```
#Load the neccessary Libraries

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

```
#Load the Dataset
df=pd.read_csv("/Analysis of Super Store .csv")
df.head()
```

| | Ship Mode | Segment | Country | City | State | Postal Code | Region | Category | Su Catego |
|---|---|---|---|---|---|---|---|---|---|
| **0** | Second Class | Consumer | United States | Henderson | Kentucky | 42420 | South | Furniture | Bookcas |
| **1** | Second Class | Consumer | United States | Henderson | Kentucky | 42420 | South | Furniture | Cha |
| **2** | Second Class | Corporate | United States | Los Angeles | California | 90036 | West | Office Supplies | Lab |
| **3** | Standard Class | Consumer | United States | Fort Lauderdale | Florida | 33311 | South | Furniture | Tabl |
| **4** | Standard | Consumer | United | Fort | Florida | 33311 | South | Office | Stora |

Start coding or generate with AI.

```
#Dropping Postal Code Coloumn
df.drop(columns="Postal Code")
```

| | Ship Mode | Segment | Country | City | State | Region | Category | Sub-Category |
|---|---|---|---|---|---|---|---|---|
| **0** | Second Class | Consumer | United States | Henderson | Kentucky | South | Furniture | Bookcases |
| **1** | Second Class | Consumer | United States | Henderson | Kentucky | South | Furniture | Chairs |
| **2** | Second Class | Corporate | United States | Los Angeles | California | West | Office Supplies | Labels |
| **3** | Standard Class | Consumer | United States | Fort Lauderdale | Florida | South | Furniture | Tables |
| **4** | Standard Class | Consumer | United States | Fort Lauderdale | Florida | South | Office Supplies | Storage |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **9989** | Second Class | Consumer | United States | Miami | Florida | South | Furniture | Furnishings |

```
df.head()
```

| | Ship Mode | Segment | Country | City | State | Postal Code | Region | Category | Su Catego |
|---|---|---|---|---|---|---|---|---|---|
| **0** | Second Class | Consumer | United States | Henderson | Kentucky | 42420 | South | Furniture | Bookcas |
| **1** | Second Class | Consumer | United States | Henderson | Kentucky | 42420 | South | Furniture | Cha |
| **2** | Second Class | Corporate | United States | Los Angeles | California | 90036 | West | Office Supplies | Lab |
| **3** | Standard Class | Consumer | United States | Fort Lauderdale | Florida | 33311 | South | Furniture | Tabl |
| **4** | Standard | Consumer | United | Fort | Florida | 33311 | South | Office | Stora |

```
df.drop(columns= "Postal Code" ,inplace=True )
df.head()
```

| | Ship Mode | Segment | Country | City | State | Region | Category | Sub-Category | Sa |
|---|---|---|---|---|---|---|---|---|---|
| **0** | Second Class | Consumer | United States | Henderson | Kentucky | South | Furniture | Bookcases | 261.9 |
| **1** | Second Class | Consumer | United States | Henderson | Kentucky | South | Furniture | Chairs | 731.9 |
| | Second | | United | Los | | | Office | | |

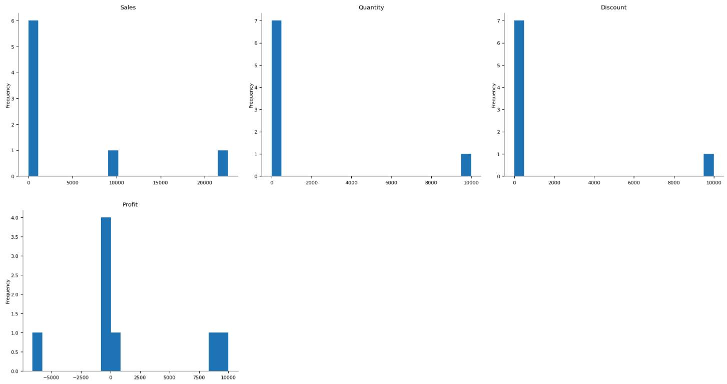
```
# Viwing the unique values from the dataset
print(df["Ship Mode"].unique())
print(df["Segment"].unique())
print(df["Region"].unique())
print(df["Category"].unique())
print(df["Country"].unique())
print(df["Sub-Category"].unique())
```

