In [1]:
```
#import packages
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from pandas import set_option
```

In [2]:
```
data = 'C:\\Users\\HP\\Documents\\WORKSPACE\\Pharm Data2.xlsx'
```

In [3]:
```
df= pd.read_excel(data)
```

In [4]:
```
df.head()
```

Out[4]:

	Distributor	Customer Name	City	Country	Latitude	Longitude	Channel	Sub-channel	Product Name	Product Class	Quantity	Price	Sales	Month	Year	Nam Sal R
0	Gottlieb-Cruickshank	Zieme, Doyle and Kunze	Lublin	Poland	51.2333	22.5667	Hospital	Private	Topipizole	Mood Stabilizers	4.0	368	1472.0	January	2018	Ma Gerra
1	Gottlieb-Cruickshank	Feest PLC	Świecie	Poland	53.4167	18.4333	Pharmacy	Retail	Choriotrisin	Antibiotics	7.0	591	4137.0	January	2018	Jess Sm
2	Gottlieb-Cruickshank	Medhurst-Beer Pharmaceutical Limited	Rybnik	Poland	50.0833	18.5000	Pharmacy	Institution	Acantaine	Antibiotics	30.0	66	1980.0	January	2018	Ste Pepp
3	Gottlieb-Cruickshank	Barton Ltd Pharma Plc	Czeladź	Poland	50.3333	19.0833	Hospital	Private	Lioletine Refliruvax	Analgesics	6.0	435	2610.0	January	2018	Ma Gerra
4	Gottlieb-Cruickshank	Keeling LLC Pharmacy	Olsztyn	Poland	53.7800	20.4942	Pharmacy	Retail	Oxymotroban Fexoformin	Analgesics	20.0	458	9160.0	January	2018	An V

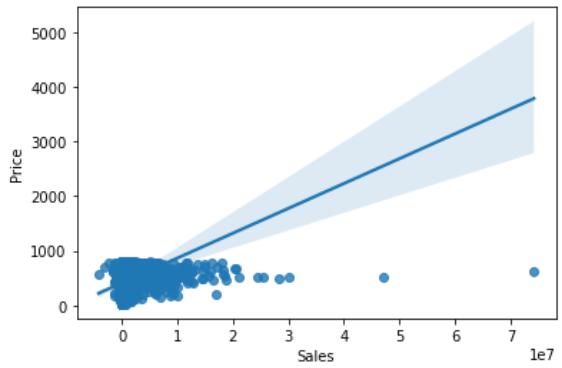
In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 254082 entries, 0 to 254081
Data columns (total 18 columns):
 #   Column             Non-Null Count   Dtype  
---  ------             --------------   -----  
 0   Distributor        254082 non-null  object 
 1   Customer Name      254082 non-null  object 
 2   City               254082 non-null  object 
 3   Country            254082 non-null  object 
 4   Latitude           254082 non-null  float64
 5   Longitude          254082 non-null  float64
 6   Channel            254082 non-null  object 
 7   Sub-channel        254082 non-null  object 
 8   Product Name       254082 non-null  object 
 9   Product Class      254082 non-null  object 
 10  Quantity           254082 non-null  float64
 11  Price              254082 non-null  int64  
 12  Sales              254082 non-null  float64
 13  Month              254082 non-null  object 
 14  Year               254082 non-null  int64  
 15  Name of Sales Rep  254082 non-null  object 
 16  Manager            254082 non-null  object 
 17  Sales Team         254082 non-null  object 
dtypes: float64(4), int64(2), object(12)
memory usage: 34.9+ MB
```

In [6]:
```
#checking for missing values
df.isnull().sum()
```

Out[6]:
```
Distributor          0
Customer Name        0
City                 0
Country              0
Latitude             0
Longitude            0
Channel              0
Sub-channel          0
Product Name         0
Product Class        0
Quantity             0
Price                0
Sales                0
Month                0
Year                 0
Name of Sales Rep    0
Manager              0
Sales Team           0
dtype: int64
```

In [7]:
```
#plt.title(" Correlation between Bill payment and Loan payment for July", fontsize = 20)
sns.regplot(x="Sales", y="Price", data=df);
```



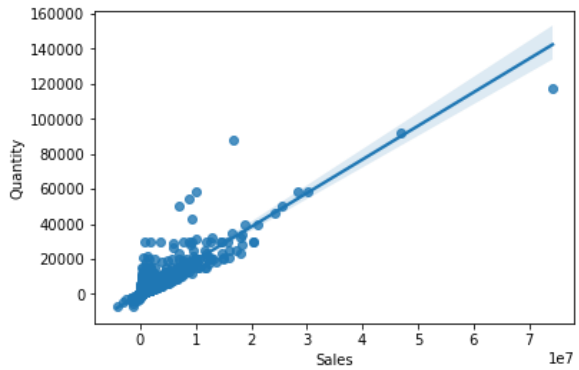
There is a postive linear relationship between Price and Sales at a lower price the higher the sales

In [8]:
```
#checking the correlation between the numeric features
set_option ('display.width', 1000)
plt.figure(figsize = (16,5))
sns.heatmap(df.corr(), annot = True)
```

Out[8]: <AxesSubplot:>

There is a strong correlation between quatity and sales

In [9]:
```
sns.regplot(x="Sales", y="Quantity", data=df);
```



We can see that there is a positive linear relationship between Quantity and Sales the higher the qunatity sold the higher the sales that is being made.

