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

GATHERING THE DATASET

```
df=pd.read_csv("/content/SampleSuperstore.csv")
```

UNDERSTANDING THE DATASET

```
df.head()
```

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Su Categ
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	Tab
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Stora

```
df.describe()
```

	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

```
df.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
```

```
df.tail()
```

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount
9989	Second Class	Consumer	United States	Miami	Florida	33180	South	Furniture	Furnishings	25.248	3	0.2
9990	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Furniture	Furnishings	91.960	2	0.0

1

```
df["Category"].unique()
```

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

```
df["Category"].value_counts()
```

```
Office Supplies    6026
Furniture          2121
Technology         1847
Name: Category, dtype: int64
```

DATA CLEANING

```
df.drop_duplicates(inplace=True)
```

```
df
```

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	0.00
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	2	0.00
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	0.45
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	2	0.20
...	...	...	...	...	...	...	...	...	...	...	...	...
9989	Second Class	Consumer	United States	Miami	Florida	33180	South	Furniture	Furnishings	25.2480	3	0.20
9990	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Furniture	Furnishings	91.9600	2	0.00
9991	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Technology	Phones	258.5760	2	0.20
9992	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Office Supplies	Paper	29.6000	4	0.00
9993	Second Class	Consumer	United States	Westminster	California	92683	West	Office Supplies	Appliances	243.1600	2	0.00

9977 rows × 13 columns

```
df.drop(columns="Postal Code")
```

	Ship Mode	Segment	Country	City	State	Region	Category	Sub-Category	Sales	Quantity	Discount	Profi
0	Second Class	Consumer	United States	Henderson	Kentucky	South	Furniture	Bookcases	261.9600	2	0.00	41.913
1	Second Class	Consumer	United States	Henderson	Kentucky	South	Furniture	Chairs	731.9400	3	0.00	219.582

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

```
df.head()
```

	Ship Mode	Segment	Country	City	State	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
0	Second Class	Consumer	United States	Henderson	Kentucky	South	Furniture	Bookcases	261.9600	2	0.00	41.9136
1	Second Class	Consumer	United States	Henderson	Kentucky	South	Furniture	Chairs	731.9400	3	0.00	219.5820
2	Second Class	Corporate	United States	Los Angeles	California	West	Office Supplies	Labels	14.6200	2	0.00	6.8714

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

```
['Second Class' 'Standard Class' 'First Class' 'Same Day']
['Consumer' 'Corporate' 'Home Office']
['United States']
['Furniture' 'Office Supplies' 'Technology']
['Henderson' 'Los Angeles' 'Fort Lauderdale' 'Concord' 'Seattle'
 'Fort Worth' 'Madison' 'West Jordan' 'San Francisco' 'Fremont'
 'Philadelphia' 'Orem' 'Houston' 'Richardson' 'Naperville' 'Melbourne'
 'Eagan' 'Westland' 'Dover' 'New Albany' 'New York City' 'Troy' 'Chicago'
 'Gilbert' 'Springfield' 'Jackson' 'Memphis' 'Decatur' 'Durham' 'Columbia'
 'Rochester' 'Minneapolis' 'Portland' 'Saint Paul' 'Aurora' 'Charlotte'
 'Orland Park' 'Urbandale' 'Columbus' 'Bristol' 'Wilmington' 'Bloomington'
 'Phoenix' 'Roseville' 'Independence' 'Pasadena' 'Newark' 'Franklin'
 'Scottsdale' 'San Jose' 'Edmond' 'Carlsbad' 'San Antonio' 'Monroe'
 'Fairfield' 'Grand Prairie' 'Redlands' 'Hamilton' 'Westfield' 'Akron'
 'Denver' 'Dallas' 'Whittier' 'Saginaw' 'Medina' 'Dublin' 'Detroit'
 'Tampa' 'Santa Clara' 'Lakeville' 'San Diego' 'Brentwood' 'Chapel Hill'
 'Morristown' 'Cincinnati' 'Inglewood' 'Tamarac' 'Colorado Springs'
 'Belleville' 'Taylor' 'Lakewood' 'Arlington' 'Arvada' 'Hackensack'
 'Saint Petersburg' 'Long Beach' 'Hesperia' 'Murfreesboro' 'Layton'
 'Austin' 'Lowell' 'Manchester' 'Harlingen' 'Tucson' 'Quincy'
 'Pembroke Pines' 'Des Moines' 'Peoria' 'Las Vegas' 'Warwick' 'Miami'
 'Huntington Beach' 'Richmond' 'Louisville' 'Lawrence' 'Canton'
 'New Rochelle' 'Gastonia' 'Jacksonville' 'Auburn' 'Norman' 'Park Ridge'
 'Amarillo' 'Lindenhurst' 'Huntsville' 'Fayetteville' 'Costa Mesa'
 'Parker' 'Atlanta' 'Gladstone' 'Great Falls' 'Lakeland' 'Montgomery'
 'Mesa' 'Green Bay' 'Anaheim' 'Marysville' 'Salem' 'Laredo' 'Grove City'
 'Dearborn' 'Warner Robins' 'Vallejo' 'Mission Viejo' 'Rochester Hills'
 'Plainfield' 'Sierra Vista' 'Vancouver' 'Cleveland' 'Tyler' 'Burlington'
 'Waynesboro' 'Chester' 'Cary' 'Palm Coast' 'Mount Vernon' 'Hialeah'
 'Oceanside' 'Evanston' 'Trenton' 'Cottage Grove' 'Bossier City'
 'Lancaster' 'Asheville' 'Lake Elsinore' 'Omaha' 'Edmonds' 'Santa Ana'
 'Milwaukee' 'Florence' 'Lorain' 'Linden' 'Salinas' 'New Brunswick'
 'Garland' 'Norwich' 'Alexandria' 'Toledo' 'Farmington' 'Riverside'
 'Torrance' 'Round Rock' 'Boca Raton' 'Virginia Beach' 'Murrieta'
 'Olympia' 'Washington' 'Jefferson City' 'Saint Peters' 'Rockford'
 'Brownsville' 'Yonkers' 'Oakland' 'Clinton' 'Encinitas' 'Roswell'
 'Jonesboro' 'Antioch' 'Homestead' 'La Porte' 'Lansing' 'Cuyahoga Falls'
 'Reno' 'Harrisonburg' 'Escondido' 'Royal Oak' 'Rockville' 'Coral Springs'
 'Buffalo' 'Boynton Beach' 'Gulfport' 'Fresno' 'Greenville' 'Macon'
 'Cedar Rapids' 'Providence' 'Pueblo' 'Deltona' 'Murray' 'Middletown'
 'Freeport' 'Pico Rivera' 'Provo' 'Pleasant Grove' 'Smyrna' 'Parma'
 'Mobile' 'New Bedford' 'Irving' 'Vineland' 'Glendale' 'Niagara Falls'
 'Thomasville' 'Westminster' 'Coppell' 'Pomona' 'North Las Vegas'
 'Allentown' 'Tempe' 'Laguna Niguel' 'Bridgeton' 'Everett' 'Watertown'
 'Appleton' 'Bellevue' 'Allen' 'El Paso' 'Grapevine' 'Carrollton' 'Kent'
 'Lafayette' 'Tigard' 'Skokie' 'Plano' 'Suffolk' 'Indianapolis' 'Bayonne'
 'Greensboro' 'Baltimore' 'Kenosha' 'Olathe' 'Tulsa' 'Redmond' 'Raleigh'
 'Muskogee' 'Meriden' 'Bowling Green' 'South Bend' 'Spokane' 'Keller'
```

```
'Port Orange' 'Medford' 'Charlottesville' 'Missoula' 'Apopka' 'Reading'
'Broomfield' 'Paterson' 'Oklahoma City' 'Chesapeake' 'Lubbock'
'Johnson City' 'San Bernardino' 'Leominster' 'Bozeman' 'Perth Amboy'
'Ontario' 'Rancho Cucamonga' 'Moorhead' 'Mesquite' 'Stockton'
'Ormond Beach' 'Sunnyvale' 'York' 'College Station' 'Saint Louis'
'Manteca' 'San Angelo' 'Salt Lake City' 'Knoxville' 'Little Rock'
'Lincoln Park' 'Marion' 'Littleton' 'Bangor' 'Southaven' 'New Castle'
'Midland' 'Sioux Falls' 'Fort Collins' 'Clarksville' 'Sacramento'
'Thousand Oaks' 'Malden' 'Holyoke' 'Albuquerque' 'Sparks' 'Coachella'
```

```
df.describe()
```

