In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as mtp
import seaborn as snm
```

In [2]:
```
sample_data=pd.read_csv(r"A:\LLM\Sample - Superstore.csv", encoding="ISO-8859-1")
sample_data.head()
```

Out[2]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region
0	1	CA-2016-152156	08-11-2016	11-11-2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South
1	2	CA-2016-152156	08-11-2016	11-11-2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South
2	3	CA-2016-138688	12-06-2016	16-06-2016	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036	West
3	4	US-2015-108966	11-10-2015	18-10-2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South
4	5	US-2015-108966	11-10-2015	18-10-2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South

5 rows × 21 columns

In [3]:
```
sample_data.shape
```

Out[3]:
```
(9994, 21)
```

In [10]:
```
sample_data.columns
```

Out[10]:
```
Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State',
       'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category',
       'Product Name', 'Sales', 'Quantity', 'Discount', 'Profit'],
      dtype='object')
```

In [4]:
```
sample_data[5:7]
```

Out[4]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	F
5	6	CA-2014-115812	09-06-2014	14-06-2014	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	...	90032	West	F 10
6	7	CA-2014-115812	09-06-2014	14-06-2014	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	...	90032	West	O 10

2 rows × 21 columns

In [5]:
```
#group by using aggregation function

data1=sample_data.groupby('Segment').agg('count')
data1.head()
```

Out[5]:

Segment	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Country	City	State	Postal Code	Region	Product ID
Consumer	5191	5191	5191	5191	5191	5191	5191	5191	5191	5191	5191	5191	5191
Corporate	3020	3020	3020	3020	3020	3020	3020	3020	3020	3020	3020	3020	3020
Home Office	1783	1783	1783	1783	1783	1783	1783	1783	1783	1783	1783	1783	1783

In [6]:
```
#if you want to save your results in csv file then we need to use to_csv function
#make sure to pass this and add csv extension

data1.to_csv("new_analysis.csv")
```

In [7]:
```
import os
print(os.getcwd())
```

```
C:\Users\Gourav Sikka
```

In [9]:
```
sample_data.head()
```

Out[9]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region
0	1	CA-2016-152156	08-11-2016	11-11-2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South
1	2	CA-2016-152156	08-11-2016	11-11-2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South
2	3	CA-2016-138688	12-06-2016	16-06-2016	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036	West
3	4	US-2015-108966	11-10-2015	18-10-2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South
4	5	US-2015-108966	11-10-2015	18-10-2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South

5 rows × 21 columns

In [10]:
```
sample_data.tail
```

Out[10]:
```
<bound method NDFrame.tail of       Row ID        Order ID  Order Date   Ship Date       Sh
ip Mode  \
0          1  CA-2016-152156  08-11-2016  11-11-2016    Second Class
1          2  CA-2016-152156  08-11-2016  11-11-2016    Second Class
2          3  CA-2016-138688  12-06-2016  16-06-2016    Second Class
3          4  US-2015-108966  11-10-2015  18-10-2015  Standard Class
4          5  US-2015-108966  11-10-2015  18-10-2015  Standard Class
...      ...             ...         ...         ...             ...
9989    9990  CA-2014-110422  21-01-2014  23-01-2014    Second Class
9990    9991  CA-2017-121258  26-02-2017  03-03-2017  Standard Class
9991    9992  CA-2017-121258  26-02-2017  03-03-2017  Standard Class
9992    9993  CA-2017-121258  26-02-2017  03-03-2017  Standard Class
9993    9994  CA-2017-119914  04-05-2017  09-05-2017    Second Class

