

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
cd drive/
```

```
/content/drive
```

```
cd MyDrive
```

```
/content/drive/MyDrive
```

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

```
df=pd.read_csv("SampleSuperstore.csv")
df.head()
```

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Cat
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Book
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	

```
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	Tat
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	South	Office Supplies	Stor
...
9989	Second Class	Consumer	United States	Miami	Florida	South	Furniture	Furnishi
9990	Standard Class	Consumer	United States	Costa Mesa	California	West	Furniture	Furnishi

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

```
df.head()
```

	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

```
print(df["Ship Mode"].unique())
print(df["Segment"].unique())
print(df["Country"].unique())
```

```

print(df["Country"].unique())
print(df["Category"].unique())
print(df["City"].unique())
print(df["State"].unique())
print(df["Region"].unique())
print(df["Sub-Category"].unique())
print(df["Sales"].unique())
print(df["Quantity"].unique())
print(df["Discount"].unique())
print(df["Profit"].unique())

```

```

'Thousand Oaks' 'Malden' 'Holyoke' 'Albuquerque' 'Sparks' 'Coachella'
'Elmhurst' 'Passaic' 'North Charleston' 'Newport News' 'Jamestown'
'Mishawaka' 'La Quinta' 'Tallahassee' 'Nashville' 'Bellingham'
'Woodstock' 'Haltom City' 'Wheeling' 'Summerville' 'Hot Springs'
'Englewood' 'Las Cruces' 'Hoover' 'Frisco' 'Vacaville' 'Waukesha'
'Bakersfield' 'Pompano Beach' 'Corpus Christi' 'Redondo Beach' 'Orlando'
'Orange' 'Lake Charles' 'Highland Park' 'Hempstead' 'Noblesville'
'Apple Valley' 'Mount Pleasant' 'Sterling Heights' 'Eau Claire' 'Pharr'
'Billings' 'Gresham' 'Chattanooga' 'Meridian' 'Bolingbrook' 'Maple Grove'
'Woodland' 'Missouri City' 'Pearland' 'San Mateo' 'Grand Rapids'
'Visalia' 'Overland Park' 'Temecula' 'Yucaipa' 'Revere' 'Conroe'
'Tinley Park' 'Dubuque' 'Dearborn Heights' 'Santa Fe' 'Hickory'
'Carol Stream' 'Saint Cloud' 'North Miami' 'Plantation'
'Port Saint Lucie' 'Rock Hill' 'Odessa' 'West Allis' 'Chula Vista'
'Manhattan' 'Altoona' 'Thornton' 'Champaign' 'Texarkana' 'Edinburg'
'Baytown' 'Greenwood' 'Woonsocket' 'Superior' 'Bedford' 'Covington'
'Broken Arrow' 'Miramar' 'Hollywood' 'Deer Park' 'Wichita' 'McAllen'
'Iowa City' 'Boise' 'Cranston' 'Port Arthur' 'Citrus Heights'
'The Colony' 'Daytona Beach' 'Bullhead City' 'Portage' 'Fargo' 'Elkhart'
'San Gabriel' 'Margate' 'Sandy Springs' 'Mentor' 'Lawton' 'Hampton'
'Rome' 'La Crosse' 'Lewiston' 'Hattiesburg' 'Danville' 'Logan'
