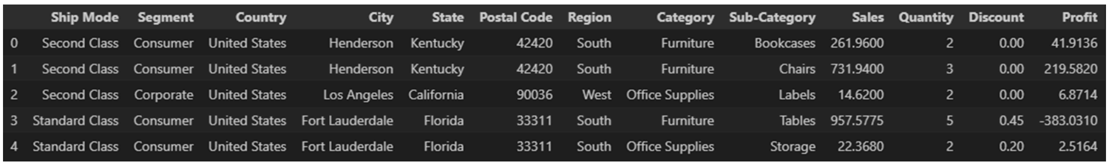
SUPER-STORE ANALYSIS

DATASET DESCRPTION

The Superstore Sales dataset contains historical sales records from supermarket branches. The dataset includes information about sales such as shipping mode, segment, quantity , city, region, discount , profit etc.

The dataset has a total of 9994 rows and 13 columns i.e. Shape of data is (9994 x 13) The dataset contains both numeric and categorical data, requiring various analysis techniques. Overall, it gives a comprehensive view of sales data, allowing analysis by product and location.

Here is a glimpse of data



	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00	41.9136
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	0.00	219.5820
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	2	0.00	6.8714
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	0.45	-383.0310
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	2	0.20	2.5164

COLUMN DESCRIPTION

The dataset contains details of the sales made by the super-store. The data file contain the following 13 columns:

- Ship mode : The mode of shipping (First class, Second class, Standard class)
- Segment : The catgory of purchaser (Consumer, Corporate , Home-office)
- Country : All the sales are made in United States
- City : Name of city (eg Los angeles, New York etc.)
- State : Name of state (eg Texas, California etc.)
- Postal code : Information about Postal address ()
- Region : info about region (east, west , south , north)
- Category : info about category of product (Technology , Office supplies, Furniture)
- Sub-category : info about sub-category of product (Tables , Envelopes, Paper, Phone etc.)
- Sales : The sales made by each purchase
- Quantity : The number of product purchased (in quantity)
- Discount : The discount given on purchase
- Profit : the profit made by the sales

IMPORTING LIBRARIES

In [36]:

```
# Importing the required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

IMPORTING DATA

In [37]:

```python
# Reading csv file into pandas DataFrame
data = pd.read_csv('super_store0.csv')
data.head()
```

Out[37]:

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

In [38]:

```python
# Shape of Data
data.shape
```

Out[38]:

```
(9994, 13)
```

DATA WRANGLING

In [39]:

```python
# Info of Data
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 [40]:

```python
# Description of data
data.describe()
```

Out[40]:

	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 [41]:

```
# Checking columns for null values
data.isnull().sum()
```

Out[41]:

```
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 [42]:

```
# Total duplicated data
data.duplicated().sum()
```

Out[42]:

```
np.int64(17)
```

In [43]:

```
# Dropping duplicates
data.drop_duplicates(inplace=True)
```

In [44]:

```
# Retrieving Column names
data.columns
```

Out[44]:

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

In [45]:

```
# Unique data-values in columns
for i in data.columns:
    print(i, data[i].nunique())
```

```
Ship Mode 4
Segment 3
Country 1
City 531
State 49
Postal Code 631
Region 4
Category 3
Sub-Category 17
Sales 5825
Quantity 14
Discount 12
Profit 7287
```

INSIGHTS

1. There are 13 column and 9994 rows .
2. There are no null values.
3. There are 5 numerical columns.
4. The column `country` contains only one type of value (hence, can be dropped)

DATA PREPARTION/CLEANING FOR ANALYSIS

In [46]:

```
# Dropping Country column as it has only one unique value
data.drop('Country', axis=1, inplace=True)
```

In [47]:

```
# Rounding off floating values to 2 decimal places
data['Discount']=round(data['Discount'],2)
data['Sales']=round(data['Sales'],2)
data['Profit']=round(data['Profit'],2)
```

In [48]:

```
data.head(5)
```

Out[48]:

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

DESCRIPTIVE STATISTICS AND POTENTIAL OUTLIERS

In [49]:

```
data.describe(include='all')
```

Out[49]:

	Ship Mode	Segment	City	State	Postal Code	Region	Category	Sub-Category
count	9977	9977	9977	9977	9977.000000	9977	9977	9977
unique	4	3	531	49	NaN	4	3	17
top	Standard Class	Consumer	New York City	California	NaN	West	Office Supplies	Binders
freq	5955	5183	914	1996	NaN	3193	6012	1522
mean	NaN	NaN	NaN	NaN	55154.964117	NaN	NaN	NaN
std	NaN	NaN	NaN	NaN	32058.266816	NaN	NaN	NaN
min	NaN	NaN	NaN	NaN	1040.000000	NaN	NaN	NaN
25%	NaN	NaN	NaN	NaN	23223.000000	NaN	NaN	NaN
50%	NaN	NaN	NaN	NaN	55901.000000	NaN	NaN	NaN
75%	NaN	NaN	NaN	NaN	90008.000000	NaN	NaN	NaN
max	NaN	NaN	NaN	NaN	99301.000000	NaN	NaN	NaN

In [50]:

```
for i in ['Ship Mode','Segment','State','Region','Category','Sub-Category']:
    print(data[i].value_counts())
    print('\n')
```

```
Ship Mode
Standard Class    5955
Second Class      1943
First Class       1537
Same Day           542
Name: count, dtype: int64


Segment
Consumer       5183
Corporate      3015
Home Office    1779
Name: count, dtype: int64


State
California              1996
New York                1127
Texas                    983
Pennsylvania             586
Washington               502
Illinois                 491
Ohio                     468
Florida                  383
Michigan                 254
North Carolina           249
Arizona                  224
Virginia                 224
Georgia                  184
Tennessee                183
Colorado                 182
Indiana                  149
Kentucky                 139
Massachusetts            135
New Jersey               130
Oregon                   123
Wisconsin                110
Maryland                 105
Delaware                  96
Minnesota                 89
Connecticut               82
Missouri                  66
Oklahoma                  66
Alabama                   61
Arkansas                  60
Rhode Island              56
Utah                      53
Mississippi               53
South Carolina            42
Louisiana                 42
Nevada                    39
Nebraska                  38
New Mexico                37
Iowa                      30
New Hampshire             27
Kansas                    24
Idaho                     21
Montana                   15
South Dakota              12
Vermont                   11
```

```
District of Columbia      10
Maine                      8
North Dakota               7
West Virginia              4
Wyoming                    1
Name: count, dtype: int64


Region
West       3193
East       2845
Central    2319
South      1620
Name: count, dtype: int64


Category
Office Supplies    6012
Furniture          2118
Technology         1847
Name: count, dtype: int64


Sub-Category
Binders        1522
Paper          1359
Furnishings     956
Phones          889
Storage         846
Art             795
Accessories     775
Chairs          615
Appliances      466
Labels          363
Tables          319
Envelopes       254
Bookcases       228
Fasteners       217
Supplies        190
Machines        115
Copiers          68
Name: count, dtype: int64
```

DATA TRENDS AND CORRELATIONS

In [70]:

