TEAM ID PNT2022TMID21264

Global Sales Data Analytics

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

```
df=pd.read_csv("Global_Superstore2.csv",encoding = "ISO-8859-1")
```

```
df
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	City
0	32298	CA-2012-124891	31-07-2012	31-07-2012	Same Day	RH-19495	Rick Hansen	Consumer	New Yorl City
1	26341	IN-2013-77878	05-02-2013	07-02-2013	Second Class	JR-16210	Justin Ritter	Corporate	Wollongong
2	25330	IN-2013-71249	17-10-2013	18-10-2013	First Class	CR-12730	Craig Reiter	Consumer	Brisbane
3	13524	ES-2013-1570342	28-01-2013	30-01-2013	First Class	KM-16375	Katherine Murray	Home Office	Berlin

```
df.shape
```

```
(51290, 24)
```

```
df.describe()
```

	Row ID	Postal Code	Sales	Quantity	Discount	Profit
count	51290.00000	9994.000000	51290.000000	51290.000000	51290.000000	51290.000000
mean	25645.50000	55190.379428	246.490581	3.476545	0.142908	28.610982
std	14806.29199	32063.693350	487.565361	2.278766	0.212280	174.340972
min	1.00000	1040.000000	0.444000	1.000000	0.000000	-6599.978000
25%	12823.25000	23223.000000	30.758625	2.000000	0.000000	0.000000
50%	25645.50000	56430.500000	85.053000	3.000000	0.000000	9.240000
75%	38467.75000	90008.000000	251.053200	5.000000	0.200000	36.810000

						Class	19795	Baird	Office

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 24 columns):
 #   Column          Non-Null Count  Dtype
```

```
---  ------          --------------  -----
 0   Row ID          51290 non-null  int64
 1   Order ID        51290 non-null  object
 2   Order Date      51290 non-null  object
 3   Ship Date       51290 non-null  object
 4   Ship Mode       51290 non-null  object
 5   Customer ID     51290 non-null  object
 6   Customer Name   51290 non-null  object
 7   Segment         51290 non-null  object
 8   City            51290 non-null  object
 9   State           51290 non-null  object
 10  Country         51290 non-null  object
 11  Postal Code     9994 non-null   float64
 12  Market          51290 non-null  object
 13  Region          51290 non-null  object
 14  Product ID      51290 non-null  object
 15  Category        51290 non-null  object
 16  Sub-Category    51290 non-null  object
 17  Product Name    51290 non-null  object
 18  Sales           51290 non-null  float64
 19  Quantity        51290 non-null  int64
 20  Discount        51290 non-null  float64
 21  Profit          51290 non-null  float64
 22  Shipping Cost   51290 non-null  float64
 23  Order Priority  51290 non-null  object
dtypes: float64(5), int64(2), object(17)
memory usage: 9.4+ MB
```

```
df['Order Date'] = pd.to_datetime(df['Order Date'])
```

```
a=df.groupby(['Order Date', 'Profit'])
a.first()
```

