```
! pip install kaggle
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub
Requirement already satisfied: kaggle in /usr/local/lib/python3.7/dist-packages (1.5.12)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (from k
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: python-slugify in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from kagg
Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages (from k
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packag
```

```
import os
os.environ["KAGGLE_CONFIG_DIR"] = "/content"
```

```
! kaggle datasets download -d apoorvaappz/global-super-store-dataset
```

```
Warning: Your Kaggle API key is readable by other users on this system! To fix this, you
Downloading global-super-store-dataset.zip to /content
 45% 5.00M/11.1M [00:00<00:00, 46.1MB/s]
100% 11.1M/11.1M [00:00<00:00, 68.2MB/s]
```

```
!unzip  \*.zip && rm.zip
```

```
Archive:  global-super-store-dataset.zip
  inflating: Global_Superstore2.csv
  inflating: Global_Superstore2.xlsx
/bin/bash: rm.zip: command not found
```

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

%matplotlib inline
```

```
df = pd.read_csv('/content/Global_Superstore2.csv', encoding = 'ISO-8859-1')
df.head()
```

| | 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 York 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- | 18-10- | First Class | CR-12730 | Craig Reiter | Consumer | Brisbane |

```
df.shape
```

```
(51290, 24)
```

```
df.describe()
```

```
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'])
```

```
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  datetime64[ns]
 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: datetime64[ns](1), float64(5), int64(2), object(16)
memory usage: 9.4+ MB
```

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

| | Row ID | Order ID | Ship Date | Ship Mode | Customer ID | Customer Name | Segment | |
|---|---|---|---|---|---|---|---|---|
| Order Date | Profit | | | | | | | |
| **2011-01-01** | **-26.055** | 11731 | IT-2011-3647632 | 05-01-2011 | Second Class | EM-14140 | Eugene Moren | Home Office | St |
| | **15.342** | 22254 | IN-2011-47883 | 08-01-2011 | Standard Class | JH-15985 | Joseph Holt | Consumer | |
| | **29.640** | 48883 | HU-2011-1220 | 05-01-2011 | Second Class | AT-735 | Annie Thurman | Consumer | B |
| | | | IN-2011- | 08- | Standard | | Joseph | | |

```
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.drop(columns='Postal Code', inplace=True)
```

```
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
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.head()
```

| | Row ID | Order ID | Order Date | Ship Date | Ship Mode | Customer ID | Customer Name | Segment | City |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 32298 | CA-2012-124891 | 2012-07-31 | 31-07-2012 | Same Day | RH-19495 | Rick Hansen | Consumer | New York City |
| **1** | 26341 | IN-2013-77878 | 2013-05-02 | 07-02-2013 | Second Class | JR-16210 | Justin Ritter | Corporate | Wollongong |
| **2** | 25330 | IN-2013-71249 | 2013-10-17 | 18-10- | First Class | CR-12730 | Craig Reiter | Consumer | Brisbane |

```
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
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')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 23 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  Market          51290 non-null  category      
 12  Region          51290 non-null  category      
 13  Product ID      51290 non-null  object        
 14  Category        51290 non-null  category      
 15  Sub-Category    51290 non-null  category      
 16  Product Name    51290 non-null  object        
 17  Sales           51290 non-null  float64       
 18  Quantity        51290 non-null  int64         
 19  Discount        51290 non-null  float64       
 20  Profit          51290 non-null  float64       
 21  Shipping Cost   51290 non-null  float64       
 22  Order Priority  51290 non-null  category      
dtypes: category(8), datetime64[ns](1), float64(4), int64(2), object(8)
memory usage: 6.3+ MB
```

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


df = remove_leading_spaces(df)


df.head(3)
```

| | Row ID | Order ID | Order Date | Ship Date | Ship Mode | Customer ID | Customer Name | Segment | City |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 32298 | CA-2012-124891 | 2012-07-31 | 31-07-2012 | Same Day | RH-19495 | Rick Hansen | Consumer | New York City |
| **1** | 26341 | IN-2013- | 2013- | 07-02- | Second | IR-16210 | Justin | Corporate | Wollongong |