```
# NET SALES AND PROFIT
x= round(data['Profit'].sum(),2)
y= round(data['Sales'].sum(),2)
z= data['Sales'].count()
print('Net Profit :',x,'\n','Total sales: ',y)
print('profit % : ', round((x/y)*100,2))
print('profit per sales', round(x/z,2))
```

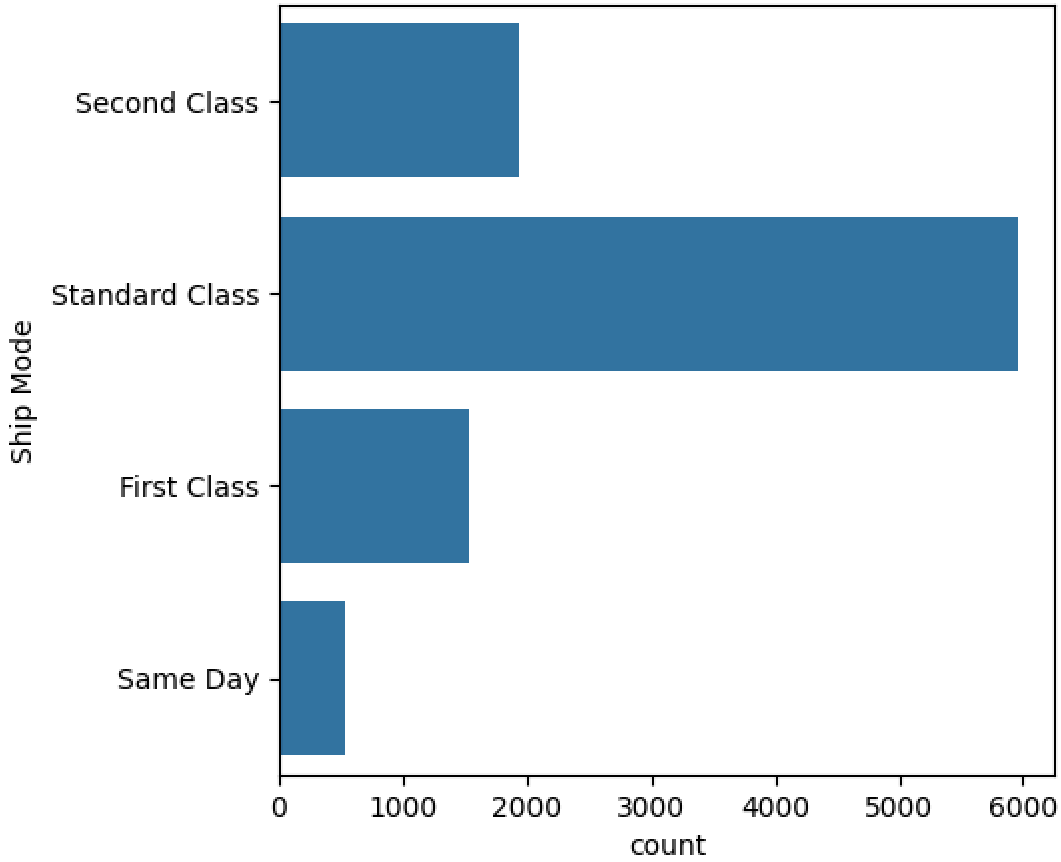
```
Net Profit : 286240.95
 Total sales:  2296195.39
profit % :  12.47
profit per sales 28.69
```

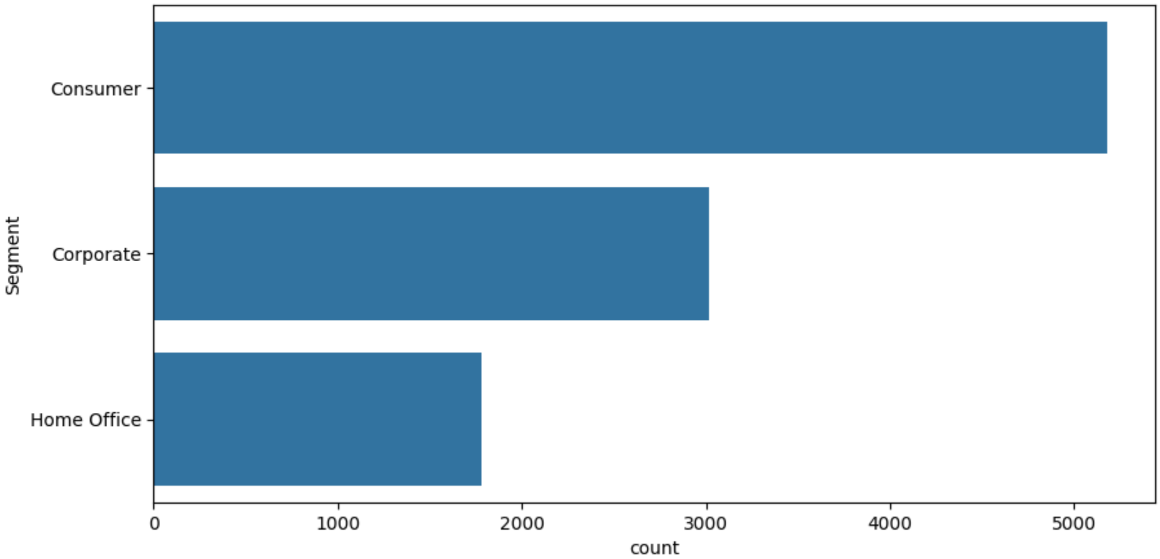
In [51]:

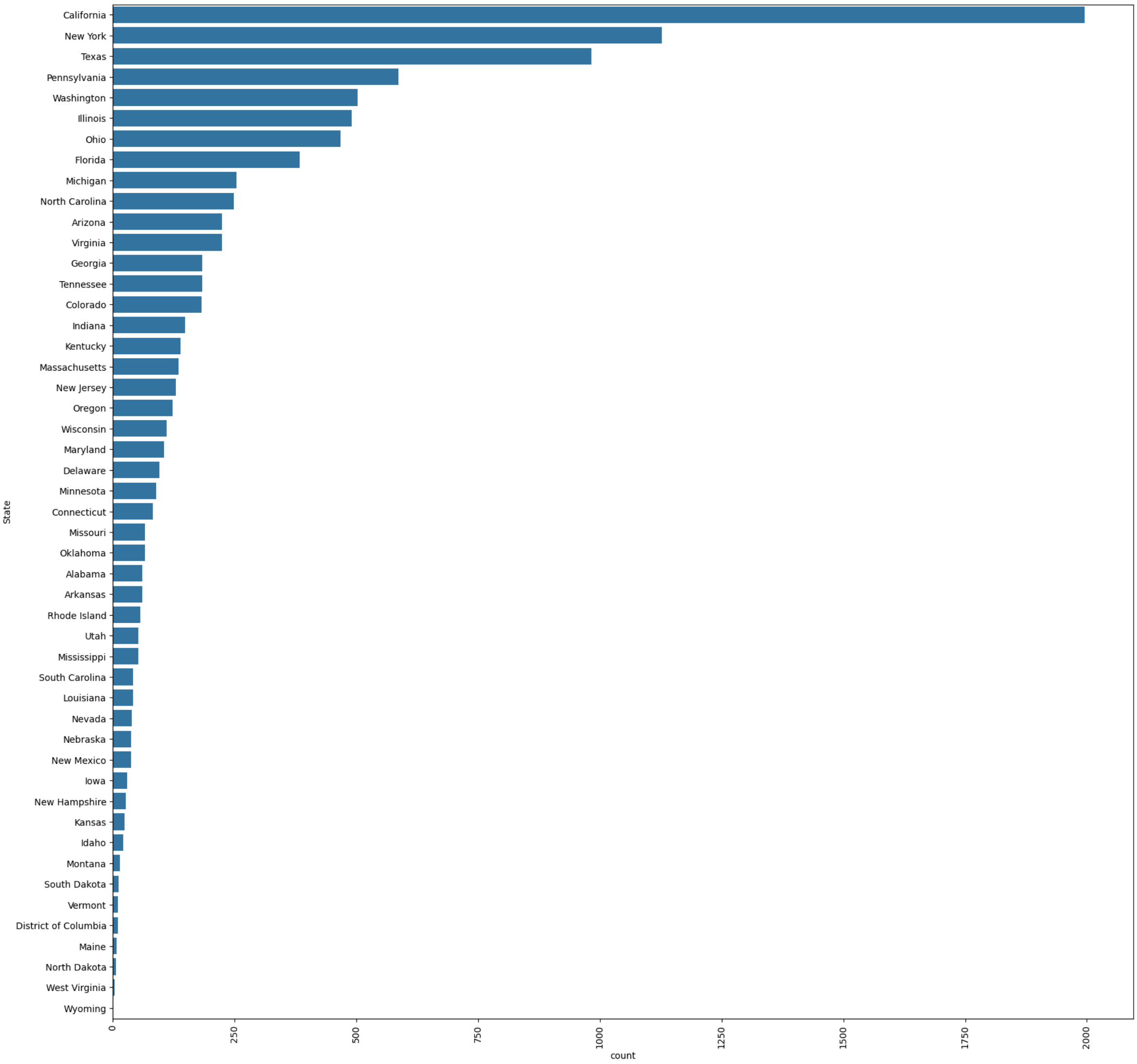
```
# Ship Mode

plt.figure(figsize=(5,5))
sns.countplot(data['Ship Mode'])

# Segment

plt.figure(figsize=(10,5))
sns.countplot(data['Segment'])

# State

plt.figure(figsize=(20,20))
sns.countplot(data['State'], order=data['State'].value_counts().index)
plt.xticks(rotation=90)


# Region

plt.figure(figsize=(10,5))
sns.countplot(data['Region'])


# Category

plt.figure(figsize=(10,5))
sns.countplot(data['Category'])