		Row ID	Order ID	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Cit
Order Date	Profit								
2011-01-01	**-26.055**	11731	IT-2011-3647632	05-01-2011	Second Class	EM-14140	Eugene Moren	Home Office	Stockholr
	15.342	22254	IN-2011-47883	08-01-2011	Standard Class	JH-15985	Joseph Holt	Consumer	Wagg Wagg
	29.640	48883	HU-2011-1220	05-01-2011	Second Class	AT-735	Annie Thurman	Consumer	Budapes
	36.036	22253	IN-2011-47883	08-01-2011	Standard Class	JH-15985	Joseph Holt	Consumer	Wagg Wagg
	37.770	22255	IN-2011-47883	08-01-2011	Standard Class	JH-15985	Joseph Holt	Consumer	Wagg Wagg
...	**...**	...	...	...	...	...	...	...	.
2014-12-31	**166.440**	42474	OD-2014-9490	05-01-2015	Standard Class	MW-8235	Mitch Willingham	Corporate	Jub
			ES-	04-					

```
df.isnull().any()
```

```
Row ID          False
Order ID        False
Order Date      False
Ship Date       False
Ship Mode       False
Customer ID     False
Customer Name   False
Segment         False
City            False
State           False
Country         False
Postal Code      True
Market          False
Region          False
```

```
Product ID        False
Category          False
Sub-Category      False
Product Name      False
Sales             False
Quantity          False
Discount          False
Profit            False
Shipping Cost     False
Order Priority    False
dtype: bool
```

```
df.isnull().sum()
```

```
Row ID              0
Order ID            0
Order Date          0
Ship Date           0
Ship Mode           0
Customer ID         0
Customer Name       0
Segment             0
City                0
State               0
Country             0
Postal Code       41296
Market              0
Region              0
Product ID          0
Category            0
Sub-Category        0
Product Name        0
Sales               0
Quantity            0
Discount            0
Profit              0
Shipping Cost       0
Order Priority      0
dtype: int64
```

```
df.nunique()
```

```
Row ID           51290
Order ID         25035
Order Date        1430
Ship Date         1464
Ship Mode            4
Customer ID       1590
Customer Name      795
Segment              3
City              3636
State             1094
Country            147
Postal Code        631
```

```
Market               7
Region              13
Product ID       10292
Category             3
Sub-Category        17
Product Name      3788
Sales            22995
Quantity            14
Discount            27
Profit           24575
Shipping Cost    10037
Order Priority       4
dtype: int64
```

```
df['Ship Mode'] = df['Ship Mode'].astype('category')
df['Segment'] = df['Segment'].astype('category')
df['Country'] = df['Country'].astype('category')
df['Market'] = df['Market'].astype('category')
df['Region'] = df['Region'].astype('category')
df['Category'] = df['Category'].astype('category')
df['Sub-Category'] = df['Sub-Category'].astype('category')
df['Order Priority'] = df['Order Priority'].astype('category')
```

```
def remove_leading_spaces(data):
    for cols in data.columns:
        if data[cols].dtypes in ['object']:
            data[cols] = data[cols].str.strip()
        return data
```

```
data = remove_leading_spaces(df)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 24 columns):
 #   Column          Non-Null Count  Dtype         
---  ------          --------------  -----         
 0   Row ID          51290 non-null  int64         
 1   Order ID        51290 non-null  object        
 2   Order Date      51290 non-null  datetime64[ns]
 3   Ship Date       51290 non-null  object        
 4   Ship Mode       51290 non-null  category      
 5   Customer ID     51290 non-null  object        
 6   Customer Name   51290 non-null  object        
 7   Segment         51290 non-null  category      
 8   City            51290 non-null  object        
 9   State           51290 non-null  object        
 10  Country         51290 non-null  category      
 11  Postal Code     9994 non-null   float64       
 12  Market          51290 non-null  category      
 13  Region          51290 non-null  category      
```

```
 14  Product ID      51290 non-null  object
 15  Category        51290 non-null  category
 16  Sub-Category    51290 non-null  category
 17  Product Name    51290 non-null  object
 18  Sales           51290 non-null  float64
 19  Quantity        51290 non-null  int64
 20  Discount        51290 non-null  float64
 21  Profit          51290 non-null  float64
 22  Shipping Cost   51290 non-null  float64
 23  Order Priority  51290 non-null  category
dtypes: category(8), datetime64[ns](1), float64(5), int64(2), object(8)
memory usage: 6.7+ MB
```

```
data.groupby(['Country']).count()[['Order ID']]
```

Country	Order ID
Afghanistan	55
Albania	16
Algeria	196
Angola	122
Argentina	390
...	...
Venezuela	194
Vietnam	265
Yemen	30
Zambia	102
Zimbabwe	80