     Customer ID     Customer Name    Segment        Country             City  \
0       CG-12520       Claire Gute   Consumer  United States        Henderson
1       CG-12520       Claire Gute   Consumer  United States        Henderson
2       DV-13045   Darrin Van Huff  Corporate  United States      Los Angeles
3       SO-20335    Sean O'Donnell   Consumer  United States  Fort Lauderdale
4       SO-20335    Sean O'Donnell   Consumer  United States  Fort Lauderdale
...          ...               ...        ...            ...              ...
9989    TB-21400  Tom Boeckenhauer   Consumer  United States            Miami
9990    DB-13060       Dave Brooks   Consumer  United States       Costa Mesa
9991    DB-13060       Dave Brooks   Consumer  United States       Costa Mesa
9992    DB-13060       Dave Brooks   Consumer  United States       Costa Mesa
9993    CC-12220      Chris Cortes   Consumer  United States      Westminster

      ... Postal Code  Region       Product ID         Category Sub-Category  \
0     ...       42420   South  FUR-BO-10001798        Furniture    Bookcases
1     ...       42420   South  FUR-CH-10000454        Furniture       Chairs
2     ...       90036    West  OFF-LA-10000240  Office Supplies       Labels
3     ...       33311   South  FUR-TA-10000577        Furniture       Tables
4     ...       33311   South  OFF-ST-10000760  Office Supplies      Storage
...   ...         ...     ...              ...              ...          ...
9989  ...       33180   South  FUR-FU-10001889        Furniture  Furnishings
9990  ...       92627    West  FUR-FU-10000747        Furniture  Furnishings
9991  ...       92627    West  TEC-PH-10003645       Technology       Phones
9992  ...       92627    West  OFF-PA-10004041  Office Supplies        Paper
9993  ...       92683    West  OFF-AP-10002684  Office Supplies   Appliances

                                        Product Name     Sales  Quantity  \
0                    Bush Somerset Collection Bookcase  261.9600         2
1     Hon Deluxe Fabric Upholstered Stacking Chairs,...  731.9400         3
2     Self-Adhesive Address Labels for Typewriters b...   14.6200         2
3         Bretford CR4500 Series Slim Rectangular Table  957.5775         5
4                       Eldon Fold 'N Roll Cart System   22.3680         2
...                                                ...       ...       ...
9989                            Ultra Door Pull Handle   25.2480         3
9990  Tenex B1-RE Series Chair Mats for Low Pile Car...   91.9600         2
9991                              Aastra 57i VoIP phone  258.5760         2
9992  It's Hot Message Books with Stickers, 2 3/4" x 5"   29.6000         4
9993  Acco 7-Outlet Masterpiece Power Center, Wihtou...  243.1600         2

      Discount    Profit
0         0.00   41.9136
1         0.00  219.5820
2         0.00    6.8714
3         0.45 -383.0310
4         0.20    2.5164
...        ...       ...
9989      0.20    4.1028
9990      0.00   15.6332
9991      0.20   19.3932
9992      0.00   13.3200
9993      0.00   72.9480

[9994 rows x 21 columns]>
```

In [11]:
```
sample_data[2:3]
```

Out[11]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Pr
2	3	CA-2016-138688	12-06-2016	16-06-2016	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036	West	OF 100

1 rows × 21 columns

In [12]:
```
sample_data.columns
```

Out[12]:
```
Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State',
       'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category',
       'Product Name', 'Sales', 'Quantity', 'Discount', 'Profit'],
      dtype='object')
```

In [15]:
```
#if we need only only column then we need to do like this

sample_data["Sales"]
```

Out[15]:
```
0       261.9600
1       731.9400
2        14.6200
3       957.5775
4        22.3680
          ...
9989     25.2480
9990     91.9600
9991    258.5760
9992     29.6000
9993    243.1600
Name: Sales, Length: 9994, dtype: float64
```

In [17]:
```
#if we need only more columns then we need to do like this