#Sub-Category
plt.figure(figsize=(20,10))
sns.countplot(data['Sub-Category'], order=data['Sub-Category'].value_counts().in
```

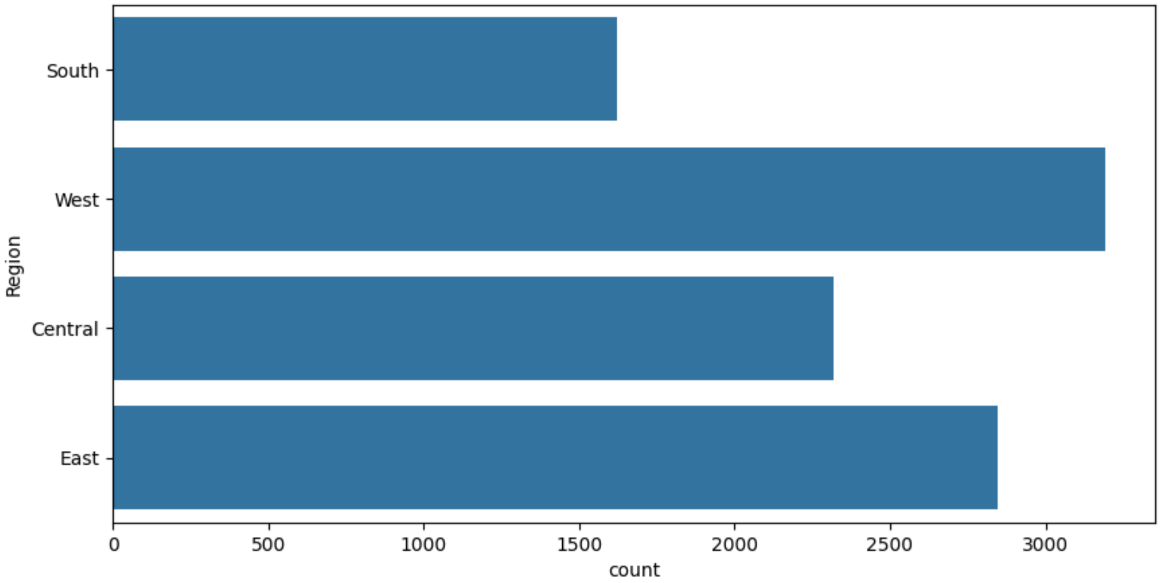
Out[51]:

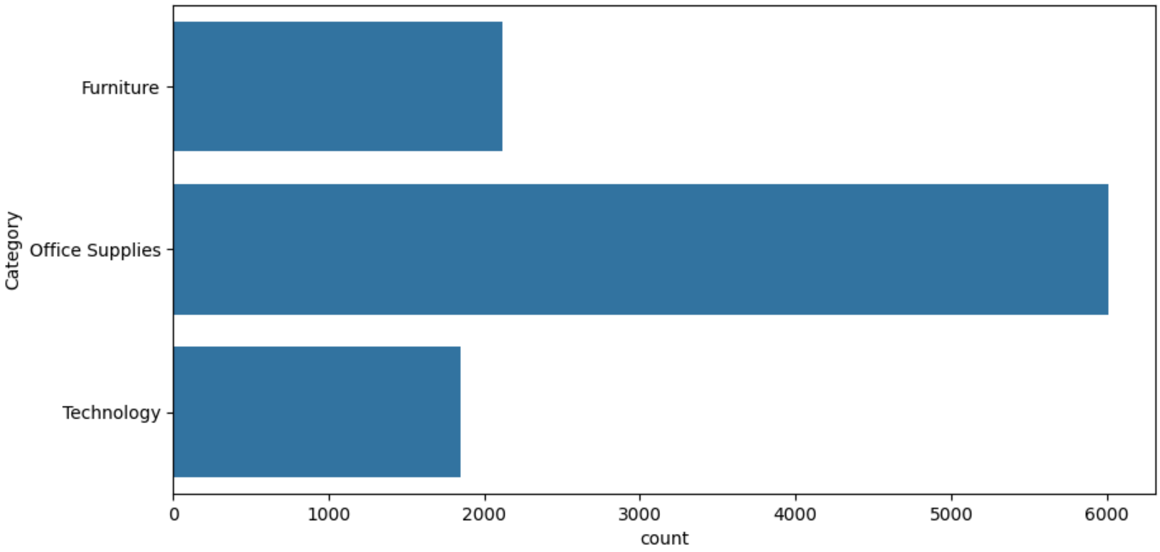
```
<Axes: xlabel='count', ylabel='Sub-Category'>
```

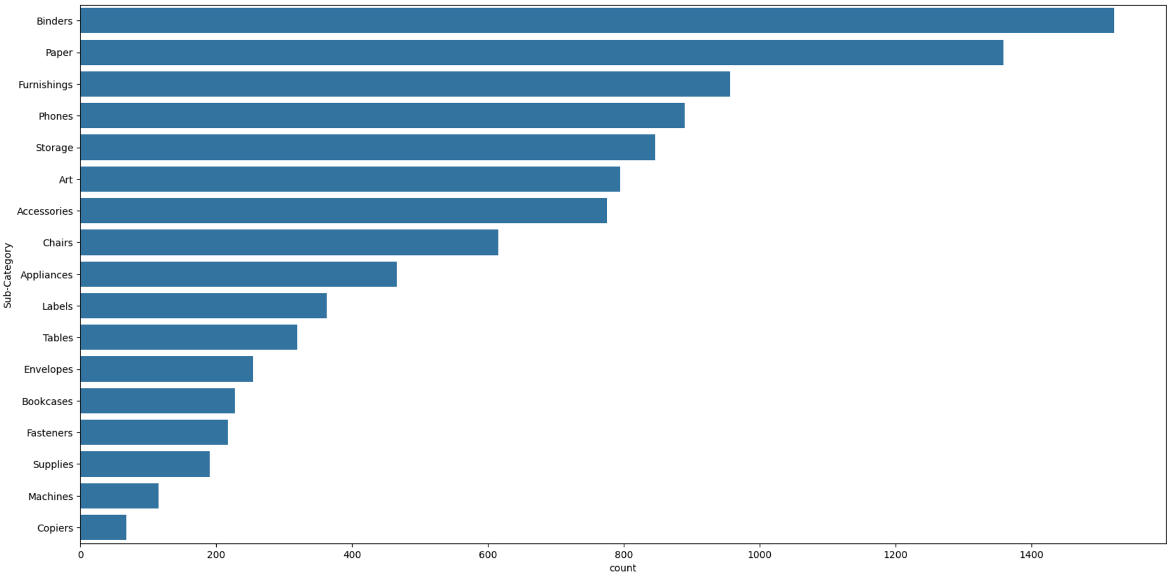

Ship Mode
Second Class
Standard Class
First Class
Same Day
0
1000
2000
3000
4000
5000
6000
count


Segment
Consumer
Corporate
Home Office
0
1000
2000
3000
4000
5000
count


State
California
New York
Texas
Pennsylvania
Washington
Illinois
Ohio
Florida
Michigan
North Carolina
Arizona
Virginia
Georgia
Tennessee
Colorado
Indiana
Kentucky
Massachusetts
New Jersey
Oregon
Wisconsin
Maryland
Delaware
Minnesota
Connecticut
Missouri
Oklahoma
Alabama
Arkansas
Rhode Island
Utah
Mississippi
South Carolina
Louisiana
Nevada
Nebraska
New Mexico
Iowa
New Hampshire
Kansas
Idaho
Montana
South Dakota
Vermont
District of Columbia
Maine
North Dakota
West Virginia
Wyoming
0
250
500
750
1000
1250
1500
1750
2000
count


Region
South
West
Central
East
0
500
1000
1500
2000
2500
3000
count





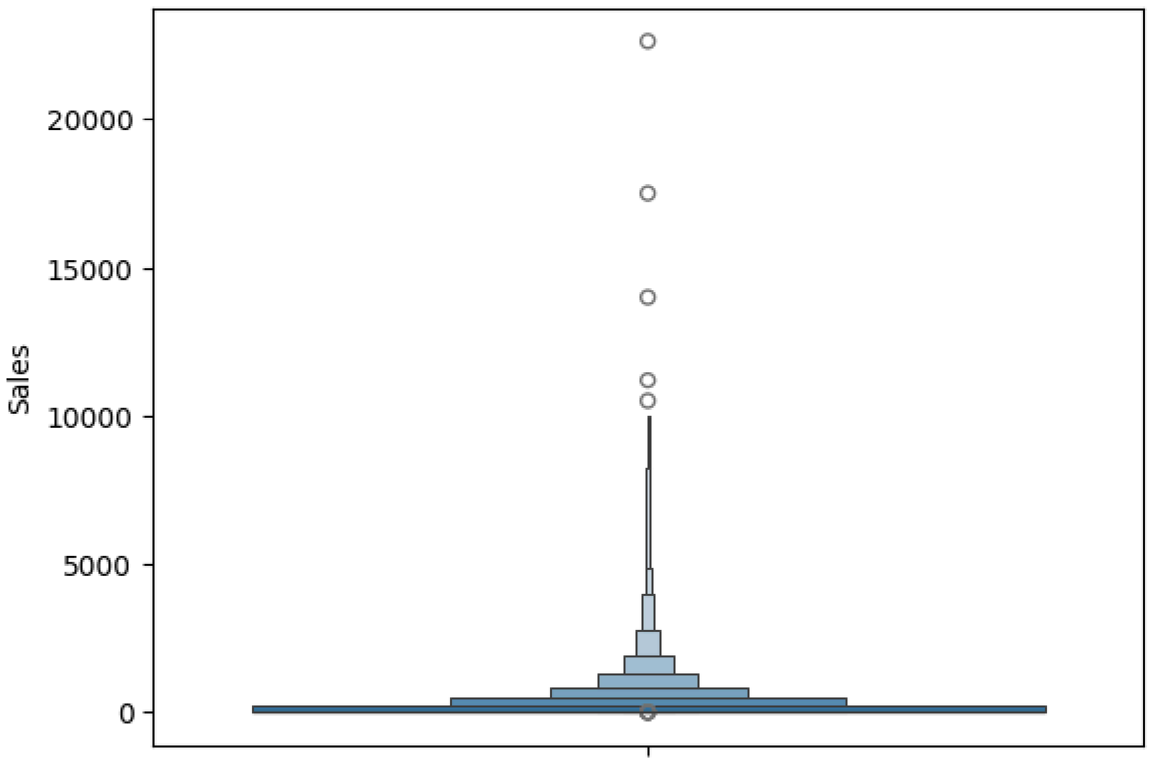
SALES AND PROFIT DISTRIBUTION AND THEIR CORRELATION

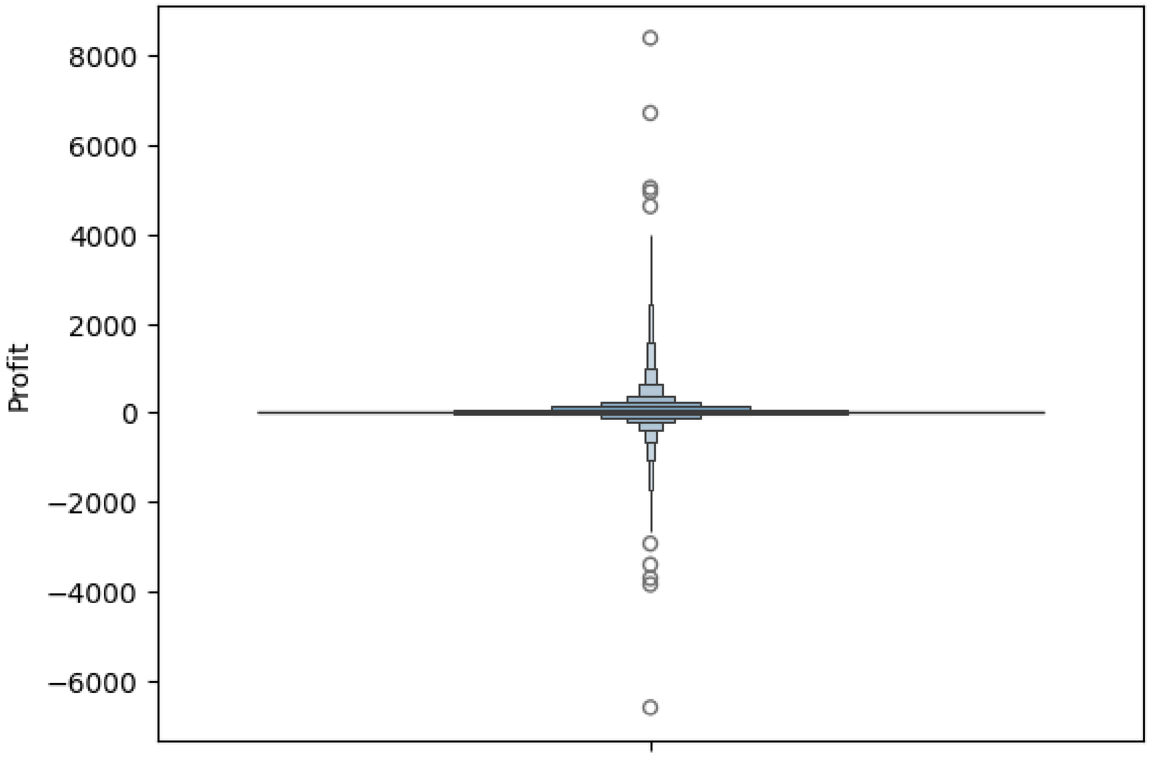
In [52]:

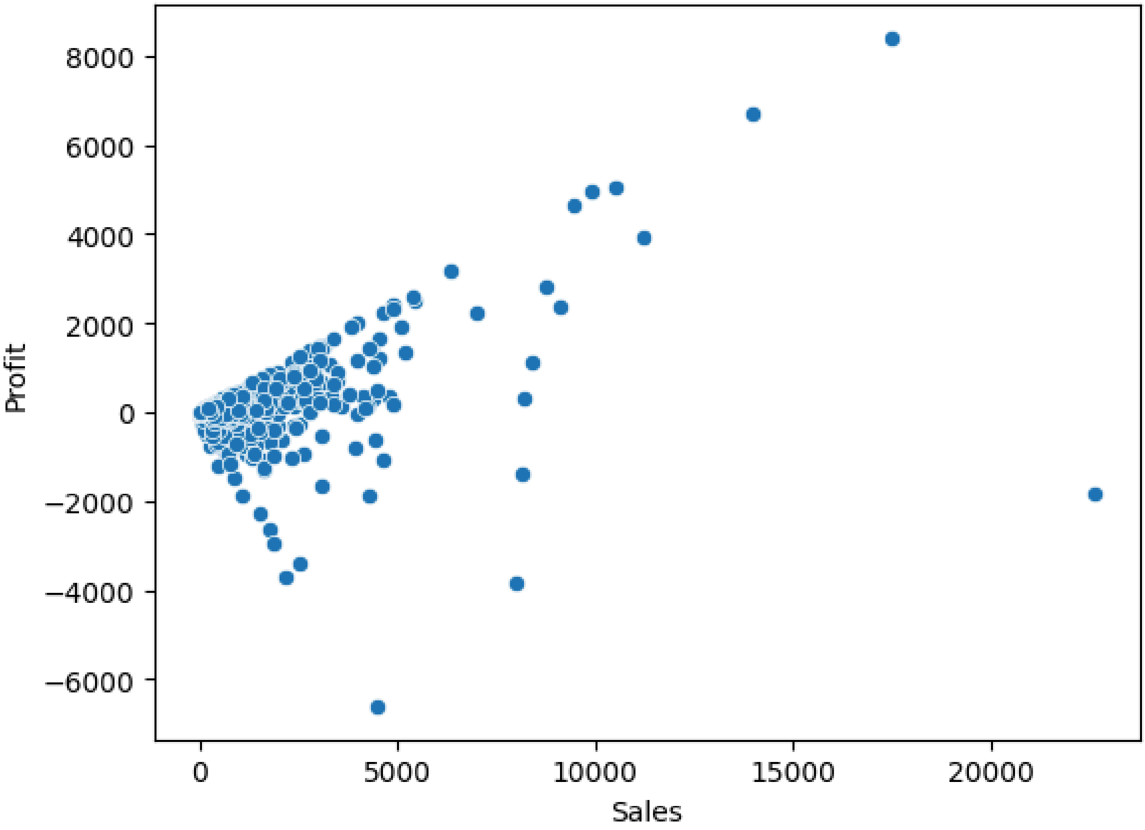
```
# Sales
sns.boxenplot(data=data['Sales'])
plt.figure()

# Profit
sns.boxenplot(data=data['Profit'])
plt.figure()

# Correlation Matrix
sns.scatterplot(data=data,x='Sales',y='Profit')
```

Out[52]:

```
<Axes: xlabel='Sales', ylabel='Profit'>
```

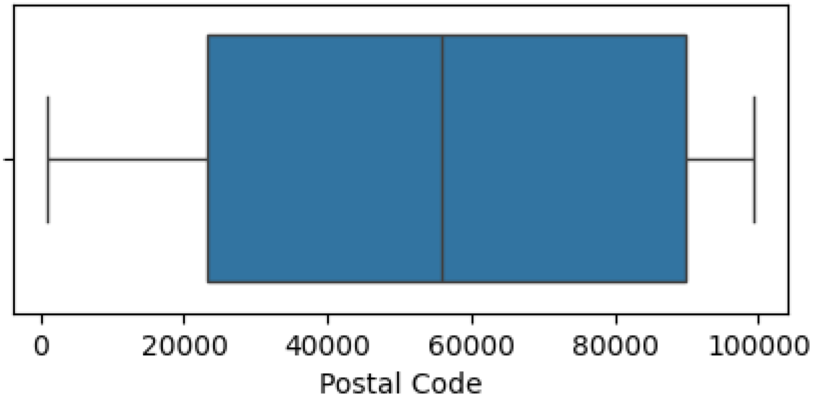

Sales
20000
15000
10000
5000
0

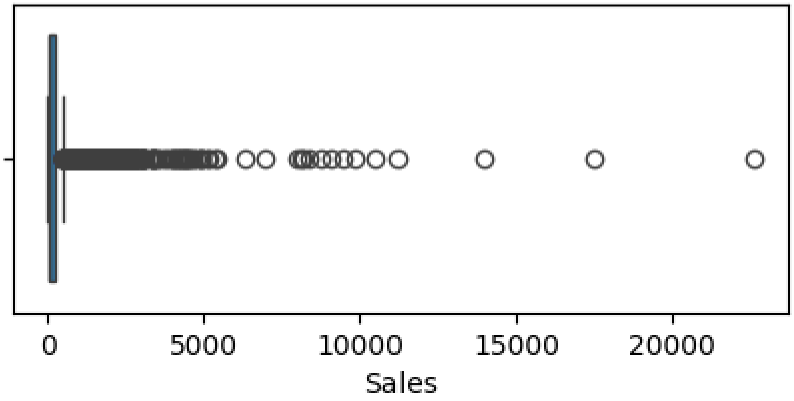

Profit
8000
6000
4000
2000
0
−2000
−4000
−6000

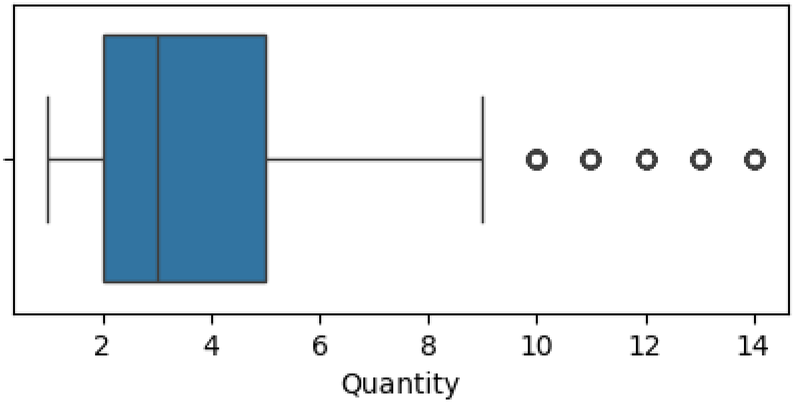


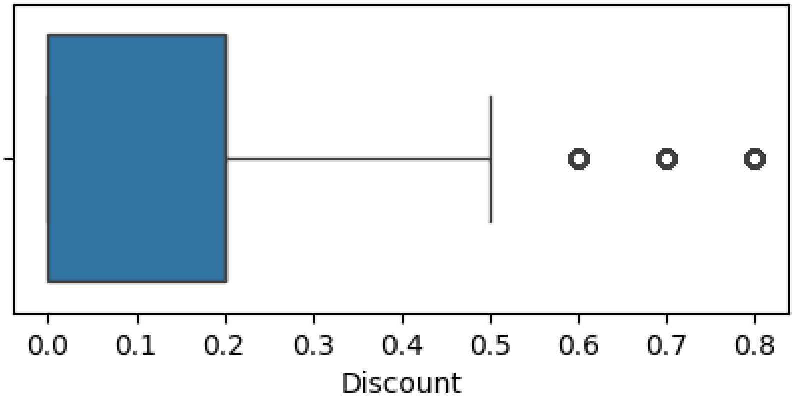
IDENTIFYING OUTLIERS

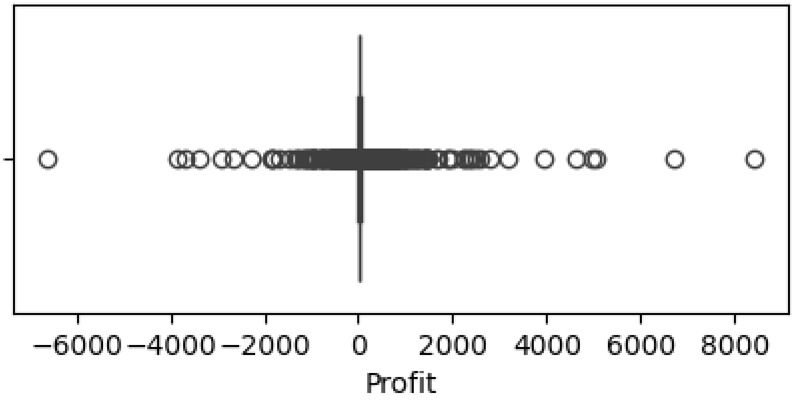
```
In [56]: x=['Postal Code','Sales','Quantity','Discount','Profit']
         for i in x:
             plt.figure(figsize=(5,2))
             sns.boxplot(x=i,data=data)
             plt.show()
```











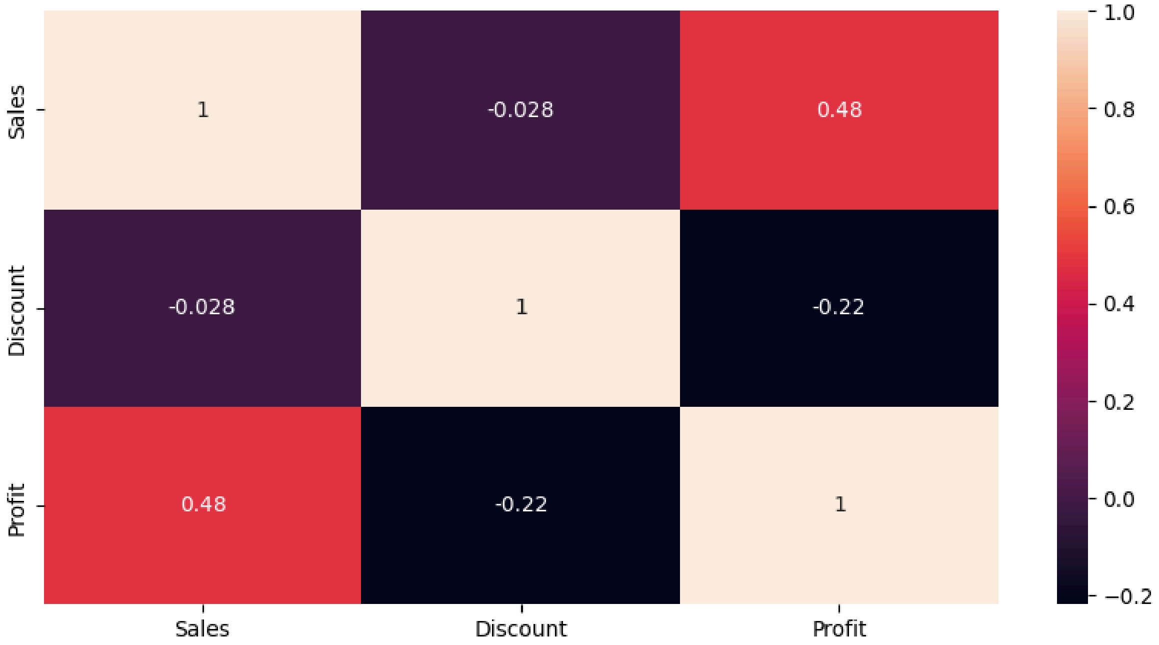
HEAT MAP

In [53]:

```
# Heatmap
x= data.select_dtypes(include='Float64')
plt.figure(figsize=(10,5))
sns.heatmap(x.corr(),annot=True)
```

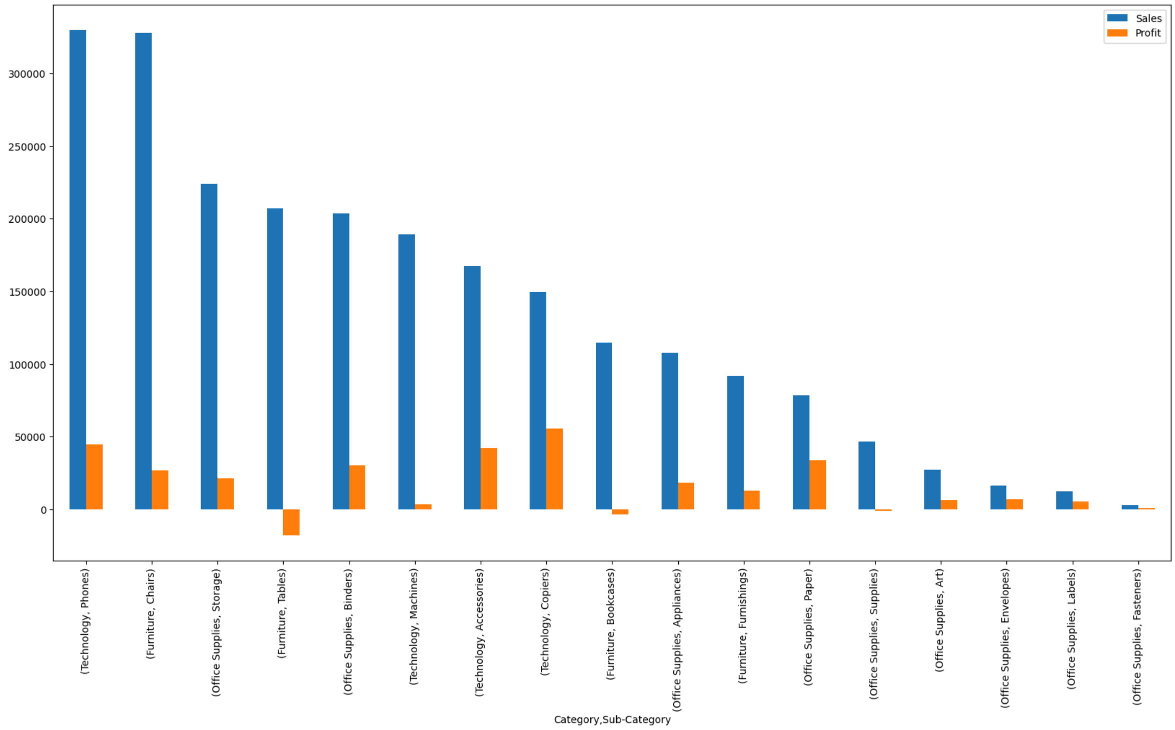
Out[53]:

```
<Axes: >
```



In [54]:
```
# Grouping data by Category and Sub-Category

x=data.groupby(['Category','Sub-Category'])[['Sales','Profit']].sum().sort_value
x.plot(kind='bar',figsize=(20,10))
```

Out[54]:
```
<Axes: xlabel='Category,Sub-Category'>
```

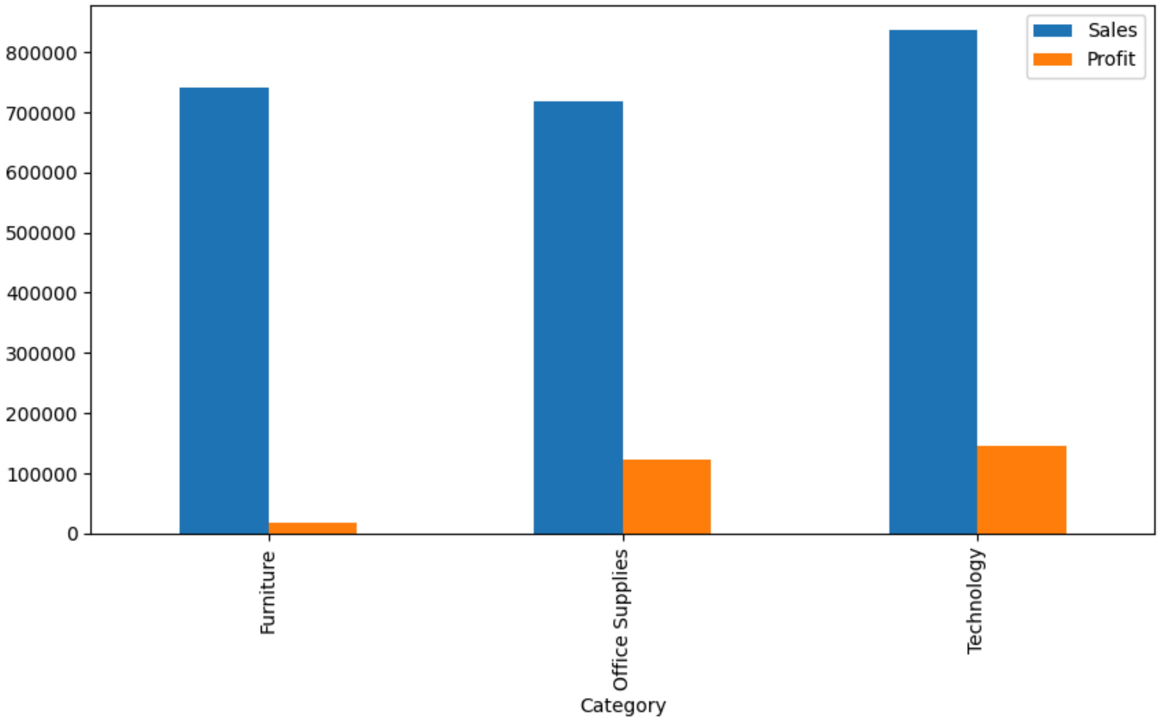


SALES AND PROFIT TRENDS

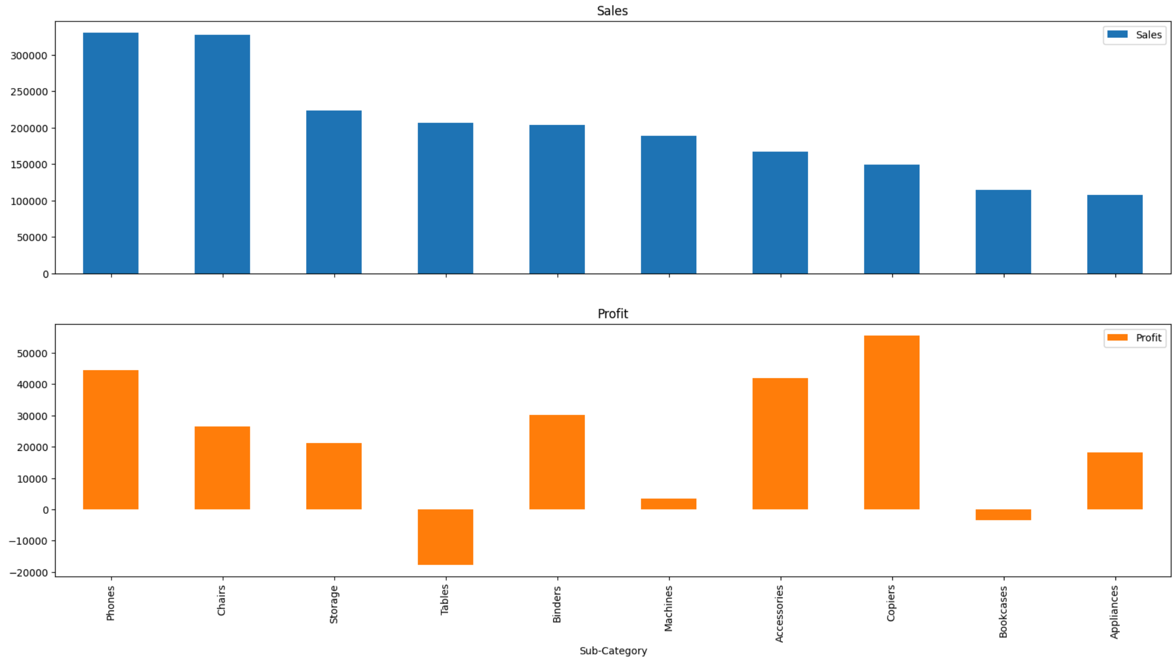
In [55]:
```
plt.figure(figsize=(10,5))
data.groupby('Category')[['Sales','Profit']].sum().plot(kind='bar',figsize=(10,5

plt.figure(figsize=(10,5))
data.groupby('Sub-Category')[['Sales','Profit']].sum().sort_values(by='Sales',as

# Grouping data by State
plt.figure(figsize=(20,10))
```

```
data.groupby('State')[['Sales','Profit']].sum().sort_values(by='Sales',ascending

# Grouping data by Region
plt.figure(figsize=(10,5))
data.groupby('Region')[['Sales','Profit']].sum().plot(kind='bar',figsize=(10,5))

# Grouping data by Segment
plt.figure(figsize=(10,5))
data.groupby('Segment')[['Sales','Profit']].sum().plot(kind='bar',figsize=(10,5)

# Grouping data by Ship Mode
plt.figure(figsize=(10,5))
data.groupby('Ship Mode')[['Sales','Profit']].sum().plot(kind='bar',figsize=(10,

# Grouping data by Discount
plt.figure(figsize=(10,5))
data.groupby('Discount')[['Sales','Profit']].sum().plot(kind='bar',figsize=(10,5

# Grouping data by Quantity
plt.figure(figsize=(10,5))
data.groupby('Quantity')[['Sales','Profit']].sum().plot(kind='bar',figsize=(10,5
```

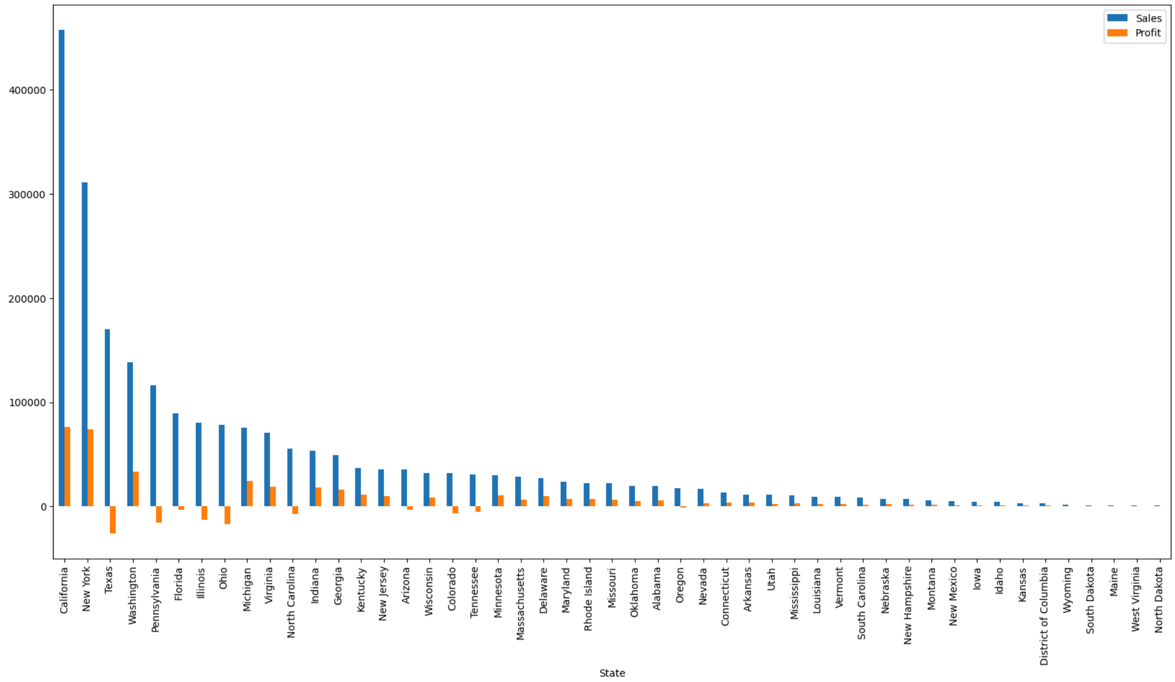
```
Out[55]:  <Axes: xlabel='Quantity'>

<Figure size 1000x500 with 0 Axes>
```



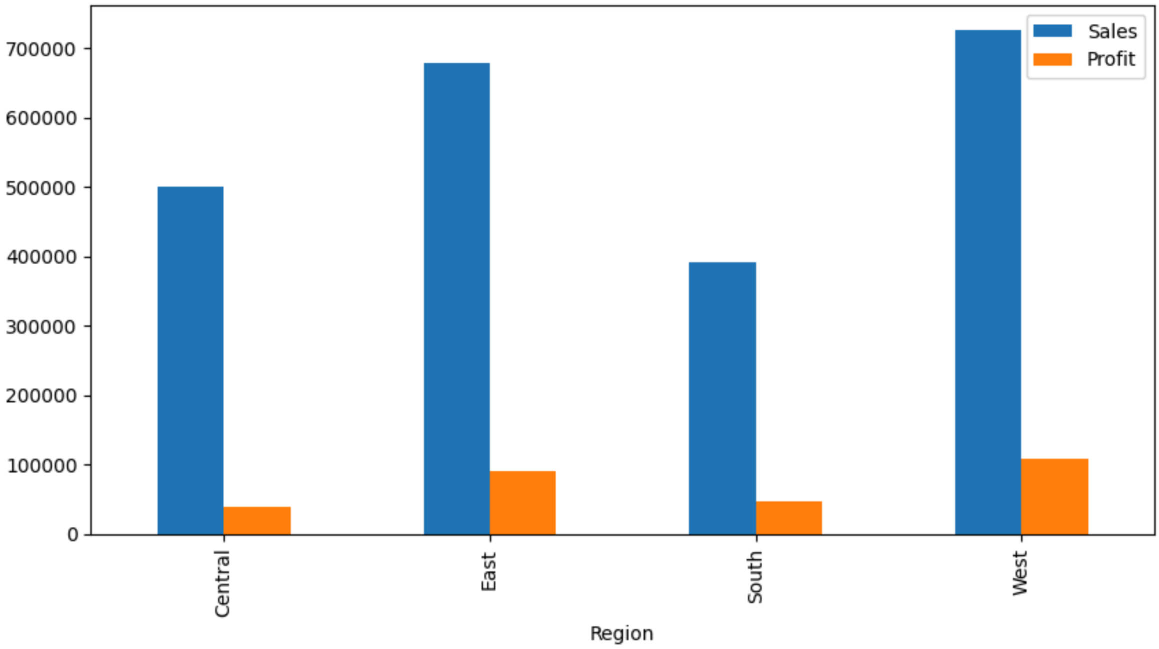
```
<Figure size 1000x500 with 0 Axes>
```


Sales
Sales
300000
250000
200000
150000
100000
50000
0
Profit
Profit
50000
40000
30000
20000
10000
0
−10000
−20000
Phones
Chairs
Storage
Tables
Binders
Machines
Accessories
Copiers
Bookcases
Appliances
Sub-Category

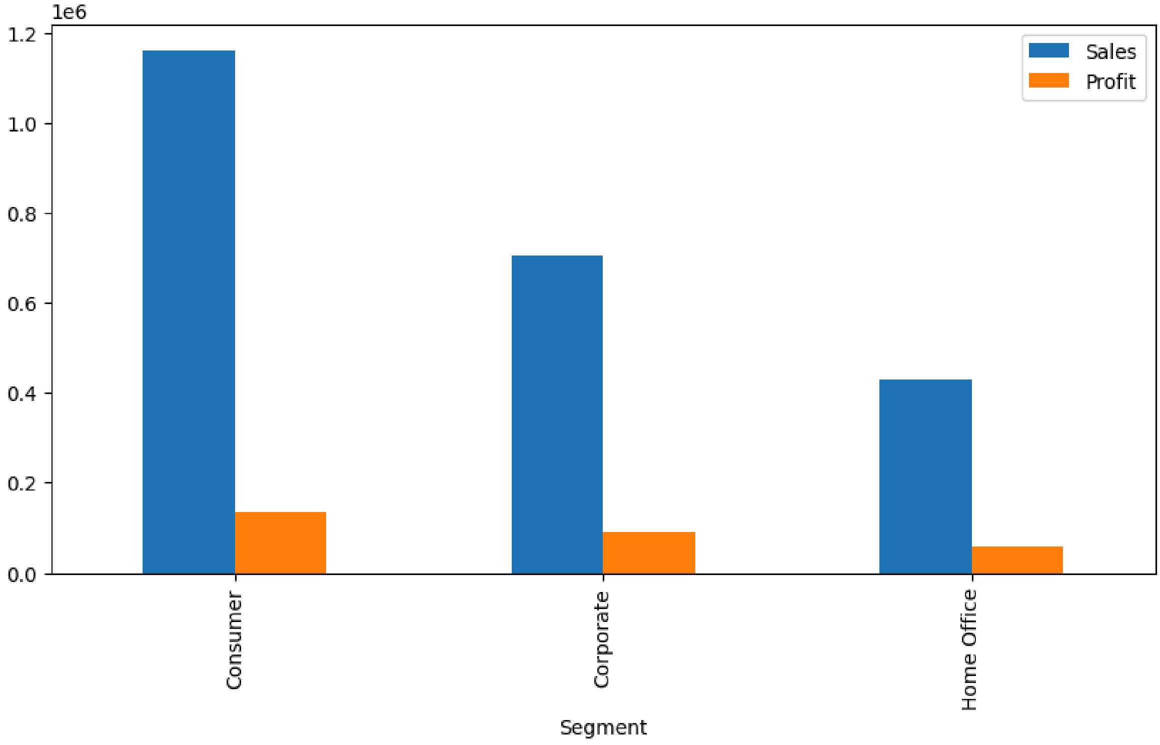
```
<Figure size 2000x1000 with 0 Axes>
```


Sales
Profit
400000
300000
200000
100000
0
California
New York
Texas
Washington
Pennsylvania
Florida
Illinois
Ohio
Michigan
Virginia
North Carolina
Indiana
Georgia
Kentucky
New Jersey
Arizona
Wisconsin
Colorado
Tennessee
Minnesota
Massachusetts
Delaware
Maryland
Rhode Island
Missouri
Oklahoma
Alabama
Oregon
Nevada
Connecticut
Arkansas
Utah
Mississippi
Louisiana
Vermont
South Carolina
Nebraska
New Hampshire
Montana
New Mexico
Iowa
Idaho
Kansas
District of Columbia
Wyoming
South Dakota
Maine
West Virginia
North Dakota
State

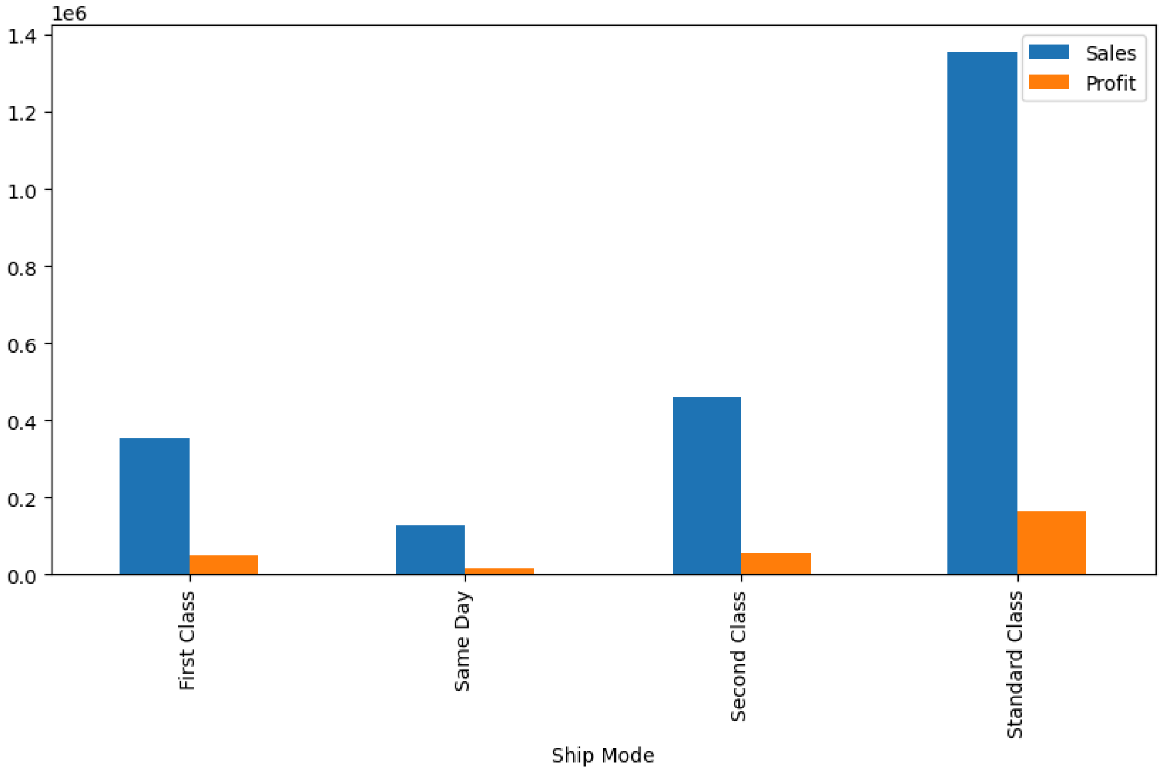
```
<Figure size 1000x500 with 0 Axes>
```


700000
600000
500000
400000
300000
200000
100000
0
Sales
Profit
Central
East
South
West
Region

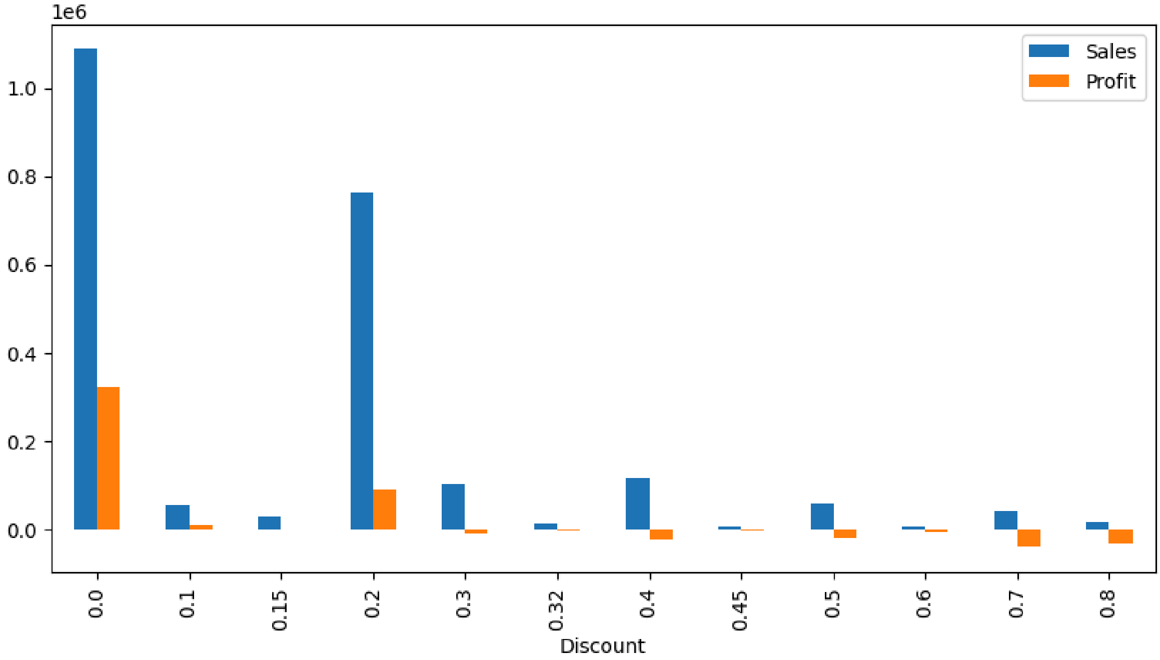
```
<Figure size 1000x500 with 0 Axes>
```


1e6
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Sales
Profit
Consumer
Corporate
Home Office
Segment

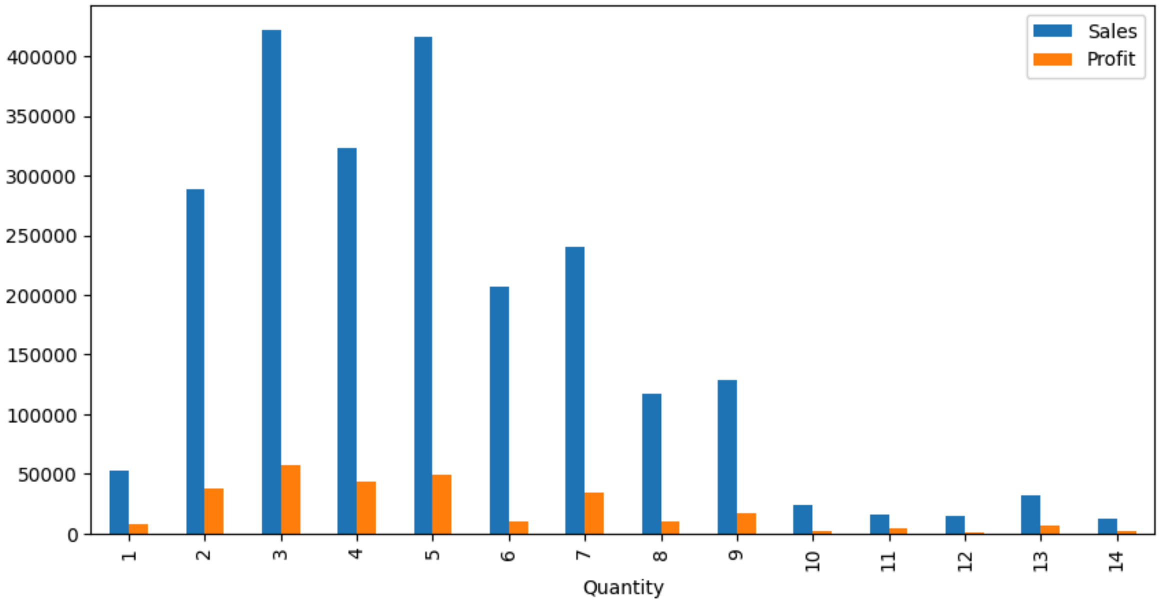
```
<Figure size 1000x500 with 0 Axes>
```


1e6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Sales
Profit
First Class
Same Day
Second Class
Standard Class
Ship Mode

```
<Figure size 1000x500 with 0 Axes>
```


1e6
1.0
0.8
0.6
0.4
0.2
0.0
Sales
Profit
0.0
0.1
0.15
0.2
0.3
0.32
0.4
0.45
0.5
0.6
0.7
0.8
Discount

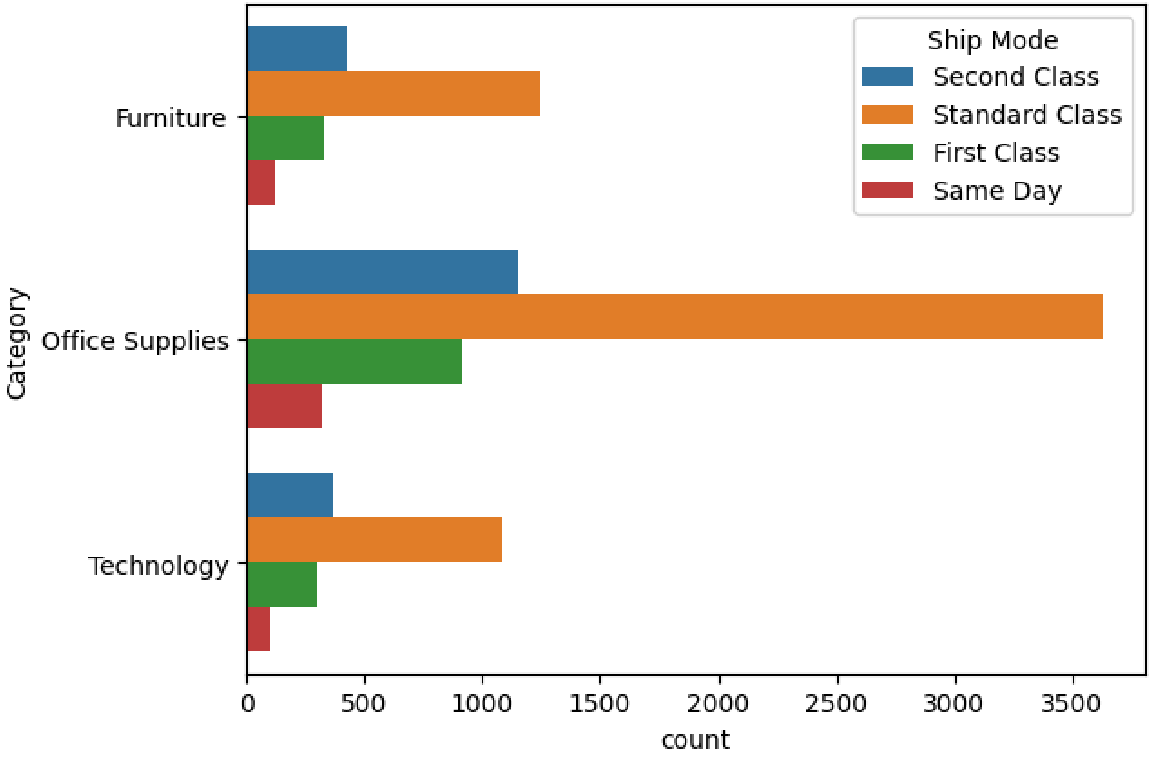
```
<Figure size 1000x500 with 0 Axes>
```

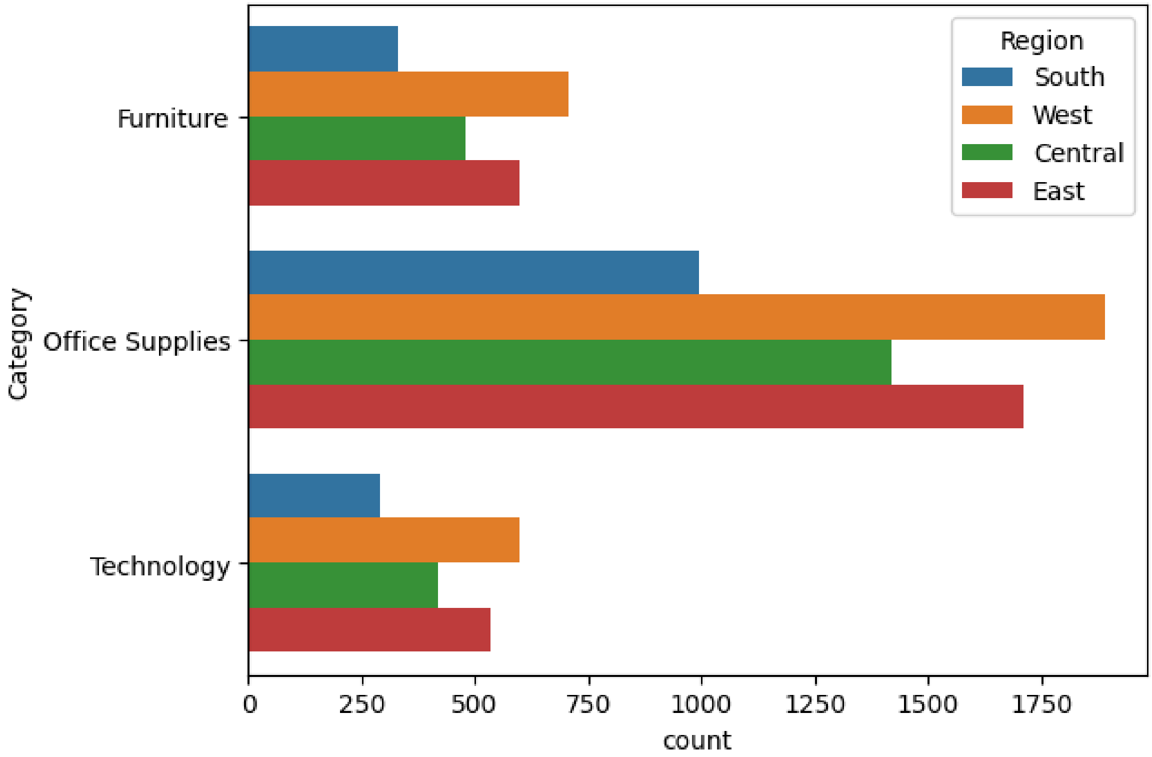


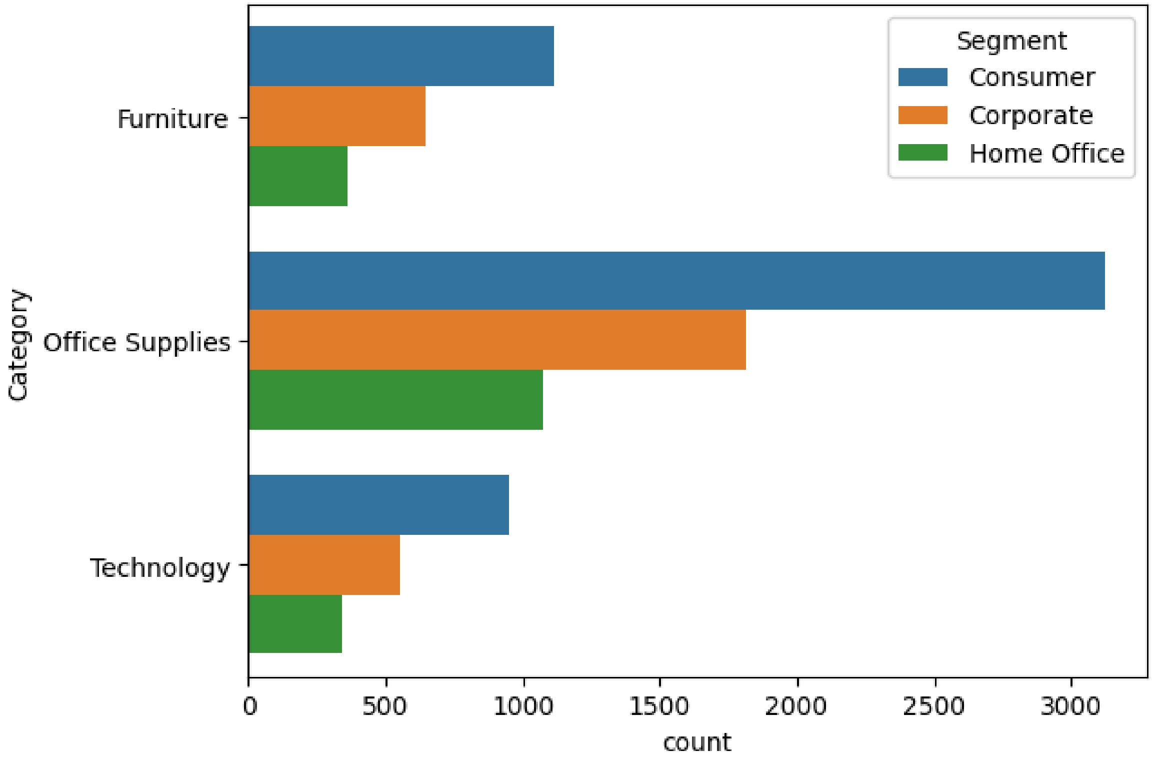
CATEGORY-WISE ANALYSIS

In [28]:
```
sns.countplot(data=data,y='Category',hue='Ship Mode')
plt.figure()