147 rows × 1 columns

```
data.groupby(['City']).count()[['Order ID']]
```

City	Order ID
Aachen	17
Aalen	1
Aalst	4
Aba	25
Abadan	11
...	...
Zwedru	1

```
top5 = data.groupby(['Country']).sum()[['Quantity']].nlargest(n=5, columns=['Quantity'])
top5
```

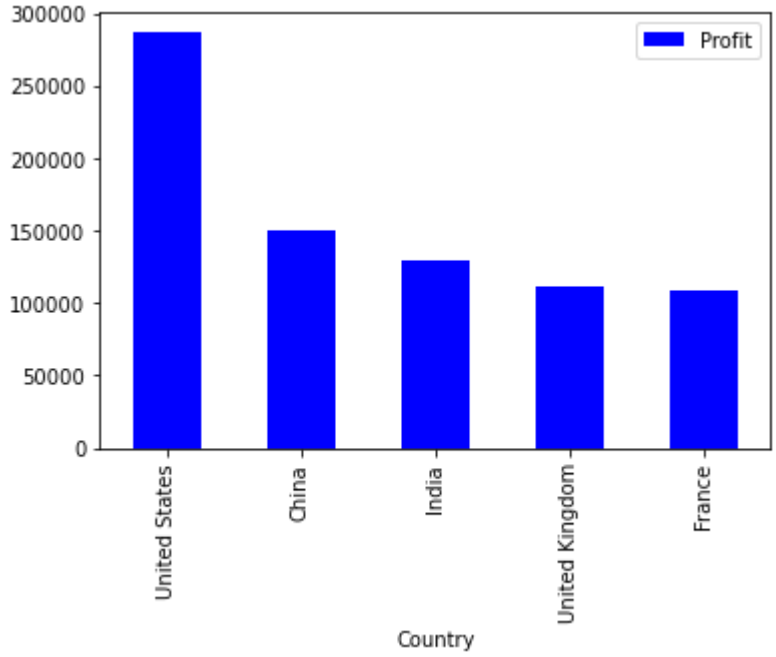
Country	Quantity
United States	37873
France	10804
Australia	10673
Mexico	10011
Germany	7745

```
topprof = data.groupby(['Product Name']).sum()[['Profit']].nlargest(n=5, columns=['Profit'])
```

```
topprof
```

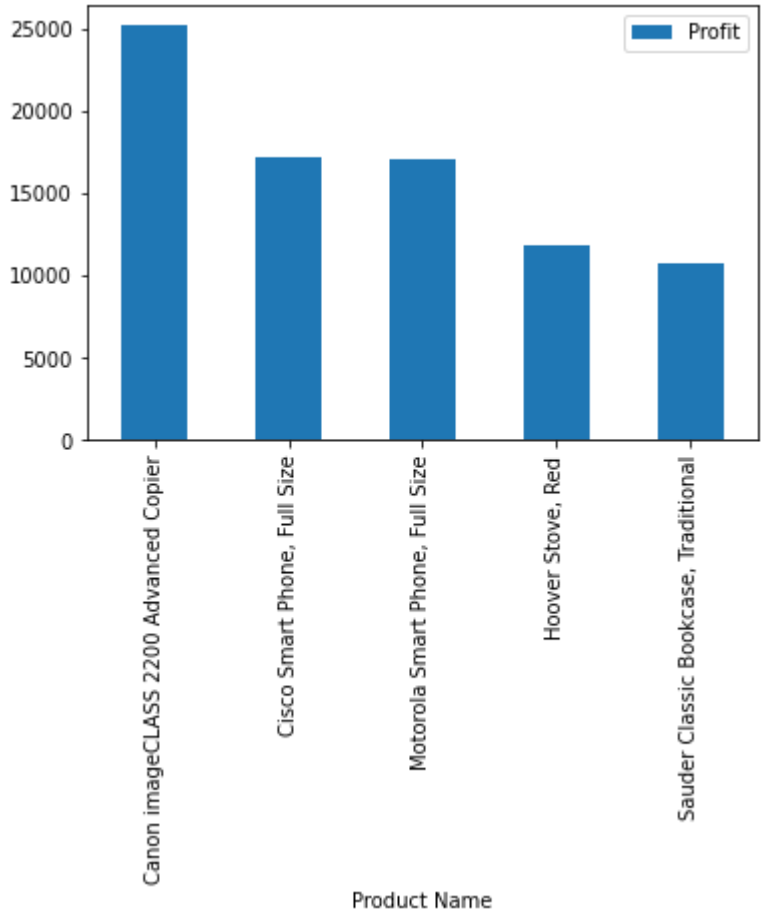
Product Name	Profit
Canon imageCLASS 2200 Advanced Copier	25199.9280
Cisco Smart Phone, Full Size	17238.5206
Motorola Smart Phone, Full Size	17027.1130
Hoover Stove, Red	11807.9690
Sauder Classic Bookcase, Traditional	10672.0730

```
data.groupby(['Country']).sum()[['Profit']].sort_values(by="Profit",ascending=False).nlargest
plt.show()
```




```
data.groupby(['Product Name']).sum()[['Profit']].sort_values(by="Profit",ascending=False).nla
```

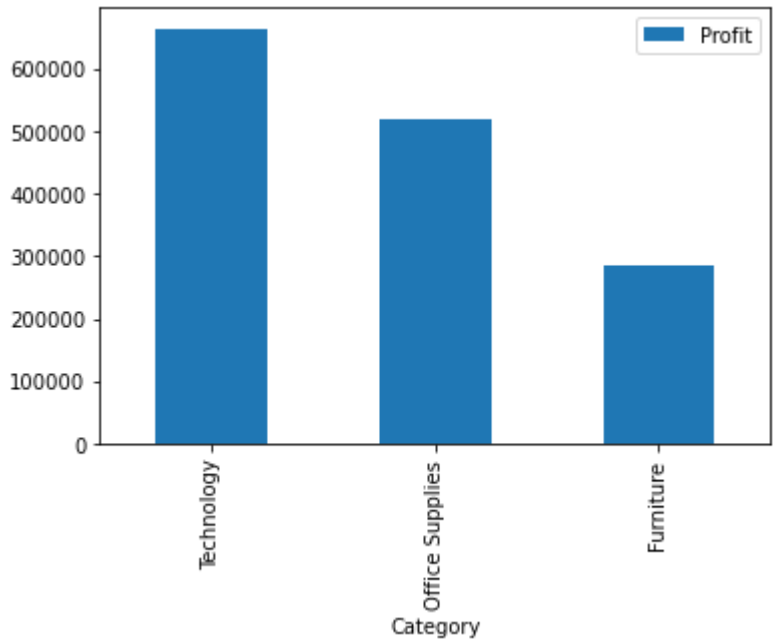
```
<AxesSubplot:xlabel='Product Name'>
```




```
data.groupby('Product Name')['Customer ID'].count().sort_values(ascending=True)
```

```
Product Name
Barricks Coffee Table, with Bottom Storage                                  1
Sanitaire Vibra Groomer IR Commercial Upright Vacuum, Replacement Belts     1
Hewlett-Packard Deskjet 5550 Printer                                        1
Hewlett-Packard Deskjet 3050a All-in-One Color Inkjet Printer               1
Grip Seal Envelopes                                                         1
                                                                          ...
Ibico Index Tab, Clear                                                     83
Rogers File Cart, Single Width                                             84
Eldon File Cart, Single Width                                              90
Cardinal Index Tab, Clear                                                  92
Staples                                                                   227
Name: Customer ID, Length: 3788, dtype: int64
```

```
data.groupby(['Category']).sum()[['Profit']].sort_values(by="Profit",ascending=False).nlarges
plt.show()
```




```
data.groupby(['Category']).sum()[['Profit']].sort_values(by="Profit",ascending=False).nlarges
plt.show()
```