TSF TASK 3 - GRIP MARCH'21

ADITYA AMBWANI

Exploratory Data Analysis - Retail

Objective:-

- Perform 'Exploratory Data Analysis' on dataset 'SampleSuperstore'
- As a business manager, try to find out the weak areas where you can work to make more profit.
- What all business problems you can derive by exploring the data?

Ques. What is EDA ? Ans. In statistics, exploratory data analysis is an approach to analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task.

In [2]:
```
# importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
# Reading dataset
data=pd.read_csv("SampleSuperstore.csv")
data
```

Out[3]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22
...	...	...	...	...	...	...	...	...	...	
9989	Second Class	Consumer	United States	Miami	Florida	33180	South	Furniture	Furnishings	25

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	
9990	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Furniture	Furnishings	91
9991	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Technology	Phones	258
9992	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Office Supplies	Paper	29
9993	Second Class	Consumer	United States	Westminster	California	92683	West	Office Supplies	Appliances	243

9994 rows × 13 columns

In [4]:
```
# Checking for rows and coloumns
data.shape
```

Out[4]:
```
(9994, 13)
```

In [5]:
```
# describing the dataset in summarized way
data.describe()
```

Out[5]:

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

In [6]:
```
# More information about dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Ship Mode     9994 non-null   object
 1   Segment       9994 non-null   object
 2   Country       9994 non-null   object
 3   City          9994 non-null   object
 4   State         9994 non-null   object
 5   Postal Code   9994 non-null   int64 
 6   Region        9994 non-null   object
 7   Category      9994 non-null   object
 8   Sub-Category  9994 non-null   object
```

```
 9   Sales         9994 non-null   float64
 10  Quantity      9994 non-null   int64
 11  Discount      9994 non-null   float64
 12  Profit        9994 non-null   float64
dtypes: float64(3), int64(2), object(8)
memory usage: 1015.1+ KB
```

In [7]:
```
# Checking for null values in our dataset
data.isnull().sum()
```

Out[7]:
```
Ship Mode       0
Segment         0
Country         0
City            0
State           0
Postal Code     0
Region          0
Category        0
Sub-Category    0
Sales           0
Quantity        0
Discount        0
Profit          0
dtype: int64
```

In [8]:
```
# Listing all coloumns in our dataset
data.columns
```

Out[8]:
```
Index(['Ship Mode', 'Segment', 'Country', 'City', 'State', 'Postal Code',
       'Region', 'Category', 'Sub-Category', 'Sales', 'Quantity', 'Discount',
       'Profit'],
      dtype='object')
```

In [9]:
```
# Checking for duplicate values in our dataset
data.duplicated().sum()
```

Out[9]:
```
17
```

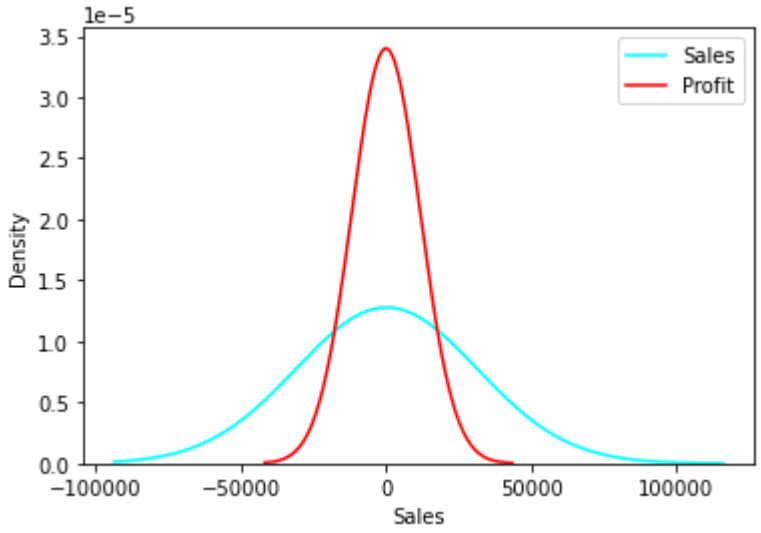
In [10]:
```
# Removing all the duplicate values
data.drop_duplicates(keep=False,inplace=True)
```

In [11]:
```
data.shape
```

Out[11]:
```
(9960, 13)
```

In [12]:
```
#EDA
sns.kdeplot(data['Sales'],color='cyan',label='Sales',bw=50)
sns.kdeplot(data['Profit'],color='red',label='Profit',bw=50)
plt.legend()
```

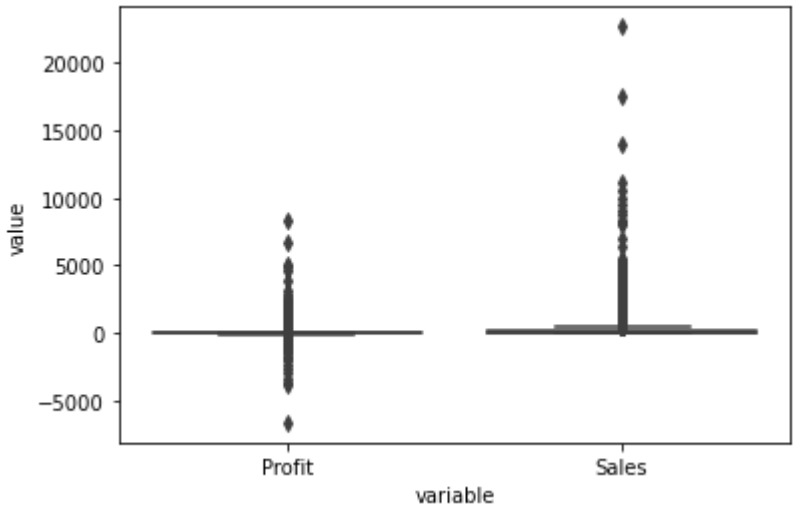
Out[12]:
```
<matplotlib.legend.Legend at 0x4fdcb66dc0>
```

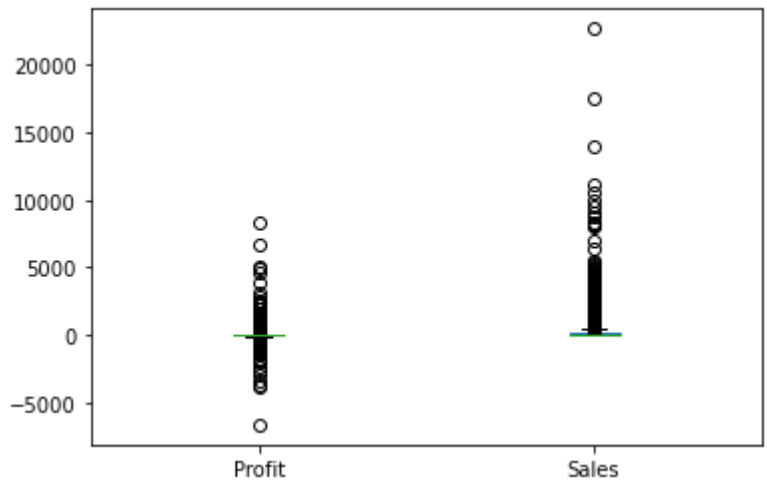


Conclusion:- We can infer from above graph that profit is more than sales in general

In [13]:
```
# Finding Outliers
df = pd.DataFrame(data = data, columns = ['Profit','Sales'])
sns.boxplot(x="variable", y="value", data=pd.melt(df))
df.plot(kind='box')
```

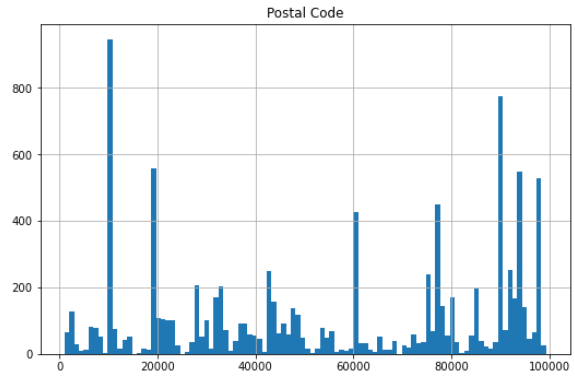
Out[13]: <AxesSubplot:>

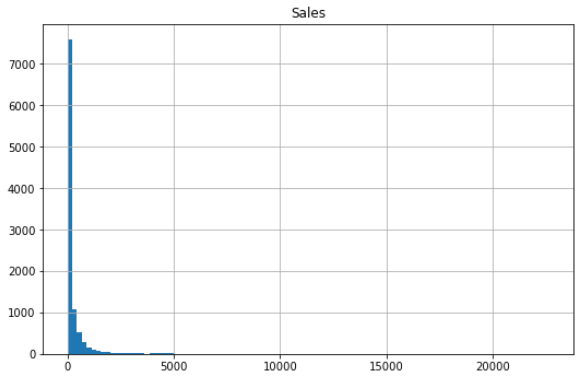


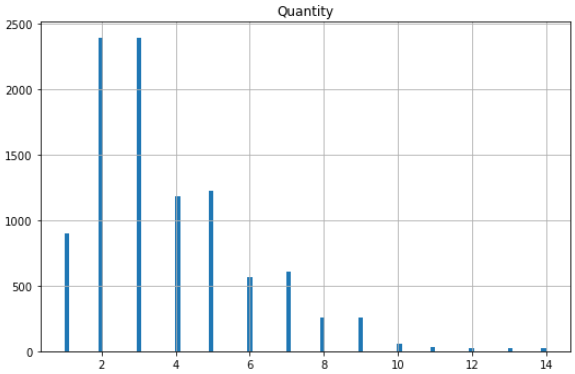


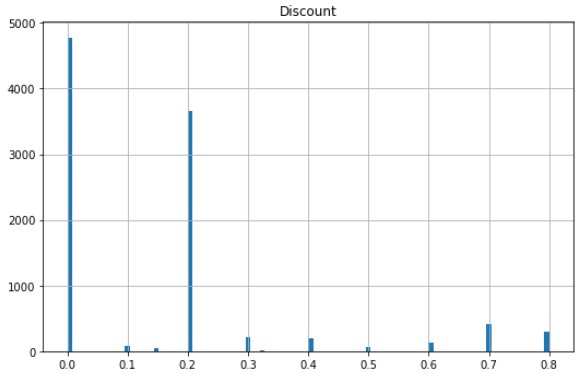
In [14]:
```
# Visualising data by plotting histograms
```

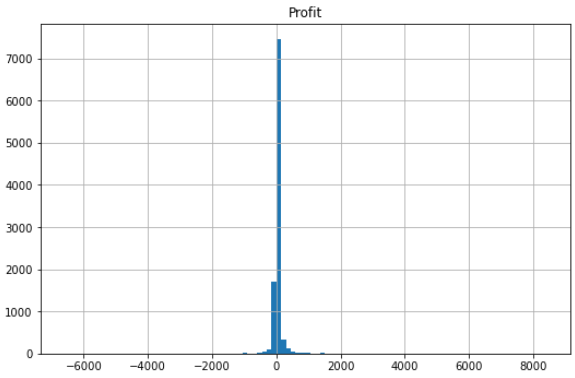
```
data.hist(bins=100,figsize=(20,20))
plt.show()
```





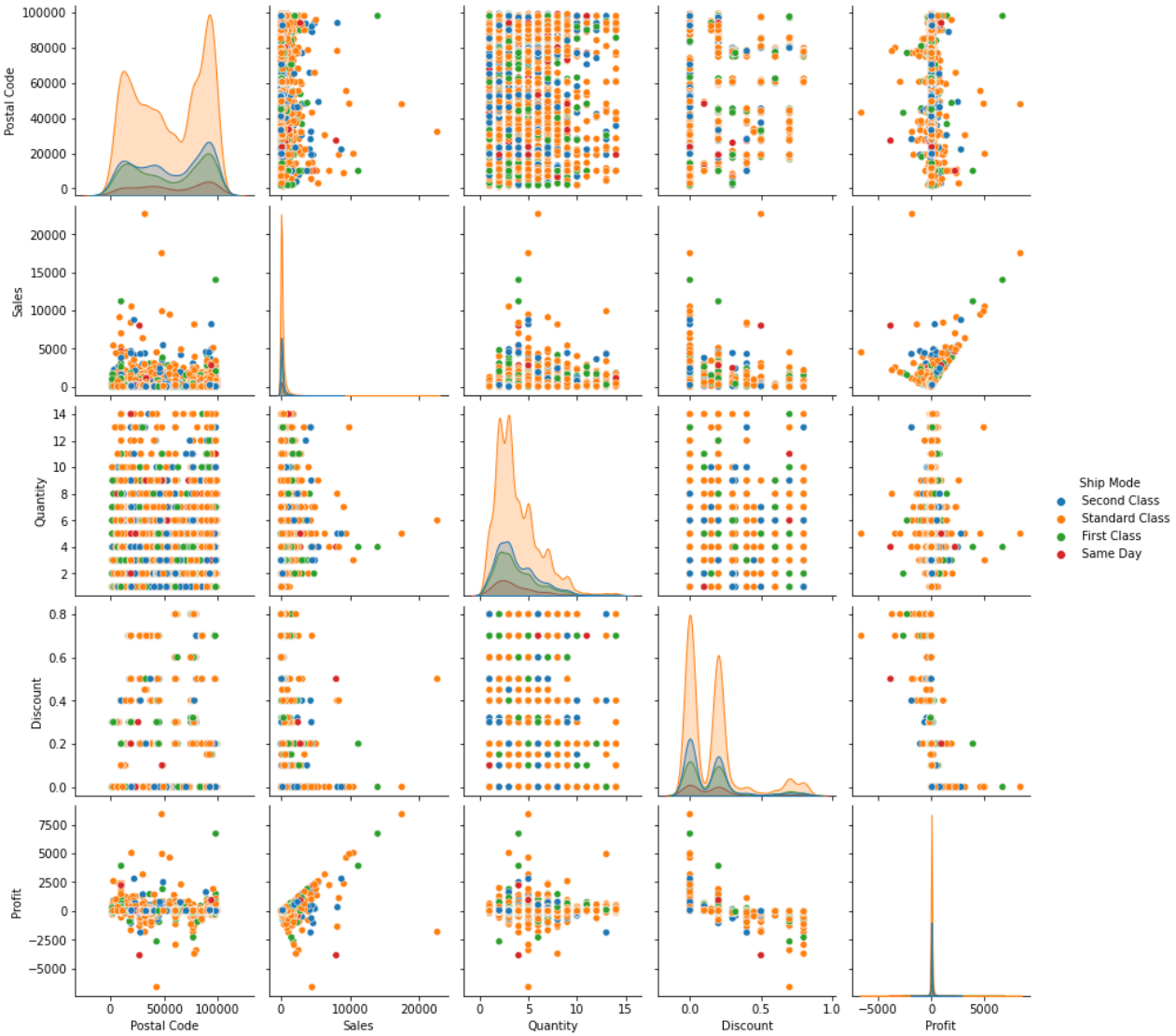






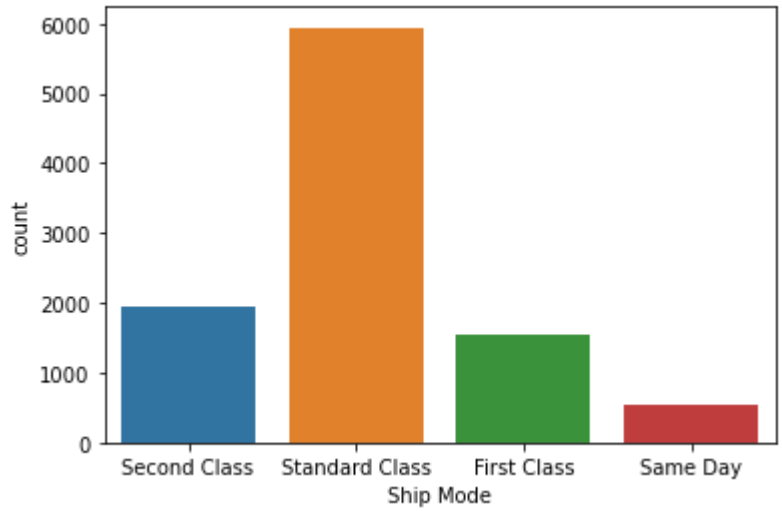
In [15]:
```
# Analysing the data based on Ship Mode
sns.pairplot(data,hue="Ship Mode")
```

Out[15]: <seaborn.axisgrid.PairGrid at 0x4fdcef1ca0>



In [16]:
```
# Analysing the data based on Ship Mode
sns.countplot(x='Ship Mode',data=data)
```

Out[16]: <AxesSubplot:xlabel='Ship Mode', ylabel='count'>

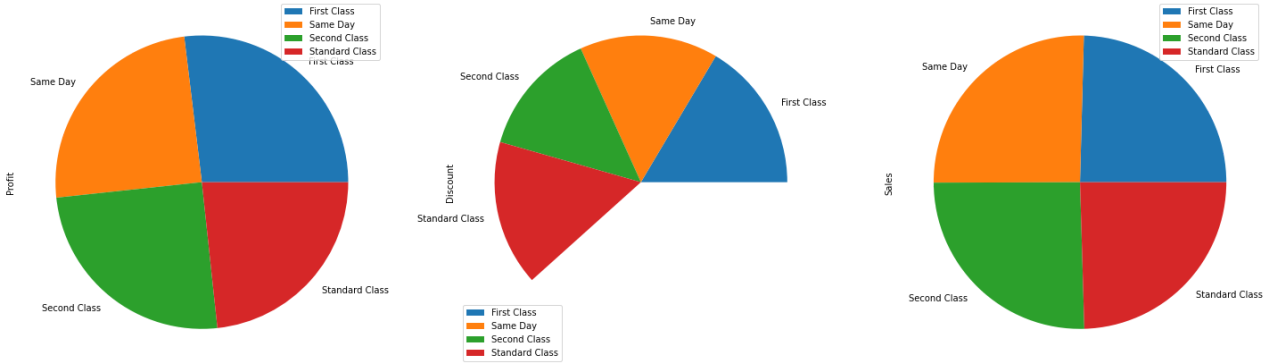


Conclusion:- The preffered ship mode is Standard Class.

In [17]:
```
# Analysing the data based on Ship Mode
```

```
dataf=data.groupby(['Ship Mode'])[['Profit','Discount','Sales']].mean()
dataf.plot.pie(subplots=True,figsize=(25,25),labels=dataf.index)
```

Out[17]:
```
array([<AxesSubplot:ylabel='Profit'>, <AxesSubplot:ylabel='Discount'>,
       <AxesSubplot:ylabel='Sales'>], dtype=object)
```



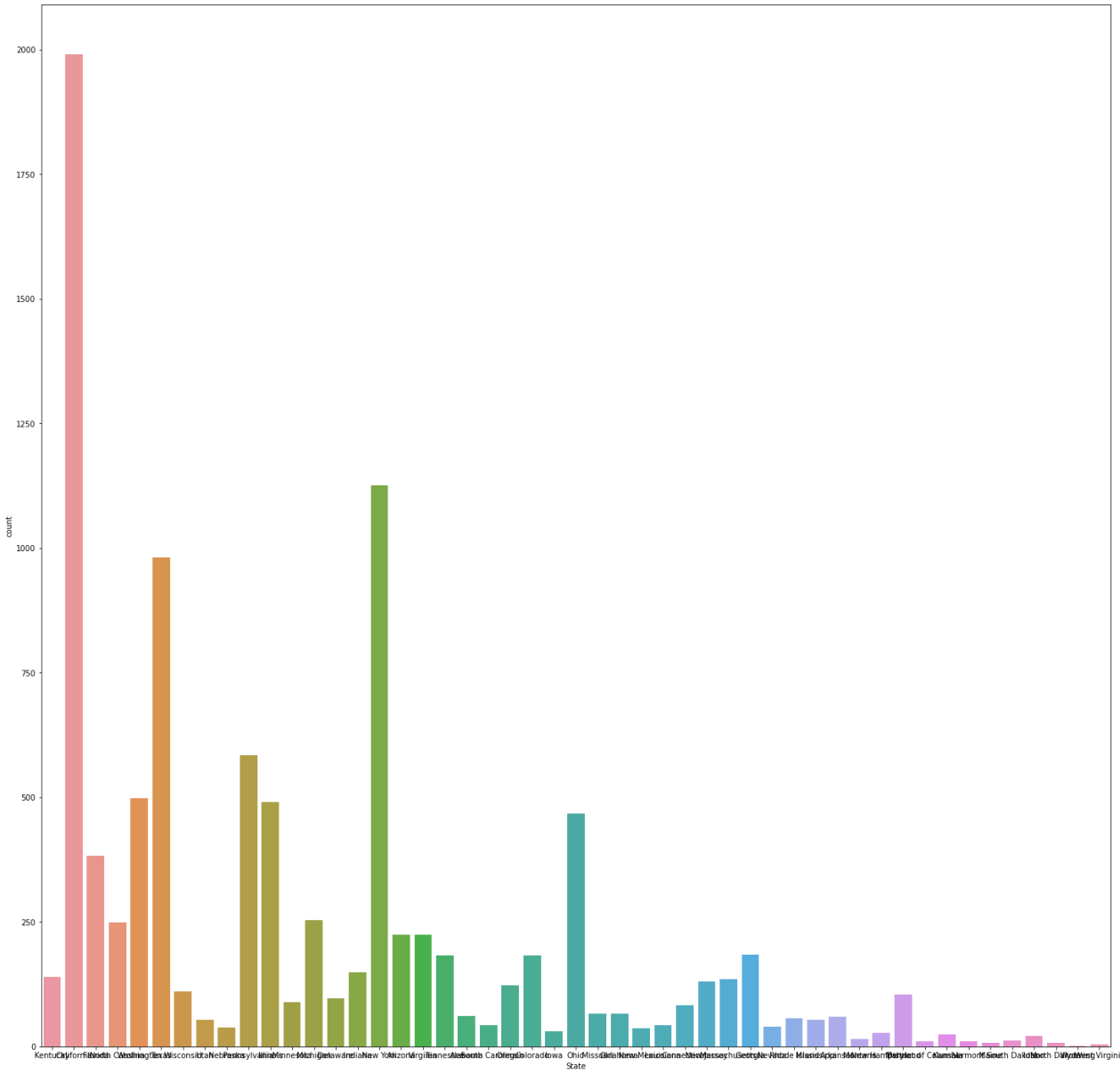
Conclusion:- The profit and sales are high in first class while discount is high in Standard Class.

In [18]:
```
# Analysing the data based on State data
sns.pairplot(data,hue='State')
```

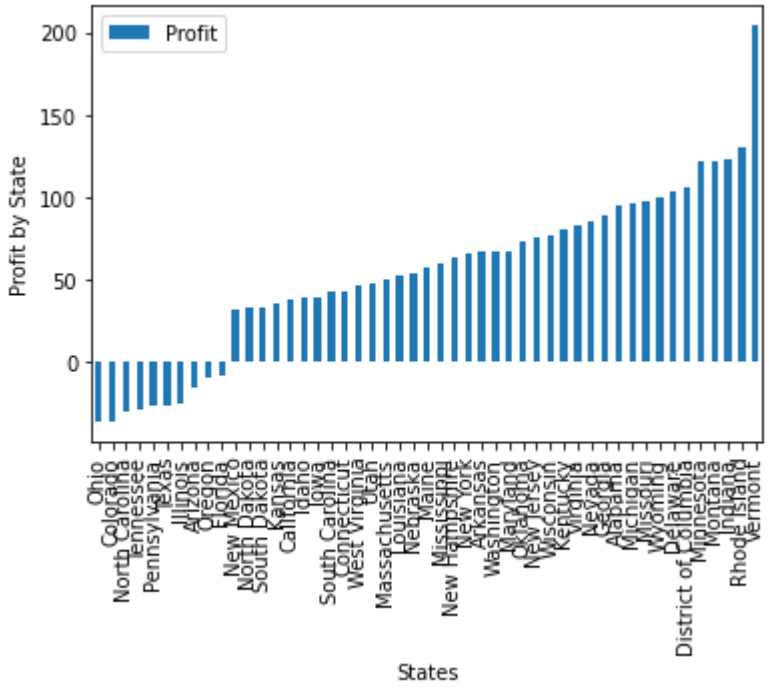
Out[18]:
```
<seaborn.axisgrid.PairGrid at 0x4fdcd23610>
```

In [19]:
```
# Analysing the data based on State data
```

```
plt.figure(figsize=(25,25))
sns.countplot(x='State',data=data)
plt.show()
```

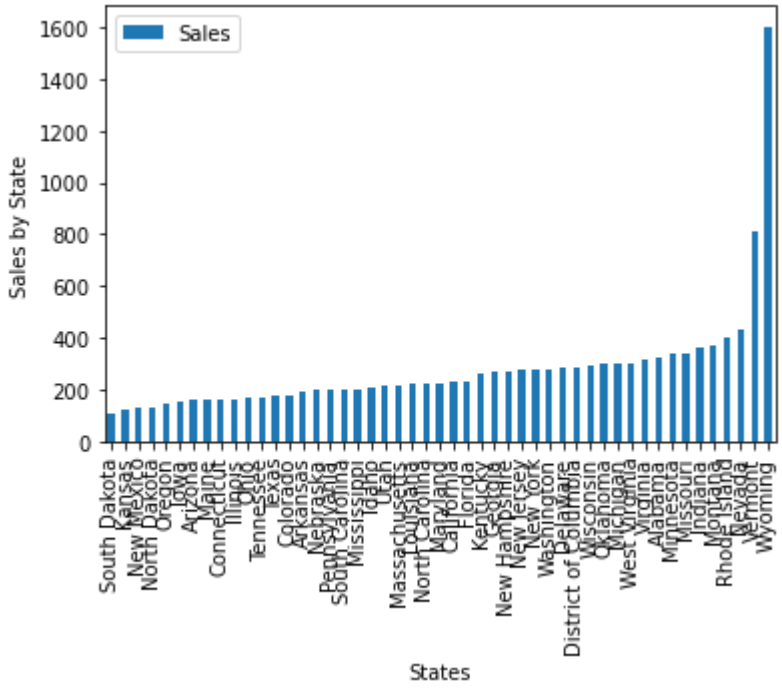


In [20]:
```
# Analysing the data based on Profit in each State
dataf1=data.groupby(['State'])[['Profit','Discount','Sales']].mean()
dataf11=dataf1.sort_values('Profit')
dataf11[['Profit']].plot(kind='bar')
plt.xlabel('States')
plt.ylabel('Profit by State')
plt.show()
```



Conclusion:- The highest profit is in Vermont State.

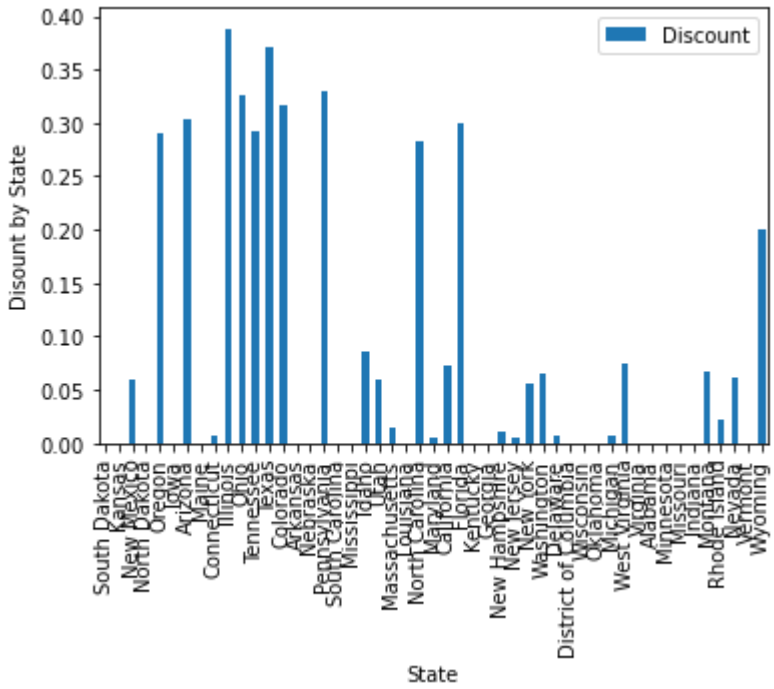
In [21]:
```
# Analysing the data based on Sale in each State
dataf12=dataf1.sort_values('Sales')
dataf12[['Sales']].plot(kind='bar')
plt.xlabel('States')
plt.ylabel('Sales by State')
plt.show()
```



Conclusion:- The highest sales is in Wyoming State.

In [22]:
```
# Analysing the data based on Discount in each State
dataf13=dataf1.sort_values('Discount')
dataf12[['Discount']].plot(kind='bar')
plt.xlabel('State')
```

```
plt.ylabel('Disount by State')
plt.show()
```

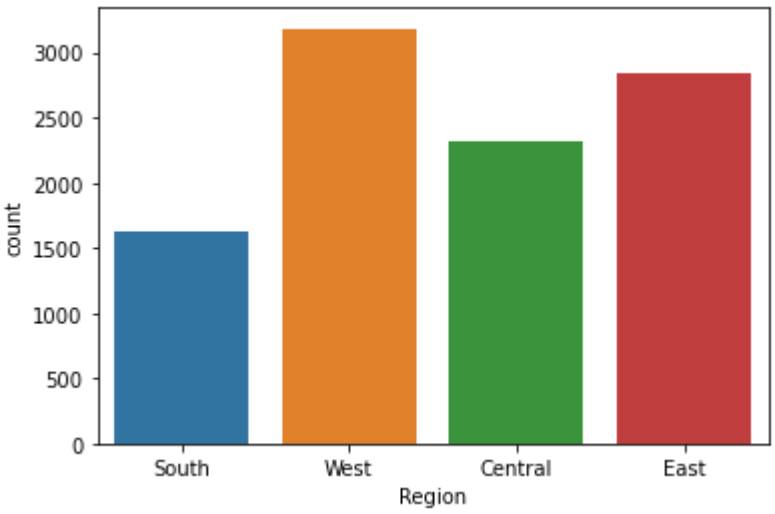


Conclusion:- The highest discount is in Illinois State.

In [23]:
```
# Analysing the data based on Region data
sns.pairplot(data,hue='Region')
```

Out[23]:
```
<seaborn.axisgrid.PairGrid at 0x4fdd05fa90>
```

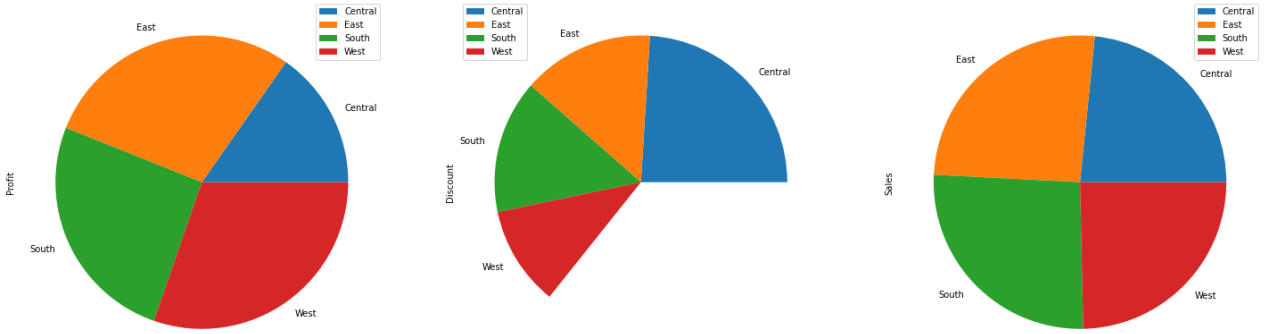
In [24]:
```
# Analysing the data based on Region data
sns.countplot(x='Region',data=data)
plt.show()
```



Conclusion:- The West has highest number of dealings.

In [25]:
```
# Analysing the data based on Region
dataf2=data.groupby(['Region'])[['Profit','Discount','Sales']].mean()
dataf2.plot.pie(subplots=True,figsize=(25,25),labels=dataf2.index)
```

Out[25]:
```
array([<AxesSubplot:ylabel='Profit'>, <AxesSubplot:ylabel='Discount'>,
       <AxesSubplot:ylabel='Sales'>], dtype=object)
```

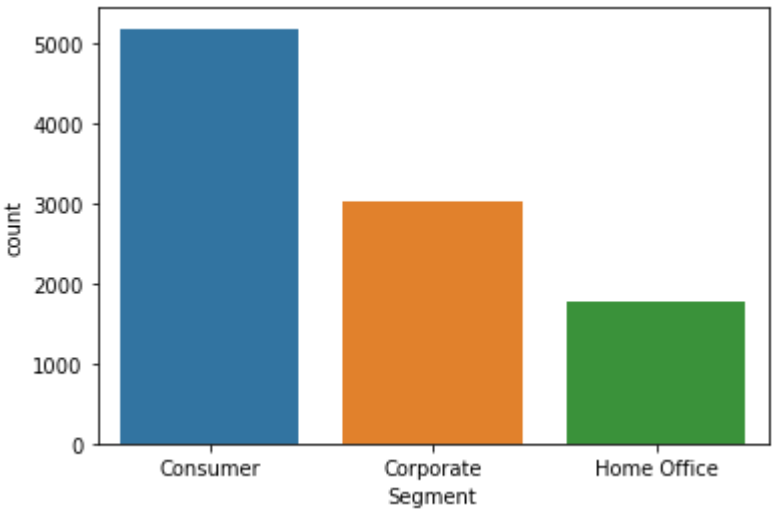


Conclusion:- Profit is highest in East. Discount is highest in Central region. East also has highest number of Sales.

In [26]:
```
# Analysing the data based on Segment data
sns.pairplot(data,hue='Segment')
```

Out[26]:
```
<seaborn.axisgrid.PairGrid at 0x4fd9c57640>
```

In [27]:
```
# Analysing the data based on Segment data
sns.countplot(x='Segment',data=data)
plt.show()
```

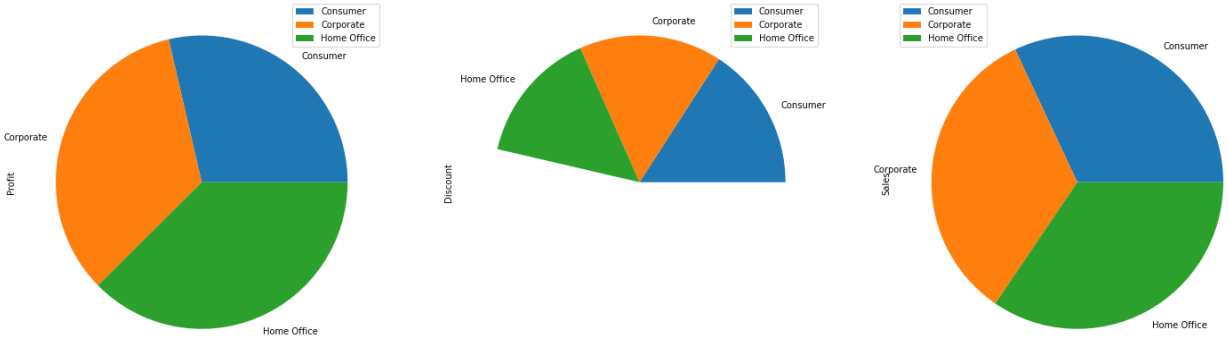


Conclusion:- THe segment which buys more product lie in Consumer section.

In [28]:
```
# Analysing the data based on Segment
```

```
dataf3=data.groupby(['Segment'])[['Profit','Discount','Sales']].mean()
dataf3.plot.pie(subplots=True,figsize=(25,25),labels=dataf3.index)
```

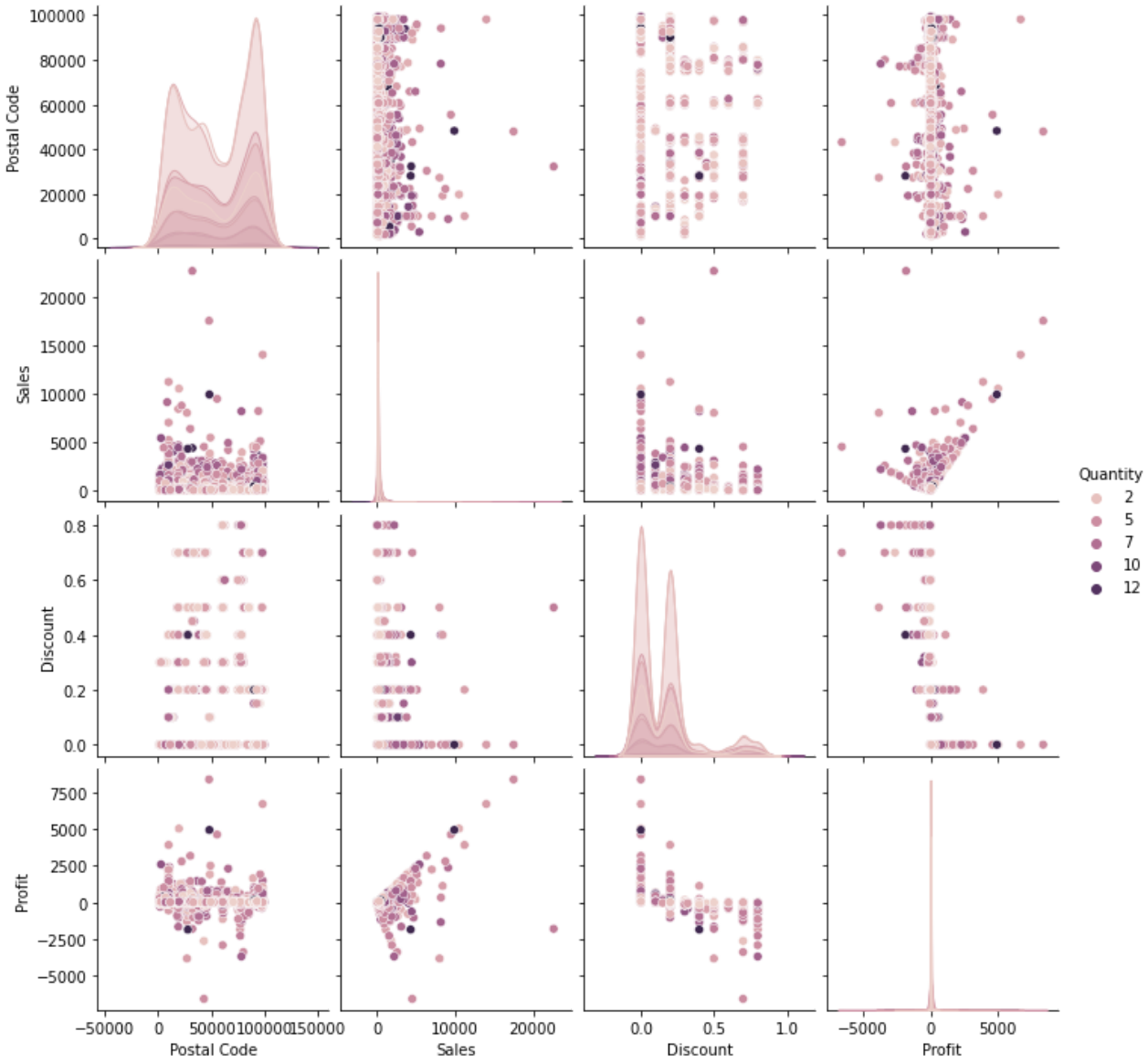
Out[28]:
```
array([<AxesSubplot:ylabel='Profit'>, <AxesSubplot:ylabel='Discount'>,
       <AxesSubplot:ylabel='Sales'>], dtype=object)
```



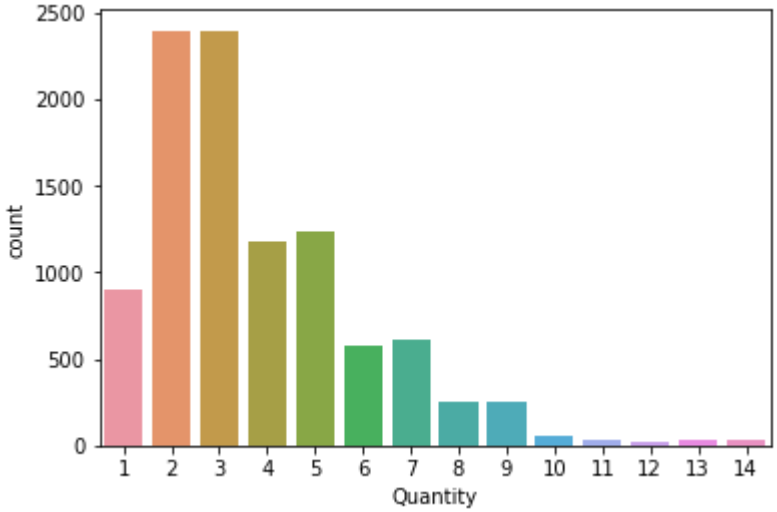
Conclusion:- Profit is highest in home-office as well as sales are also highest here.

In [29]:
```
# Analysing the data based on Quantity data
sns.pairplot(data,hue='Quantity')
```

Out[29]:
```
<seaborn.axisgrid.PairGrid at 0x4fe8453dc0>
```

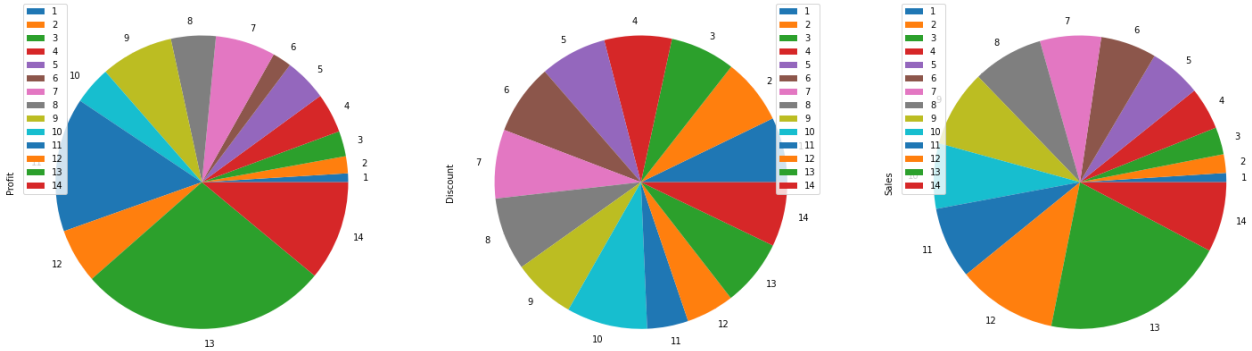


In [30]:
```
# Analysing the data based on Quantity data
sns.countplot(x='Quantity',data=data)
plt.show()
```



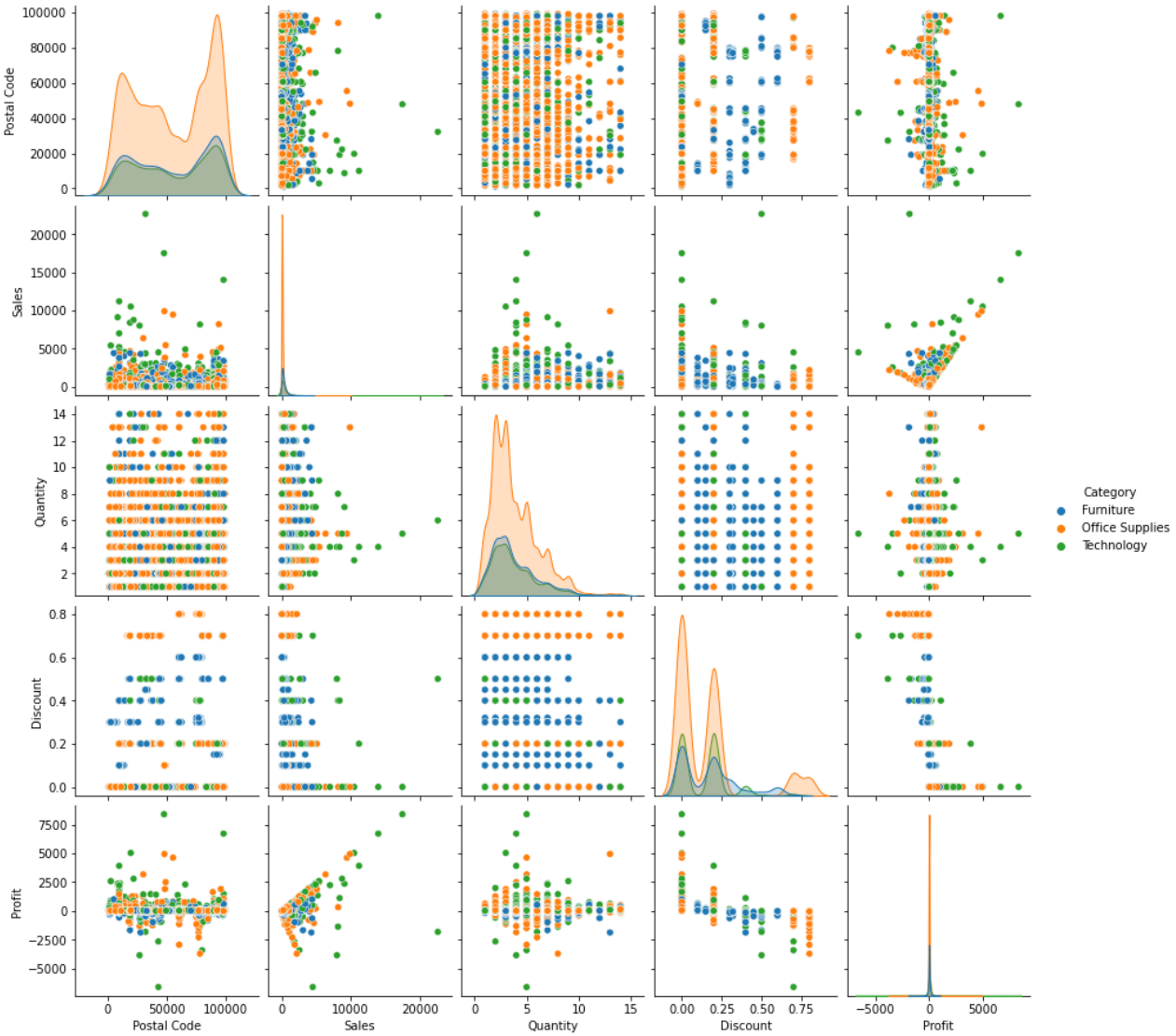
In [31]:
```
# Analysing the data based on Quantity
dataf4=data.groupby(['Quantity'])[['Profit','Discount','Sales']].mean()
dataf4.plot.pie(subplots=True,figsize=(25,25))
```

Out[31]:
```
array([<AxesSubplot:ylabel='Profit'>, <AxesSubplot:ylabel='Discount'>,
       <AxesSubplot:ylabel='Sales'>], dtype=object)
```

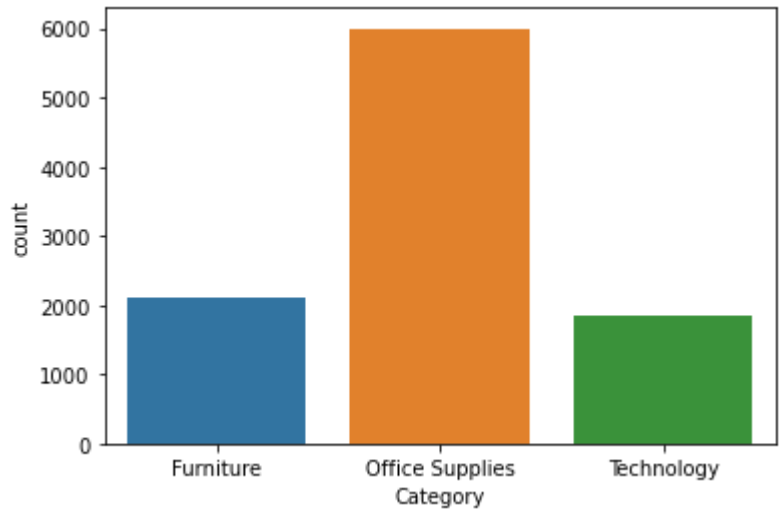


In [32]:
```
# Analysing the data based on Category data
sns.pairplot(data,hue='Category')
```

Out[32]:
```
<seaborn.axisgrid.PairGrid at 0x4fe90edcd0>
```



In [33]:
```
# Analysing the data based on Category data
sns.countplot(x='Category',data=data)
plt.show()
```

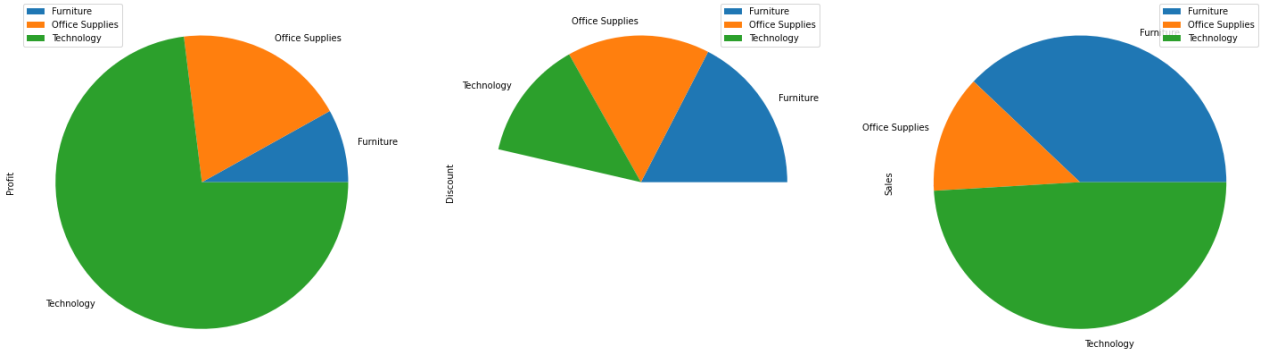


Conclusion:- Office Supplies are more compared to furniture and technology.

In [34]:
```
# Analysing the data based on Category
```

```
dataf5=data.groupby(['Category'])[['Profit','Discount','Sales']].mean()
dataf5.plot.pie(subplots=True,figsize=(25,25),labels=dataf5.index)
```

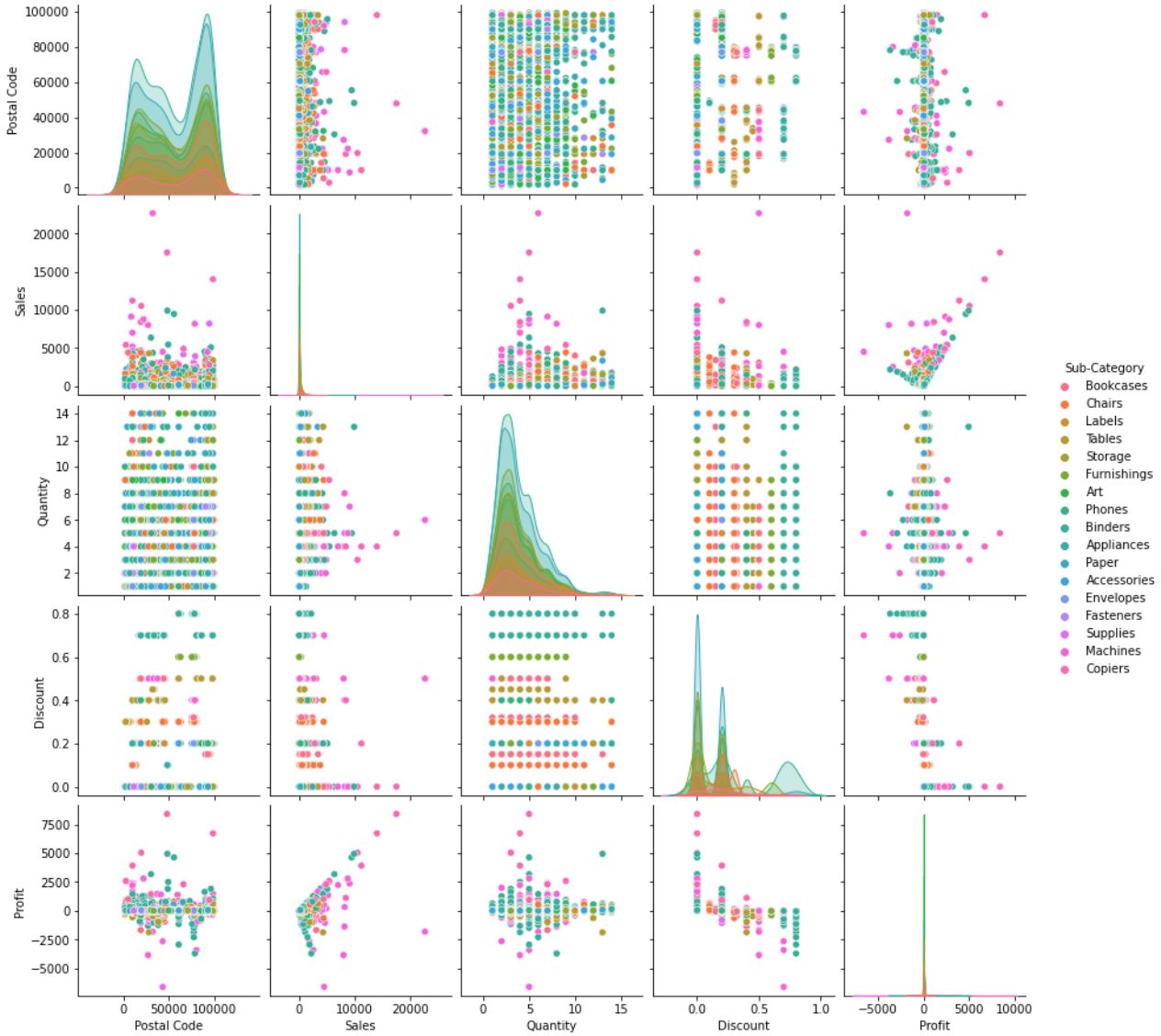
Out[34]:
```
array([<AxesSubplot:ylabel='Profit'>, <AxesSubplot:ylabel='Discount'>,
       <AxesSubplot:ylabel='Sales'>], dtype=object)
```



Conclusion:- Profit is highest in Technology and also the most number of sales.

In [35]:
```
# Analysing the data based on Sub-Category data
sns.pairplot(data,hue='Sub-Category')
```

Out[35]:
```
<seaborn.axisgrid.PairGrid at 0x4feb47bdc0>
```



In [36]:
```
# Analysing the data based on Sub-Category data
sns.countplot(x='Sub-Category',data=data)
plt.show()
```