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 23 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  Market          51290 non-null  category
 12  Region          51290 non-null  category
 13  Product ID      51290 non-null  object
 14  Category        51290 non-null  category
 15  Sub-Category    51290 non-null  category
 16  Product Name    51290 non-null  object
 17  Sales           51290 non-null  float64
 18  Quantity        51290 non-null  int64
 19  Discount        51290 non-null  float64
 20  Profit          51290 non-null  float64
 21  Shipping Cost   51290 non-null  float64
 22  Order Priority  51290 non-null  category
dtypes: category(8), datetime64[ns](1), float64(4), int64(2), object(8)
memory usage: 6.3+ MB
```

```
df.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

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

| City | Order ID |
|---|---|
| **Aachen** | 17 |
| **Aalen** | 1 |
| **Aalst** | 4 |
| **Aba** | 25 |
| **Abadan** | 11 |
| **...** | ... |
| **Zwedru** | 1 |
| **Zwickau** | 3 |
| **Zwolle** | 2 |
| **eMbalenhle** | 2 |

```
df.groupby(['Product ID']).count()[['Order ID']]
```

| Product ID | Order ID |
|---|---|
| **FUR-ADV-10000002** | 2 |
| **FUR-ADV-10000108** | 3 |
| **FUR-ADV-10000183** | 8 |
| **FUR-ADV-10000188** | 5 |
| **FUR-ADV-10000190** | 1 |
| **...** | ... |
| **TEC-STA-10004181** | 6 |
| **TEC-STA-10004536** | 5 |
| **TEC-STA-10004542** | 5 |
| **TEC-STA-10004834** | 2 |
| **TEC-STA-10004927** | 1 |

10292 rows × 1 columns

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

Double-click (or enter) to edit

```
top5
```

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

```
df.groupby(['Product ID']).count()[['Order ID']].nlargest(n=5, columns=['Order ID'])
```

| Product ID | Order ID |
|---|---|
| **OFF-AR-10003651** | 35 |
| **OFF-AR-10003829** | 31 |
| **OFF-BI-10002799** | 30 |
| **OFF-BI-10003708** | 30 |
| **FUR-CH-10003354** | 28 |

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

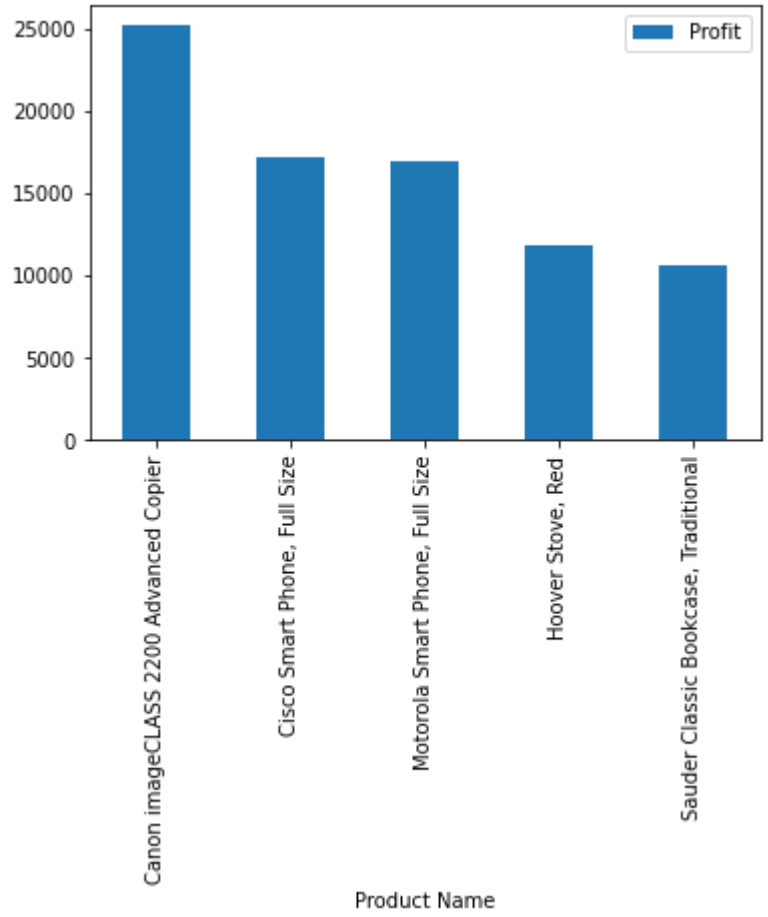
```
df2 = df.groupby(['Product Name']).sum()[['Profit']].nlargest(n=5, columns=['Profit'])
df2
```

| 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 |

## TOP 5 PRODUCT BY TOTAL PROFIT

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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0398aeb350>
```




## TOP 5 COUNTRY BY TOTAL PROFIT

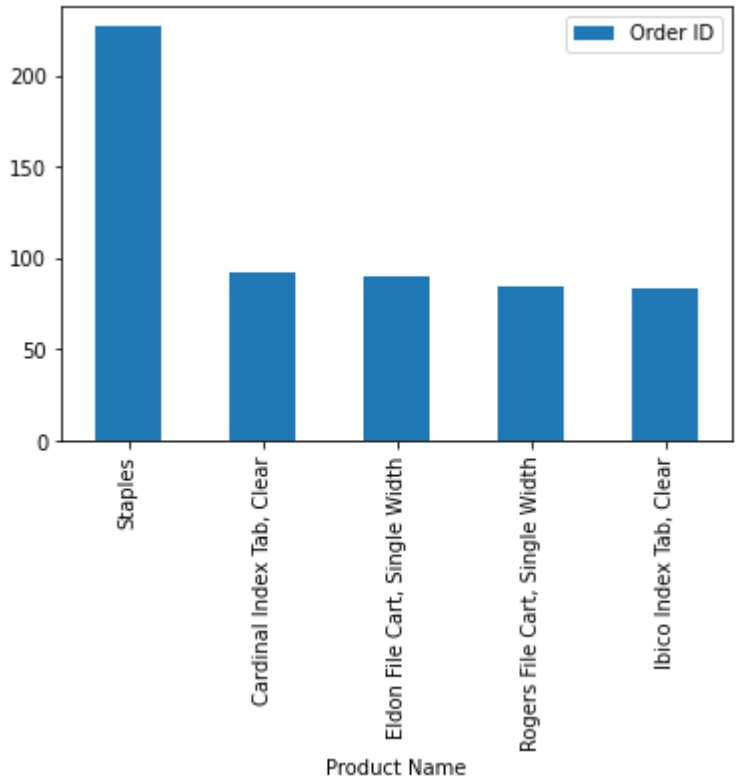
```
df.groupby(['Country']).sum()[['Profit']].sort_values(by="Profit",ascending=False).nlargest(n
plt.show()
```


```
df.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
```

## TOP 5 PRODUCT BY TOTAL ORDER

```
df.groupby(['Product Name']).count()[['Order ID']].sort_values(by="Order ID",ascending=False)
plt.show()
```

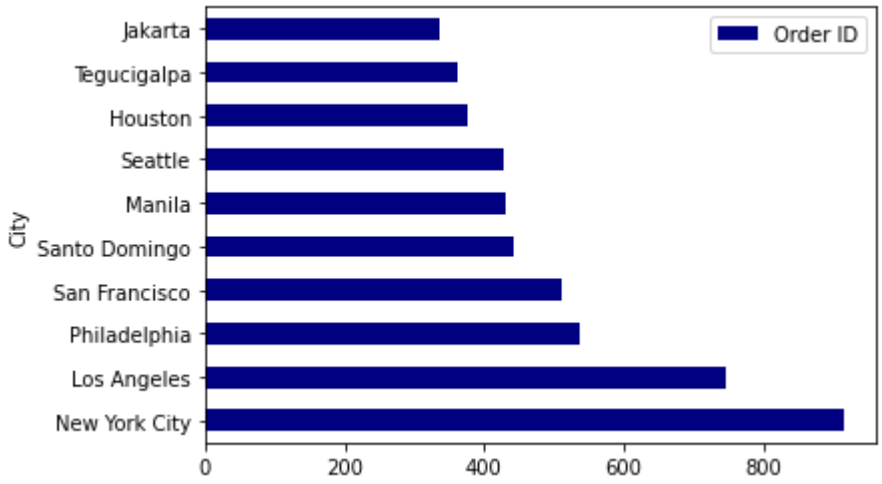