'Waterbury' 'Athens' 'Avondale' 'Marietta' 'Yuma' 'Wausau' 'Pasco'
'Oak Park' 'Pensacola' 'League City' 'Gaithersburg' 'Lehi' 'Tuscaloosa'
'Moreno Valley' 'Georgetown' 'Loveland' 'Chandler' 'Helena' 'Kirkwood'
'Waco' 'Frankfort' 'Bethlehem' 'Grand Island' 'Woodbury' 'Rogers'
'Clovis' 'Jupiter' 'Santa Barbara' 'Cedar Hill' 'Norfolk' 'Draper'
'Ann Arbor' 'La Mesa' 'Pocatello' 'Holland' 'Milford' 'Buffalo Grove'
'Lake Forest' 'Redding' 'Chico' 'Utica' 'Conway' 'Cheyenne' 'Owensboro'
'Caldwell' 'Kenner' 'Nashua' 'Bartlett' 'Redwood City' 'Lebanon'
'Santa Maria' 'Des Plaines' 'Longview' 'Hendersonville' 'Waterloo'
'Cambridge' 'Palatine' 'Beverly' 'Eugene' 'Oxnard' 'Renton' 'Glenview'
'Delray Beach' 'Commerce City' 'Texas City' 'Wilson' 'Rio Rancho'
'Goldsboro' 'Montebello' 'El Cajon' 'Beaumont' 'West Palm Beach'
'Abilene' 'Normal' 'Saint Charles' 'Camarillo' 'Hillsboro' 'Burbank'
'Modesto' 'Garden City' 'Atlantic City' 'Longmont' 'Davis' 'Morgan Hill'
'Clifton' 'Sheboygan' 'East Point' 'Rapid City' 'Andover' 'Kissimmee'
'Shelton' 'Danbury' 'Sanford' 'San Marcos' 'Greeley' 'Mansfield' 'Elyria'
'Twin Falls' 'Coral Gables' 'Romeoville' 'Marlborough' 'Laurel' 'Bryan'
'Pine Bluff' 'Aberdeen' 'Hagerstown' 'East Orange' 'Arlington Heights'
'Oswego' 'Coon Rapids' 'San Clemente' 'San Luis Obispo' 'Springdale'
'Lodi' 'Mason']
['Kentucky' 'California' 'Florida' 'North Carolina' 'Washington' 'Texas'
'Wisconsin' 'Utah' 'Nebraska' 'Pennsylvania' 'Illinois' 'Minnesota'
'Michigan' 'Delaware' 'Indiana' 'New York' 'Arizona' 'Virginia'
'Tennessee' 'Alabama' 'South Carolina' 'Oregon' 'Colorado' 'Iowa' 'Ohio'
'Missouri' 'Oklahoma' 'New Mexico' 'Louisiana' 'Connecticut' 'New Jersey'
'Massachusetts' 'Georgia' 'Nevada' 'Rhode Island' 'Mississippi'
'Arkansas' 'Montana' 'New Hampshire' 'Maryland' 'District of Columbia'
'Kansas' 'Vermont' 'Maine' 'South Dakota' 'Idaho' 'North Dakota'
'Wyoming' 'West Virginia']
['South' 'West' 'Central' 'East']
['Bookcases' 'Chairs' 'Labels' 'Tables' 'Storage' 'Furnishings' 'Art'
'Phones' 'Binders' 'Appliances' 'Paper' 'Accessories' 'Envelopes'
'Fasteners' 'Supplies' 'Machines' 'Copiers']
[261.96 731.94 14.62 ... 437.472 97.98 243.16 ]
[ 2  3  5  7  4  6  9  1  8 14 11 13 10 12]
[0.   0.45 0.2  0.8  0.3  0.5  0.7  0.6  0.32 0.1  0.4  0.15]
[ 41.9136 219.582   6.8714 ... 16.124   4.1028 72.948 ]

```

```
df.describe()
```



```
df.info()
```

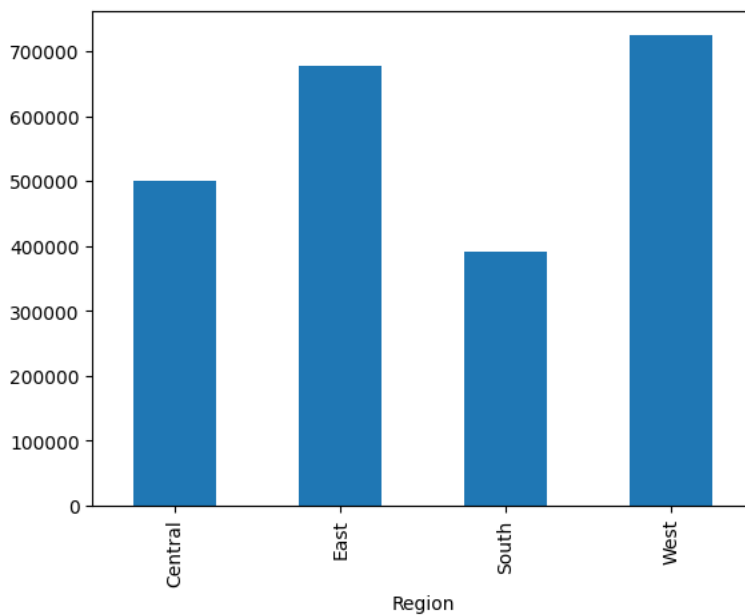
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype  
---  --
 0   Ship Mode       9994 non-null   object  
 1   Segment         9994 non-null   object  
 2   Country         9994 non-null   object  
 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
```