sns.countplot(data=data,y='Category',hue='Region',)
plt.figure()

sns.countplot(data=data,y='Category',hue='Segment')
plt.figure()
```

Out[28]:
```
<Figure size 640x480 with 0 Axes>
```





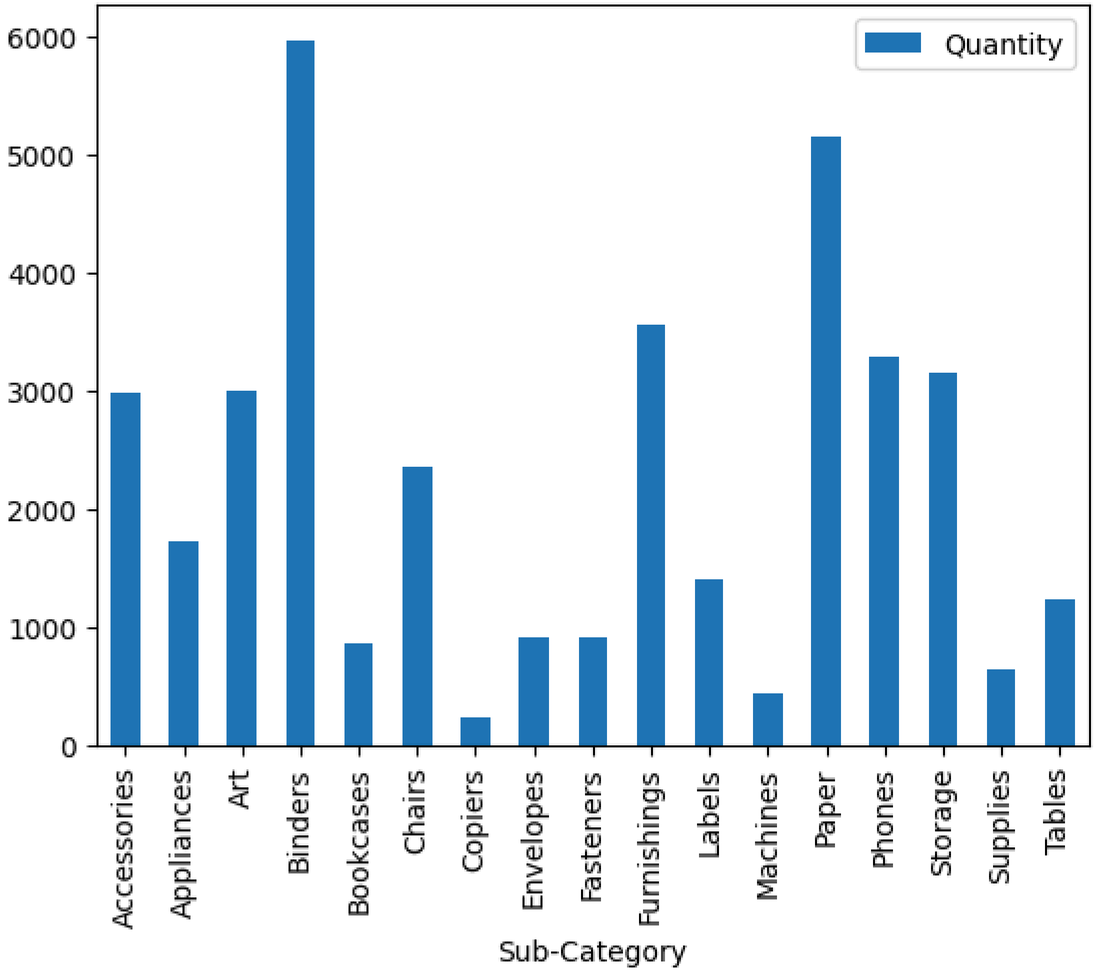


```
<Figure size 640x480 with 0 Axes>
```

SUB-CATEGORY WISE ANALYSIS

In [29]:
```
dd=data[['Sub-Category','Quantity']]
x=dd.groupby('Sub-Category').sum()
x.plot(kind='bar')
```

Out[29]:
```
<Axes: xlabel='Sub-Category'>
```

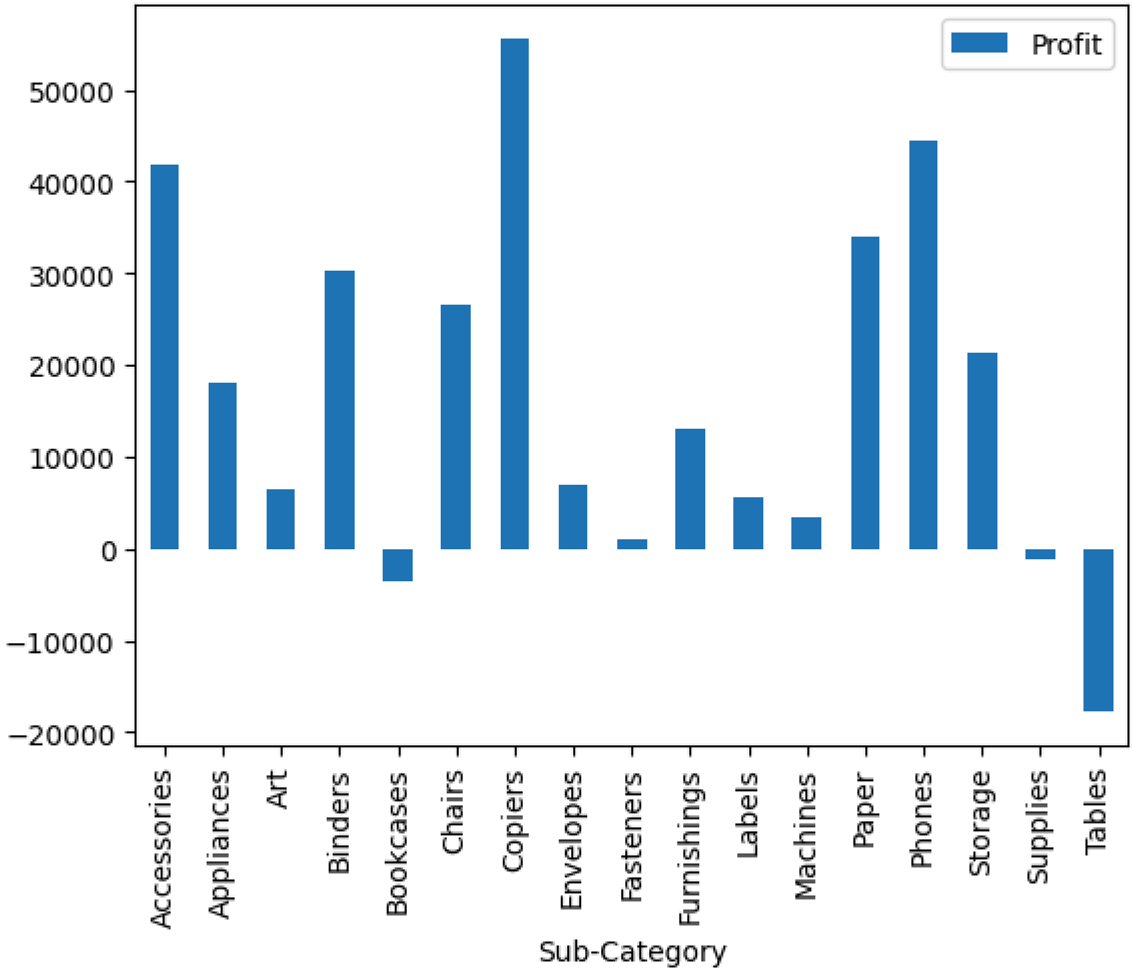


In [30]:

```
dd= data[['Sub-Category','Profit']]
x=dd.groupby('Sub-Category').sum()
x.plot(kind='bar')
```

Out[30]:

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



INSIGHTS

1. Preffered ship mode -> STANDARD
2. Largest share of users -> CONSUMER
3. Largest state by sales -> CALIFORNIA, NEW YORK, TEXAS
4. Net sales -> 2296195.39
5. Net profit -> 286240.95
6. Primary sales category -> Ofiice supplies
7. Category producing most profit -> Technology
8. States showing negative profit -> Texas, Pensylvania, Ohio
9. Most sold product (by quantity) -> Binders, Papers
10. Product showing neative profit -> Tables, Bookcases