```
['Second Class' 'Standard Class' 'First Class' 'Same Day']
['Consumer' 'Corporate' 'Home Office']
['South' 'West' 'Central' 'East']
['Furniture' 'Office Supplies' 'Technology']
['United States']
['Bookcases' 'Chairs' 'Labels' 'Tables' 'Storage' 'Furnishings' 'Art'
 'Phones' 'Binders' 'Appliances' 'Paper' 'Accessories' 'Envelopes'
 'Fasteners' 'Supplies' 'Machines' 'Copiers']
```

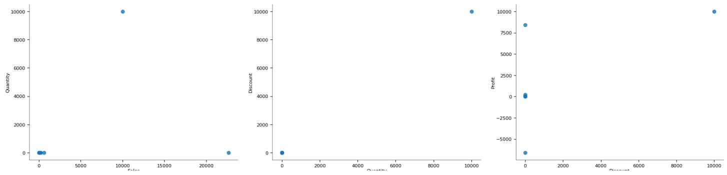
```
  # Describing the  Statistical data
df.describe()
```

| | Sales | Quantity | Discount | Profit |
|---|---|---|---|---|
| **count** | 9994.000000 | 9994.000000 | 9994.000000 | 9994.000000 |
| **mean** | 229.858001 | 3.789574 | 0.156203 | 28.656896 |
| **std** | 623.245101 | 2.225110 | 0.206452 | 234.260108 |
| **min** | 0.444000 | 1.000000 | 0.000000 | -6599.978000 |
| **25%** | 17.280000 | 2.000000 | 0.000000 | 1.728750 |
| **50%** | 54.490000 | 3.000000 | 0.200000 | 8.666500 |
| **75%** | 209.940000 | 5.000000 | 0.200000 | 29.364000 |
| **max** | 22638.480000 | 14.000000 | 0.800000 | 8399.976000 |

**Distributions**



**2-d distributions**



```
# viwing the data types of every coloumns

df.info()
```

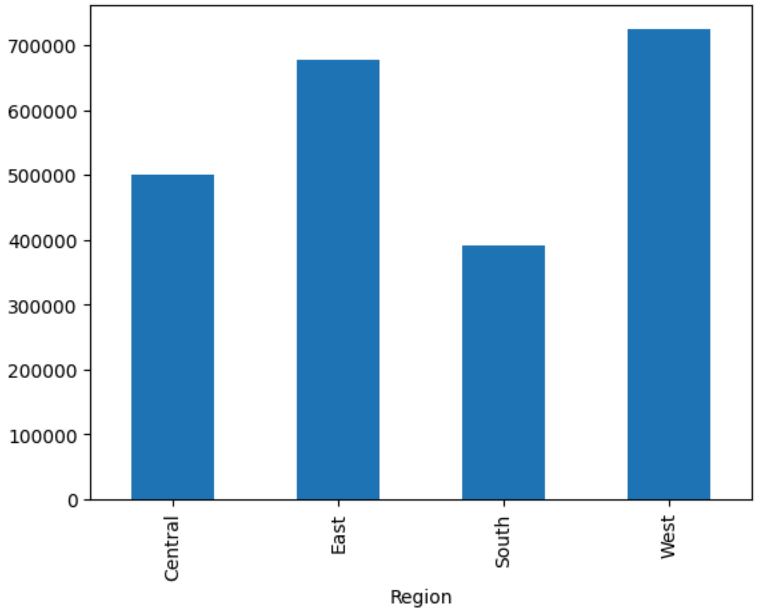
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Ship Mode     9994 non-null   object 
 1   Segment       9994 non-null   object 
 2   Country       9994 non-null   object 
 3   City          9994 non-null   object 
 4   State         9994 non-null   object 
 5   Region        9994 non-null   object 
 6   Category      9994 non-null   object 
 7   Sub-Category  9994 non-null   object 
 8   Sales         9994 non-null   float64
 9   Quantity      9994 non-null   int64  
 10  Discount      9994 non-null   float64
 11  Profit        9994 non-null   float64
dtypes: float64(3), int64(1), object(8)
memory usage: 937.1+ KB
```

```
# viwing for  any na values in dataset
df.isna().sum()
```

```
Ship Mode       0
Segment         0
Country         0
City            0
State           0
Region          0
Category        0
Sub-Category    0
Sales           0
Quantity        0
Discount        0
Profit          0
dtype: int64
```