```
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
```

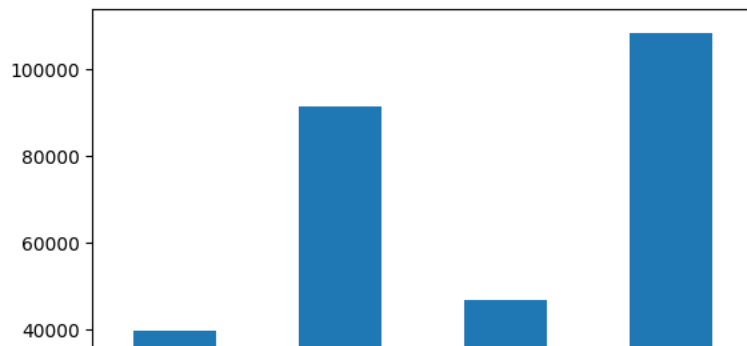
```
df.groupby("Region")["Sales"].sum().plot.bar()
```

<Axes: xlabel='Region'>



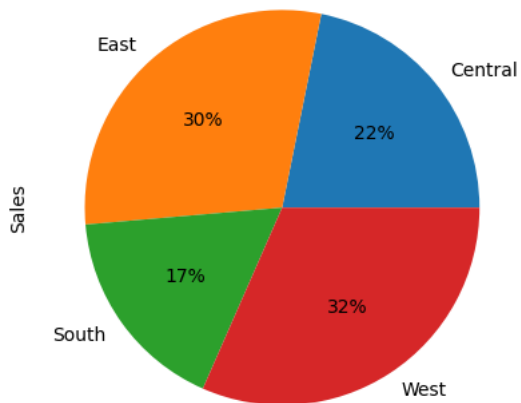
```
df.groupby("Region")["Profit"].sum().plot.bar()
```

<Axes: xlabel='Region'>



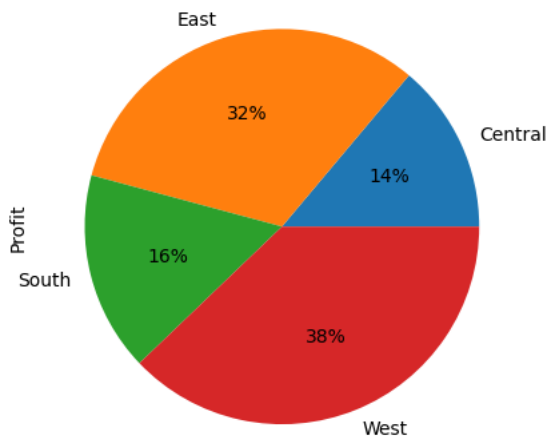
```
df.groupby("Region")["Sales"].sum().plot.pie(autopct="%1.0f%%")
```

<Axes: ylabel='Sales'>



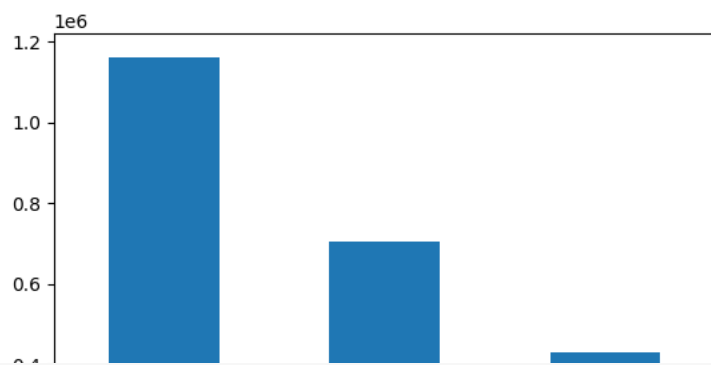
```
df.groupby("Region")["Profit"].sum().plot.pie(autopct="%1.0f%%")
```

<Axes: ylabel='Profit'>



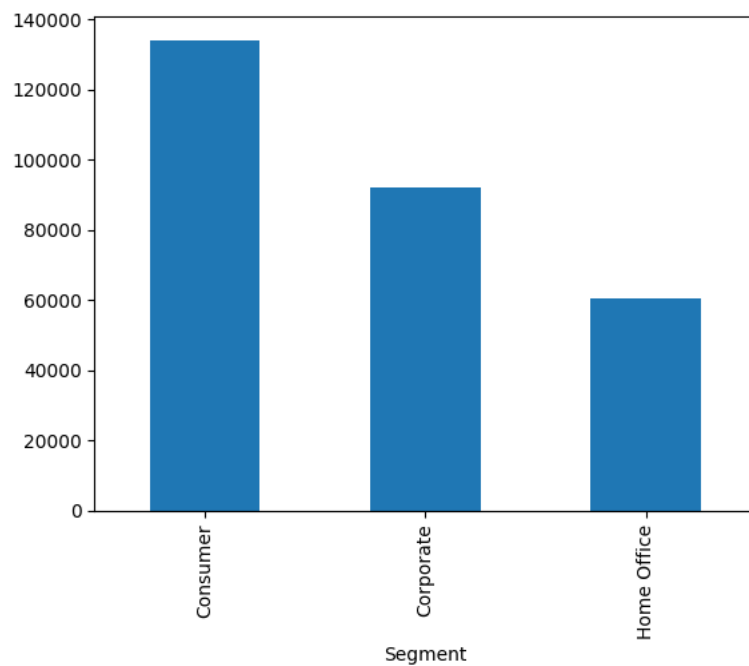
```
df.groupby("Segment")["Sales"].sum().plot.bar()
```

<Axes: xlabel='Segment'>



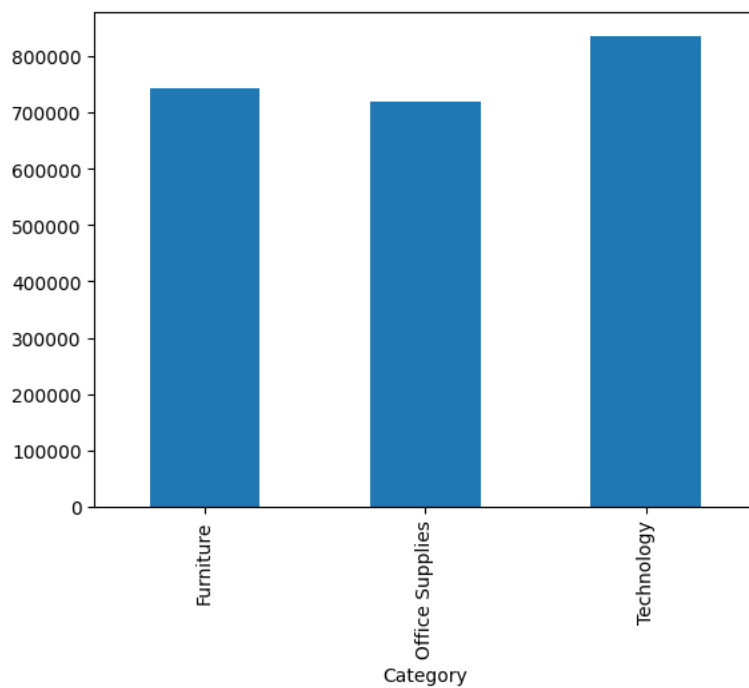
```
df.groupby("Segment")["Profit"].sum().plot.bar()
```

<Axes: xlabel='Segment'>



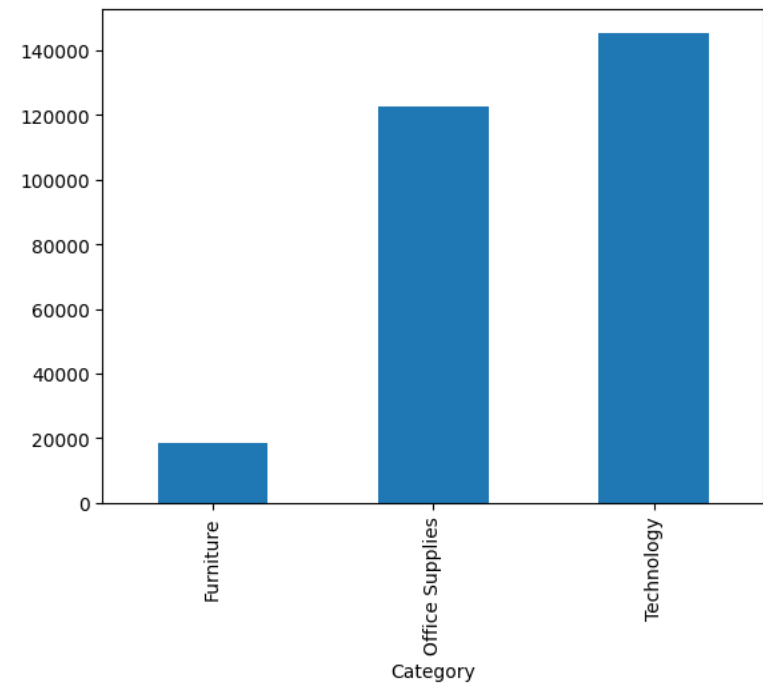
```
df.groupby("Category")["Sales"].sum().plot.bar()
```

<Axes: xlabel='Category'>



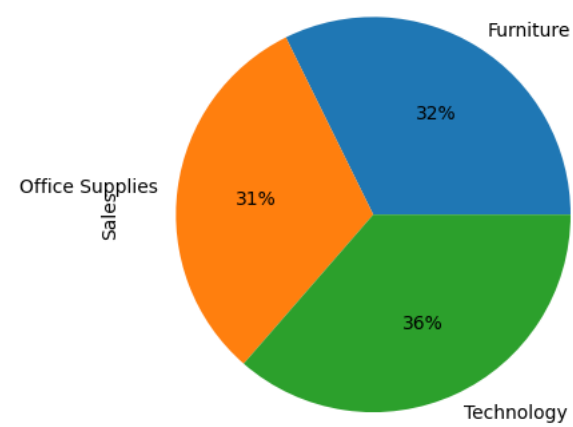
```
df.groupby("Category")["Profit"].sum().plot.bar()
```

<Axes: xlabel='Category'>



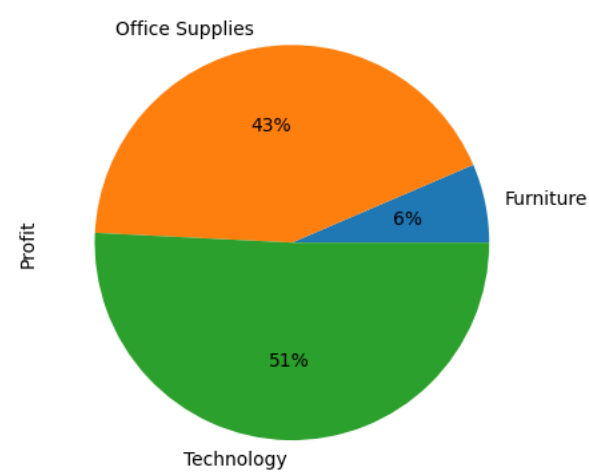
```
df.groupby("Category")["Sales"].sum().plot.pie(autopct="%1.0f%%")
