	2019-12-22 15:13:00	410 6th St, San Francisco, CA 94016	12	149.99	San Francisco	15	2019-12-22 15:13:00
4	4	295669	USB-C Charging Cable	1	11.95	2019-12-18 12:38:00	43 Hill St, Atlanta, GA 30301	12	11.95	Atlanta	12	2019-12-18 12:38:00

In [34]:
```
data['month']
```

Out[34]:
```
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
```

In [35]:
```
data.dtypes
```

Out[35]:
```
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
month               object
dtype: object
```

In [37]:
```
data['Price Each'] = data['Price Each'].astype(float)
```

In [38]:
```
data['Quantity Ordered'] = data['Quantity Ordered'].astype(int)
```

In [39]:
```
data['Sales'] = data['Quantity Ordered']* data['Price Each']
data.head()
```

Out[39]:

	Unnamed: 0	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour	month
0	0	295665	Macbook Pro Laptop	1	1700.00	2019-12-30 00:01:00	136 Church St, New York City, NY 10001	12	1700.00	New York City	0	2019-12-30 00:01:00
1	1	295666	LG Washing Machine	1	600.00	2019-12-29 07:03:00	562 2nd St, New York City, NY 10001	12	600.00	New York City	7	2019-12-29 07:03:00
2	2	295667	USB-C Charging Cable	1	11.95	2019-12-12 18:21:00	277 Main St, New York City, NY 10001	12	11.95	New York City	18	2019-12-12 18:21:00
3	3	295668	27in FHD Monitor	1	149.99	2019-12-22 15:13:00	410 6th St, San Francisco, CA 94016	12	149.99	San Francisco	15	2019-12-22 15:13:00
4	4	295669	USB-C Charging Cable	1	11.95	2019-12-18 12:38:00	43 Hill St, Atlanta, GA 30301	12	11.95	Atlanta	12	2019-12-18 12:38:00

In [40]:
```
data.groupby('Month')['Sales'].sum()
```

Out[40]:
```
Month
1     1822256.73
2     2202022.42
3     2807100.38
4     3390670.24
5     3152606.75
6     2577802.26
7     2647775.76
8     2244467.88
9     2097560.13
10    3736726.88
11    3199603.20
12    4613443.34
Name: Sales, dtype: float64
```

In [44]:
```
'917 1st St, Dallas , TX 75001'.split(',')[1]
```

Out[44]:
```
' Dallas '
```

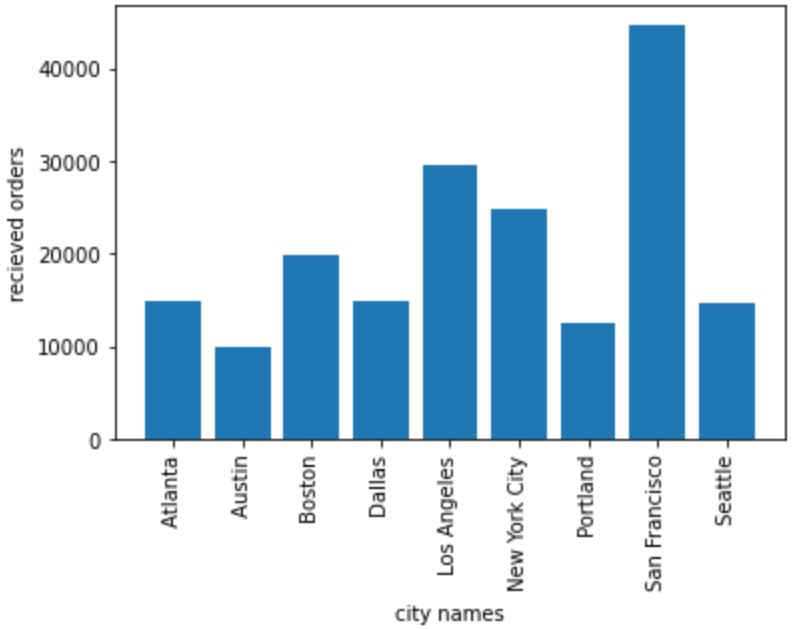
In [45]:
```
def city(x):
    return x.split(',')[1]
```

In [46]:
```
data['city'] = data['Purchase Address'].apply(city)
```

In [48]:
```
data.groupby('city')['city'].count()
```

Out[48]:
```
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
```

In [49]:
```
plt.bar(data.groupby('city')['city'].count().index,data.groupby('city')['city'].count())
plt.xticks(rotation='vertical')
plt.ylabel('recieved orders')
plt.xlabel('city names')
plt.show()
```

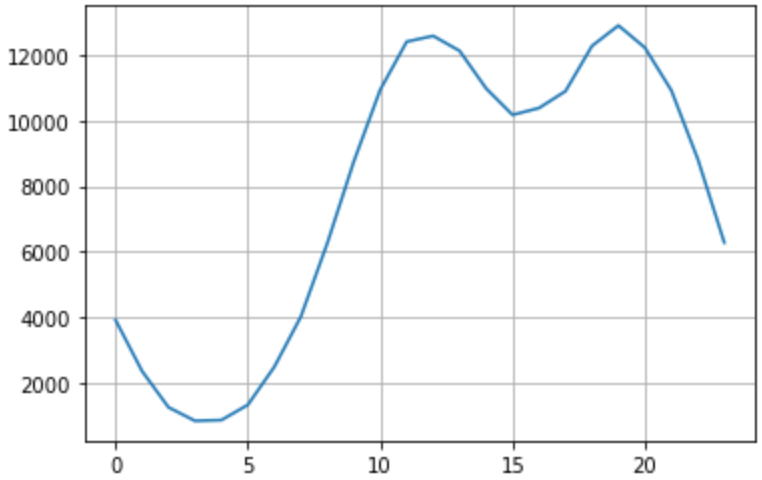


In [50]:
```
data['Hour'] = pd.to_datetime(data['Order Date']).dt.hour
```

In [53]:
```
keys = []
hour = []
for key,hour_df in data.groupby('Hour'):
    keys.append(key)
    hour.append(len(hour_df))
```

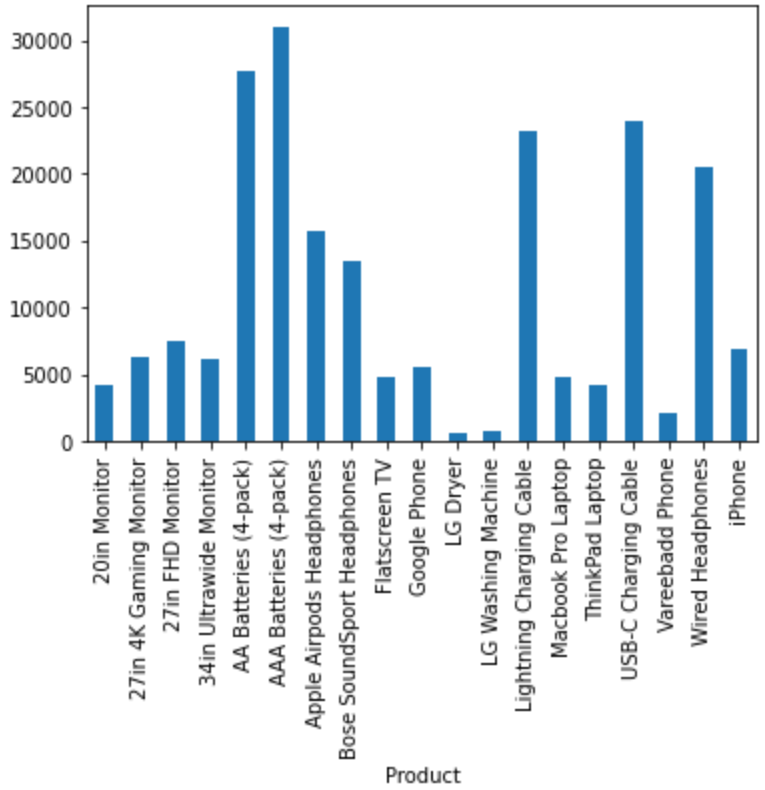
In [54]:
```
    plt.grid()
    plt.plot(keys,hour)
```

Out[54]:
```
[<matplotlib.lines.Line2D at 0x7fd67aaaec10>]
```



In [55]:
```
data.groupby('Product')['Quantity Ordered'].sum().plot(kind = 'bar')
```

Out[55]: <AxesSubplot:xlabel='Product'>



In [56]:
```
data.groupby('Product')["Price Each"].mean()
```

Out[56]:
```
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
```

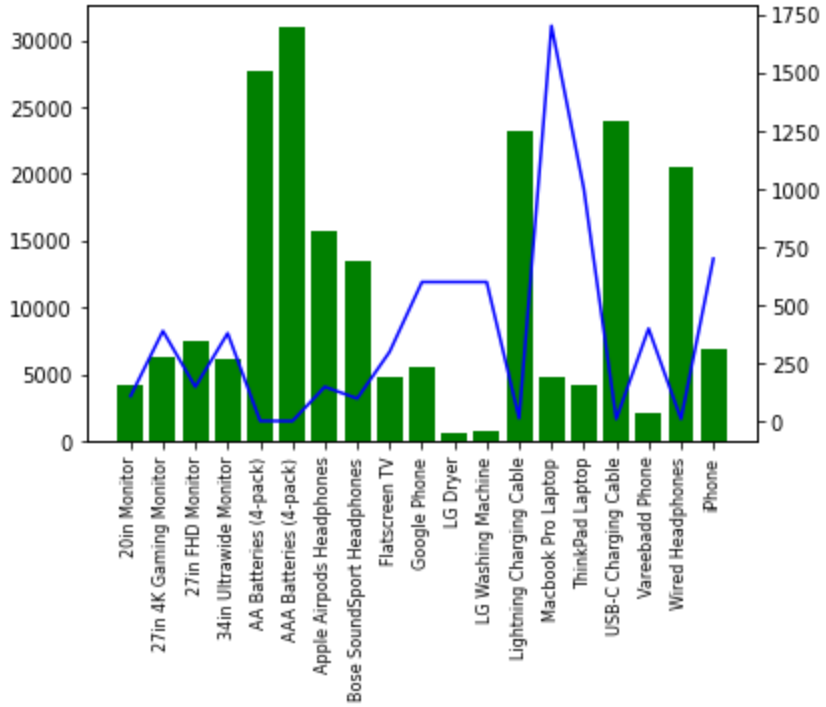
In [61]:
```
products = data.groupby('Product')['Quantity Ordered'].sum().index
quantity = data.groupby('Product')['Quantity Ordered'].sum()
prices = data.groupby('Product')['Price Each'].mean()
```

In [62]:
```
    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)
```

```
/var/folders/3r/c5cgn0vs3hgds4crgx6db9vr0000gp/T/ipykernel_1706/1869245129.py:6: UserWarning: FixedF
ormatter should only be used together with FixedLocator
  ax1.set_xticklabels(products, rotation ='vertical',size = 8)
```

Out[62]:
```
[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 2880x1728 with 0 Axes>
```



In [63]:
```
data.shape
```

Out[63]: (185950, 13)

In [65]:
```
df = data[data['Order ID'].duplicated(keep=False)]
df.head(20)
```

Out[65]:

	Unnamed: 0	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour	month	city
16	16	295681	Google Phone	1	600.00	2019-12-25 12:37:00	79 Elm St, Boston, MA 02215	12	600.00	Boston	12	2019-12-25 12:37:00	Boston
17	17	295681	USB-C Charging Cable	1	11.95	2019-12-25 12:37:00	79 Elm St, Boston, MA 02215	12	11.95	Boston	12	2019-12-25 12:37:00	Boston
18	18	295681	Bose SoundSport Headphones	1	99.99	2019-12-25 12:37:00	79 Elm St, Boston, MA 02215	12	99.99	Boston	12	2019-12-25 12:37:00	Boston
19	19	295681	Wired Headphones	1	11.99	2019-12-25 12:37:00	79 Elm St, Boston, MA 02215	12	11.99	Boston	12	2019-12-25 12:37:00	Boston
36	36	295698	Vareebadd Phone	1	400.00	2019-12-13 14:32:00	175 1st St, New York City, NY 10001	12	400.00	New York City	14	2019-12-13 14:32:00	New York City
37	37	295698	USB-C Charging Cable	2	11.95	2019-12-13 14:32:00	175 1st St, New York City, NY 10001	12	23.90	New York City	14	2019-12-13 14:32:00	New York City
42	42	295703	AA Batteries (4-pack)	1	3.84	2019-12-17 12:27:00	502 Jefferson St, Austin, TX 73301	12	3.84	Austin	12	2019-12-17 12:27:00	Austin
43	43	295703	Bose SoundSport Headphones	1	99.99	2019-12-17 12:27:00	502 Jefferson St, Austin, TX 73301	12	99.99	Austin	12	2019-12-17 12:27:00	Austin
66	66	295726	iPhone	1	700.00	2019-12-25 14:49:00	203 Lakeview St, Boston, MA 02215	12	700.00	Boston	14	2019-12-25 14:49:00	Boston
67	67	295726	Lightning Charging Cable	1	14.95	2019-12-25 14:49:00	203 Lakeview St, Boston, MA 02215	12	14.95	Boston	14	2019-12-25 14:49:00	Boston
76	76	295735	iPhone	1	700.00	2019-12-22 18:25:00	374 Lincoln St, New York City, NY 10001	12	700.00	New York City	18	2019-12-22 18:25:00	New York City