In [10]:
```
#descriptive analysis of the data
df.describe()
```

Out[10]:

	Latitude	Longitude	Quantity	Price	Sales	Year
count	254082.000000	254082.000000	254082.000000	254082.000000	2.540820e+05	254082.000000
mean	50.962222	10.803212	112.872139	412.207720	4.643772e+04	2018.385187
std	1.625526	4.143311	744.310385	224.963687	3.491918e+05	1.041352
min	47.514200	6.083800	-7200.000000	22.000000	-4.161600e+06	2017.000000
25%	49.805600	7.891100	5.000000	195.000000	1.704000e+03	2018.000000
50%	51.133300	9.397800	20.000000	430.000000	5.850000e+03	2018.000000
75%	52.083300	12.133300	50.000000	605.000000	2.156525e+04	2019.000000
max	54.781900	23.566700	117600.000000	794.000000	7.420560e+07	2020.000000

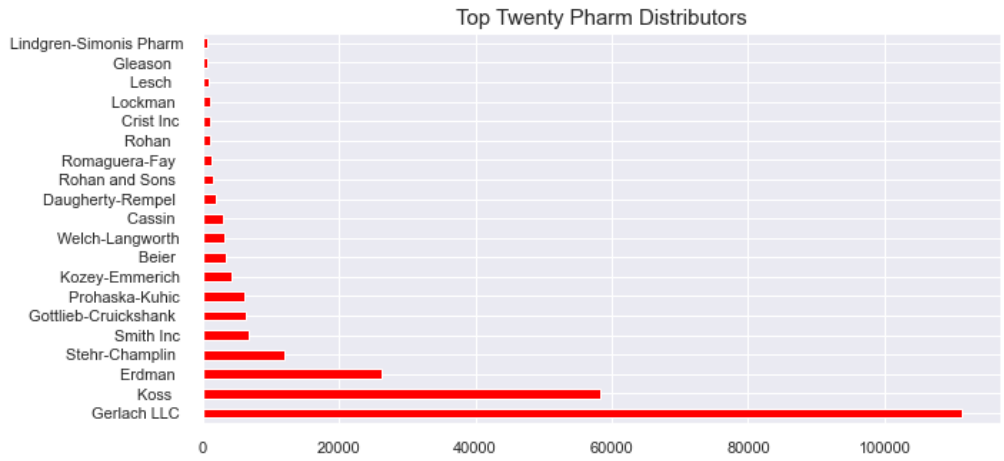
In [11]:
```
#checking for unique disributors
df['Distributor'].unique()
```

Out[11]:
```
array(['Gottlieb-Cruickshank  ', 'Carter-Conn  ', 'Prohaska-Kuhic ',
       'Smith Inc ', 'Rohan   ', 'Schuppe Inc ', 'Cassin  ',
       'Graham and Sons ', 'Stehr-Champlin  ', 'Kris LLC  ',
       'Rogahn-Klein ', 'Lindgren-Simonis Pharm', 'Beier  ',
       'Gerlach LLC ', 'Erdman  ', 'Koss   ', 'Schaefer LLC ',
       'Crist Inc ', 'Rohan and Sons  ', 'Lockman  ', 'Kozey-Emmerich ',
       'Gleason   ', 'Romaguera-Fay  ', 'Daugherty-Rempel  ',
       'Welch-Langworth ', 'Bashirian-Kassulke  ', 'Nader-Gaylord  ',
       'Hansen Group Pharm', 'Lesch   '], dtype=object)
```

In [12]:
```
#count of major distributros
(df['Distributor'].value_counts())
```

Out[12]:
```
Gerlach LLC               111364
Koss                       58360
Erdman                     26238
Stehr-Champlin             11942
Smith Inc                   6802
Gottlieb-Cruickshank        6427
Prohaska-Kuhic              6109
Kozey-Emmerich              4305
Beier                       3311
Welch-Langworth             3125
Cassin                      3007
Daugherty-Rempel            1960