	Sales	Quantity	Discount	Profit
count	9977.000000	9977.000000	9977.000000	9977.00000
mean	230.148902	3.790719	0.156278	28.69013
std	623.721409	2.226657	0.206455	234.45784
min	0.444000	1.000000	0.000000	-6599.97800
25%	17.300000	2.000000	0.000000	1.72620
50%	54.816000	3.000000	0.200000	8.67100
75%	209.970000	5.000000	0.200000	29.37200
max	22638.480000	14.000000	0.800000	8399.97600

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9977 entries, 0 to 9993
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Ship Mode     9977 non-null   object
 1   Segment       9977 non-null   object
 2   Country       9977 non-null   object
 3   City          9977 non-null   object
 4   State         9977 non-null   object
 5   Region        9977 non-null   object
 6   Category      9977 non-null   object
 7   Sub-Category  9977 non-null   object
 8   Sales         9977 non-null   float64
 9   Quantity      9977 non-null   int64
 10  Discount      9977 non-null   float64
 11  Profit        9977 non-null   float64
dtypes: float64(3), int64(1), object(8)
memory usage: 1013.3+ KB
```

HANDLING MISSING VALUES

```
df.dropna()
```

	Ship Mode	Segment	Country	City	State	Region	Category	Sub-Category	Sales	Quantity	Discount	Profi
0	Second Class	Consumer	United States	Henderson	Kentucky	South	Furniture	Bookcases	261.9600	2	0.00	41.913
1	Second Class	Consumer	United States	Henderson	Kentucky	South	Furniture	Chairs	731.9400	3	0.00	219.582
2	Second Class	Corporate	United States	Los Angeles	California	West	Office Supplies	Labels	14.6200	2	0.00	6.871
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	South	Furniture	Tables	957.5775	5	0.45	-383.031
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	South	Office Supplies	Storage	22.3680	2	0.20	2.516
...	...	...	...	...	...	...	...	...	...	...	...	...
9989	Second Class	Consumer	United States	Miami	Florida	South	Furniture	Furnishings	25.2480	3	0.20	4.102
9990	Standard Class	Consumer	United States	Costa Mesa	California	West	Furniture	Furnishings	91.9600	2	0.00	15.633

```
df.drop_duplicates(inplace=True)
```

```
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
```

STATISTICAL ANALYSIS

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

```
Region
Central    500724.2708
East       677843.9080
South      391721.9050
West       724309.2935
Name: Sales, dtype: float64
```

```
df.groupby("Region")["Sales"].mean()
```

```
Region
Central    216.295581
East       239.351662
South      241.803645
West       227.985299
Name: Sales, dtype: float64
```

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

```
Region
Central     39662.0474
East        91471.0909
South       46749.4303
West       108214.9893
Name: Profit, dtype: float64
```

```
df.groupby("Region")["Profit"].mean()
```

```
Region
Central    17.132634
East       32.299114
South      28.857673
West       34.062005
Name: Profit, dtype: float64
```

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

```
Segment
Consumer       1.159418e+06
Corporate      7.059702e+05
Home Office    4.292117e+05
Name: Sales, dtype: float64
```

```
df.groupby("Segment")["Sales"].mean()
```

```
Segment
Consumer       224.693317
Corporate      234.697538
Home Office    241.673237
Name: Sales, dtype: float64
```

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

```
Segment
Consumer       133939.5218
Corporate       91911.1606
Home Office     60246.8755
Name: Profit, dtype: float64
```

```
df.groupby("Segment")["Profit"].mean()
```

```
Segment
Consumer       25.957272
Corporate      30.555572
Home Office    33.922790
Name: Profit, dtype: float64
```

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

```
Category
Furniture          740568.6663
Office Supplies    718256.0700
Technology         835774.6410
Name: Sales, dtype: float64
```

```
df.groupby("Category")["Sales"].mean()
```

```
Category
Furniture          350.316304
Office Supplies    119.989320
Technology         453.240044
Name: Sales, dtype: float64
```

```
df.groupby("Sub-Category")["Sales"].sum()
```

```
Sub-Category
Accessories    167303.3020
Appliances     107532.1610
Art             27065.7320
Binders        203328.8590
Bookcases      114166.9293
Chairs         327777.7610
Copiers        149528.0300
Envelopes       16476.4020
Fasteners        3024.2800
Furnishings     91658.4440
Labels          12385.7640
Machines       189238.6310
Paper           77960.8940
Phones         329704.6780
Storage        223808.4400
Supplies        46673.5380
Tables         206965.5320
Name: Sales, dtype: float64
```

```
df.groupby("Sub-Category")["Profit"].sum()
```

```
Sub-Category
Accessories    41920.9052
Appliances     18138.0054
Art             6511.8388
Binders        30205.2603
Bookcases      -3398.7542
Chairs         26567.1278
Copiers        55617.8249
Envelopes       6964.1767
Fasteners        949.5182
Furnishings    13042.0534
Labels          5499.1558
Machines        3384.7569
Paper          33830.0381
Phones         44493.0524
Storage        21287.1788
Supplies       -1189.0995
Tables        -17725.4811
Name: Profit, dtype: float64
```

```
df.groupby("Sub-Category")["Sales"].mean()
```

```
Sub-Category
Accessories     216.433767
Appliances      230.755710
Art              34.130810
Binders         133.945230
Bookcases       505.163404
Chairs          532.971969
Copiers        2198.941618
Envelopes        64.867724
Fasteners        13.936774
Furnishings      96.078034
Labels           34.500735
Machines       1645.553313
Paper            58.006618
Phones          371.289052
Storage         264.862059
Supplies        245.650200
```

```
Tables          648.794771
Name: Sales, dtype: float64
```

```
df.groupby("Sub-Category")["Profit"].mean()
```

```
Sub-Category
Accessories     54.231443
Appliances      38.922758
Art              8.211650
Binders         19.898063
Bookcases      -15.038735
Chairs          43.198582
Copiers        817.909190
Envelopes       27.418019
Fasteners        4.375660
Furnishings     13.670916
Labels          15.317983
Machines        29.432669
Paper           25.171159
Phones          50.104789
Storage         25.191928
Supplies        -6.258418
Tables         -55.565771
Name: Profit, dtype: float64
```

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

```
State
Alabama                  19510.6400
Arizona                  35282.0010
Arkansas                 11678.1300
California              456629.9285
Colorado                 32108.1180
Connecticut              13384.3570
Delaware                 27451.0690
District of Columbia      2865.0200
Florida                  89473.7080
Georgia                  49095.8400
Idaho                     4382.4860
Illinois                 80127.3690
Indiana                  53555.3600
Iowa                      4579.7600
Kansas                    2914.3100
Kentucky                 36591.7500
Louisiana                 9217.0300
Maine                     1270.5300
Maryland                 23705.5230
Massachusetts            28634.4340
Michigan                 75879.6440
Minnesota                29863.1500
Mississippi              10771.3400
Missouri                 22205.1500
Montana                   5589.3520
Nebraska                  7464.9300
Nevada                   16729.1020
New Hampshire             7292.5240
New Jersey               35764.3120
New Mexico                4783.5220
New York                310349.2150
North Carolina           55603.1640
North Dakota               919.9100
Ohio                     77976.7640
Oklahoma                 19683.3900
Oregon                   17420.7820
Pennsylvania            116383.0100
Rhode Island             22627.9560
South Carolina            8481.7100
South Dakota              1315.5600
Tennessee                30661.8730
Texas                   170101.1278
Utah                     11220.0560
Vermont                   8929.3700
Virginia                 70636.7200
Washington              138560.8100
West Virginia             1209.8240
Wisconsin                32114.6100
Wyoming                   1603.1360
Name: Sales, dtype: float64
```

```
df.groupby("State")["Sales"].mean()
```

```
State
Alabama                  319.846557
Arizona                  157.508933
Arkansas                 194.635500
California               230.621176
Colorado                 176.418231
```

```
Connecticut             163.223866
Delaware                285.948635
District of Columbia    286.502000
Florida                 233.612815
Georgia                 266.825217
Idaho                   208.689810
Illinois                163.525243
Indiana                 359.431946
Iowa                    152.658667
Kansas                  121.429583
Kentucky                263.250000
Louisiana               219.453095
Maine                   158.816250
Maryland                225.766886
Massachusetts           212.106919
Michigan                298.738756
Minnesota               335.541011
Mississippi             203.232830
Missouri                336.441667
Montana                 372.623467
Nebraska                196.445526
Nevada                  428.951333
New Hampshire           270.093481
New Jersey              275.110092
New Mexico              129.284378
New York                277.345143
North Carolina          223.305880
North Dakota            131.415714
Ohio                    166.617017
Oklahoma                298.233182
Oregon                  141.632374
Pennsylvania            200.314991
Rhode Island            404.070643
South Carolina          201.945476
South Dakota            109.630000
Tennessee               167.551219
Texas                   173.572579
Utah                    211.699170
Vermont                 811.760909
Virginia                315.342500
Washington              276.017550
West Virginia           302.456000
Wisconsin               291.951000
Wyoming                1603.136000
Name: Sales, dtype: float64
```

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

```
State
Alabama                 5786.8253
Arizona                -3427.9246
Arkansas                4008.6871
California             76215.9705
Colorado               -6527.8579
Connecticut             3511.4918
Delaware                9977.3748
District of Columbia    1059.5893
Florida                -3399.3017
Georgia                16250.0433
Idaho                    826.7231
Illinois             -12593.2976
Indiana                18382.9363
Iowa                    1183.8119
Kansas                   836.4435
Kentucky               11199.6966
Louisiana               2196.1023
Maine                    454.4862
Maryland                7031.1788
Massachusetts           6785.5016
Michigan               24428.0903
Minnesota              10823.1874
Mississippi             3172.9762
Missouri                6436.2105
Montana                 1833.3285
Nebraska                2037.0942
Nevada                  3316.7659
New Hampshire           1706.5028
New Jersey              9772.9138
New Mexico              1157.1161
New York               74006.1552
North Carolina         -7490.9122
North Dakota             230.1497
Ohio                 -16959.3178
Oklahoma                4853.9560
Oregon                 -1194.0993
Pennsylvania         -15591.3148
Rhode Island            7285.6293
South Carolina          1769.0566
South Dakota             394.8283
```

```
Tennessee              -5341.6936
Texas                 -25753.1635
Utah                   2546.5335
Vermont                2244.9783
Virginia              18597.9504
Washington            33368.2375
West Virginia           185.9216
Wisconsin              8401.8004
Wyoming                 100.1960
Name: Profit, dtype: float64
```

```
df.groupby("State")["Profit"].mean()
```

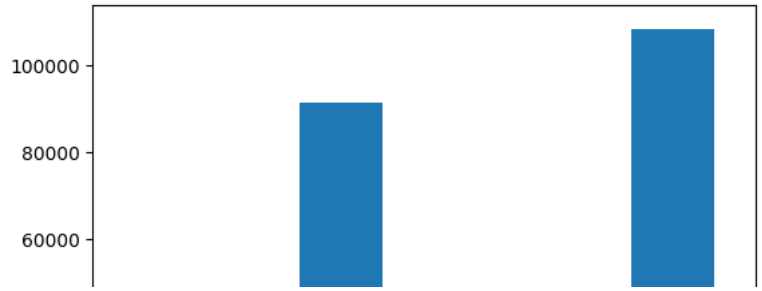
```
State
Alabama                  94.865989
Arizona                 -15.303235
Arkansas                 66.811452
California               38.492914
Colorado                -35.867351
Connecticut              42.823071
Delaware                103.930988
District of Columbia    105.958930
Florida                  -8.875461
Georgia                  88.315453
Idaho                    39.367767
Illinois                -25.700607
Indiana                 123.375411
Iowa                     39.460397
Kansas                   34.851813
Kentucky                 80.573357
Louisiana                52.288150
Maine                    56.810775
Maryland                 66.963608
Massachusetts            50.262975
Michigan                 96.173584
Minnesota               121.608847
Mississippi              59.867475
Missouri                 97.518341
Montana                 122.221900
Nebraska                 53.607742
Nevada                   85.045279
New Hampshire            63.203807
New Jersey               75.176260
New Mexico               31.273408
New York                 66.135974
North Carolina          -30.083985
North Dakota             32.878529
Ohio                    -36.237859
Oklahoma                 73.544788
Oregon                   -9.708124
Pennsylvania            -26.835309
Rhode Island            130.100523
South Carolina           42.120395
South Dakota             32.902358
Tennessee               -29.189583
Texas                   -26.278738
Utah                     48.047802
Vermont                 204.088936
Virginia                 83.026564
Washington               66.470593
West Virginia            46.480400
Wisconsin                76.380004
Wyoming                 100.196000
Name: Profit, dtype: float64
```

VISUALIZATIONS

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

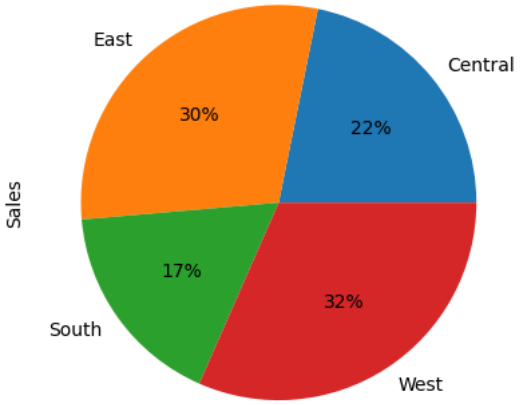
```
df.groupby("Region")["Profit"].sum().plot.bar()
```

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



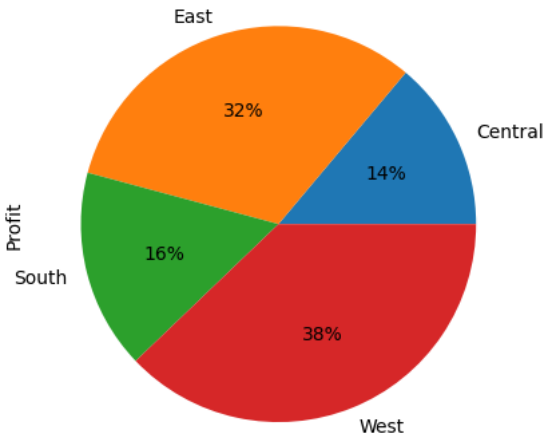

```
df.groupby("Region")["Sales"].sum().plot.pie(autopct="%1.0f%%")
```

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




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

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

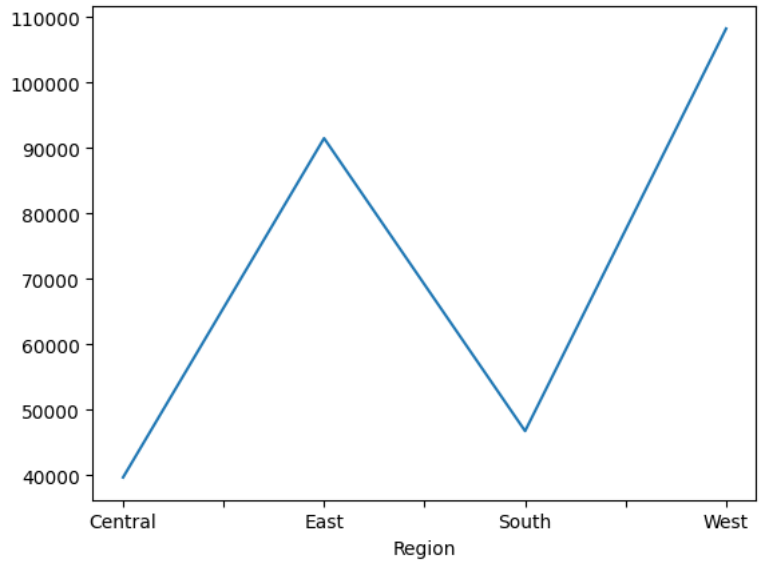



```
df.groupby("Region")["Sales"].sum().plot.line()
```

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

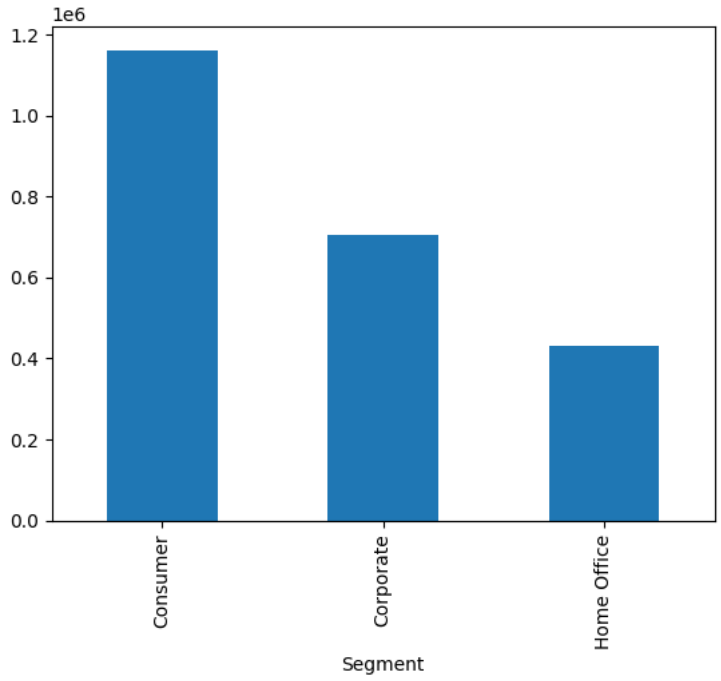

```
df.groupby("Region")["Profit"].sum().plot.line()
```

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



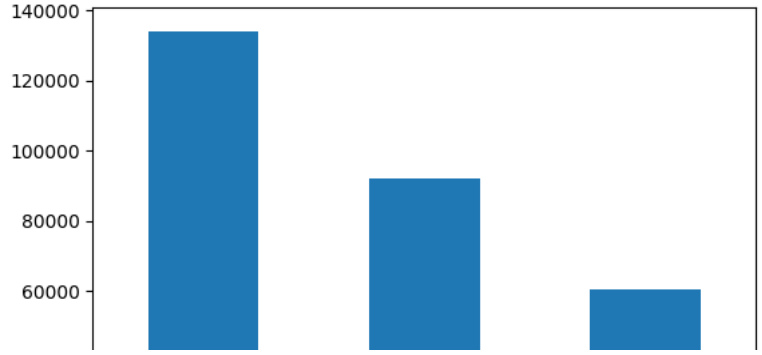

```
df.groupby("Segment")["Sales"].sum().plot.bar()
```

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



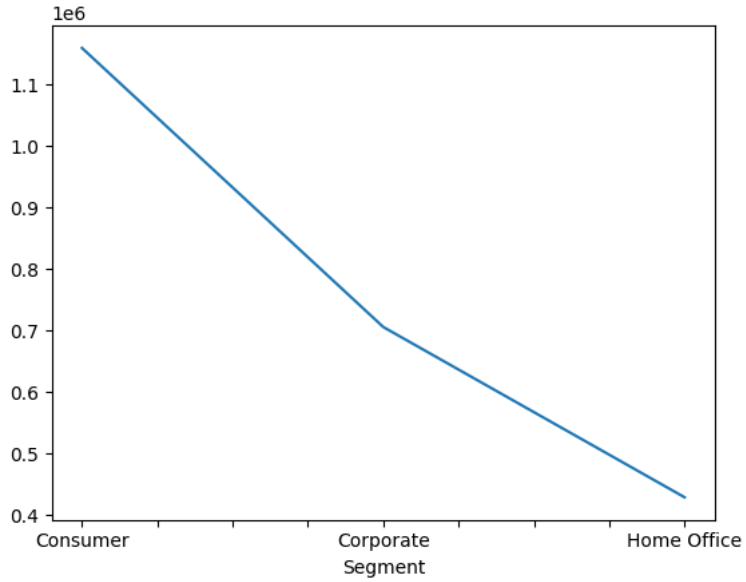

```
df.groupby("Segment")["Profit"].sum().plot.bar()
```

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



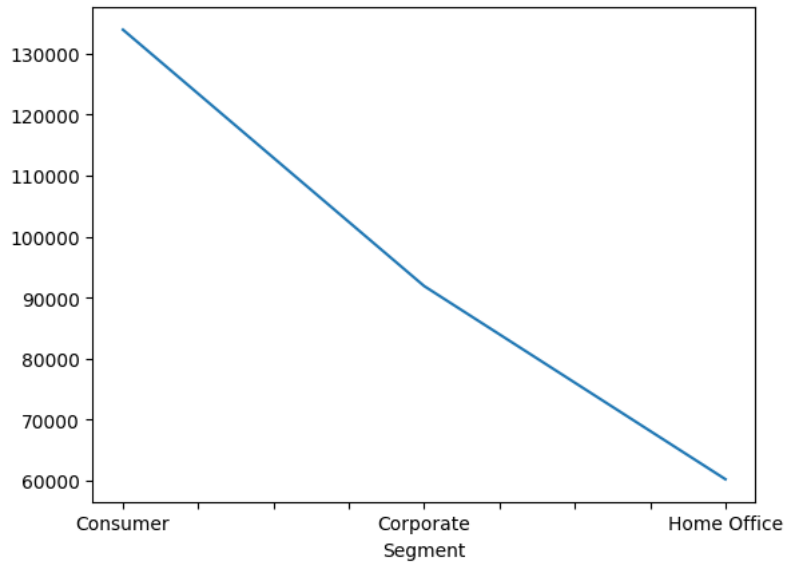

```
df.groupby("Segment")["Sales"].sum().plot.line()
```

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



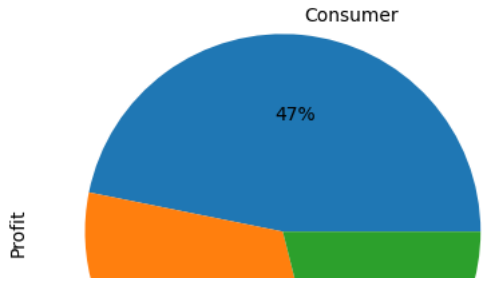

```
df.groupby("Segment")["Profit"].sum().plot.line()
```

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



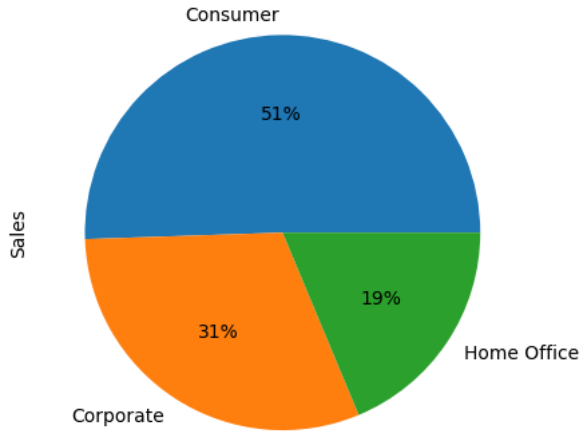

```
df.groupby("Segment")["Profit"].sum().plot.pie(autopct="%1.0f%%")
```

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



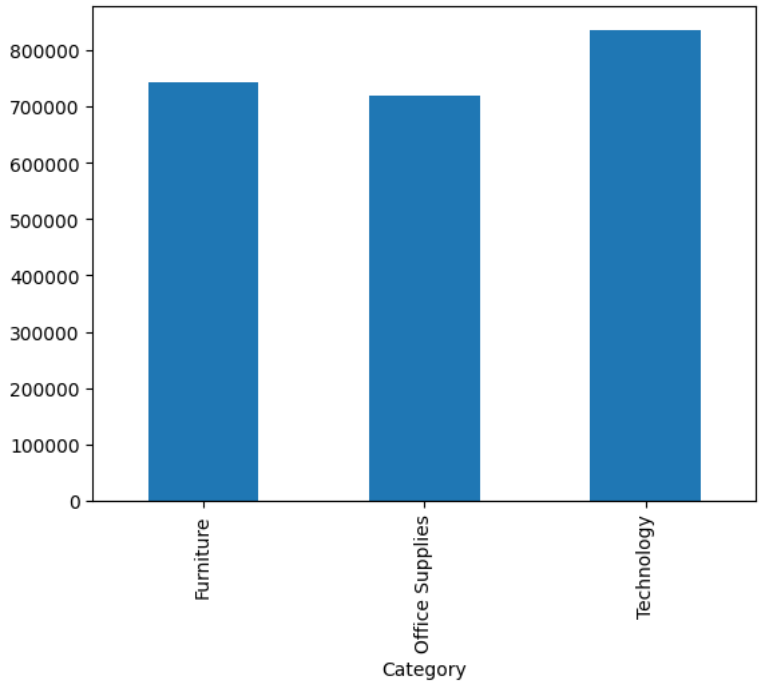

```
df.groupby("Segment")["Sales"].sum().plot.pie(autopct="%1.0f%%")
```

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



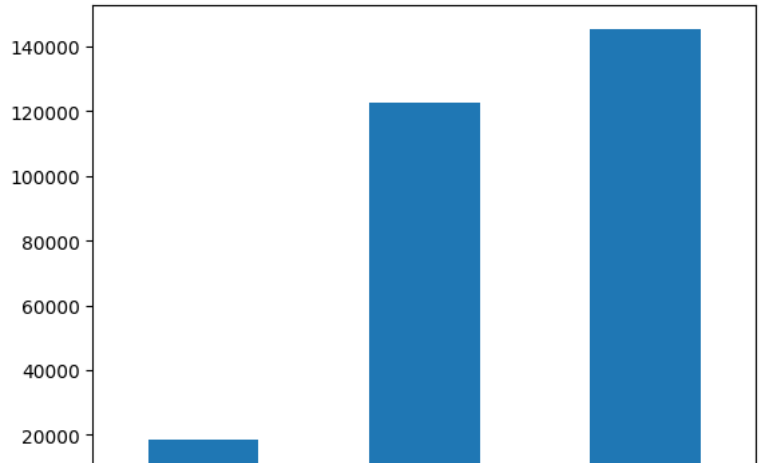

```
df.groupby("Category")["Sales"].sum().plot.bar()
```

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



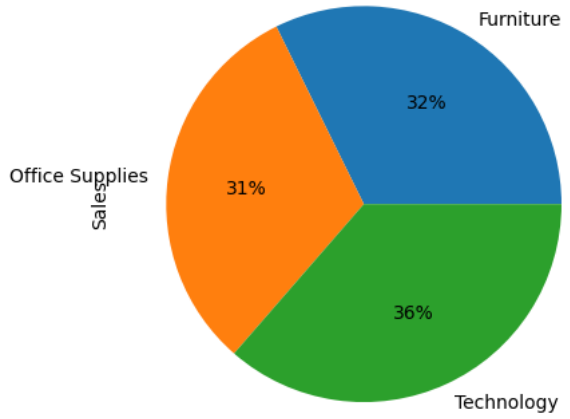

```
df.groupby("Category")["Profit"].sum().plot.bar()
```

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



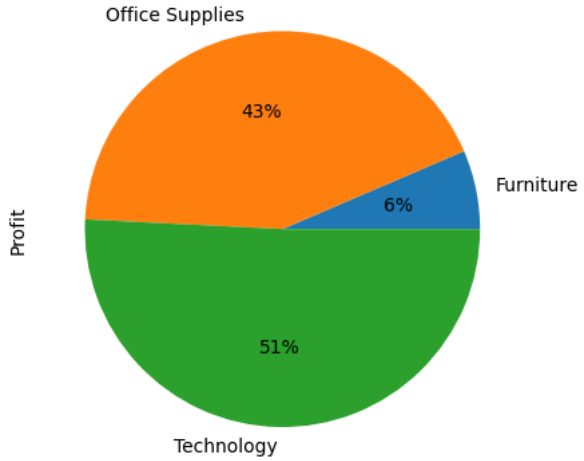

```
df.groupby("Category")["Sales"].sum().plot.pie(autopct="%1.0f%%")
```

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



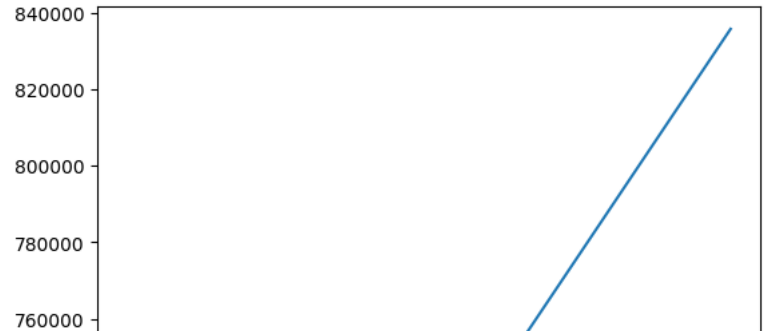

```
df.groupby("Category")["Profit"].sum().plot.pie(autopct="%1.0f%%")
```

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



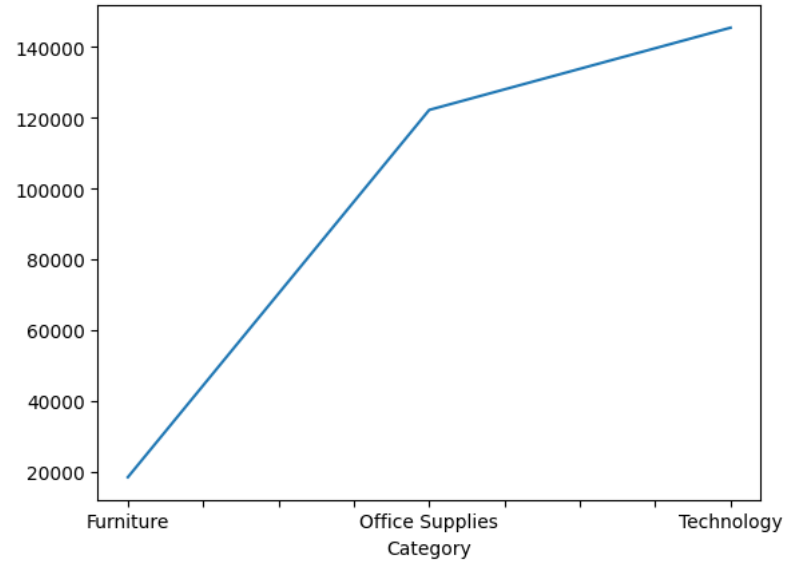

```
df.groupby("Category")["Sales"].sum().plot.line()
```

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



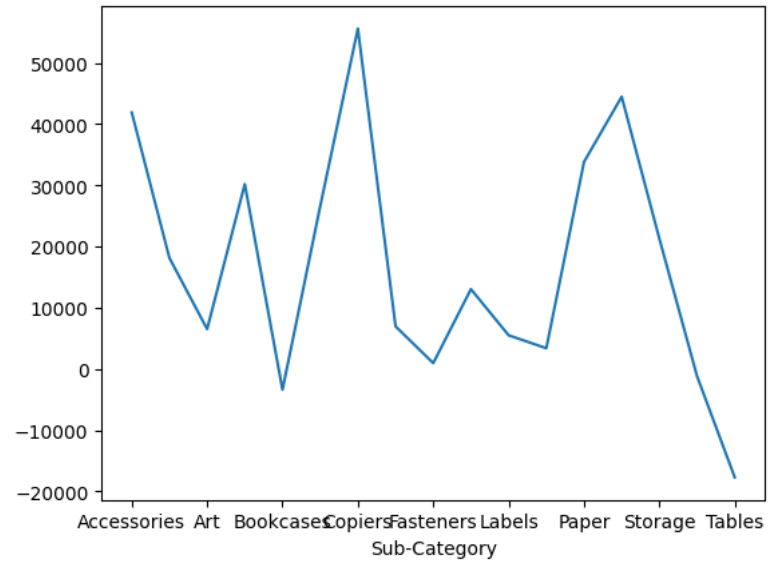

```
df.groupby("Category")["Profit"].sum().plot.line()
```

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



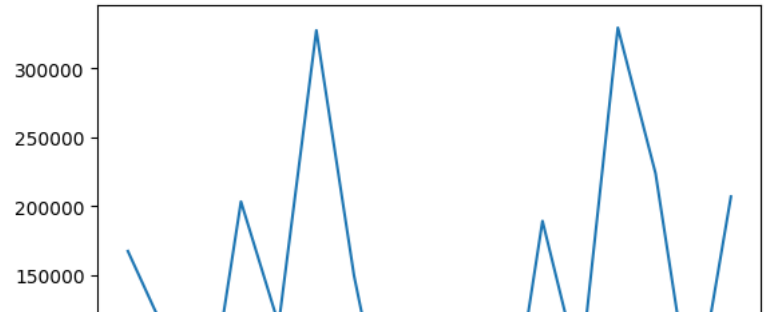

```
df.groupby("Sub-Category")["Profit"].sum().plot.line()
```

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



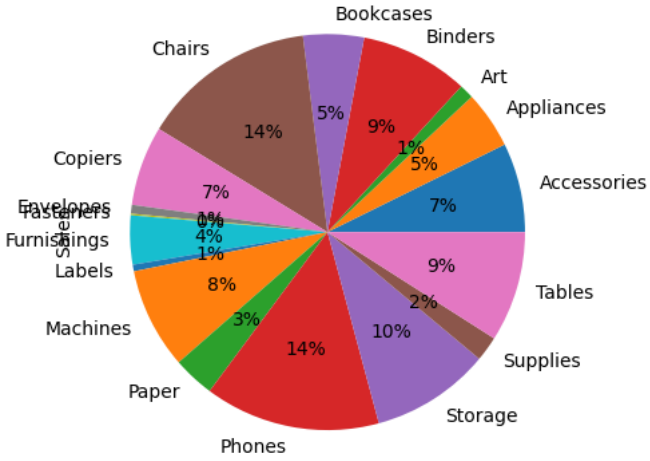

```
df.groupby("Sub-Category")["Sales"].sum().plot.line()
```

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



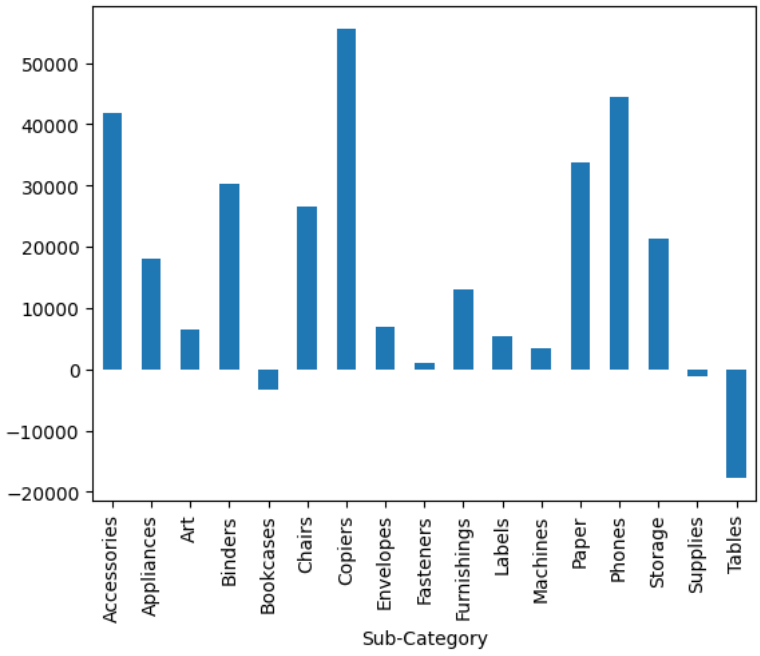

```
df.groupby("Sub-Category")["Sales"].sum().plot.pie(autopct="%1.0f%%")
```

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



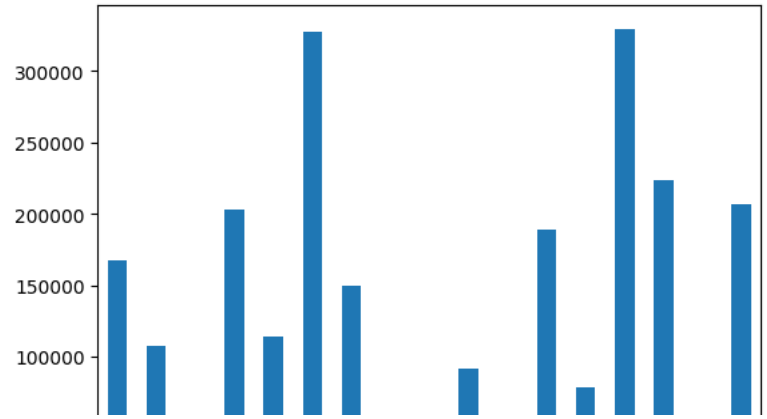

```
df.groupby("Sub-Category")["Profit"].sum().plot.bar()
```

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



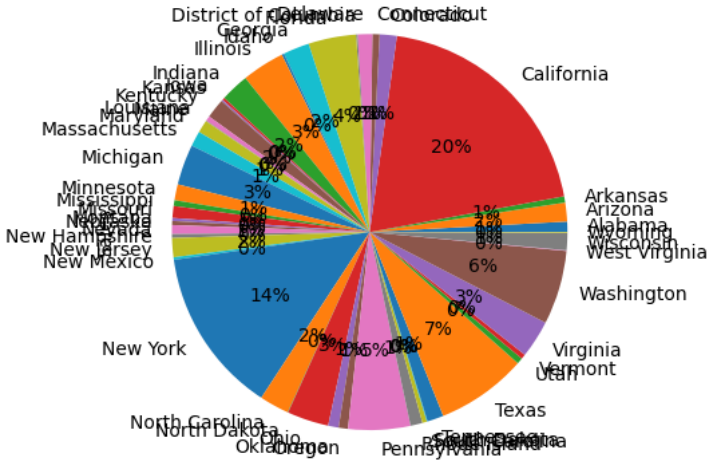

```
df.groupby("Sub-Category")["Sales"].sum().plot.bar()
```

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



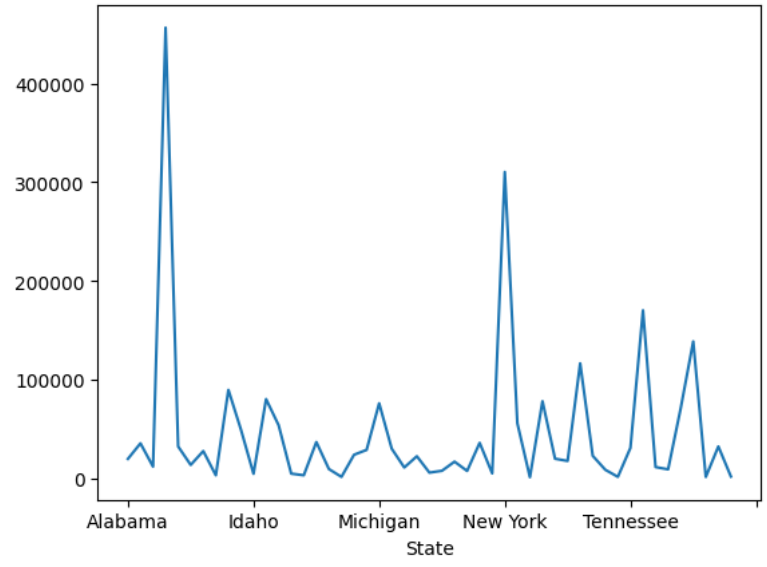

```
df.groupby("State")["Sales"].sum().plot.pie(autopct="%1.0f%%")
```

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



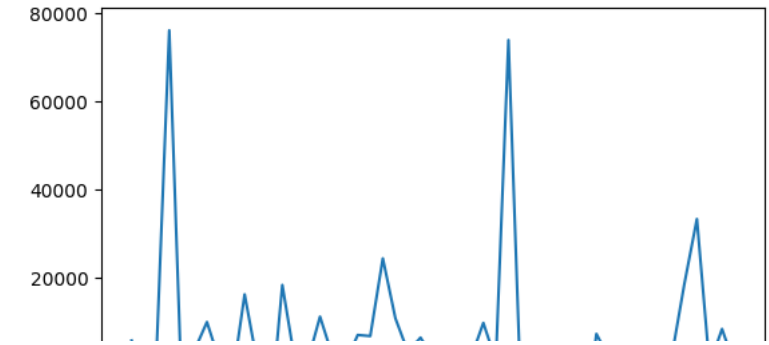

```
df.groupby("State")["Sales"].sum().plot.line()
```

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



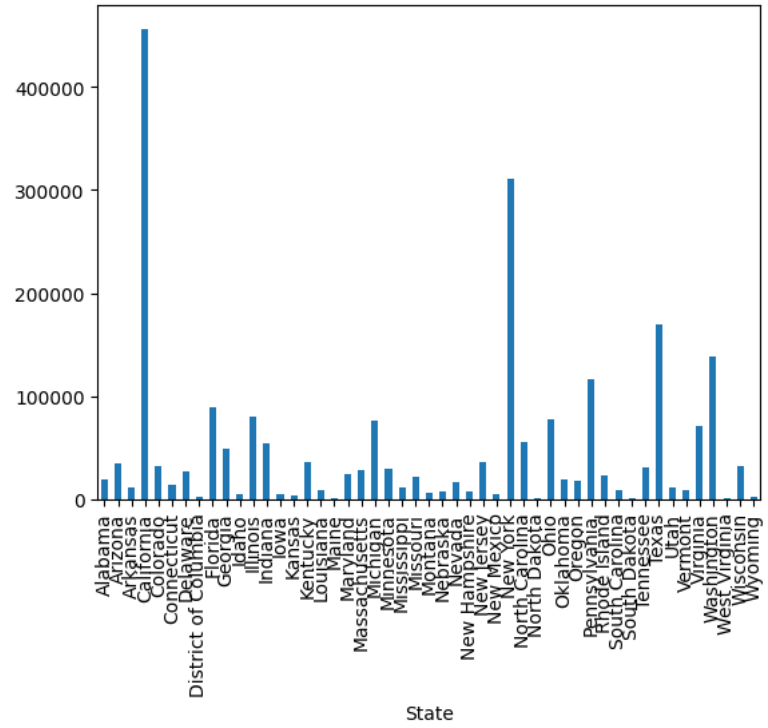

```
df.groupby("State")["Profit"].sum().plot.line()
```

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




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

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




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

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