Description of columns from dataset documentation

Attributes:-

```
Row ID           int64
Order ID          object
Order Date        object
Ship Date         object
Ship Mode         object
Customer ID       object
Customer Name     object
Segment           object
Country           object
City              object
State             object
Postal Code       float64
Region            object
Product ID        object
Category          object
Sub-Category      object
Product Name      object
Sales             float64
```

reference of dataset:-

https://www.kaggle.com/datasets/rohitsahoo/sales-forecasting
(https://www.kaggle.com/datasets/rohitsahoo/sales-forecasting)

IMPORTING LIBRARY

In [1]:
```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

Reading dataset

In [2]:
```
df = pd.read_csv('train.csv')
```

```
In [3]: df
```

Out[3]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City
0	1	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson
1	2	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson
2	3	CA-2017-138688	12/06/2017	16/06/2017	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles
3	4	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale
4	5	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale
...	...	...	...	...	...	...	...	...	...	...
9795	9796	CA-2017-125920	21/05/2017	28/05/2017	Standard Class	SH-19975	Sally Hughsby	Corporate	United States	Chicago
9796	9797	CA-2016-128608	12/01/2016	17/01/2016	Standard Class	CS-12490	Cindy Schnelling	Corporate	United States	Toledo
9797	9798	CA-2016-128608	12/01/2016	17/01/2016	Standard Class	CS-12490	Cindy Schnelling	Corporate	United States	Toledo
9798	9799	CA-2016-128608	12/01/2016	17/01/2016	Standard Class	CS-12490	Cindy Schnelling	Corporate	United States	Toledo
9799	9800	CA-2016-128608	12/01/2016	17/01/2016	Standard Class	CS-12490	Cindy Schnelling	Corporate	United States	Toledo

9800 rows × 18 columns

In [4]:
```
df.head()
```

Out[4]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	
0	1	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Ke
1	2	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Ke
2	3	CA-2017-138688	12/06/2017	16/06/2017	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	Ca
3	4	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	
4	5	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	

In [ ]:
```
## we can see there is 9800 rows 18 columns
```

In [5]:
```
df.shape
```

Out[5]:
```
(9800, 18)
```

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9800 entries, 0 to 9799
Data columns (total 18 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Row ID         9800 non-null   int64  
 1   Order ID       9800 non-null   object 
 2   Order Date     9800 non-null   object 
 3   Ship Date      9800 non-null   object 
 4   Ship Mode      9800 non-null   object 
 5   Customer ID    9800 non-null   object 
 6   Customer Name  9800 non-null   object 
 7   Segment        9800 non-null   object 
 8   Country        9800 non-null   object 
 9   City           9800 non-null   object 
 10  State          9800 non-null   object 
 11  Postal Code    9789 non-null   float64
 12  Region         9800 non-null   object 
 13  Product ID     9800 non-null   object 
 14  Category       9800 non-null   object 
 15  Sub-Category   9800 non-null   object 
 16  Product Name   9800 non-null   object 
 17  Sales          9800 non-null   float64
dtypes: float64(2), int64(1), object(15)
memory usage: 1.3+ MB
```

In [36]:
```
## In above cell we can obserb there is 2 float values, 1 integer value, 15 object
```

Checking the null values in dataset

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
Row ID            0
Order ID          0
Order Date        0
Ship Date         0
Ship Mode         0
Customer ID       0
Customer Name     0
Segment           0
Country           0
City              0
State             0
Postal Code      11
Region            0
Product ID        0
Category          0
Sub-Category      0
Product Name      0
Sales             0
dtype: int64
```

In [8]:
```
df.drop(['Row ID','Order ID','Postal Code','Product ID'],inplace=True,axis=1)
```

In [9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9800 entries, 0 to 9799
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Order Date     9800 non-null   object 
 1   Ship Date      9800 non-null   object 
 2   Ship Mode      9800 non-null   object 
 3   Customer ID    9800 non-null   object 
 4   Customer Name  9800 non-null   object 
 5   Segment        9800 non-null   object 
 6   Country        9800 non-null   object 
 7   City           9800 non-null   object 
 8   State          9800 non-null   object 
 9   Region         9800 non-null   object 
 10  Category       9800 non-null   object 
 11  Sub-Category   9800 non-null   object 
 12  Product Name   9800 non-null   object 
 13  Sales          9800 non-null   float64
dtypes: float64(1), object(13)
memory usage: 1.0+ MB
```

In [10]:
```
df.dtypes
```

Out[10]:
```
Order Date        object
Ship Date         object
Ship Mode         object
Customer ID       object
Customer Name     object
Segment           object
Country           object
City              object
State             object
Region            object
Category          object
Sub-Category      object
Product Name      object
Sales            float64
dtype: object
```

In [11]:
```
df.columns
```

Out[11]:
```
Index(['Order Date', 'Ship Date', 'Ship Mode', 'Customer ID', 'Customer Name',
       'Segment', 'Country', 'City', 'State', 'Region', 'Category',
       'Sub-Category', 'Product Name', 'Sales'],
      dtype='object')
```

In [14]:
```
df.shape
```

Out[14]:
```
(9800, 14)
```

In [16]:
```
df.drop(['Ship Date','Customer ID','Customer Name','Product Name'],inplace=True,axis=1
```

In [17]:
```
df.shape
```

Out[17]:
```
(9800, 10)
```

In [18]:
```
df.isnull().sum()
```

Out[18]:
```
Order Date      0
Ship Mode       0
Segment         0
Country         0
City            0
State           0
Region          0
Category        0
Sub-Category    0
Sales           0
dtype: int64
```

In [ ]:
```
## In avobe cell there is no null value
```

Checking dublicate values in dataset

In [19]:
```
df.duplicated().sum()
```

Out[19]:
```
2
```

In [37]:
```
### Droping duplicate value
```

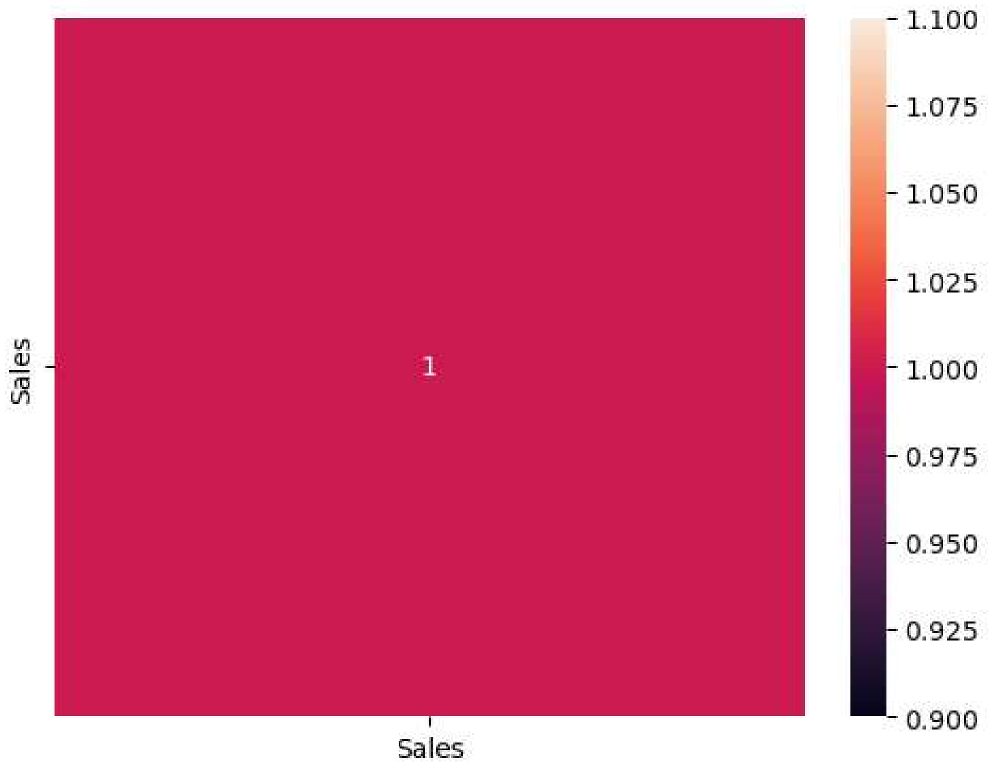
In [20]:
```
df.drop_duplicates(inplace=True)
```

In [21]:
```
df.duplicated().sum()
```

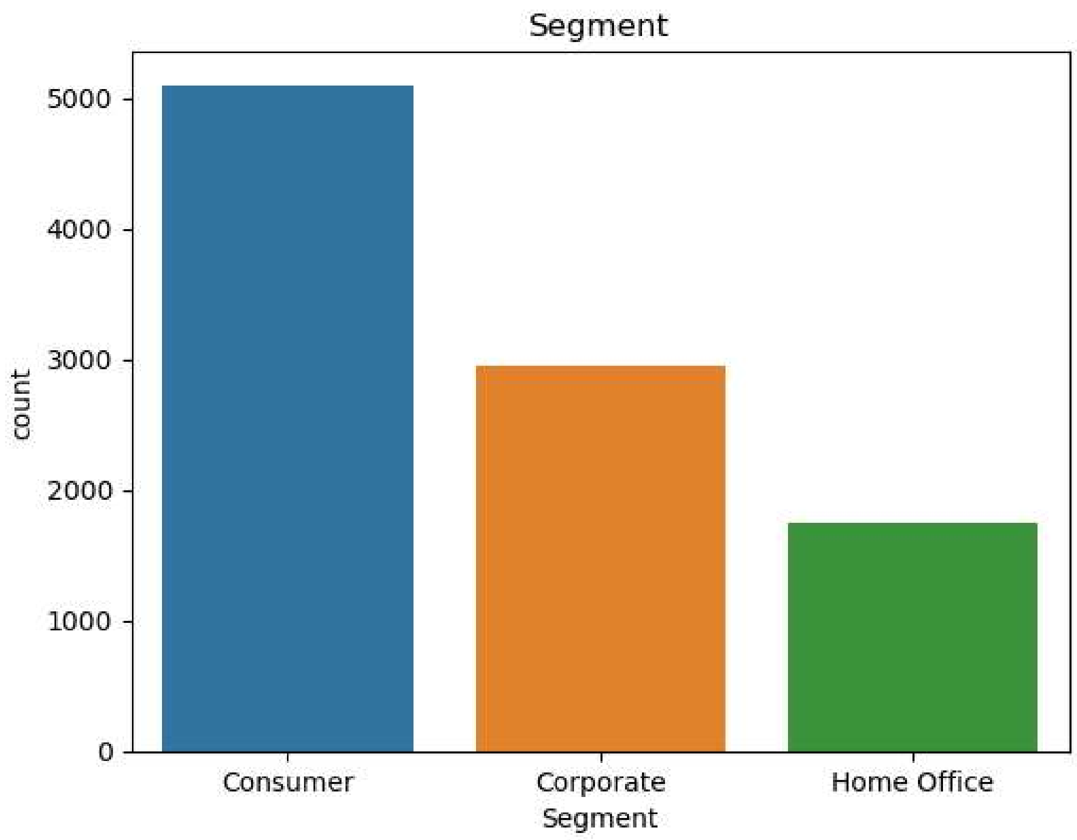
Out[21]:
```
0
```

EDA

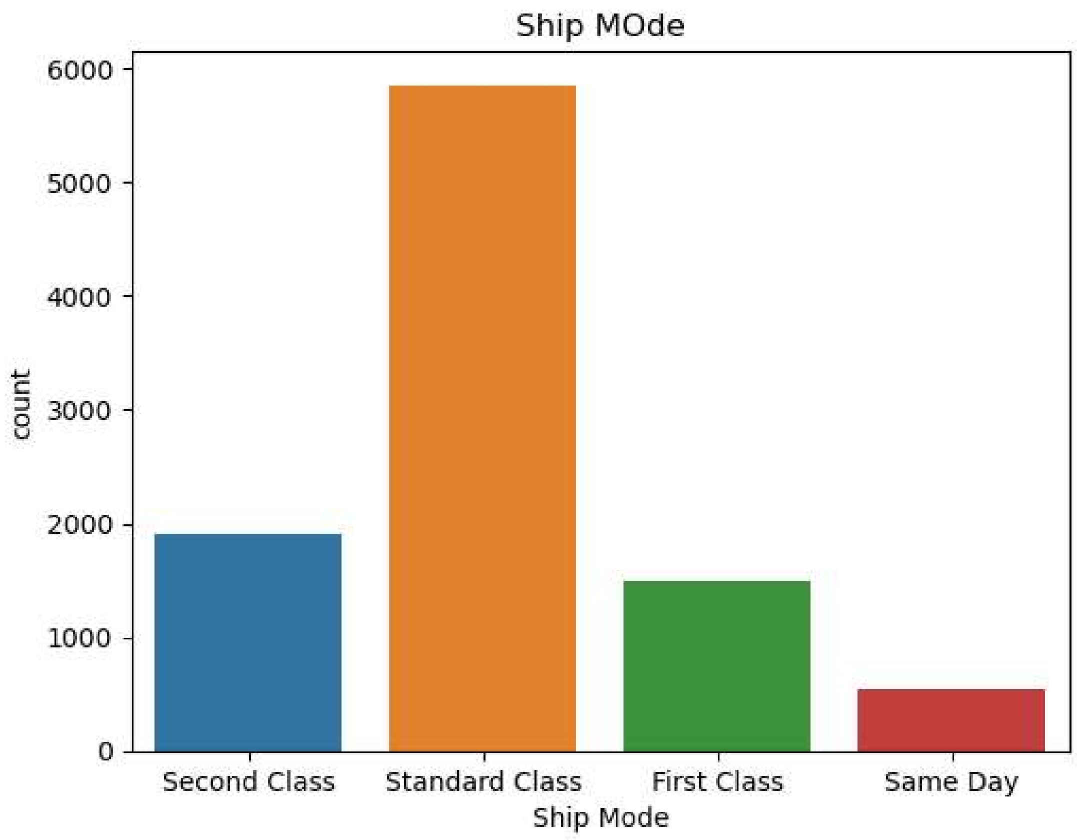
```
In [31]: sns.heatmap(df.corr(),annot=True)
         plt.show()
```



```
In [24]: sns.countplot(data=df,x='Segment')
         plt.title('Segment')
         plt.show()
```



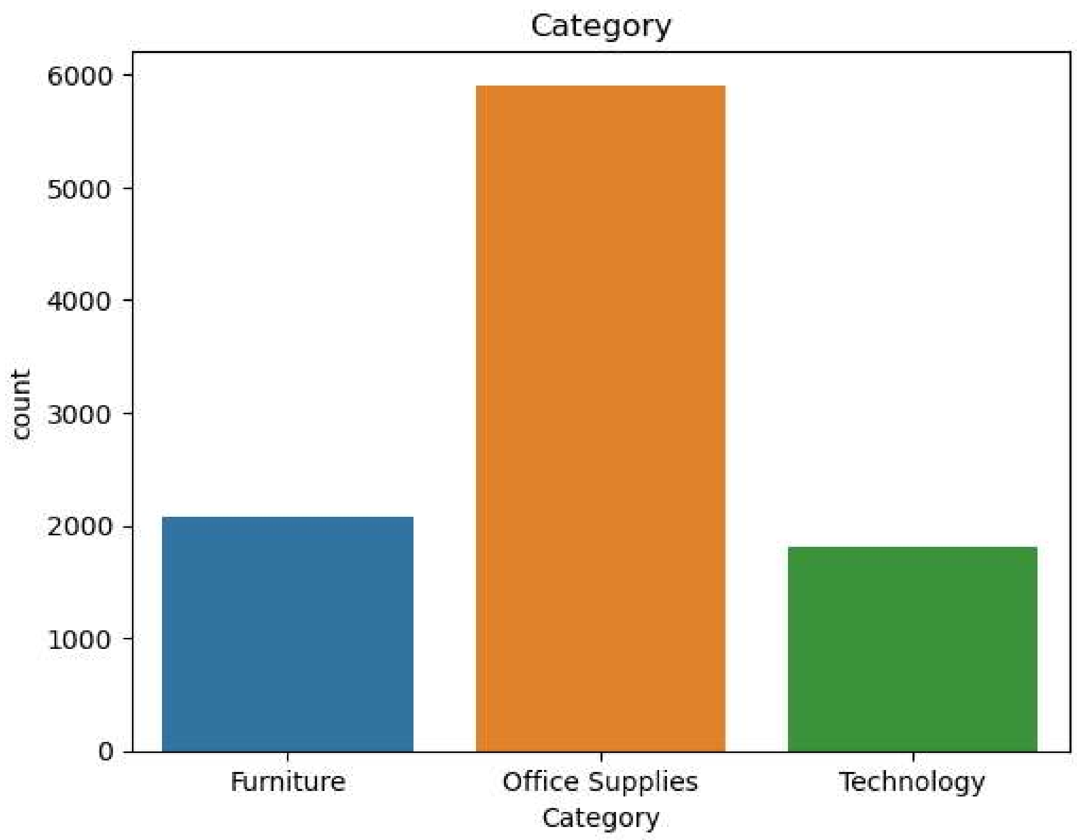
In [26]:
```
sns.countplot(data=df,x='Ship Mode')
plt.title('Ship MOde')
plt.show()
```



From the above plot we can easily see that:-

- items can be send throuugh the standard class is more

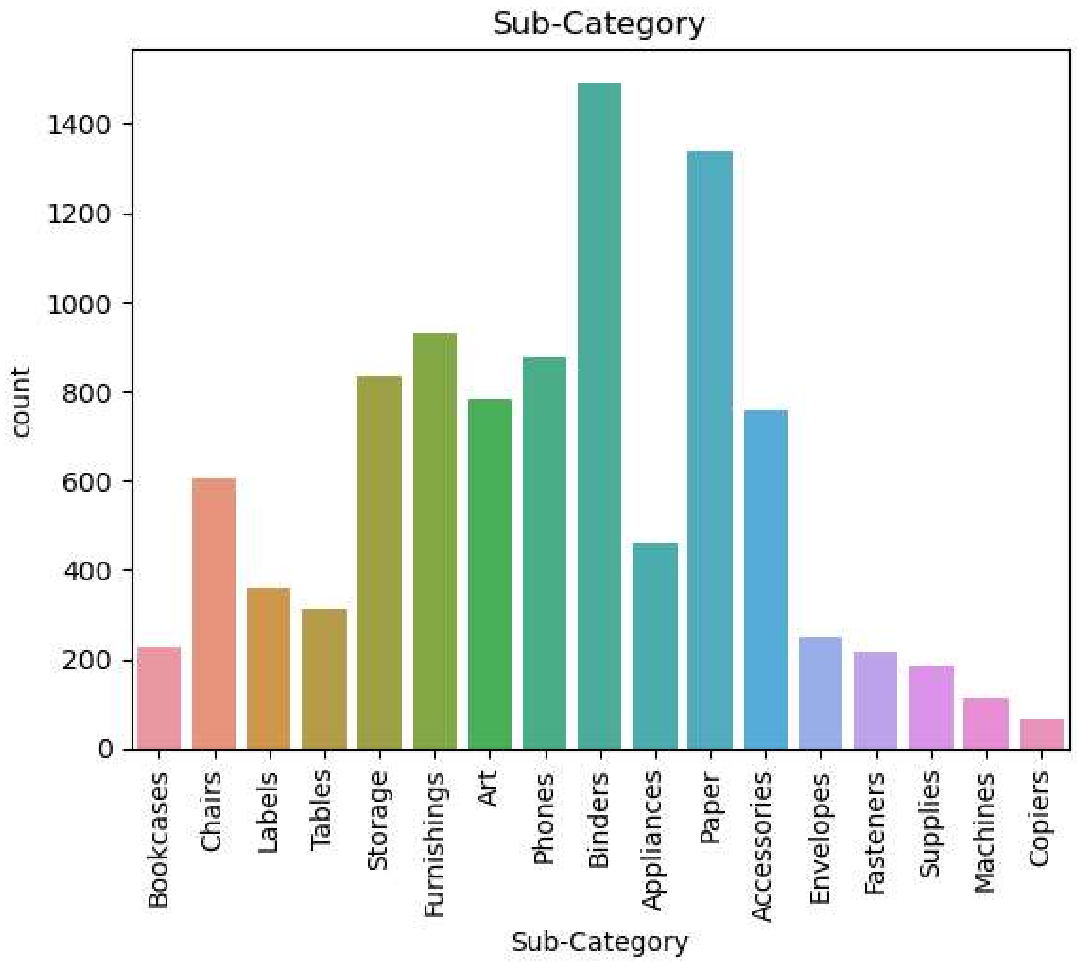
In [27]:
```
sns.countplot(data=df,x='Category')
plt.title('Category')
plt.show()
```



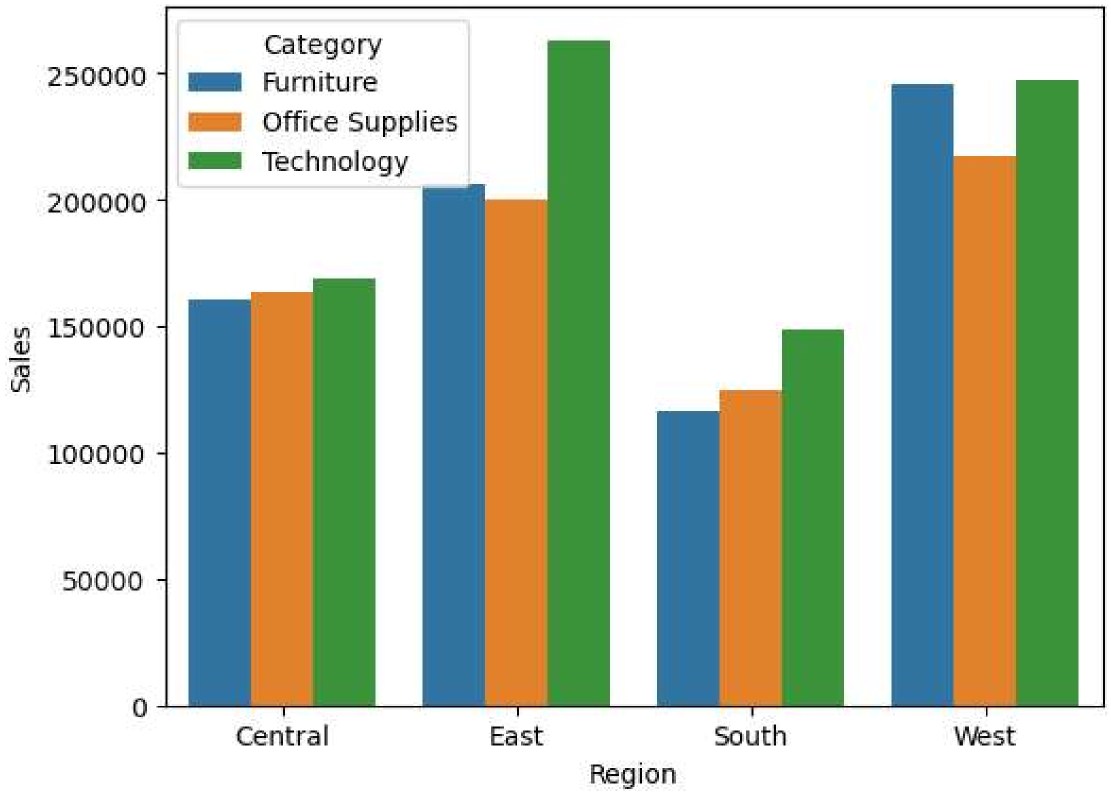
From the above plot we can see that-

- In the category of the Items the office supplies is more than other

```
In [30]: sns.countplot(data=df,x='Sub-Category')
         plt.title('Sub-Category')
         plt.xticks(rotation=90)
         plt.show()
```



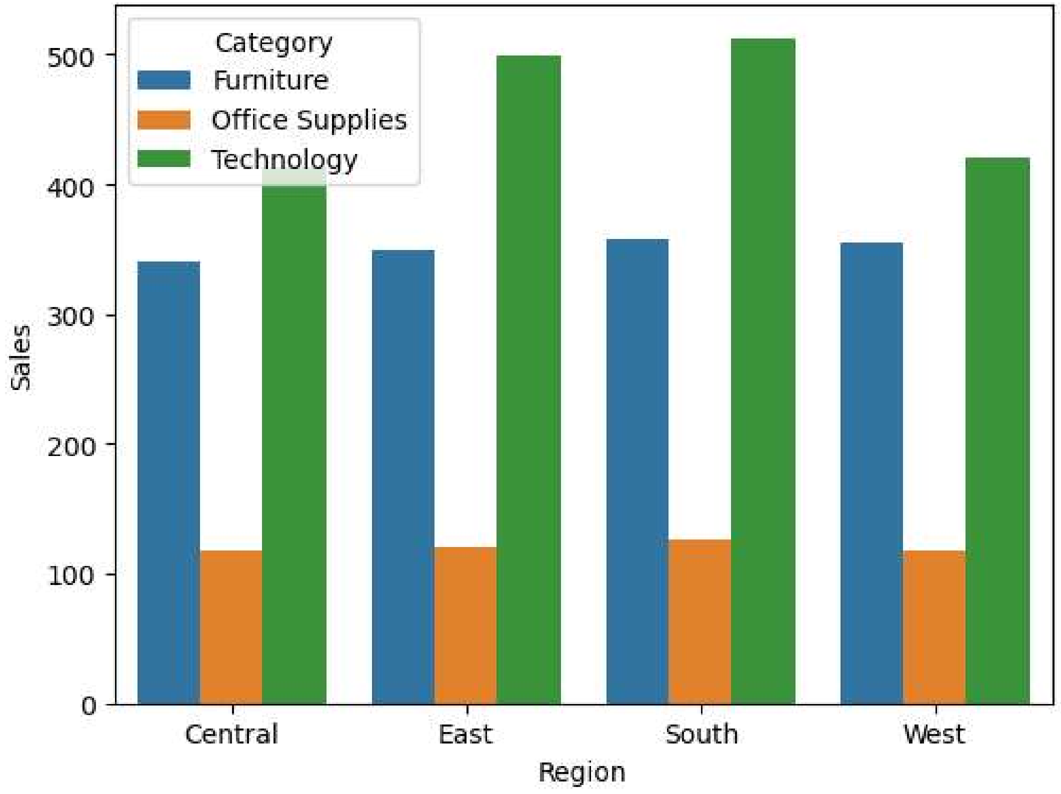
In [22]:
```
Category  = df.groupby(['Region',"Category"], as_index=False).agg({'Sales': "sum"})
Category.head
#so.Plot(data= Region, x="Region", y= "Sales", alpha = 'Category').add(so.Bar())
sns.barplot(data=Category, x="Region", y="Sales", hue="Category")
plt.show()
```



From the above plot we can easily see that-

- in east region technology have more sales
- in south region furniture ,office supplies,Technology have less sales

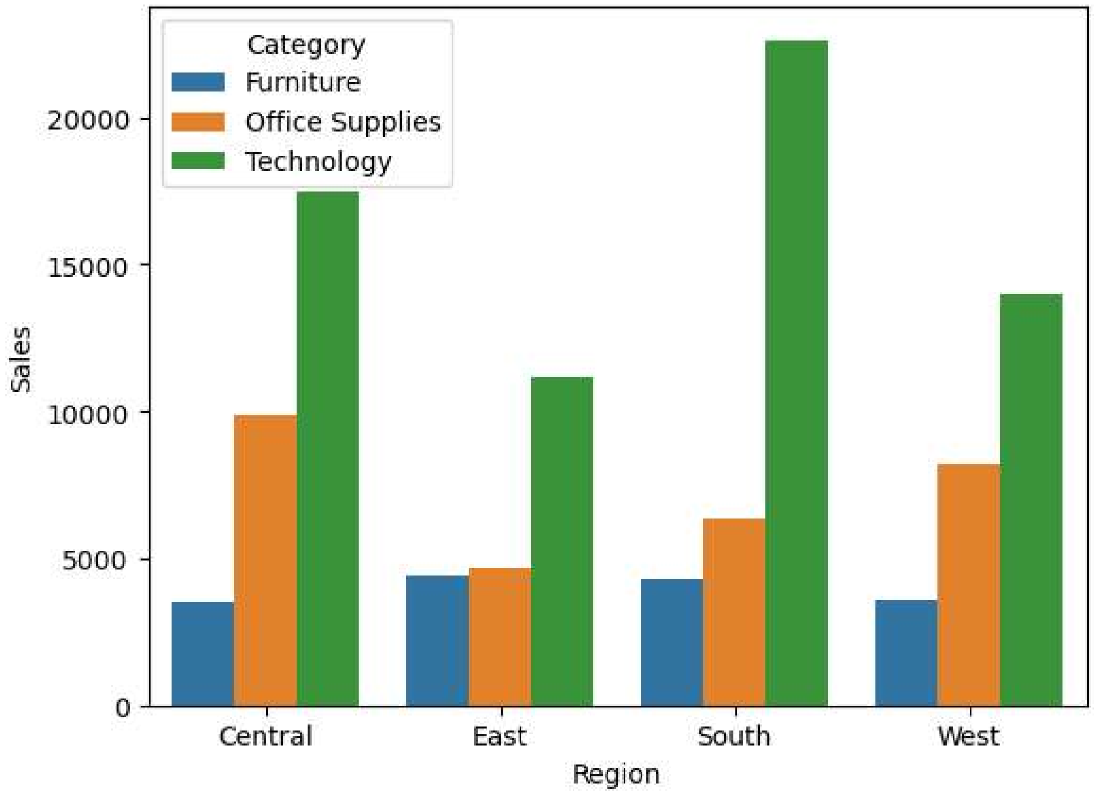
```
In [32]: Category  = df.groupby(['Region',"Category"], as_index=False).agg({'Sales': "mean"})
         Category.head
         #so.Plot(data= Region, x="Region", y= "Sales", alpha = 'Category').add(so.Bar())
         sns.barplot(data=Category, x="Region", y="Sales", hue="Category")
         plt.show()
```



From the above plot we can easily see that-

- In all regions technology have high sales and specially in south region technology have more sales
- in all the region office supplies have less sales

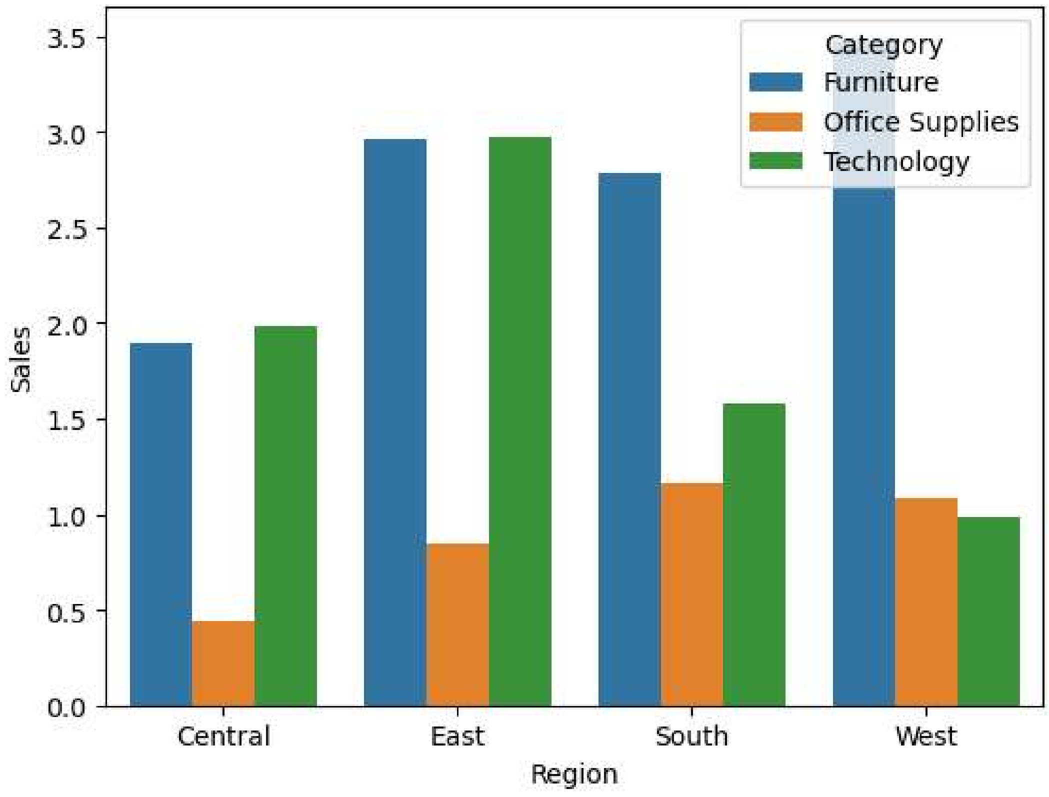
```
In [33]: Category  = df.groupby(['Region',"Category"], as_index=False).agg({'Sales': "max"})
         Category.head
         #so.Plot(data= Region, x="Region", y= "Sales", alpha = 'Category').add(so.Bar())
         sns.barplot(data=Category, x="Region", y="Sales", hue="Category")
         plt.show()
```



From the above plot we can easily see that-

- In south region technology have maximum sales
- In west region furniture have less sales

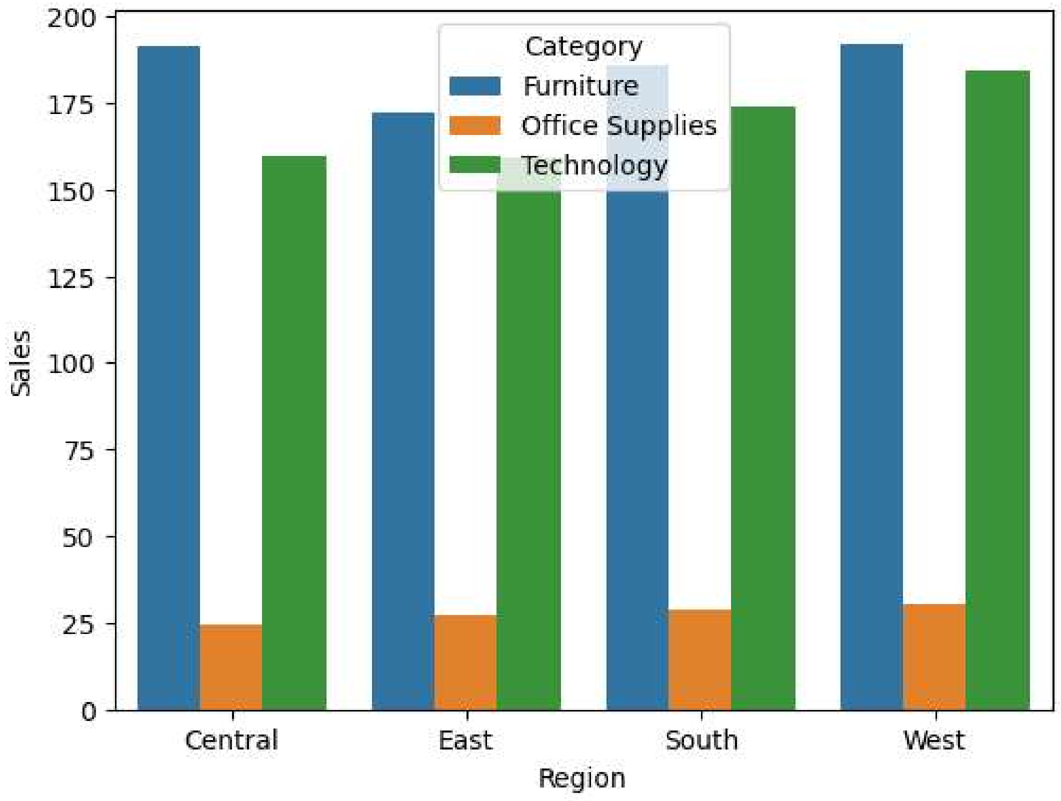
```
In [34]: Category  = df.groupby(['Region',"Category"], as_index=False).agg({'Sales': "min"})
         Category.head
         #so.Plot(data= Region, x="Region", y= "Sales", alpha = 'Category').add(so.Bar())
         sns.barplot(data=Category, x="Region", y="Sales", hue="Category")
         plt.show()
```



From the above plot we can easily see that-

- in west region furniture sales is more than office supplies and technology
- office supply sales is very less in central region

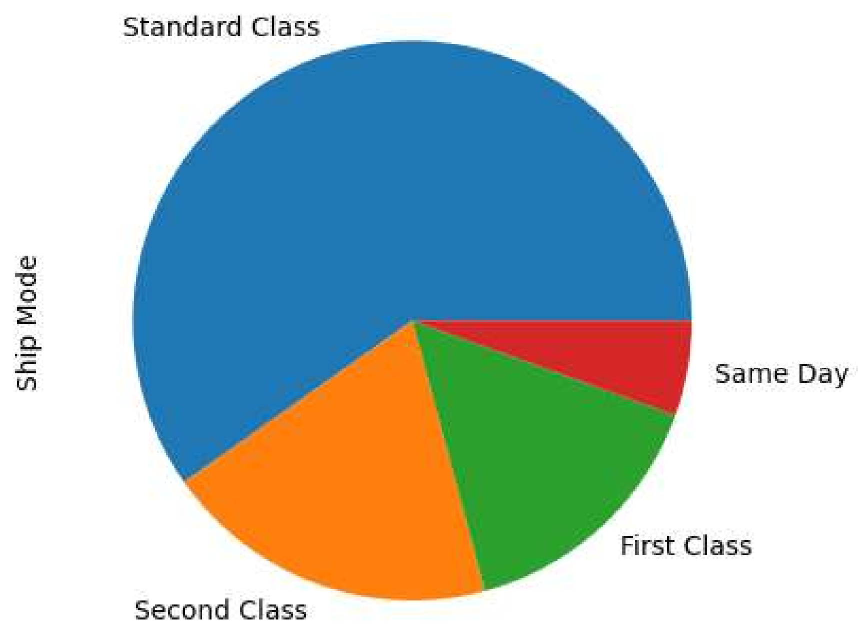
```
In [35]: Category  = df.groupby(['Region',"Category"], as_index=False).agg({'Sales': "median"})
         Category.head
         #so.Plot(data= Region, x="Region", y= "Sales", alpha = 'Category').add(so.Bar())
         sns.barplot(data=Category, x="Region", y="Sales", hue="Category")
         plt.show()
```



From the above plot we can easily see that-

- Furniture has more sales in all region

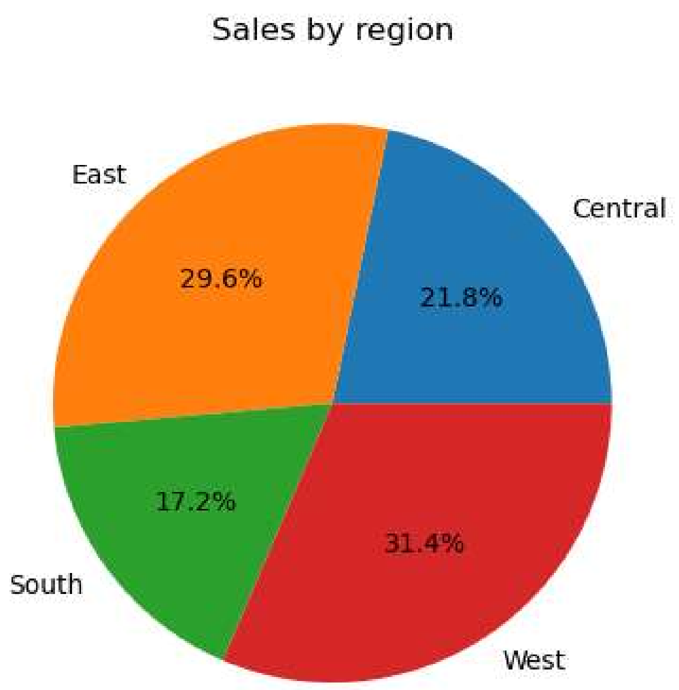
In [41]:
```
df["Ship Mode"].value_counts().plot.pie()
plt.show()
```



From the above plot we can easily see that-

- items can be send through stadard class is more

```
In [23]: Region = df.groupby(['Region'], as_index=False).agg({'Sales': "sum"})
         Region.head
         #sns.color_palette(palette= 'bright')
         #so.Plot(data= Region, x="Region", y= "Sales").add(so.Bar())
         #sns.lineplot(x='year',y= 'Sales', data= Region, hue='Region')
         plt.pie(Region['Sales'],labels=Region['Region'], autopct='%1.1f%%')
         #Region.plot.pie(y='Sales')
         plt.title("Sales by region")
         plt.show()
```



From the above plot we can easily see that-

- West region has high sales among all the region

```
In [ ]:
```