```

<Axes: ylabel='Sales'>



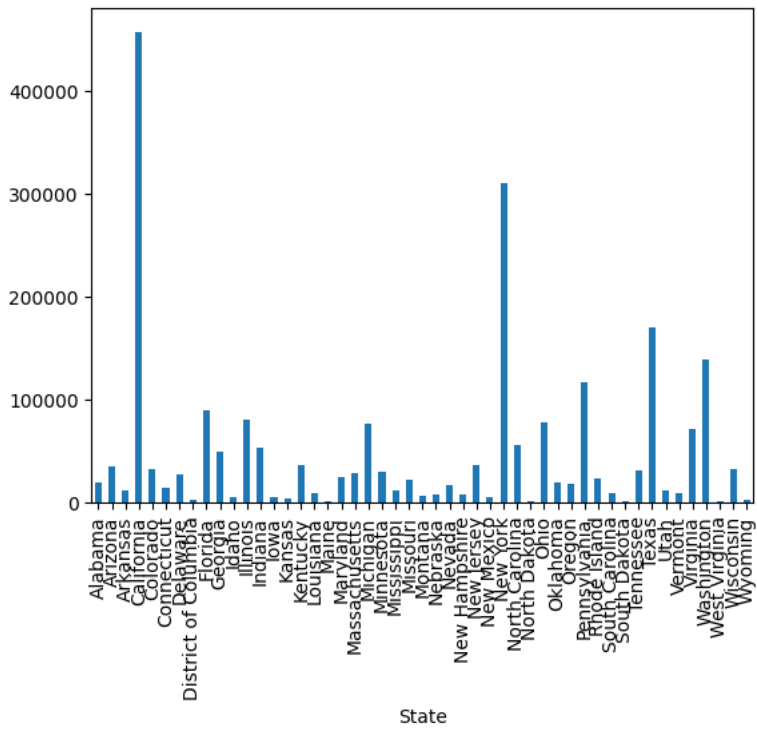
```
df.groupby("Category")["Profit"].sum().plot.pie(autopct="%1.0f%%")
```

<Axes: ylabel='Profit'>



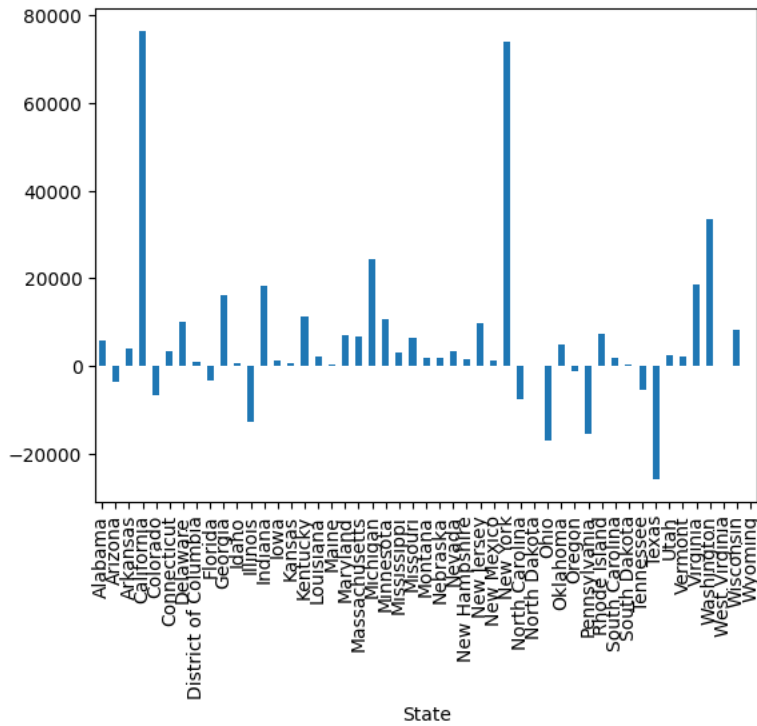
```
df.groupby("State")["Sales"].sum().plot.bar()
```

<Axes: xlabel='State'>



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

<Axes: xlabel='State'>



✓ 1s completed at 3:05 PM