	Unnamed: 0	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour	month	city
77	77	295735	Apple Airpods Headphones	1	150.00	2019-12-22 18:25:00	374 Lincoln St, New York City, NY 10001	12	150.00	New York City	18	2019-12-22 18:25:00	New York City
78	78	295735	Wired Headphones	1	11.99	2019-12-22 18:25:00	374 Lincoln St, New York City, NY 10001	12	11.99	New York City	18	2019-12-22 18:25:00	New York City
80	80	295737	iPhone	1	700.00	2019-12-19 08:51:00	966 10th St, Atlanta, GA 30301	12	700.00	Atlanta	8	2019-12-19 08:51:00	Atlanta
81	81	295737	Lightning Charging Cable	1	14.95	2019-12-19 08:51:00	966 10th St, Atlanta, GA 30301	12	14.95	Atlanta	8	2019-12-19 08:51:00	Atlanta
97	97	295753	34in Ultrawide Monitor	1	379.99	2019-12-25 06:26:00	365 Washington St, Dallas, TX 75001	12	379.99	Dallas	6	2019-12-25 06:26:00	Dallas
98	98	295753	Lightning Charging Cable	1	14.95	2019-12-25 06:26:00	365 Washington St, Dallas, TX 75001	12	14.95	Dallas	6	2019-12-25 06:26:00	Dallas
104	104	295759	Bose SoundSport Headphones	1	99.99	2019-12-25 06:53:00	15 Pine St, New York City, NY 10001	12	99.99	New York City	6	2019-12-25 06:53:00	New York City
105	105	295759	Wired Headphones	1	11.99	2019-12-25 06:53:00	15 Pine St, New York City, NY 10001	12	11.99	New York City	6	2019-12-25 06:53:00	New York City
129	129	295783	Vareebadd Phone	1	400.00	2019-12-06 12:41:00	87 5th St, San Francisco, CA 94016	12	400.00	San Francisco	12	2019-12-06 12:41:00	San Francisco

In [66]:
```
df["Grouped"] = df.groupby('Order ID')['Product'].transform(lambda x : ','.join(x))
```

```
/var/folders/3r/c5cgn0vs3hgds4crgx6db9vr0000gp/T/ipykernel_1706/768347393.py:1: SettingWithCopyWarni
ng:
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/indexi
ng.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/inde
xing.html#returning-a-view-versus-a-copy)
  df["Grouped"] = df.groupby('Order ID')['Product'].transform(lambda x : ','.join(x))
```

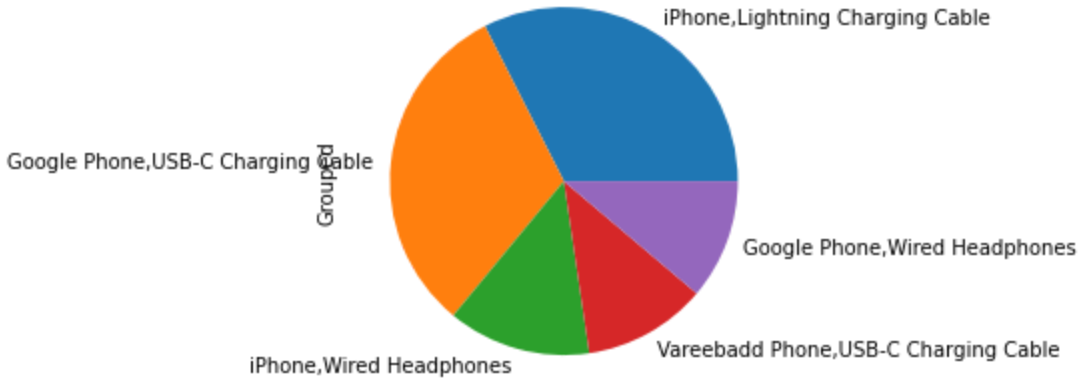
In [67]:
```
df.shape
```

Out[67]:
```
(14649, 14)
```

In [68]:
```
df2 = df.drop_duplicates(subset=['Order ID'])
```

In [69]:
```
df2['Grouped'].value_counts()[0:5].plot.pie()
```

Out[69]:
```
<AxesSubplot:ylabel='Grouped'>
```



In [70]:
```
values=df2['Grouped'].value_counts()[0:5]
labels=df['Grouped'].value_counts()[0:5].index
```

In [75]:
```
trace= go.Pie(labels=labels,values=values,
              hoverinfo='label+percent',textinfo='value',
              textfont=dict(size = 25),
              pull=[0,0,0,0.2,0]
              )
```

In [80]:
```
iplot([trace])
```