sample_data[["Sales","Category"]]
```

Out[17]:

	Sales	Category
0	261.9600	Furniture
1	731.9400	Furniture
2	14.6200	Office Supplies
3	957.5775	Furniture
4	22.3680	Office Supplies
...	...	...
9989	25.2480	Furniture
9990	91.9600	Furniture
9991	258.5760	Technology
9992	29.6000	Office Supplies
9993	243.1600	Office Supplies

9994 rows × 2 columns

In [18]:
```
sample_data.describe()
```

Out[18]:

	Row ID	Postal Code	Sales	Quantity	Discount	Profit
count	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000
mean	4997.500000	55190.379428	229.858001	3.789574	0.156203	28.656896
std	2885.163629	32063.693350	623.245101	2.225110	0.206452	234.260108
min	1.000000	1040.000000	0.444000	1.000000	0.000000	-6599.978000
25%	2499.250000	23223.000000	17.280000	2.000000	0.000000	1.728750
50%	4997.500000	56430.500000	54.490000	3.000000	0.200000	8.666500
75%	7495.750000	90008.000000	209.940000	5.000000	0.200000	29.364000
max	9994.000000	99301.000000	22638.480000	14.000000	0.800000	8399.976000

In [20]:
```
#describe for particulat 1 column
sample_data["Sales"].describe()
```

Out[20]:
```
count     9994.000000
mean       229.858001
std        623.245101
min          0.444000
25%         17.280000
50%         54.490000
75%        209.940000
max      22638.480000
Name: Sales, dtype: float64
```

In [21]:
```
#min for one variable
sample_data["Sales"].min()
```

Out[21]:
```
0.444
```

In [22]:
```
#max for one variable
sample_data["Sales"].max()
```

Out[22]:
```
22638.48
```

In [23]:
```
#I dodnot want maxium value, I want comple row
sample_data["Sales"]==sample_data["Sales"].max()
```

Out[23]:
```
0       False
1       False
2       False
3       False
4       False
        ...
9989    False
9990    False
9991    False
9992    False
9993    False
Name: Sales, Length: 9994, dtype: bool
```

In [24]:
```
#I do want maxium value, I want comple row
#great code
sample_data[sample_data["Sales"]==sample_data["Sales"].max()]
```

Out[24]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Regi
2697	2698	CA-2014-145317	18-03-2014	23-03-2014	Standard Class	SM-20320	Sean Miller	Home Office	United States	Jacksonville	...	32216	Sou

1 rows × 21 columns

In [26]:
```
#I do want maxium value, I want comple row
#great code
sample_data['Ship Mode'][sample_data["Sales"]==sample_data["Sales"].max()]
```

Out[26]:
```
2697    Standard Class
Name: Ship Mode, dtype: object
```

In [30]:
```
#how to apply filter in pandas
sample_data[sample_data["Sales"]==258.5760]
```

Out[30]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region
5179	5180	US-2015-120502	13-04-2015	19-04-2015	Standard Class	BT-11395	Bill Tyler	Corporate	United States	Los Angeles	...	90036	West
9991	9992	CA-2017-121258	26-02-2017	03-03-2017	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	92627	West

2 rows × 21 columns

In [33]:
```
sample_data.columns
```

Out[33]:
```
Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State',
       'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category',
       'Product Name', 'Sales', 'Quantity', 'Discount', 'Profit'],
      dtype='object')
```

In [38]:
```
#group by function with max function

group_date_category=sample_data.groupby('Category').max()
group_date_category
```

Out[38]:

Category	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Post Co
Furniture	9991	US-2017-169551	31-12-2015	31-12-2017	Standard Class	ZD-21925	Zuschuss Donatelli	Home Office	United States	York	Wyoming	993(
Office Supplies	9994	US-2017-169551	31-12-2016	31-12-2017	Standard Class	ZD-21925	Zuschuss Donatelli	Home Office	United States	Yuma	Wisconsin	993(
Technology	9992	US-2017-169551	31-12-2016	31-12-2016	Standard Class	ZD-21925	Zuschuss Donatelli	Home Office	United States	Yuma	Wisconsin	992(

In [39]:
```
group_date_category.head()
```

Out[39]:

Category	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Post Co
Furniture	9991	US-2017-169551	31-12-2015	31-12-2017	Standard Class	ZD-21925	Zuschuss Donatelli	Home Office	United States	York	Wyoming	993(
Office Supplies	9994	US-2017-169551	31-12-2016	31-12-2017	Standard Class	ZD-21925	Zuschuss Donatelli	Home Office	United States	Yuma	Wisconsin	993(
Technology	9992	US-2017-169551	31-12-2016	31-12-2016	Standard Class	ZD-21925	Zuschuss Donatelli	Home Office	United States	Yuma	Wisconsin	992(

In [41]:
```
#group by function with max function
sample_data.groupby("City").min()
```

Out[41]:

City	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	State	Postal Code
Aberdeen	9261	CA-2017-167976	11-11-2017	14-11-2017	Second Class	JL-15505	Jeremy Lonsdale	Consumer	United States	South Dakota	57401
Abilene	6990	CA-2017-165099	11-12-2017	13-12-2017	First Class	DK-13375	Dennis Kane	Consumer	United States	Texas	79605
Akron	201	CA-2014-111360	04-09-2017	09-09-2017	First Class	AT-10435	Alyssa Tate	Consumer	United States	Ohio	44312
Albuquerque	2417	CA-2014-100881	10-10-2014	01-01-2018	Same Day	BF-11215	Benjamin Farhat	Consumer	United States	New Mexico	87105
Alexandria	921	CA-2014-102988	04-01-2015	09-01-2015	Second Class	AG-10495	Andrew Gjertsen	Consumer	United States	Virginia	22304
...	...	...	...	...	...	...	...	...	...	...	...
Woonsocket	3990	CA-2014-122588	12-09-2017	15-09-2017	First Class	AR-10540	Andy Reiter	Consumer	United States	Rhode Island	2895
Yonkers	1083	CA-2014-135755	01-03-2016	05-03-2016	Second Class	AB-10150	Aimee Bixby	Consumer	United States	New York	10701
York	2146	CA-2017-169404	09-04-2017	14-04-2017	Standard Class	NC-18625	Noah Childs	Corporate	United States	Pennsylvania	17403
Yucaipa	3348	CA-2015-112014	13-08-2015	20-08-2015	Standard Class	ON-18715	Odella Nelson	Corporate	United States	California	92399
Yuma	4823	CA-2016-106621	10-07-2016	01-10-2016	Same Day	DM-12955	Dario Medina	Corporate	United States	Arizona	85364

531 rows × 20 columns

In [47]:
```
#How to concatenate_data in pandas

data1=sample_data.groupby('Category').count()
data1

data2=sample_data.groupby("Sub-Category").count()
data2
```

Out[47]:

Sub-Category	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Postal Code	Regi
Accessories	775	775	775	775	775	775	775	775	775	775	775	775	7
Appliances	466	466	466	466	466	466	466	466	466	466	466	466	4
Art	796	796	796	796	796	796	796	796	796	796	796	796	7
Binders	1523	1523	1523	1523	1523	1523	1523	1523	1523	1523	1523	1523	15
Bookcases	228	228	228	228	228	228	228	228	228	228	228	228	2
Chairs	617	617	617	617	617	617	617	617	617	617	617	617	6
Copiers	68	68	68	68	68	68	68	68	68	68	68	68	
Envelopes	254	254	254	254	254	254	254	254	254	254	254	254	2
Fasteners	217	217	217	217	217	217	217	217	217	217	217	217	2
Furnishings	957	957	957	957	957	957	957	957	957	957	957	957	9
Labels	364	364	364	364	364	364	364	364	364	364	364	364	3
Machines	115	115	115	115	115	115	115	115	115	115	115	115	1
Paper	1370	1370	1370	1370	1370	1370	1370	1370	1370	1370	1370	1370	13
Phones	889	889	889	889	889	889	889	889	889	889	889	889	8
Storage	846	846	846	846	846	846	846	846	846	846	846	846	8
Supplies	190	190	190	190	190	190	190	190	190	190	190	190	1
Tables	319	319	319	319	319	319	319	319	319	319	319	319	3

In [50]:
```
data_concate=pd.concat([data1,data2])
data_concate
```

Out[50]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region
Furniture	2121	2121	2121	2121	2121	2121	2121	2121	2121	2121	...	2121	2121
Office Supplies	6026	6026	6026	6026	6026	6026	6026	6026	6026	6026	...	6026	6026
Technology	1847	1847	1847	1847	1847	1847	1847	1847	1847	1847	...	1847	1847
Accessories	775	775	775	775	775	775	775	775	775	775	...	775	775
Appliances	466	466	466	466	466	466	466	466	466	466	...	466	466
Art	796	796	796	796	796	796	796	796	796	796	...	796	796
Binders	1523	1523	1523	1523	1523	1523	1523	1523	1523	1523	...	1523	1523
Bookcases	228	228	228	228	228	228	228	228	228	228	...	228	228
Chairs	617	617	617	617	617	617	617	617	617	617	...	617	617
Copiers	68	68	68	68	68	68	68	68	68	68	...	68	68
Envelopes	254	254	254	254	254	254	254	254	254	254	...	254	254
Fasteners	217	217	217	217	217	217	217	217	217	217	...	217	217
Furnishings	957	957	957	957	957	957	957	957	957	957	...	957	957
Labels	364	364	364	364	364	364	364	364	364	364	...	364	364
Machines	115	115	115	115	115	115	115	115	115	115	...	115	115
Paper	1370	1370	1370	1370	1370	1370	1370	1370	1370	1370	...	1370	1370
Phones	889	889	889	889	889	889	889	889	889	889	...	889	889
Storage	846	846	846	846	846	846	846	846	846	846	...	846	846
Supplies	190	190	190	190	190	190	190	190	190	190	...	190	190
Tables	319	319	319	319	319	319	319	319	319	319	...	319	319

20 rows × 21 columns

In [51]:
```
#merge or join in pandas

sample_data.columns
```

Out[51]:
```
Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State',
       'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category',
       'Product Name', 'Sales', 'Quantity', 'Discount', 'Profit'],
      dtype='object')
```

In [52]:
```
sample_data.shape
```

Out[52]:
```
(9994, 21)
```

In [53]:
```
sample_retrun_data=pd.read_csv("A:\LLM\Return_Superstore.csv")
```

In [54]:
```
sample_retrun_data.columns
```

Out[54]:
```
Index(['Returned', 'Order ID'], dtype='object')
```

In [55]:
```
#Let perform merge or join in pandas

data3=pd.merge(sample_data,sample_retrun_data, on='Order ID' )
```

In [58]:
```
data3.head()
data3.columns
```

Out[58]:
```
Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State',
       'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category',
       'Product Name', 'Sales', 'Quantity', 'Discount', 'Profit', 'Returned'],
      dtype='object')
```

In [60]:
```
#Let perform merge or join in pandas right join

data4=pd.merge(sample_data,sample_retrun_data, on='Order ID',how='right' )
```

In [61]:
```
data4
```

Out[61]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Region	Produ
0	6971	CA-2017-153822	19-09-2017	25-09-2017	Standard Class	AB-10105	Adrian Barton	Consumer	United States	Phoenix	...	West	OFF-S 100003
1	6972	CA-2017-153822	19-09-2017	25-09-2017	Standard Class	AB-10105	Adrian Barton	Consumer	United States	Phoenix	...	West	TEC-A 100001
2	6973	CA-2017-153822	19-09-2017	25-09-2017	Standard Class	AB-10105	Adrian Barton	Consumer	United States	Phoenix	...	West	TEC-P 100024
3	6974	CA-2017-153822	19-09-2017	25-09-2017	Standard Class	AB-10105	Adrian Barton	Consumer	United States	Phoenix	...	West	OFF-I 100014
4	5434	CA-2017-129707	25-04-2017	29-04-2017	Standard Class	LH-16750	Larry Hughes	Consumer	United States	Chandler	...	West	OFF-A 100002
...	...	...	...	...	...	...	...	...	...	...	...	...	...
795	993	CA-2016-105585	26-08-2016	27-08-2016	First Class	RF-19735	Roland Fjeld	Consumer	United States	San Jose	...	West	OFF-F 100029
796	994	CA-2016-105585	26-08-2016	27-08-2016	First Class	RF-19735	Roland Fjeld	Consumer	United States	San Jose	...	West	OFF-F 100036
797	418	CA-2016-148796	14-04-2016	18-04-2016	Standard Class	PB-19150	Philip Brown	Consumer	United States	Los Angeles	...	West	FUR-C 100048
798	7717	CA-2015-149636	06-01-2015	12-01-2015	Standard Class	SP-20620	Stefania Perrino	Corporate	United States	Colorado Springs	...	West	OFF-F 100040
799	7718	CA-2015-149636	06-01-2015	12-01-2015	Standard Class	SP-20620	Stefania Perrino	Corporate	United States	Colorado Springs	...	West	OFF-I 100027

800 rows × 22 columns

In [64]:
```
#let perform merge or join in pandas left join

data5=pd.merge(sample_data,sample_retrun_data, on='Order ID',how='left' )
data5.head()
```

Out[64]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Region	Produ
0	1	CA-2016-152156	08-11-2016	11-11-2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	South	FUR-BO 1000179
1	2	CA-2016-152156	08-11-2016	11-11-2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	South	FUR-CH 1000045
2	3	CA-2016-138688	12-06-2016	16-06-2016	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	West	OFF-LA 1000024
3	4	US-2015-108966	11-10-2015	18-10-2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	South	FUR-TA 1000057
4	5	US-2015-108966	11-10-2015	18-10-2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	South	OFF-ST 1000076

5 rows × 22 columns

In [65]:
```
data5.shape
```

Out[65]: (9994, 22)

In [72]:
```
#if you want to select particular coloumns in dataframes
#make sure of this comma
data5.iloc[:,[0,1,2]]
```

Out[72]:

	Row ID	Order ID	Order Date
0	1	CA-2016-152156	08-11-2016
1	2	CA-2016-152156	08-11-2016
2	3	CA-2016-138688	12-06-2016
3	4	US-2015-108966	11-10-2015
4	5	US-2015-108966	11-10-2015
...	...	...	...
9989	9990	CA-2014-110422	21-01-2014
9990	9991	CA-2017-121258	26-02-2017
9991	9992	CA-2017-121258	26-02-2017
9992	9993	CA-2017-121258	26-02-2017
9993	9994	CA-2017-119914	04-05-2017

9994 rows × 3 columns

In [78]:
```
#if you want to apply data sort by order by after join or merge

data5=pd.merge(sample_data,sample_retrun_data, on='Order ID',how='left' )
data5.head()
data5.sort_values(by='Sales', ascending=False)
```

Out[78]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Region	P
2697	2698	CA-2014-145317	18-03-2014	23-03-2014	Standard Class	SM-20320	Sean Miller	Home Office	United States	Jacksonville	...	South	TE 10(
6826	6827	CA-2016-118689	02-10-2016	09-10-2016	Standard Class	TC-20980	Tamara Chand	Corporate	United States	Lafayette	...	Central	TE 10(
8153	8154	CA-2017-140151	23-03-2017	25-03-2017	First Class	RB-19360	Raymond Buch	Consumer	United States	Seattle	...	West	TE 10(
2623	2624	CA-2017-127180	22-10-2017	24-10-2017	First Class	TA-21385	Tom Ashbrook	Home Office	United States	New York City	...	East	TE 10(
4190	4191	CA-2017-166709	17-11-2017	22-11-2017	Standard Class	HL-15040	Hunter Lopez	Consumer	United States	Newark	...	East	TE 10(
...	...	...	...	...	...	...	...	...	...	...	...	...	
2106	2107	US-2014-152723	26-09-2014	26-09-2014	Same Day	HG-14965	Henry Goldwyn	Corporate	United States	Mesquite	...	Central	C 10(
4711	4712	CA-2014-112403	31-03-2014	31-03-2014	Same Day	JO-15280	Jas O'Carroll	Consumer	United States	Philadelphia	...	East	C 10(
8658	8659	CA-2016-168361	21-06-2016	25-06-2016	Standard Class	KB-16600	Ken Brennan	Corporate	United States	Chicago	...	Central	C 10(
9292	9293	CA-2017-124114	02-03-2017	02-03-2017	Same Day	RS-19765	Roland Schwarz	Corporate	United States	Waco	...	Central	C 10(
4101	4102	US-2017-102288	19-06-2017	23-06-2017	Standard Class	ZC-21910	Zuschuss Carroll	Consumer	United States	Houston	...	Central	OI 10(

9994 rows × 22 columns

In [81]:
```
#for example if we want to implement row_number here

sample_data_row_number=sample_data["Sales"].rank(method='first', ascending=False)
```

In [82]:
```
sample_data_row_number.shape
```

Out[82]: (9994,)

In [83]:
```
sample_data_row_number.columns
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
C:\Users\GOURAV~1\AppData\Local\Temp/ipykernel_21332/3095762944.py in <module>
----> 1 sample_data_row_number.columns

~\anaconda3\lib\site-packages\pandas\core\generic.py in __getattr__(self, name)
   5485         ):
   5486             return self[name]
-> 5487         return object.__getattribute__(self, name)
   5488
   5489     def __setattr__(self, name: str, value) -> None:

AttributeError: 'Series' object has no attribute 'columns'
```

In [93]:
```
sample_data["ranking"]=sample_data["Sales"].rank(method='first', ascending=False).astype(int
```

In [94]:
```
sample_data.shape
```

Out[94]:
```
(9994, 22)
```

In [95]:
```
sample_data.columns
```

Out[95]:
```
Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State',
       'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category',
       'Product Name', 'Sales', 'Quantity', 'Discount', 'Profit', 'ranking'],
      dtype='object')
```

In [98]:
```
sample_data['ranking']
```

Out[98]:
```
0       2121
1        719
2       7871
3        500
4       6920
        ...
9989    6662
9990    3937
9991    2140
9992    6344
9993    2242
Name: ranking, Length: 9994, dtype: int32
```

In [99]:
```
sample_data['ranking']==5
```

Out[99]:
```
0       False
1       False
2       False
3       False
4       False
        ...
9989    False
9990    False
9991    False
9992    False
9993    False
Name: ranking, Length: 9994, dtype: bool
```

In [101]:
```
sample_data[sample_data['ranking']==1]
```

Out[101]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Region	Pr
2697	2698	CA-2014-145317	18-03-2014	23-03-2014	Standard Class	SM-20320	Sean Miller	Home Office	United States	Jacksonville	...	South	TEC 1000

1 rows × 22 columns

In [106]:
```
#for example if I need 1 and 2 ranking then we can use isin function

sample_data[sample_data['ranking'].isin([1,2])]
```

Out[106]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Region	Pr
2697	2698	CA-2014-145317	18-03-2014	23-03-2014	Standard Class	SM-20320	Sean Miller	Home Office	United States	Jacksonville	...	South	TE 100
6826	6827	CA-2016-118689	02-10-2016	09-10-2016	Standard Class	TC-20980	Tamara Chand	Corporate	United States	Lafayette	...	Central	TE 100

2 rows × 22 columns

In [ ]: