

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

```
#Reading data
```

```
USA_data = pd.read_excel('Superstore_USA.xlsx')
```

```
USA_data.head()
```

	Row ID	Order ID	Priority	Discount	Unit Price	Shipping Cost
Customer ID \	0	18606	Not Specified	0.01	2.88	0.50
2	1	20847	High	0.01	2.84	0.93
3	2	23086	Not Specified	0.03	6.68	6.15
3	3	23087	Not Specified	0.01	5.68	3.60
3	4	23088	Not Specified	0.00	205.99	2.50
3						

	Customer Name	Ship Mode	Customer Segment	Product Category	...
\	0	Janice Fletcher	Regular Air	Corporate	Office Supplies ...
1	Bonnie Potter	Express Air	Corporate	Office Supplies ...	
2	Bonnie Potter	Express Air	Corporate	Office Supplies ...	
3	Bonnie Potter	Regular Air	Corporate	Office Supplies ...	
4	Bonnie Potter	Express Air	Corporate	Technology ...	

	Region	State or Province	City	Postal Code	Order Date	Ship Date
Date \	0	Central	Illinois	Addison	60101	2012-05-28 2012-05-30
1	West	Washington	Anacortes	98221	2010-07-07	2010-07-08
2	West	Washington	Anacortes	98221	2011-07-27	2011-07-28
3	West	Washington	Anacortes	98221	2011-07-27	2011-07-28
4	West	Washington	Anacortes	98221	2011-07-27	2011-07-27

	Profit	Quantity ordered	new	Sales	Order ID
0	1.3200		2	5.90	88525
1	4.5600		4	13.01	88522

2	-47.6400	7	49.92	88523
3	-30.5100	7	41.64	88523
4	998.2023	8	1446.67	88523

[5 rows x 24 columns]

#Checked Whether there are null values or not

USA_data.isnull().sum()

Row ID	0
Order Priority	0
Discount	0
Unit Price	0
Shipping Cost	0
Customer ID	0
Customer Name	0
Ship Mode	0
Customer Segment	0
Product Category	0
Product Sub-Category	0
Product Container	0
Product Name	0
Product Base Margin	72
Region	0
State or Province	0
City	0
Postal Code	0
Order Date	0
Ship Date	0
Profit	0
Quantity ordered new	0
Sales	0
Order ID	0

dtype: int64

There are 72 null values in Product Base Margin Column

#Checking datatypes of each column

USA_data.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 9426 entries, 0 to 9425

Data columns (total 24 columns):

#	Column	Non-Null Count	Dtype
0	Row ID	9426 non-null	int64
1	Order Priority	9426 non-null	object
2	Discount	9426 non-null	float64
3	Unit Price	9426 non-null	float64
4	Shipping Cost	9426 non-null	float64

```

5 Customer ID          9426 non-null    int64
6 Customer Name        9426 non-null    object
7 Ship Mode            9426 non-null    object
8 Customer Segment     9426 non-null    object
9 Product Category     9426 non-null    object
10 Product Sub-Category 9426 non-null    object
11 Product Container    9426 non-null    object
12 Product Name        9426 non-null    object
13 Product Base Margin  9354 non-null    float64
14 Region              9426 non-null    object
15 State or Province   9426 non-null    object
16 City                9426 non-null    object
17 Postal Code         9426 non-null    int64
18 Order Date          9426 non-null    datetime64[ns]
19 Ship Date           9426 non-null    datetime64[ns]
20 Profit              9426 non-null    float64
21 Quantity ordered new 9426 non-null    int64
22 Sales               9426 non-null    float64
23 Order ID            9426 non-null    int64
dtypes: datetime64[ns](2), float64(6), int64(5), object(11)
memory usage: 1.7+ MB

```

```

#Replacing null values with 0
USA_data['Product Base Margin'].fillna(0)

```

```

0      0.36
1      0.54
2      0.37
3      0.56
4      0.59
...
9421   0.40
9422   0.54
9423   0.71
9424   0.65
9425   0.38

```

```
Name: Product Base Margin, Length: 9426, dtype: float64
```

```
#Checking unique values in Product Category
```

```
USA_data['Product Category'].unique()
```

```
array(['Office Supplies', 'Technology', 'Furniture'], dtype=object)
```

```
#Checking Whether there are any null values left or not
```

```
USA_data['Product Base Margin'].isnull().sum()
```

```
0
```

```
USA_data.describe()
```

	Row ID	Discount	Unit Price	Shipping Cost	Customer
ID \					
count	9426.000000	9426.000000	9426.000000	9426.000000	
9426.000000					
mean	20241.015277	0.049628	88.303686	12.795142	
1738.422236					
min	2.000000	0.000000	0.990000	0.490000	
2.000000					
25%	19330.250000	0.020000	6.480000	3.192500	
898.000000					
50%	21686.500000	0.050000	20.990000	6.050000	
1750.000000					
75%	24042.750000	0.080000	85.990000	13.990000	
2578.750000					
max	26399.000000	0.250000	6783.020000	164.730000	
3403.000000					
std	6101.890965	0.031798	281.540982	17.181203	
979.167197					
	Product Base Margin	Postal Code			Order
Date \					
count	9426.000000	9426.000000			
9426					
mean	0.508277	52446.327286	2012-03-05		
18:33:59.465308672					
min	0.000000	1001.000000		2010-01-01	
00:00:00					
25%	0.380000	29406.000000		2011-03-07	
06:00:00					
50%	0.520000	52302.000000		2012-04-08	
00:00:00					
75%	0.590000	78516.000000		2013-03-26	
00:00:00					
max	0.850000	99362.000000		2013-12-31	
00:00:00					
std	0.141901	29374.597802			
NaN					
	Ship Date	Profit	Quantity ordered		
new \					
count	9426	9426.000000			
9426.000000					
mean	2012-03-07 19:16:18.485041408	139.236410			
13.798430					
min	2010-01-02 00:00:00	-16476.838000			
1.000000					
25%	2011-03-09 00:00:00	-74.017375			
5.000000					
50%	2012-04-09 00:00:00	2.567600			
10.000000					

75%	2013-03-28 00:00:00	140.243850
17.000000		
max	2014-01-17 00:00:00	16332.414000
170.000000		
std	NaN	998.486483
15.107688		

	Sales	Order ID	order year
count	9426.000000	9426.000000	9426.000000
mean	949.706272	82318.489073	2011.678655
min	1.320000	6.000000	2010.000000
25%	61.282500	86737.250000	2011.000000
50%	203.455000	88344.500000	2012.000000
75%	776.402500	89987.750000	2013.000000
max	100119.160000	91591.000000	2013.000000
std	2598.019818	19149.448857	1.131542

#Checking unique values in Product Sub-Category

```
USA_data['Product Sub-Category'].unique()
```

```
array(['Labels', 'Pens & Art Supplies', 'Paper',
       'Scissors, Rulers and Trimmers', 'Telephones and
       Communication',
       'Office Machines', 'Chairs & Chairmats', 'Appliances',
       'Bookcases',
       'Tables', 'Envelopes', 'Office Furnishings', 'Rubber Bands',
       'Binders and Binder Accessories', 'Storage & Organization',
       'Computer Peripherals', 'Copiers and Fax'], dtype=object)
```

#Checking total Order Count

```
USA_data['Order ID'].count()
```

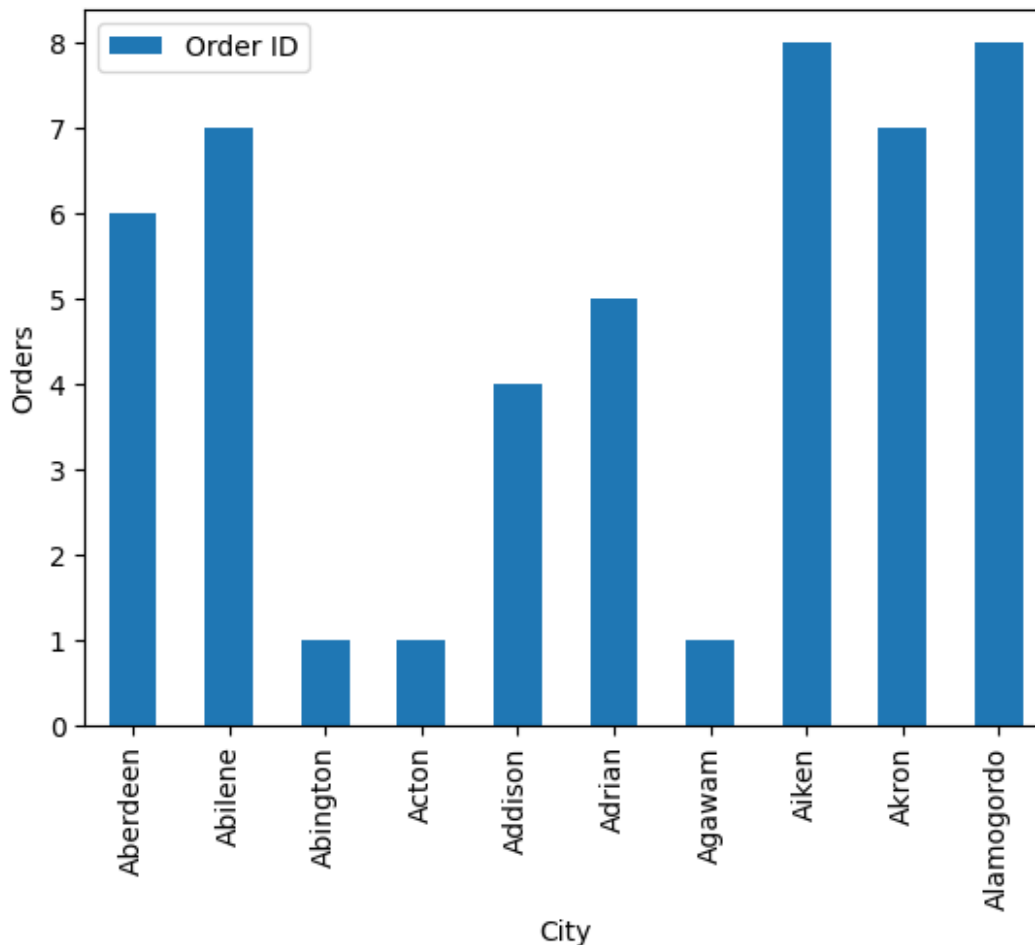
9426

#Checking Orders per city

```
orders_per_city = USA_data.groupby(['City']).agg({'Order ID':
'count'})[:10]
orders_per_city
```

	Order ID
City	
Aberdeen	6
Abilene	7
Abington	1
Acton	1
Addison	4
Adrian	5
Agawam	1
Aiken	8
Akron	7
Alamogordo	8

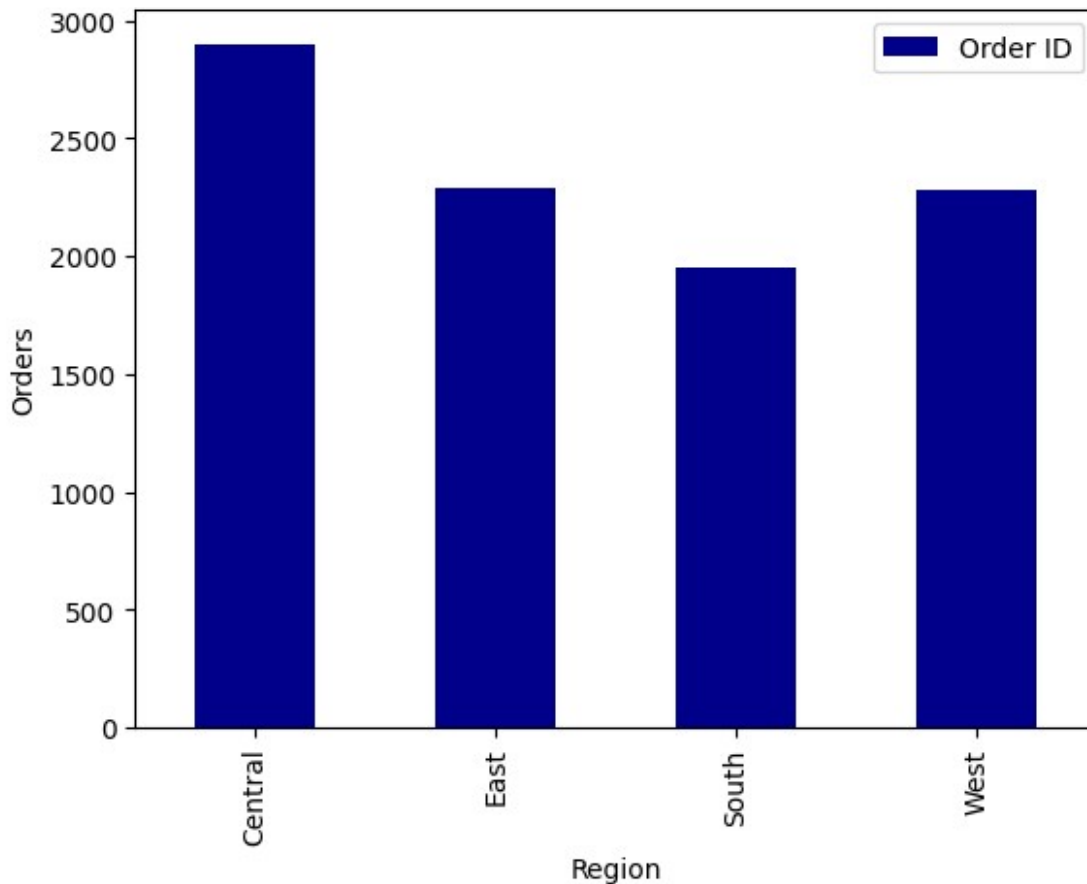
```
#Plotting Bar Chart of orders_
orders_per_city.plot(kind='bar')
plt.xlabel('City')
plt.ylabel('Orders')
plt.xticks(rotation = 90)
plt.show()
```



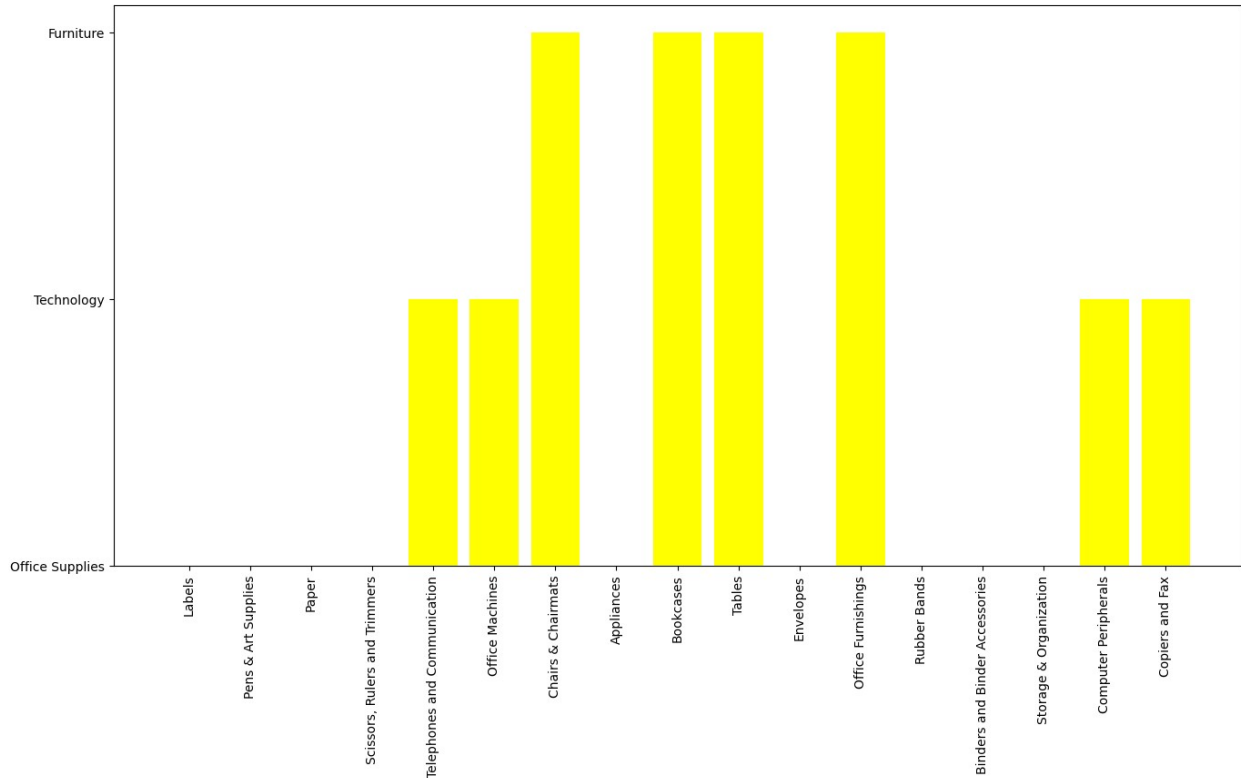
```
#Checking Orders_per_region
orders_per_region = USA_data.groupby(['Region']).agg({'Order ID':
'count'})
orders_per_region
```

	Order ID
Region	
Central	2899
East	2289
South	1954
West	2284

```
#Plotting Bar Chart of orders_
orders_per_region.plot(kind='bar',color = 'darkblue')
plt.xlabel('Region')
plt.ylabel('Orders')
plt.xticks(rotation = 90)
plt.show()
```



```
#Lets see how sub-categories are distributed wrt to category
plt.figure(figsize=(16,8))
plt.bar('Product Sub-Category','Product
Category',data=USA_data,color='yellow')
plt.xticks(rotation = 90)
plt.show()
```



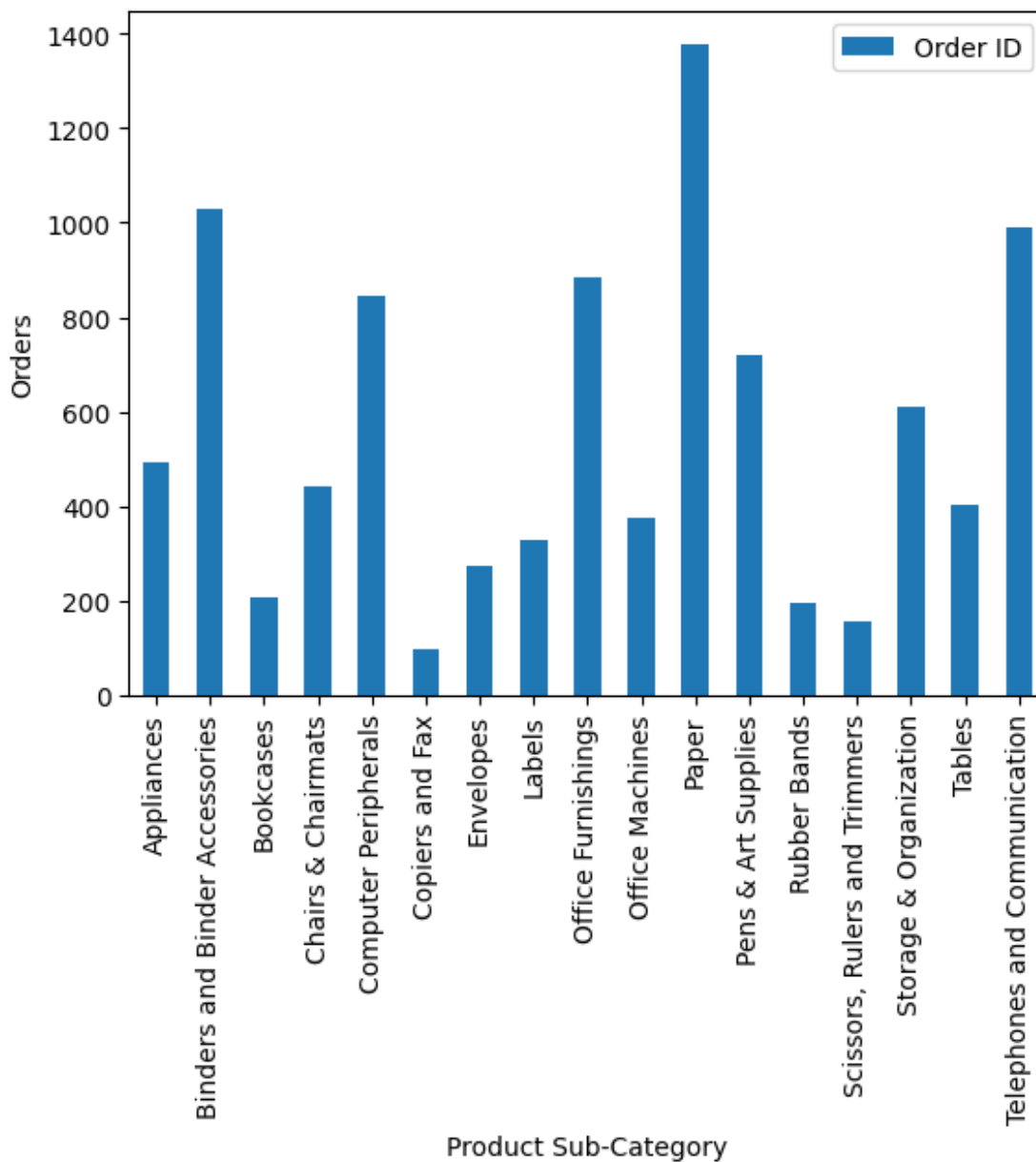
```
USA_data['Product Sub-Category'].unique()
array(['Labels', 'Pens & Art Supplies', 'Paper',
      'Scissors, Rulers and Trimmers', 'Telephones and
Communication',
      'Office Machines', 'Chairs & Chairmats', 'Appliances',
      'Bookcases',
      'Tables', 'Envelopes', 'Office Furnishings', 'Rubber Bands',
      'Binders and Binder Accessories', 'Storage & Organization',
      'Computer Peripherals', 'Copiers and Fax'], dtype=object)
```

```
orders_sub_category = USA_data.groupby(['Product Sub-
Category']).agg({"Order ID": "count"})
orders_sub_category
```

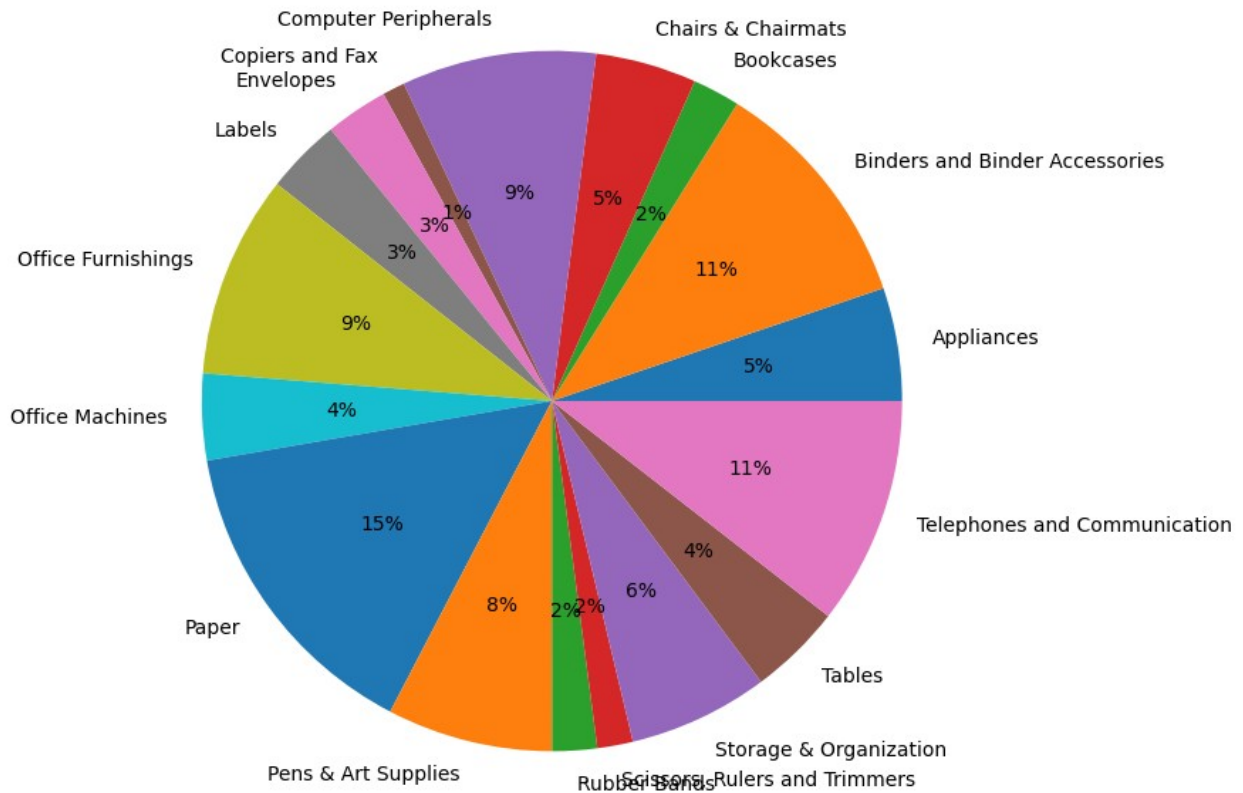
Product Sub-Category	Order ID
Appliances	492
Binders and Binder Accessories	1028
Bookcases	206
Chairs & Chairmats	440
Computer Peripherals	846
Copiers and Fax	98
Envelopes	272
Labels	329
Office Furnishings	883

Office Machines	376
Paper	1379
Pens & Art Supplies	721
Rubber Bands	195
Scissors, Rulers and Trimmers	155
Storage & Organization	610
Tables	404
Telephones and Communication	992

```
orders_sub_category.plot(kind='bar')
plt.ylabel('Orders')
plt.show()
```



```
plt.figure(figsize=(14,8))
plt.pie(orders_sub_category['Order ID'],
labels=orders_sub_category.index,autopct='%1.0f%%')
plt.show()
```



The store has wide variety of Office Supplies especially in Binders and Paper department.

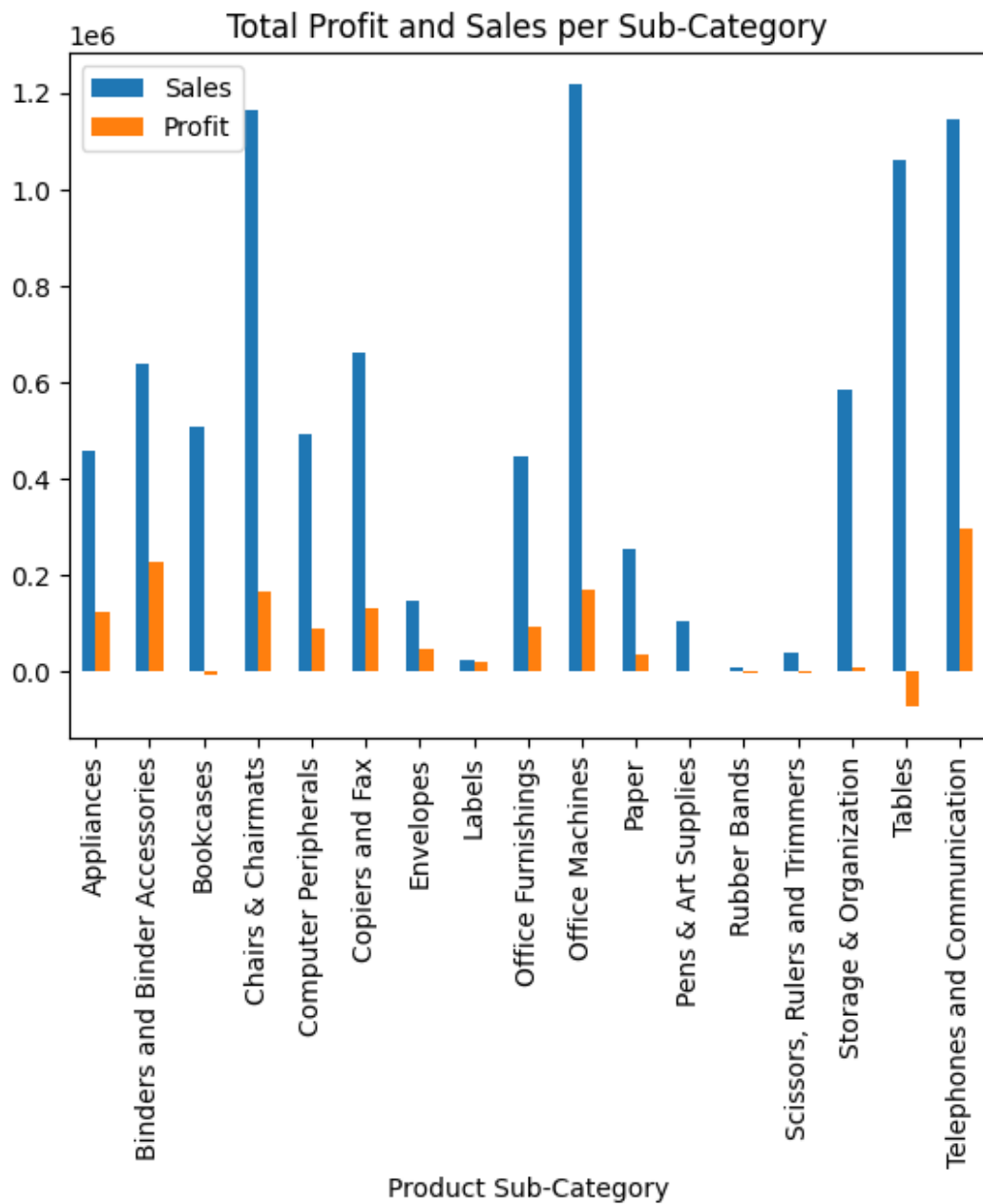
```
#Total Profit and Sales per Sub-Category
a=USA_data.groupby('Product Sub-Category')[['Sales',
'Profit']].agg('sum')
a
```

Product Sub-Category	Sales	Profit
Appliances	456723.08	121651.391380
Binders and Binder Accessories	638582.09	226572.523042
Bookcases	507494.49	-7708.748564
Chairs & Chairmats	1164584.16	165348.882760
Computer Peripherals	490840.53	87917.842513
Copiers and Fax	661211.93	129156.684030
Envelopes	147921.03	46133.223840
Labels	23449.90	17775.320505

Office Furnishings	444624.03	92209.225867
Office Machines	1218656.59	168072.833340
Paper	253600.31	35361.621738
Pens & Art Supplies	103251.59	1195.902840
Rubber Bands	8663.77	-2841.722459
Scissors, Rulers and Trimmers	40428.87	-1936.849120
Storage & Organization	585704.91	8078.804727
Tables	1061921.06	-72495.061875
Telephones and Communication	1144272.98	297950.524866

#Total Profit and Sales per Sub-Category

```
a.plot(kind='bar')
plt.title('Total Profit and Sales per Sub-Category')
plt.legend(['Sales', 'Profit'])
plt.show()
```



```
USA_data['Product Name'].nunique()
```

```
1263
```

```
USA_data['Product Name'].value_counts()
```

```
Product Name
```

```
Global High-Back Leather Tilter, Burgundy
```

```
27
```

```
Bevis 36 x 72 Conference Tables
```

```
26
```

```
BoxOffice By Design Rectangular and Half-Moon Meeting Room Tables
```

```

24
Master Giant Foot® Doorstop, Safety Yellow
24
Wilson Jones Hanging View Binder, White, 1"
23
.
.
Fellowes High-Stak® Drawer Files
1
Computer Printout Index Tabs
1
Hewlett Packard 610 Color Digital Copier / Printer
1
Global Comet™ Stacking Armless Chair
1
Alliance Rubber Bands
1
Name: count, Length: 1263, dtype: int64

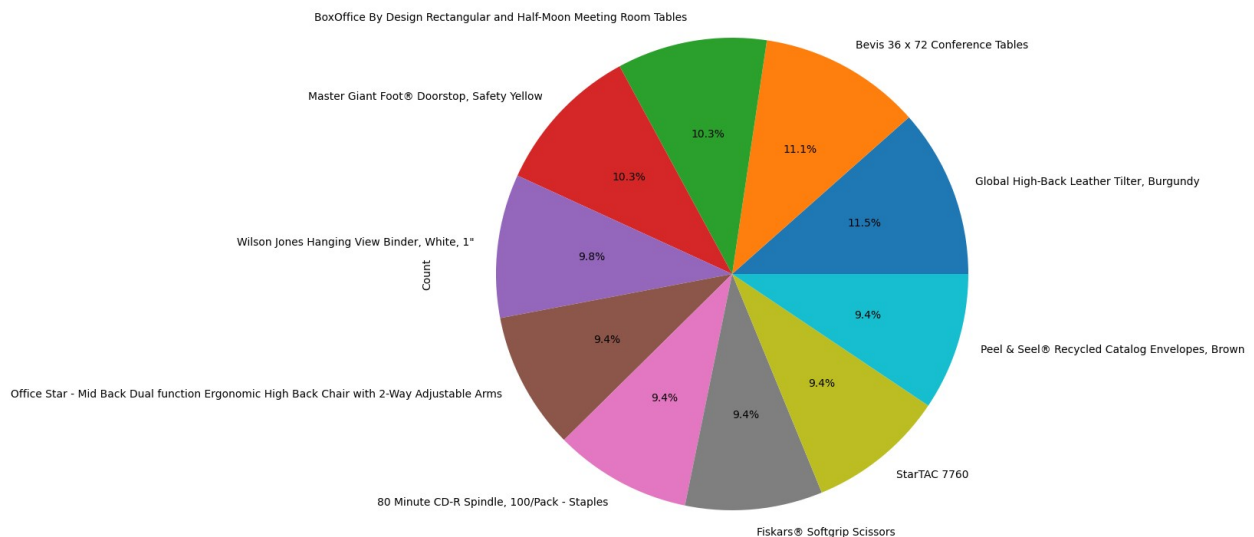
```

#Distribution of Top 10 Products

```

plt.figure(figsize=(12,10))
USA_data['Product
Name'].value_counts().head(10).plot.pie(autopct="%1.1f%%")
plt.ylabel('Count')
plt.show()

```



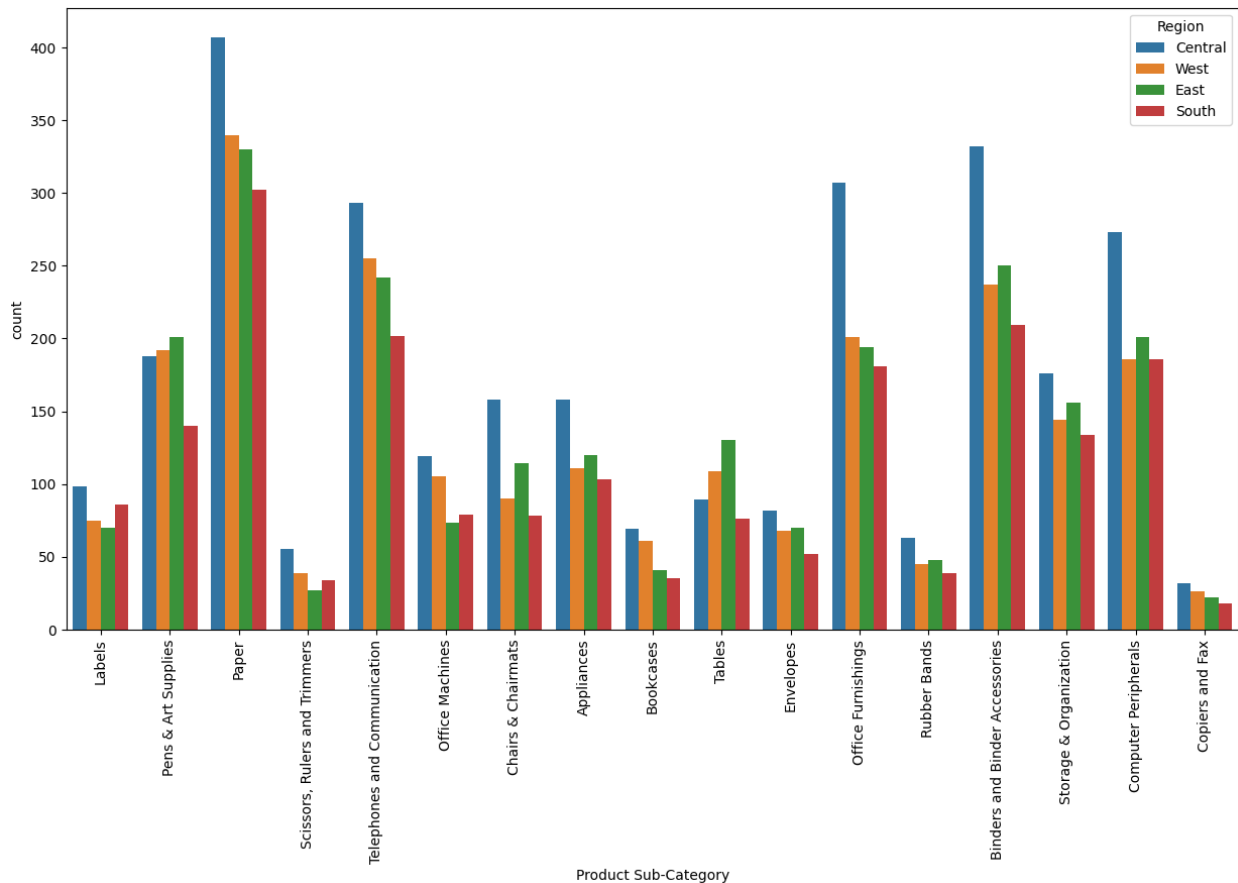
#Count of Sub-Category region wise

```

plt.figure(figsize=(15,8))
sns.countplot(x="Product Sub-Category", hue="Region", data=USA_data)

```

```
plt.xticks(rotation=90)
plt.show()
```



People residing in Central part of US tend to order more from superstore.

```
#extracting the year of order
```

```
USA_data['order year']=USA_data['Order Date'].dt.year
```

```
USA_data['order year'].head()
```

```
0    2012
```

```
1    2010
```

```
2    2011
```

```
3    2011
```

```
4    2011
```

```
Name: order year, dtype: int32
```

```
#Sales per year
```

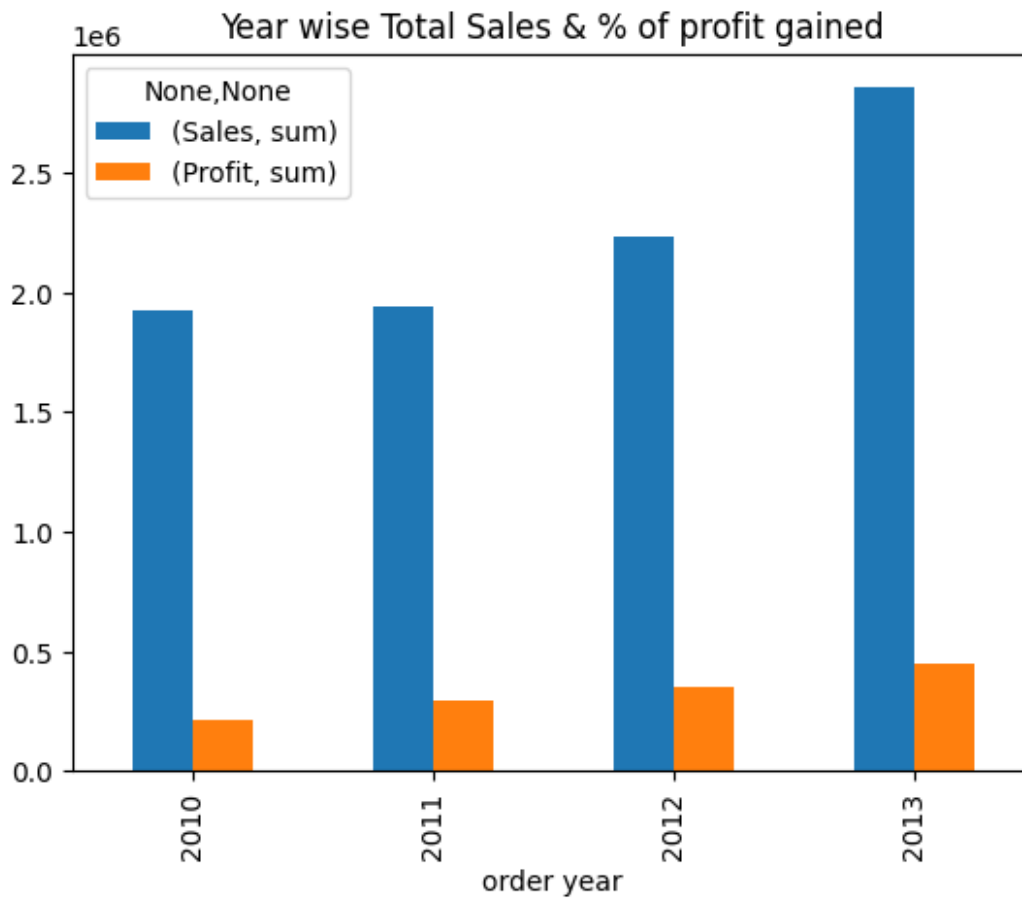
```
b= USA_data.groupby('order year')[['Sales','Profit']].agg(['sum'])
```

```
b
```

	Sales sum	Profit sum
order year		

2010	1924332.88	213324.139337
2011	1944507.43	297847.743198
2012	2230731.18	354073.573280
2013	2852359.83	447196.943612

```
b.plot(kind='bar')
plt.title('Year wise Total Sales & % of profit gained')
plt.show()
```

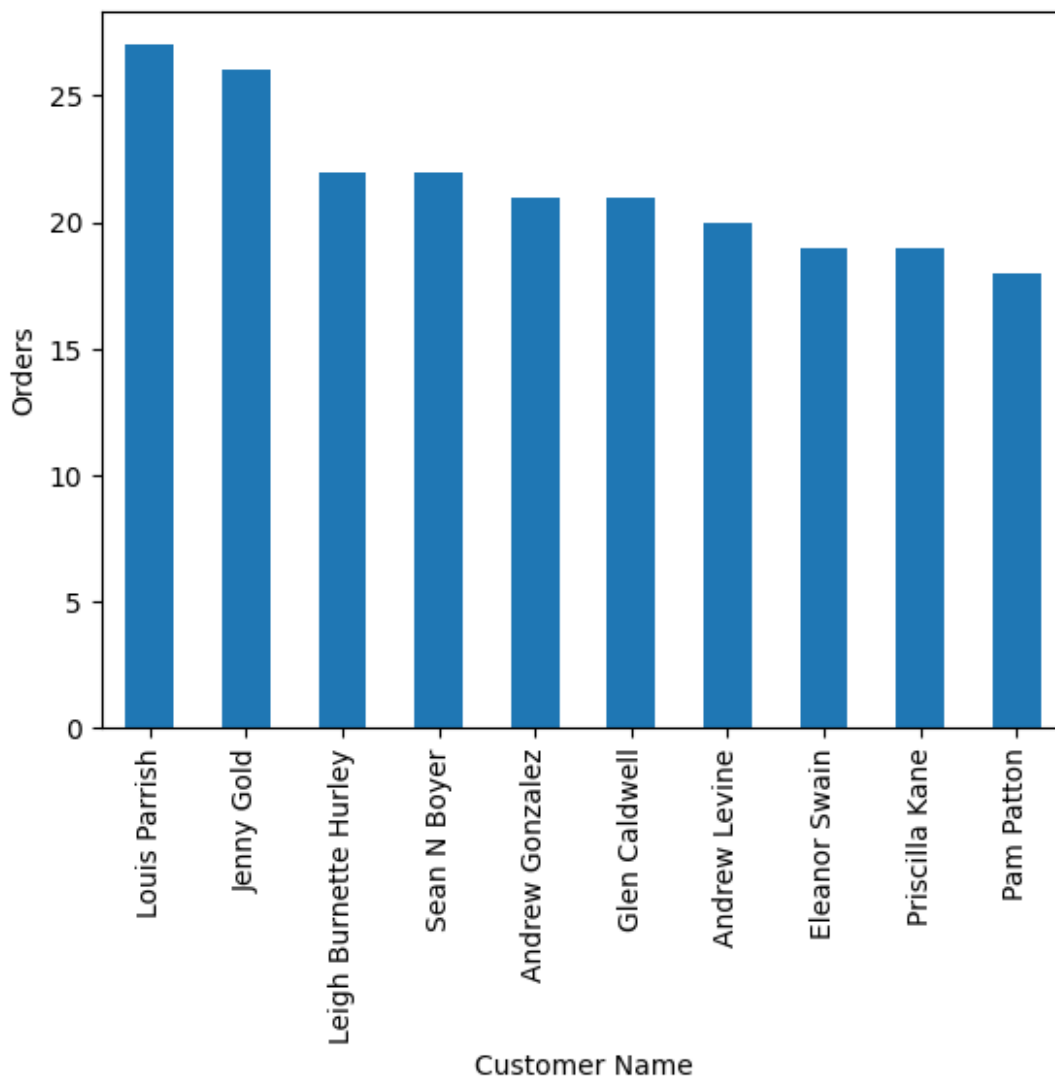


```
#Top 10 customers who order frequently
top10_cust = USA_data['Customer Name'].value_counts().head(10)
top10_cust
```

Customer Name	
Louis Parrish	27
Jenny Gold	26
Leigh Burnette Hurley	22
Sean N Boyer	22
Andrew Gonzalez	21
Glen Caldwell	21
Andrew Levine	20

```
Eleanor Swain          19
Priscilla Kane         19
Pam Patton             18
Name: count, dtype: int64
```

```
top10_cust.plot(kind='bar')
plt.ylabel('Orders')
plt.show()
```

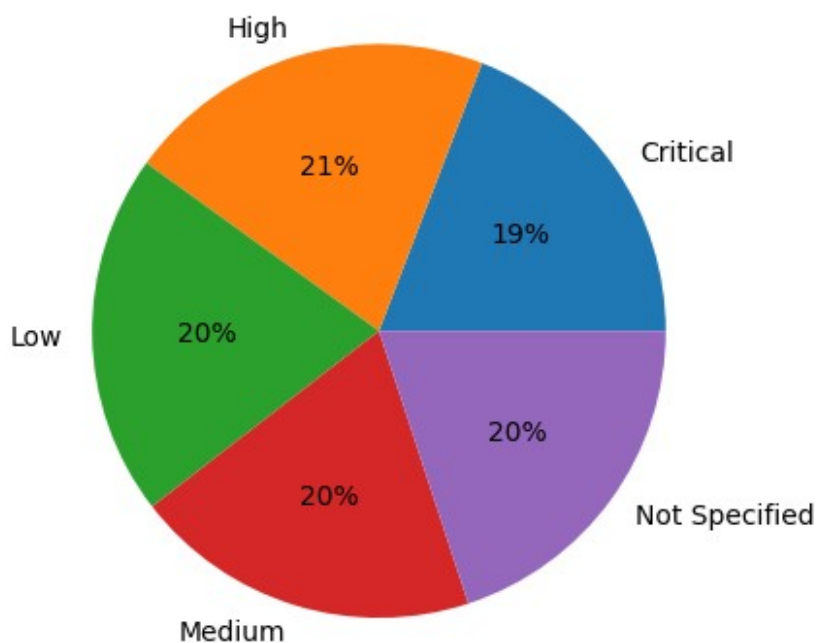


```
USA_data['Order Priority'] = USA_data['Order
Priority'].replace("Critical ", "Critical")

#Orders with respect to their priority
order_priority = USA_data.groupby('Order Priority').agg({"Order ID" :
'count'})
order_priority
```


	Order ID
Order Priority	
Critical	1805
High	1970
Low	1926
Medium	1844
Not Specified	1881

```
plt.pie(x=order_priority['Order ID'], labels
=order_priority.index, autopct='%1.0f%%' )
plt.show()
```

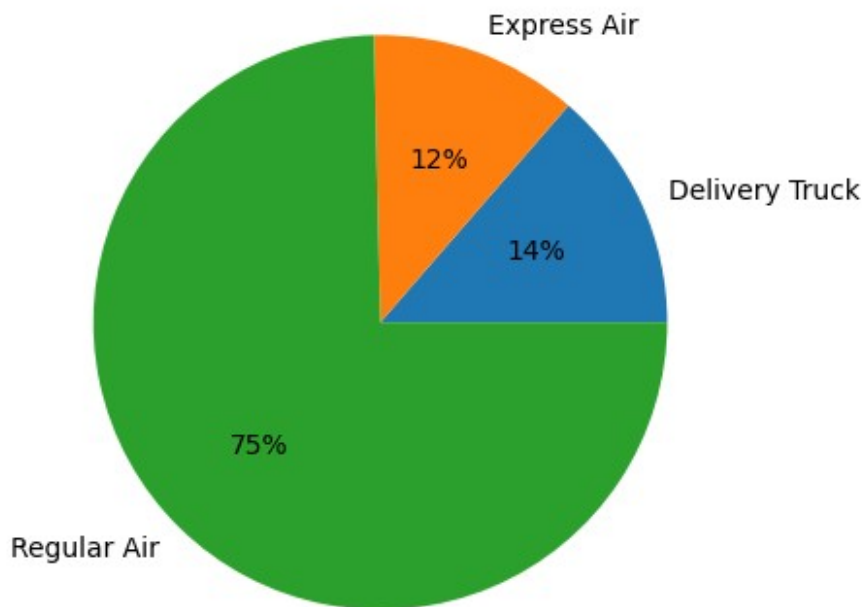


#orders w.r.t ship mode

```
orders_mode = USA_data.groupby("Ship Mode").agg({'Order ID': 'count'})
orders_mode
```

	Order ID
Ship Mode	
Delivery Truck	1283
Express Air	1107
Regular Air	7036

```
plt.pie(x = orders_mode['Order ID'], labels = orders_mode.index,
autopct = '%1.0f%%')
plt.show()
```



```
#Orders w.r.t customer segment
orders_segment = USA_data.groupby('Customer Segment').agg({'Order ID':
'count'})
orders_segment
```

Customer Segment	Order ID
Consumer	1894
Corporate	3375
Home Office	2316
Small Business	1841

```
plt.pie(x = orders_segment['Order ID'], labels = orders_segment.index,
autopct = '%1.0f%%')
plt.show()
```