Rohan and Sons              1575
Romaguera-Fay               1356
Rohan                       1185
Crist Inc                   1157
Lockman                     1013
Lesch                        974
Gleason                      779
Lindgren-Simonis Pharm       759
Schaefer LLC                 572
Nader-Gaylord                324
Hansen Group Pharm           318
Schuppe Inc                  269
Rogahn-Klein                 227
Graham and Sons              213
Carter-Conn                  181
Bashirian-Kassulke           178
Kris LLC                      52
Name: Distributor, dtype: int64
```

In [13]:
```
#visual representaion of major distiributors
plt.style.use('seaborn')
sns.set(style="darkgrid")
plt.figure(figsize = (10, 5))
plt.title('Top Twenty Pharm Distributors', fontsize = 15)
df['Distributor'].value_counts()[:20].plot(kind='barh', color='red')
plt.show()
```



From the above analysis we can see that Gerlach LLC of all the distributor had the highest occurrence with a count of 111,364. we can then say for a fact that Gerlach LLC was a major dististributor amongts others.

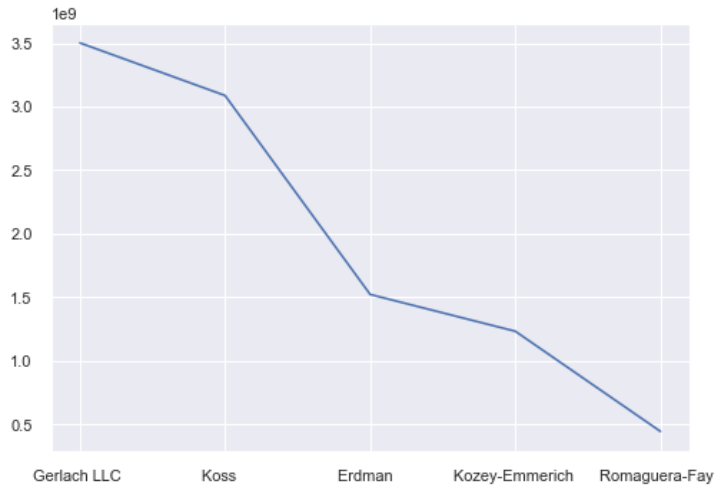
In [48]:
```
#checkiong the distributor with the highest amount of sales
dist_sales =df.groupby('Distributor')[['Sales']].sum().sort_values(by=['Sales'],ascending=False)[:10];
```

In [49]:
```
dist_sales
```

Out[49]:

Distributor	Sales
Gerlach LLC	3.501834e+09
Koss	3.087827e+09
Erdman	1.522610e+09
Kozey-Emmerich	1.232932e+09
Romaguera-Fay	4.449925e+08
Bashirian-Kassulke	3.493407e+08
Welch-Langworth	2.606324e+08
Daugherty-Rempel	2.321302e+08
Beier	1.508344e+08
Rohan and Sons	1.425120e+08

In [51]:
```
#plot showing the analysis of sales by distributor
plt.plot(df.groupby('Distributor')[['Sales']].sum().sort_values(by=['Sales'],ascending=False)[:5]);
```



From the above table we can see that Gerlach LLC had the highest number of sales with a total number of 3.5 billion.

In [18]:
```
df.groupby(['Distributor','Month']).agg({'Sales':'sum'}).reset_index().sort_values(by=['Sales'],ascending=False).head(50)
```

Out[18]:

	Distributor	Month	Sales
79	Gerlach LLC	July	3.622057e+08
83	Gerlach LLC	November	3.580536e+08
80	Gerlach LLC	June	3.570305e+08
122	Koss	June	3.055939e+08
123	Koss	March	3.028968e+08
81	Gerlach LLC	March	3.007010e+08
117	Koss	August	3.001932e+08
118	Koss	December	2.932541e+08
82	Gerlach LLC	May	2.927934e+08
125	Koss	November	2.894637e+08
84	Gerlach LLC	October	2.874368e+08
85	Gerlach LLC	September	2.847042e+08
74	Gerlach LLC	April	2.745771e+08
127	Koss	September	2.656886e+08
1	Bashirian-Kassulke	August	2.595039e+08
121	Koss	July	2.563525e+08
126	Koss	October	2.516082e+08
77	Gerlach LLC	February	2.507439e+08
78	Gerlach LLC	January	2.504595e+08
75	Gerlach LLC	August	2.502074e+08
119	Koss	February	2.353290e+08
76	Gerlach LLC	December	2.329213e+08
124	Koss	May	2.324940e+08
116	Koss	April	2.101942e+08
73	Erdman	September	1.726316e+08
135	Kozey-Emmerich	March	1.617111e+08
64	Erdman	December	1.557550e+08
68	Erdman	June	1.540488e+08
120	Koss	January	1.447594e+08
137	Kozey-Emmerich	November	1.415153e+08
138	Kozey-Emmerich	October	1.402210e+08
71	Erdman	November	1.372861e+08
67	Erdman	July	1.342499e+08
72	Erdman	October	1.278108e+08
65	Erdman	February	1.265052e+08
69	Erdman	March	1.254089e+08
129	Kozey-Emmerich	August	1.174838e+08
133	Kozey-Emmerich	July	1.167387e+08
130	Kozey-Emmerich	December	1.145090e+08
62	Erdman	April	1.113199e+08
70	Erdman	May	1.068488e+08
63	Erdman	August	9.821901e+07
139	Kozey-Emmerich	September	9.491943e+07
131	Kozey-Emmerich	February	9.057764e+07
128	Kozey-Emmerich	April	8.762715e+07
136	Kozey-Emmerich	May	8.432686e+07
224	Romaguera-Fay	March	7.332049e+07
66	Erdman	January	7.252580e+07
221	Romaguera-Fay	January	7.234593e+07
220	Romaguera-Fay	February	5.701198e+07

In [21]:
```
#analysis of products based on slaes
product_sales = df.groupby('Product Name')[['Sales']].sum().sort_values(by=['Sales'],ascending=False)
```

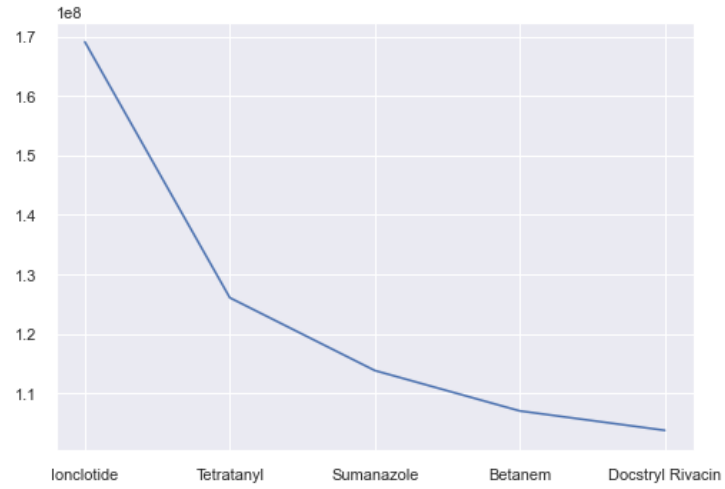
In [22]:
```
product_sales.head(10)
```

Out[22]:

Product Name	Sales
Ionclotide	169083391.0
Tetratanyl	126091294.0
Sumanazole	113861431.0
Betanem	107073473.0
Docstryl Rivacin	103811886.0
Travoloride	101167660.0
Propratecan	100878712.0
Ketastadil	97313783.0
Nevanide Actozide	96643552.0
Cephozumab Synmethate	95320320.0

Ionciotide was the product with the most sales with a total sale of 169,083,391.

In [55]:
```
#visul analysis to show the product with the most sale.
plt.plot(df.groupby('Product Name')[['Sales']].sum().sort_values(by=['Sales'],ascending=False).head());
plt.show()
```



In [23]:
```
#product by quantiy
product_quant = df.groupby('Product Name')[['Quantity']].sum().sort_values(by=['Quantity'],ascending= False)
```

In [24]:
```
product_quant.head(10)
```

Out[24]:

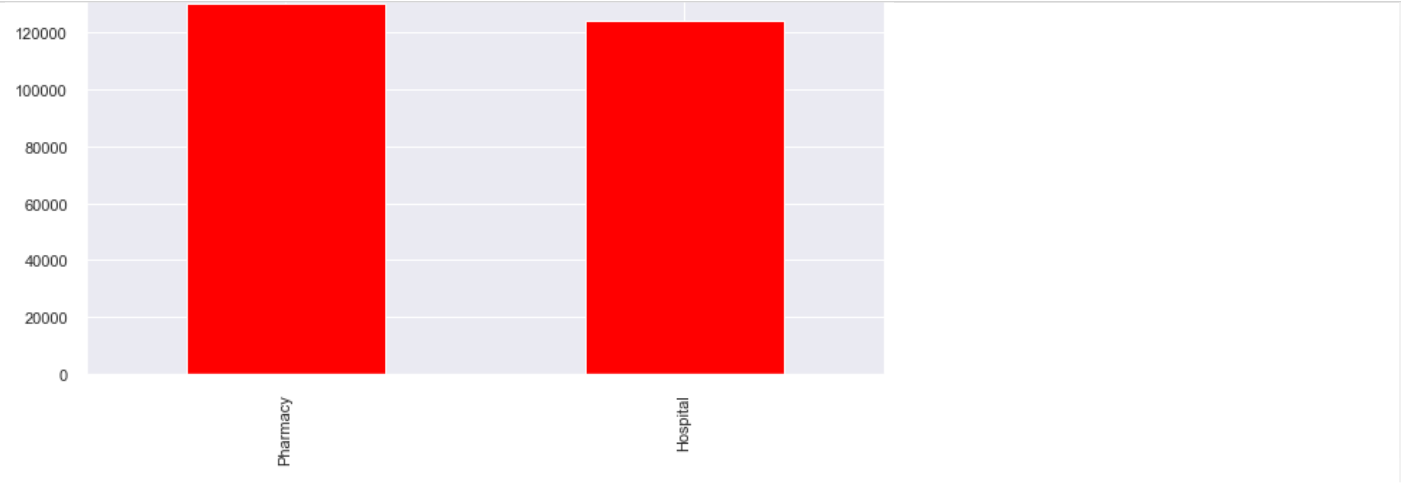
Product Name	Quantity
Ionclotide	267961.0
Tetratanyl	246754.0
Sumanazole	215239.0
Formolovir Amanferon	193393.0
Symdocet	193282.0
Hepavice	177452.0
Amavirase	177196.0
Betanem	175243.0
Dantocept Ferurenone	173490.0
Zyvance	172260.0

Ionciotide was the product with the most quantity distributed

In [25]:
```
(df['Channel'].value_counts())
```

Out[25]:
```
Pharmacy    129971
Hospital    124111
Name: Channel, dtype: int64
```

In [26]:
```
plt.style.use('seaborn')
sns.set(style="darkgrid")
plt.figure(figsize = (10, 5))
plt.title('Top Distribution Channel', fontsize = 15)
df['Channel'].value_counts()[:20].plot(kind='bar', color='red')
plt.show()
```



In [27]:
```
#channerl of distribution by sales
chan_sales = df.groupby('Channel')[['Sales']].sum().sort_values(by=['Sales'],ascending=False)
```

In [28]:
```
chan_sales
```

Out[28]:

Channel	Sales
Pharmacy	6.218312e+09
Hospital	5.580676e+09

From the above table we can see that Pharmacy was the channnel where the most sales were made with a total of 6.2 Billion

In [29]:
```
year_sales = df.groupby('Year')[['Sales']].sum().sort_values(by=['Sales'],ascending=True)
```

In [30]:
```
year_sales
```

Out[30]:

Year	Sales
2020	2.659672e+09
2017	2.701481e+09
2019	2.930937e+09
2018	3.506897e+09

In [31]:
```
(df['Country'].value_counts())
```

Out[31]:
```
Germany    213598
Poland      40484
Name: Country, dtype: int64
```

In [32]:
```
df.groupby('Sub-channel')[['Sales']].sum().sort_values(by=['Sales'],ascending=False)
```

Out[32]:

Sub-channel	Sales
Retail	3.343097e+09
Government	3.058240e+09
Institution	2.875215e+09
Private	2.522435e+09

In [33]:
```
(df['City'].value_counts()).head(10)
```

Out[33]:
```
Friedberg    796
Neustadt     760
Bergheim     440
Ettlingen    437
Kehl         436
Merseburg    436
Oldenburg    434
Zeitz        431
Hamburg      430
Apolda       429
Name: City, dtype: int64
```

In [34]:
```
df.groupby('City')[['Sales']].sum().sort_values(by=['Sales'],ascending=False)[:10]
```

Out[34]:

City	Sales
Butzbach	9.356178e+07