```
df.groupby(['Product Name']).count()[['Order ID']].nlargest(n=5, columns=['Order ID'])
```

| Product Name | Order ID |
|---|---|
| **Staples** | 227 |
| **Cardinal Index Tab, Clear** | 92 |
| **Eldon File Cart, Single Width** | 90 |
| **Rogers File Cart, Single Width** | 84 |
| **Ibico Index Tab, Clear** | 83 |

## TOP 10 CITY BY TOTAL ORDER

```
df.groupby(['City']).count()[['Order ID']].sort_values(by="Order ID",ascending=True).nlargest
plt.show()
```




```
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
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.dropna(axis=0, inplace=True)
```

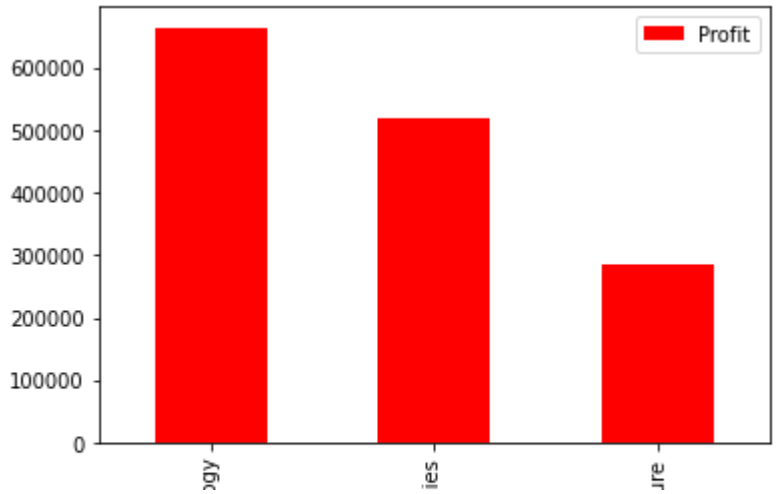
```
df.isnull().sum()
```

```
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
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.shape
```

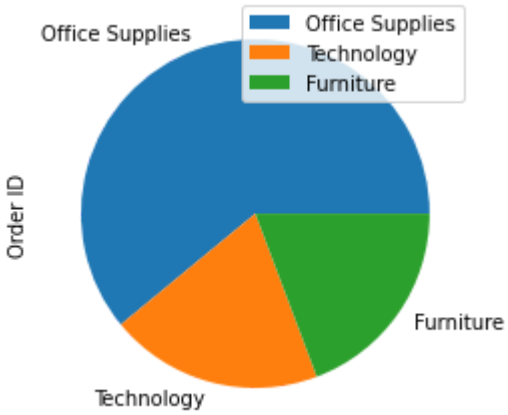
```
(51290, 23)
```

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




## TOTAL ORDER BY CATEGORY

```
df.groupby(['Category']).count()[['Order ID']].sort_values(by="Order ID",ascending=False).nla
plt.show()
```




## TOTAL PROFIT BY CATEGORY

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

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-47-6738504bc90a> in <module>
----> 1 choromap = go.Figure(data = [country_map],layout = layout)
      2 iplot(choromap)

NameError: name 'go' is not defined
```

SEARCH STACK OVERFLOW