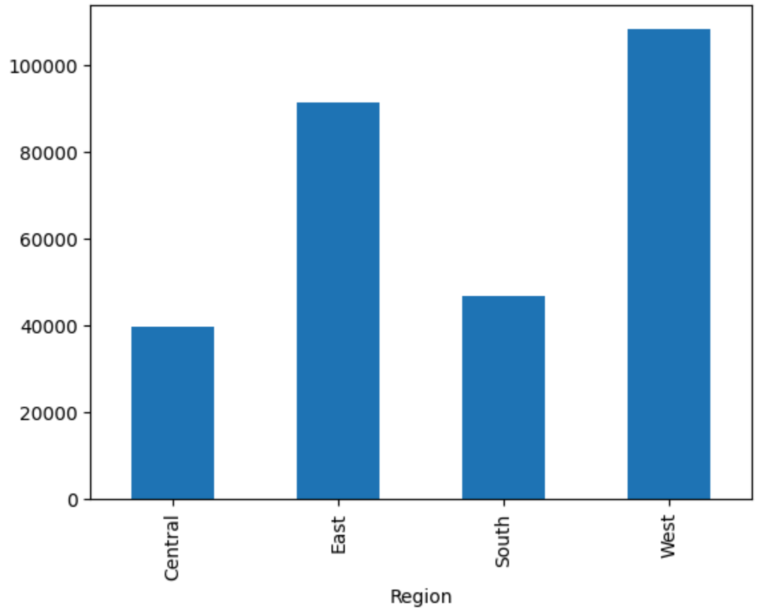
```
  # Sales Analysis based on Region
df.groupby("Region")["Sales"].sum().plot.bar()
```

```
<Axes: xlabel='Region'>
```



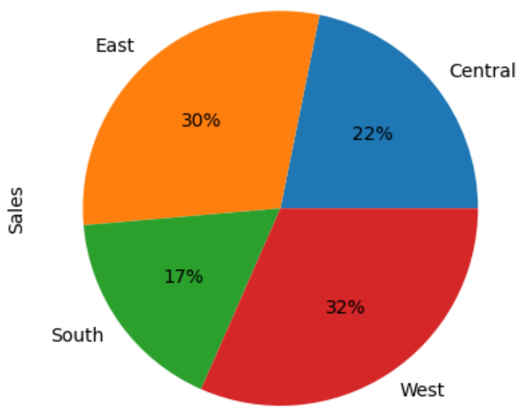
```
# Profit Analysis Based on region

df.groupby("Region")["Profit"].sum().plot.bar()
```

```
<Axes: xlabel='Region'>
```



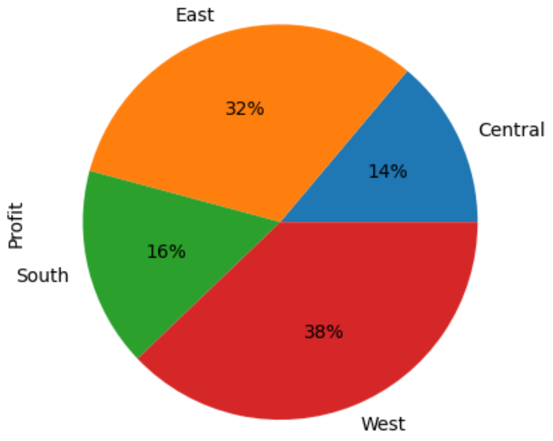
```
# Sales Analysis based on Region pie chart
df.groupby("Region")["Sales"].sum().plot.pie(autopct="%1.0f%%")
```

```
<Axes: ylabel='Sales'>
```



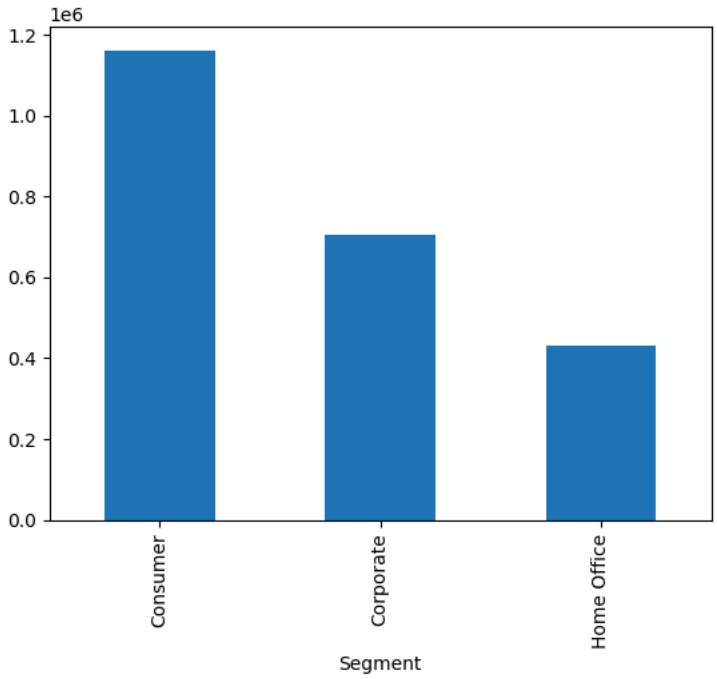
```
# Profit  Analysis based on Region pie chart
df.groupby("Region")["Profit"].sum().plot.pie(autopct="%1.0f%%")
```

```
<Axes: ylabel='Profit'>
```



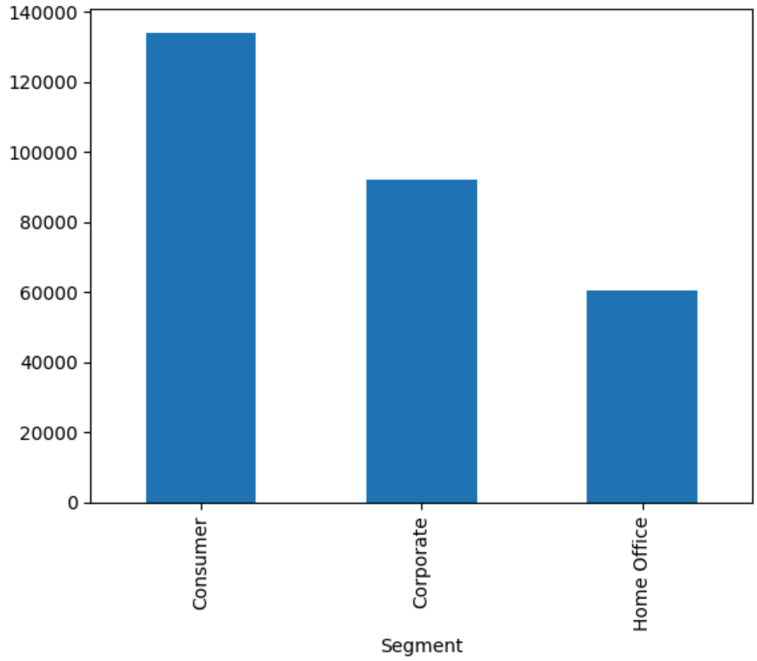
```
#Sales Analysis based on Customer segment
df.groupby("Segment")["Sales"].sum().plot.bar()
```

```
<Axes: xlabel='Segment'>
```



```
#Profit Analysis based on customer Segment
df.groupby("Segment")["Profit"].sum().plot.bar()
```

```
<Axes: xlabel='Segment'>
```

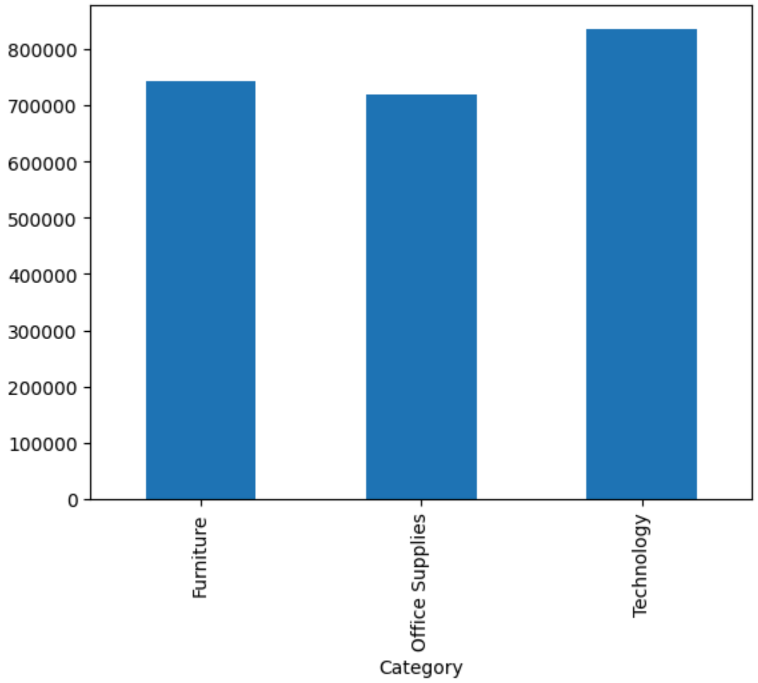


Start coding or generate with AI.

Start coding or generate with AI.

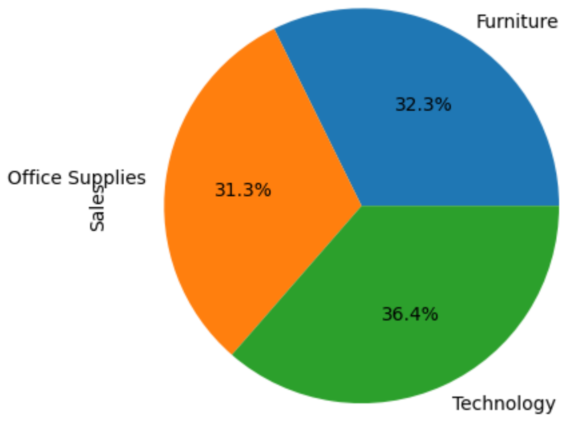
```
# sales Analysis based on category
df.groupby("Category")["Sales"].sum().plot.bar()
```

```
<Axes: xlabel='Category'>
```



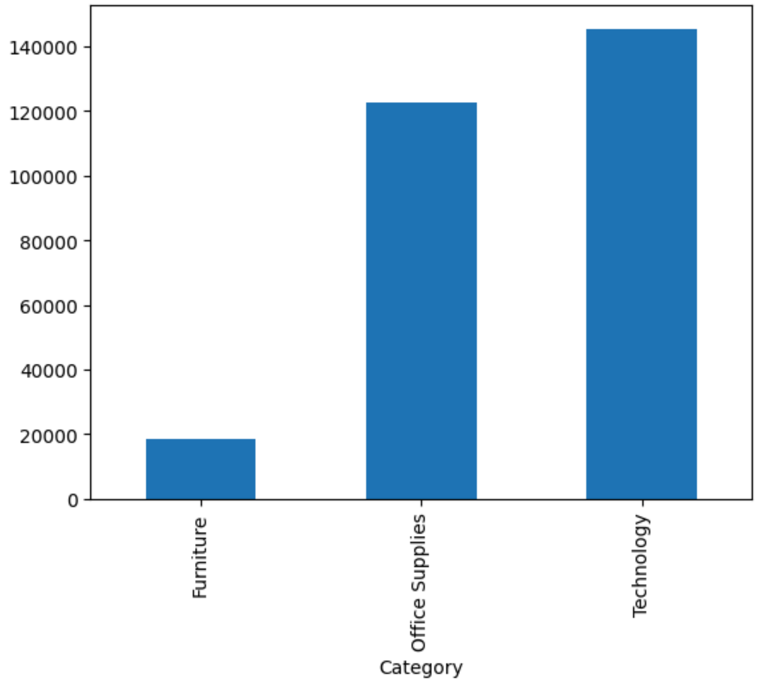
```
# sales Analysis based on category  using piechart
df.groupby("Category")["Sales"].sum().plot.pie(autopct="%0.1f%%")
```

```
<Axes: ylabel='Sales'>
```



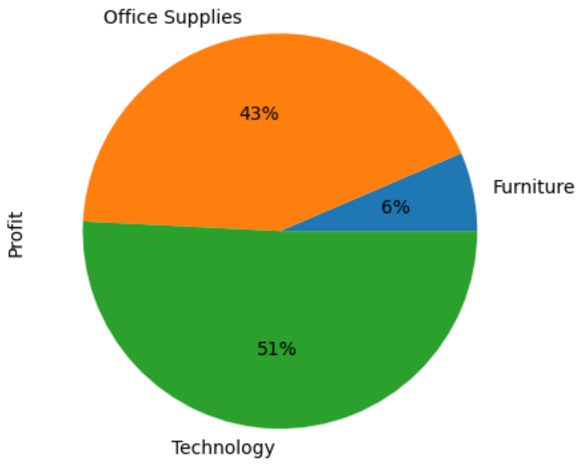
```
#Profit Analysis Based on Category
df.groupby("Category")["Profit"].sum().plot.bar()
```

```
<Axes: xlabel='Category'>
```



```
#Profit Analysis Based on Category using Pie chart
df.groupby("Category")["Profit"].sum().plot.pie(autopct="%1.0f%%")
```

```
<Axes: ylabel='Profit'>
```

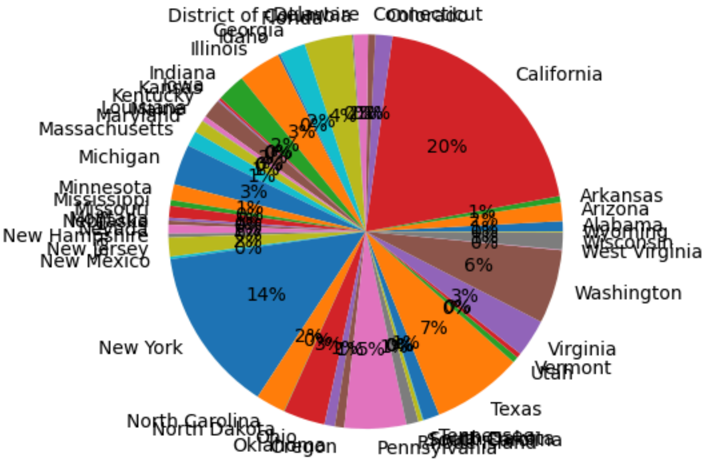


```
# Sales Analysis Based on State
df.groupby("State")["Sales"].sum().plot.bar()
```

```
<Axes: xlabel='State'>
```

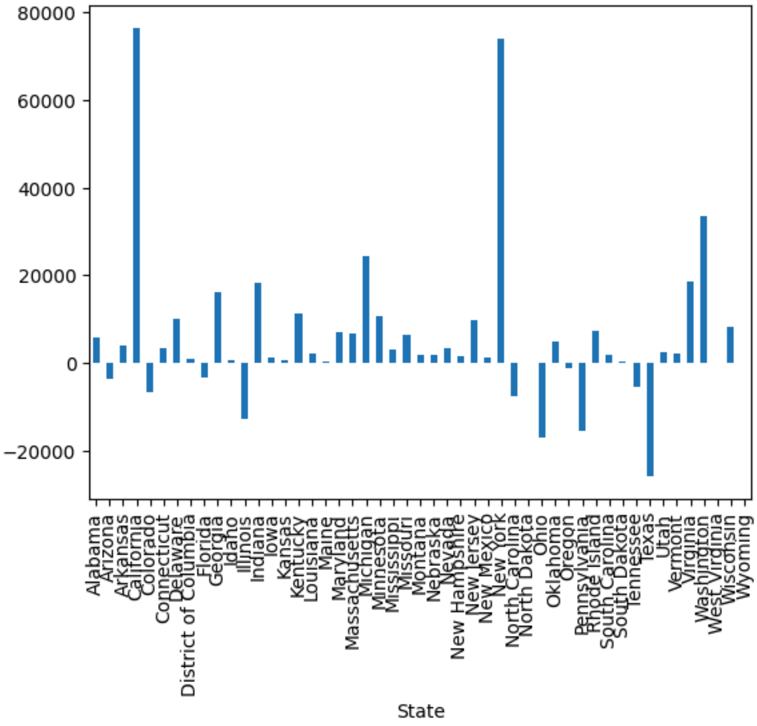
```
# Sales Analysis Based on State
df.groupby("State")["Sales"].sum().plot.pie(autopct="%1.0f%%")
```

```
<Axes: ylabel='Sales'>
```



```
#Profit Analysis based on State
df.groupby("State")["Profit"].sum().plot.bar()
```

```
<Axes: xlabel='State'>
```



```
# max used shipping mode
df["Ship Mode"].value_counts()
```