Baesweiler	6.489050e+07
Cuxhaven	5.600668e+07
Friedberg	5.218363e+07
Altenburg	5.088532e+07
Emsdetten	4.593901e+07
Bottrop	4.445462e+07
Freising	4.377938e+07
Trier	4.349563e+07
Castrop-Rauxel	4.206666e+07

In [35]:
```
(df['Sub-channel'].value_counts())
```

Out[35]:
```
Retail         68351
Government     65605
Institution    61620
Private        58506
Name: Sub-channel, dtype: int64
```

In [36]:
```
(df['Product Class'].value_counts())
```

Out[36]:
```
Antiseptics         52037
Mood Stabilizers    46415
Analgesics          44751
Antibiotics         36979
Antipiretics        36955
Antimalarial        36945
Name: Product Class, dtype: int64
```

In [37]:
```
df.groupby('Product Class')[['Quantity']].sum().sort_values(by=['Quantity'],ascending=False)
```

Out[37]:

Product Class	Quantity
Analgesics	5.553144e+06
Antiseptics	5.499913e+06
Mood Stabilizers	5.169781e+06
Antimalarial	4.249075e+06
Antibiotics	4.154322e+06
Antipiretics	4.052544e+06

In [38]:
```
df.groupby('Sub-channel')[['Sales']].sum().sort_values(by=['Sales'],ascending=False)
```

Out[38]:

Sub-channel	Sales
Retail	3.343097e+09
Government	3.058240e+09
Institution	2.875215e+09
Private	2.522435e+09

In [39]:
```
df.groupby('Month')[['Sales']].sum().sort_values(by=['Sales'],ascending=False)
```

Out[39]:

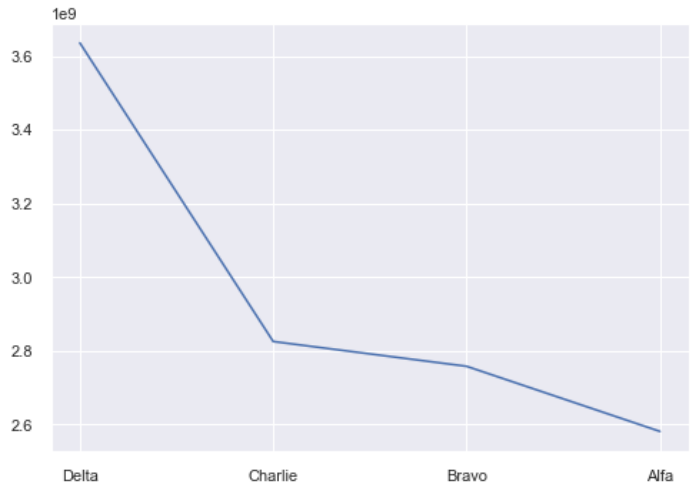
Month	Sales
August	1.186627e+09
November	1.108803e+09
March	1.108802e+09
June	1.064033e+09
July	1.042537e+09
September	1.029988e+09
December	9.750071e+08
February	9.721298e+08
October	9.716484e+08
May	8.651872e+08
April	8.000346e+08
January	6.741911e+08

In [40]:
```
df.groupby('Sales Team')[['Sales']].sum().sort_values(by=['Sales'],ascending=False)
```

Out[40]:

Sales Team	Sales
Delta	3.635341e+09
Charlie	2.824970e+09
Bravo	2.757702e+09
Alfa	2.580974e+09

In [41]:
```
plt.plot(df.groupby('Sales Team')[['Sales']].sum().sort_values(by=['Sales'],ascending=False))
plt.show()
```



In [ ]:
```
(df.groupby('Sales Team')[['Quantity']].sum().sort_values(by=['Quantity'],ascending=False))
```

In [ ]:
```
(df['Manager'].value_counts())
```

In [ ]:
```
df.groupby(['Manager', 'Country'])[['Sales']].sum().sort_values(by=['Sales'],ascending=False)
```

In [ ]:

In [ ]:

In [ ]:
```
#dist_sales = df.groupby('Distributor')[['Sales']].count().sort_values(by= ['Sales'], ascending=False)
#dist_sales.sort_values(ascending=True)
```

In [ ]: