

# GRIP OCTOBER

## THE SPARKS FOUNDATION

### TASK 3: Exploratory data analysis of superstore

This is the task performed by Saksham Sharma in the intership #GRIPOCTOBER under THE SPARKS FOUNDATION

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

```
data=pd.read_csv(r'D:/Saksham 5th sem/Sparks internship tasks/SampleSuperstore.csv')
data.head()
```

	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

```
data=data.drop(['Country','Postal Code'], axis =1)
data.head()
```

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

```
data.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

```
data.shape
```

```
(9994, 11)
```

```
data.nunique()
```

```
Ship Mode      4
Segment        3
City           531
State          49
Region         4
Category       3
Sub-Category   17
Sales          5825
Quantity       14
Discount       12
Profit         7287
dtype: int64

data.isnull().sum()

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

data['Segment'].unique()

array(['Consumer', 'Corporate', 'Home Office'], dtype=object)

data['Ship Mode'].unique()

array(['Second Class', 'Standard Class', 'First Class', 'Same Day'],
      dtype=object)

data['Category'].unique()

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

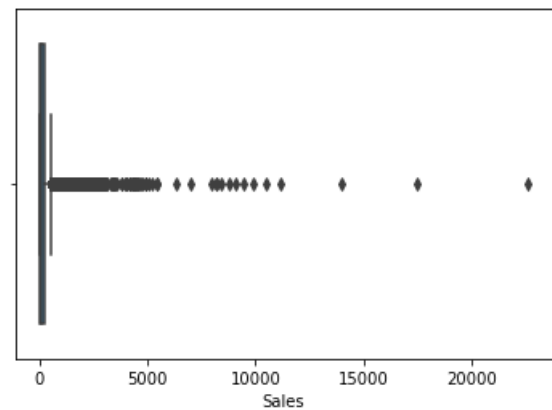
data['Region'].unique()

array(['South', 'West', 'Central', 'East'], dtype=object)

for column in data.columns[7:11]:
    sns.boxplot(data[column])
    plt.show()
```

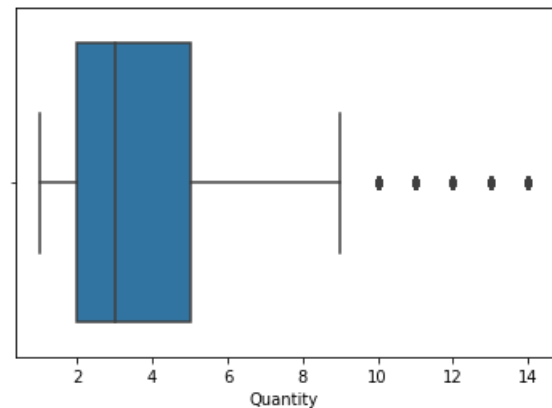
C:\Users\15264\Anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn()



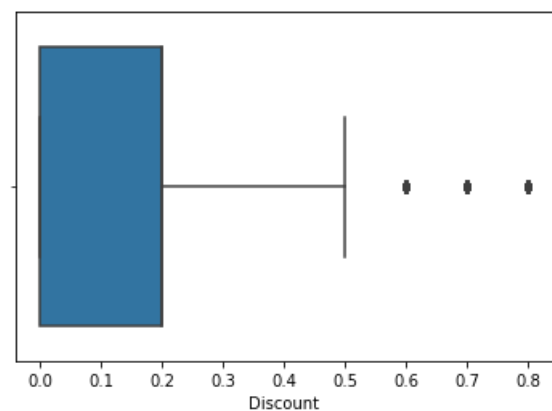
C:\Users\15264\Anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn()



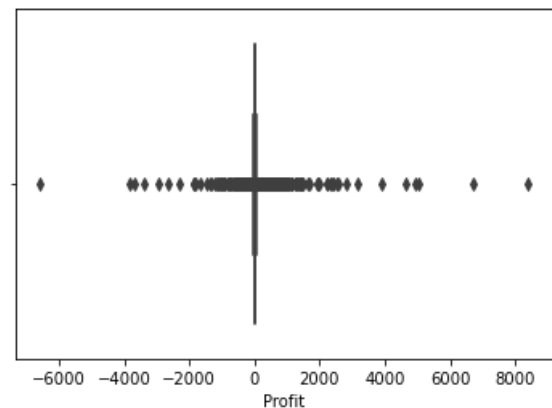
C:\Users\15264\Anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn()



C:\Users\15264\Anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn()



In [175]:

```
df_outlier_removed=pd.DataFrame()
def remove_outlier_IQR(df):
    Q1=df.quantile(0.25)
    Q3=df.quantile(0.75)
    IQR=Q3-Q1
```

```
width=1.5
lower=Q1-(width*IQR)
upper=Q3+(width*IQR)

df_final=np.where(df>upper,upper,np.where(df<lower,lower,df))
return df_final

for column in data.columns[7:11]:
    df_outlier_removed[column]=remove_outlier_IQR(data[column])

df_outlier_removed.shape
df_outlier_removed
```

Out[175]:

	Sales	Quantity	Discount	Profit
0	261.960	2.0	0.00	41.913600
1	498.930	3.0	0.00	70.816875
2	14.620	2.0	0.00	6.871400
3	498.930	5.0	0.45	-39.724125
4	22.368	2.0	0.20	2.516400
...	...	...	...	...
9989	25.248	3.0	0.20	4.102800
9990	91.960	2.0	0.00	15.633200
9991	258.576	2.0	0.20	19.393200
9992	29.600	4.0	0.00	13.320000
9993	243.160	2.0	0.00	70.816875

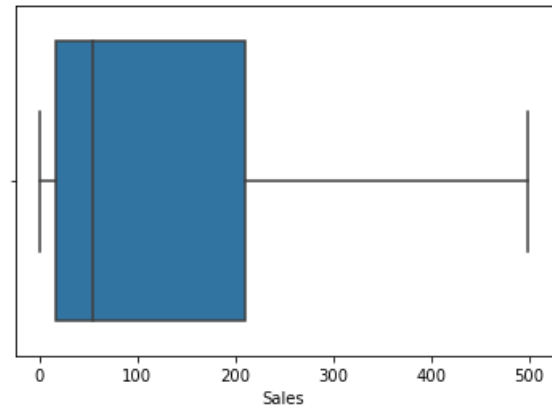
9994 rows × 4 columns

In [176]:

```
for column in df_outlier_removed.loc[8:11]:
    sns.boxplot(df_outlier_removed[column])
    plt.show()
```

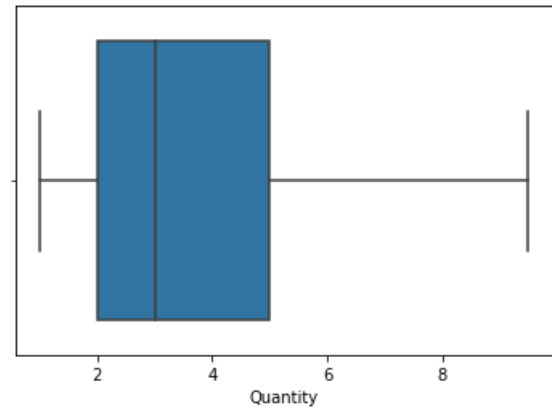
C:\Users\15264\Anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(



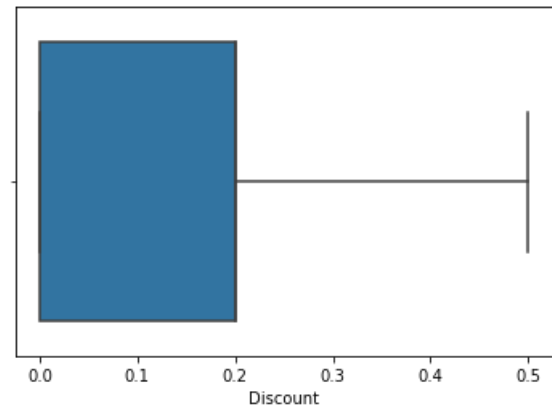
C:\Users\15264\Anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(



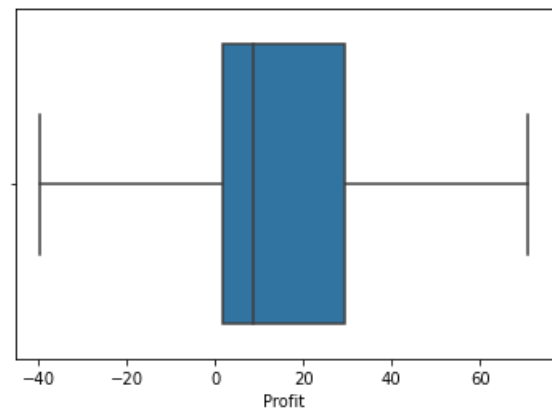
C:\Users\15264\Anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(



C:\Users\15264\Anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(



In [183]:

```
data = pd.concat([data.drop(data.columns[7:11], axis=1), df_outlier_removed], axis=1, join='inner')
display(result)
```

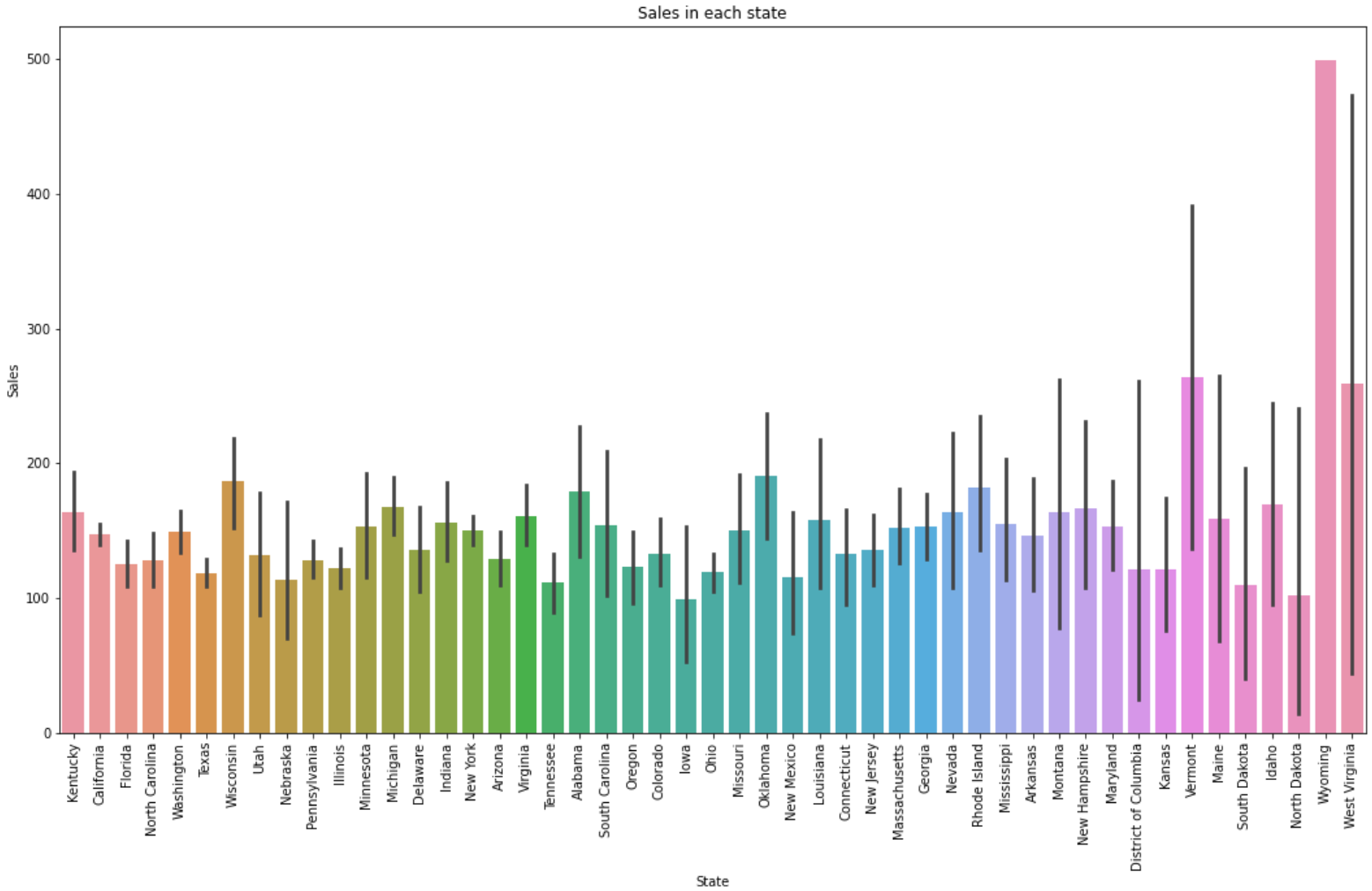
	Ship Mode	Segment	City	State	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
0	Second Class	Consumer	Henderson	Kentucky	South	Furniture	Bookcases	261.960	2.0	0.00	41.913600
1	Second Class	Consumer	Henderson	Kentucky	South	Furniture	Chairs	498.930	3.0	0.00	70.816875
2	Second Class	Corporate	Los Angeles	California	West	Office Supplies	Labels	14.620	2.0	0.00	6.871400
3	Standard Class	Consumer	Fort Lauderdale	Florida	South	Furniture	Tables	498.930	5.0	0.45	-39.724125
4	Standard Class	Consumer	Fort Lauderdale	Florida	South	Office Supplies	Storage	22.368	2.0	0.20	2.516400
...	...	...	...	...	...	...	...	...	...	...	...
9989	Second Class	Consumer	Miami	Florida	South	Furniture	Furnishings	25.248	3.0	0.20	4.102800
9990	Standard Class	Consumer	Costa Mesa	California	West	Furniture	Furnishings	91.960	2.0	0.00	15.633200
9991	Standard Class	Consumer	Costa Mesa	California	West	Technology	Phones	258.576	2.0	0.20	19.393200
9992	Standard Class	Consumer	Costa Mesa	California	West	Office Supplies	Paper	29.600	4.0	0.00	13.320000
9993	Second Class	Consumer	Westminster	California	West	Office Supplies	Appliances	243.160	2.0	0.00	70.816875

9994 rows × 11 columns

```
fig = plt.figure(figsize = (18, 10))
chart=sns.barplot('State','Sales',data=data)
chart.set_xticklabels(chart.get_xticklabels(), rotation=90)
plt.xlabel("State")
plt.ylabel("Sales")
plt.title("Sales in each state")
plt.show()
```

C:\Users\15264\Anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

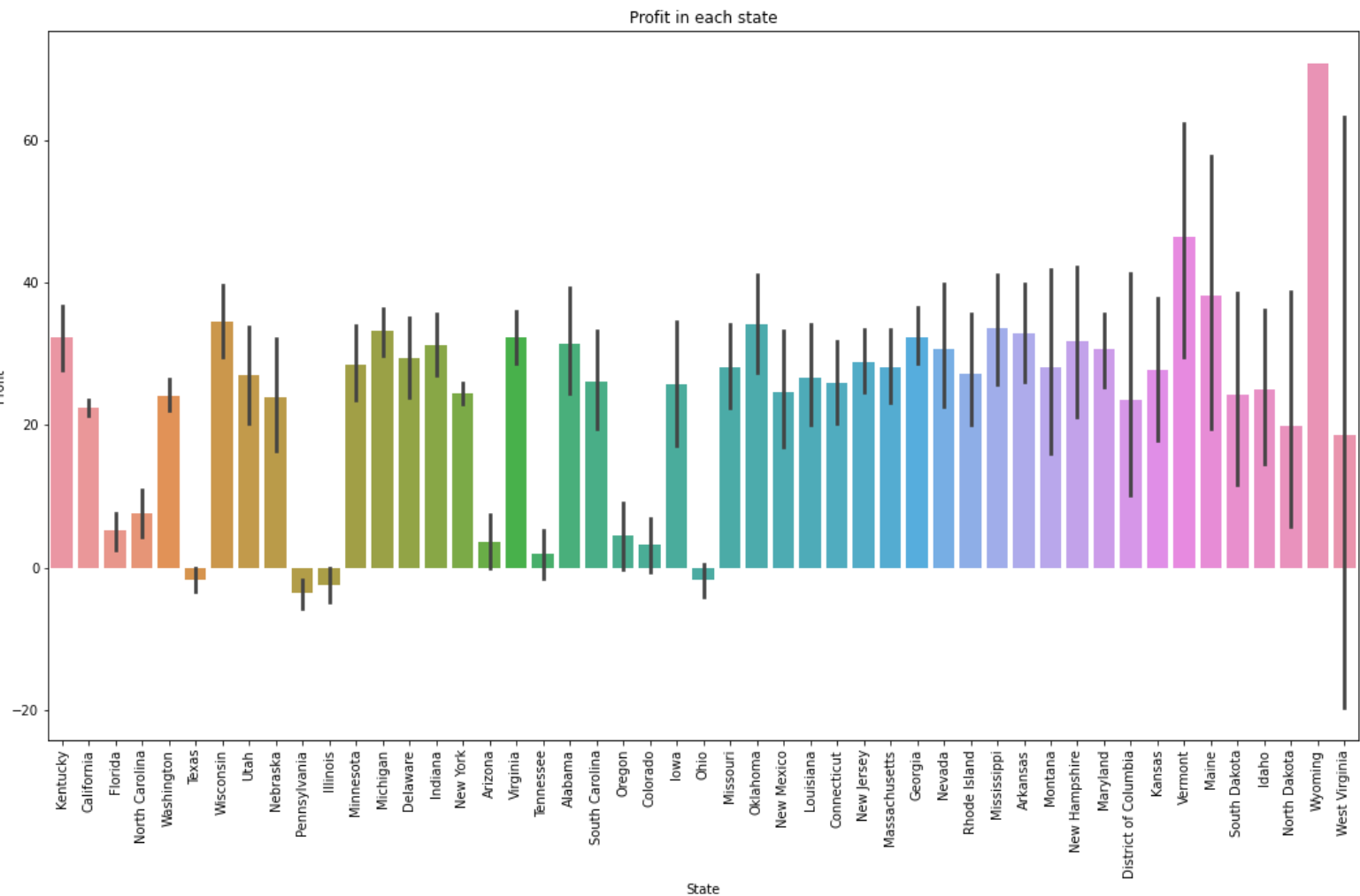
warnings.warn(



```
fig = plt.figure(figsize = (18, 10))
chart=sns.barplot('State','Profit',data=data)
chart.set_xticklabels(chart.get_xticklabels(), rotation=90)
plt.xlabel("State")
plt.ylabel("Profit")
plt.title("Profit in each state")
plt.show()
```

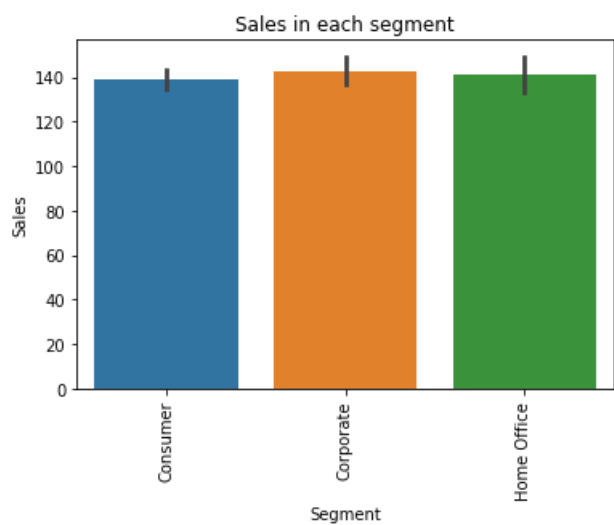
In [184]:

In [185]:



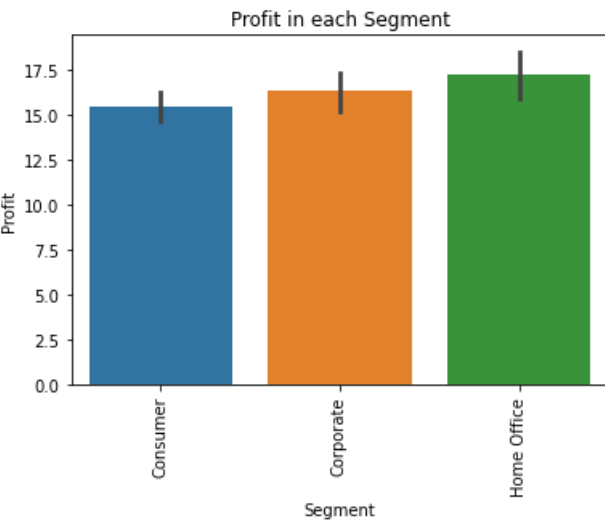
In [186]:

```
chart=sns.barplot(x='Segment',y='Sales',data=data)
chart.set_xticklabels(chart.get_xticklabels(), rotation=90)
plt.xlabel("Segment")
plt.ylabel("Sales")
plt.title("Sales in each segment")
plt.show()
```



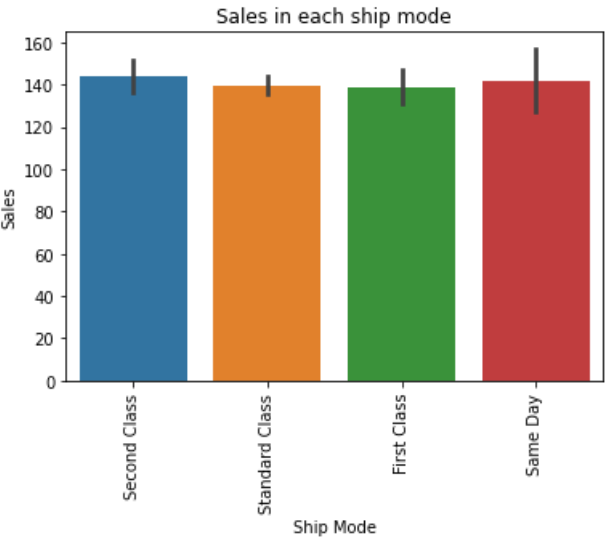
In [187]:

```
chart=sns.barplot(x='Segment',y='Profit',data=data)
chart.set_xticklabels(chart.get_xticklabels(), rotation=90)
plt.xlabel("Segment")
plt.ylabel("Profit")
plt.title("Profit in each Segment")
plt.show()
```



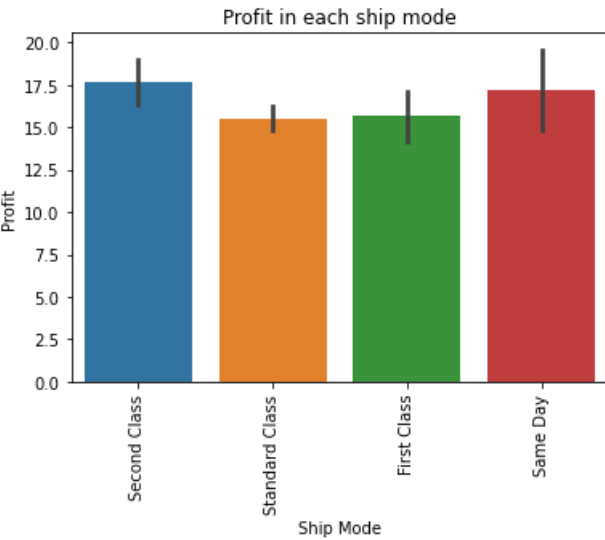
In [188]:

```
chart=sns.barplot(x='Ship Mode',y='Sales',data=data)
chart.set_xticklabels(chart.get_xticklabels(), rotation=90)
plt.xlabel("Ship Mode")
plt.ylabel("Sales")
plt.title("Sales in each ship mode")
plt.show()
```



In [189]:

```
chart=sns.barplot(x='Ship Mode',y='Profit',data=data)
chart.set_xticklabels(chart.get_xticklabels(), rotation=90)
plt.xlabel("Ship Mode")
plt.ylabel("Profit")
plt.title("Profit in each ship mode")
plt.show()
```

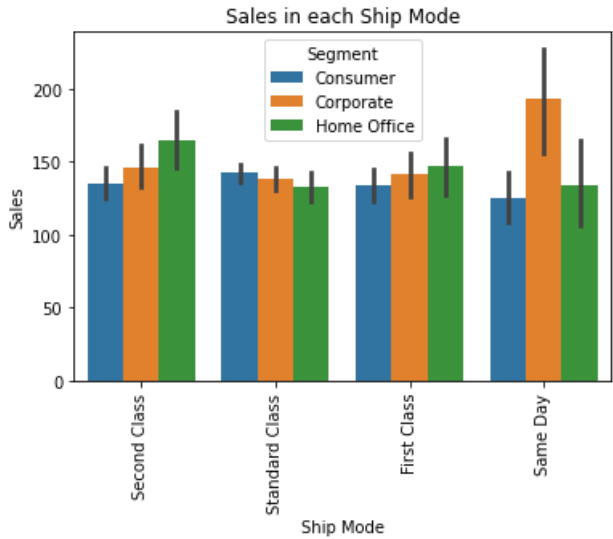


In [190]:

```
chart=sns.barplot(x='Ship Mode',y='Sales',hue='Segment',data=data)
```



```
chart.set_xticklabels(chart.get_xticklabels(), rotation=90)
plt.xlabel("Ship Mode")
plt.ylabel("Sales")
plt.title("Sales in each Ship Mode")
plt.show()
```



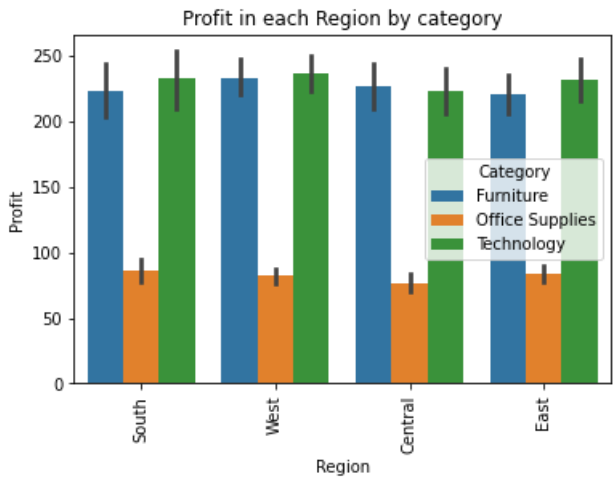
In [191]:

```
chart=sns.barplot(x='Region',y='Sales',hue='Segment',data=data)
chart.set_xticklabels(chart.get_xticklabels(), rotation=90)
plt.xlabel("Region")
plt.ylabel("Sales")
plt.title("Sales in each Region")
plt.show()
```



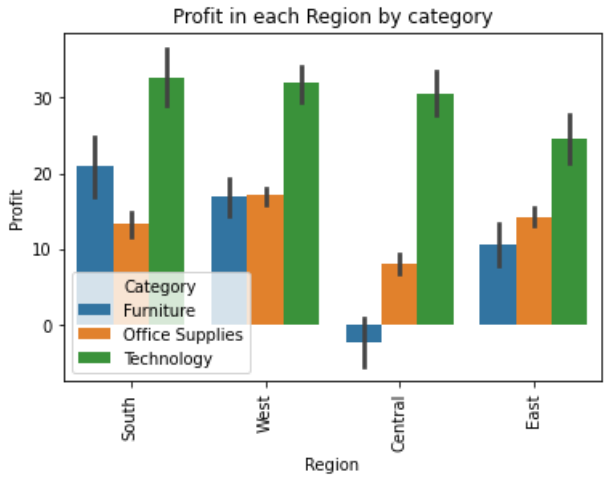
In [216]:

```
chart=sns.barplot(x='Region',y='Sales',hue='Category',data=data)
chart.set_xticklabels(chart.get_xticklabels(), rotation=90)
plt.xlabel("Region")
plt.ylabel("Sales")
plt.title("Sales in each Region by category")
plt.show()
```



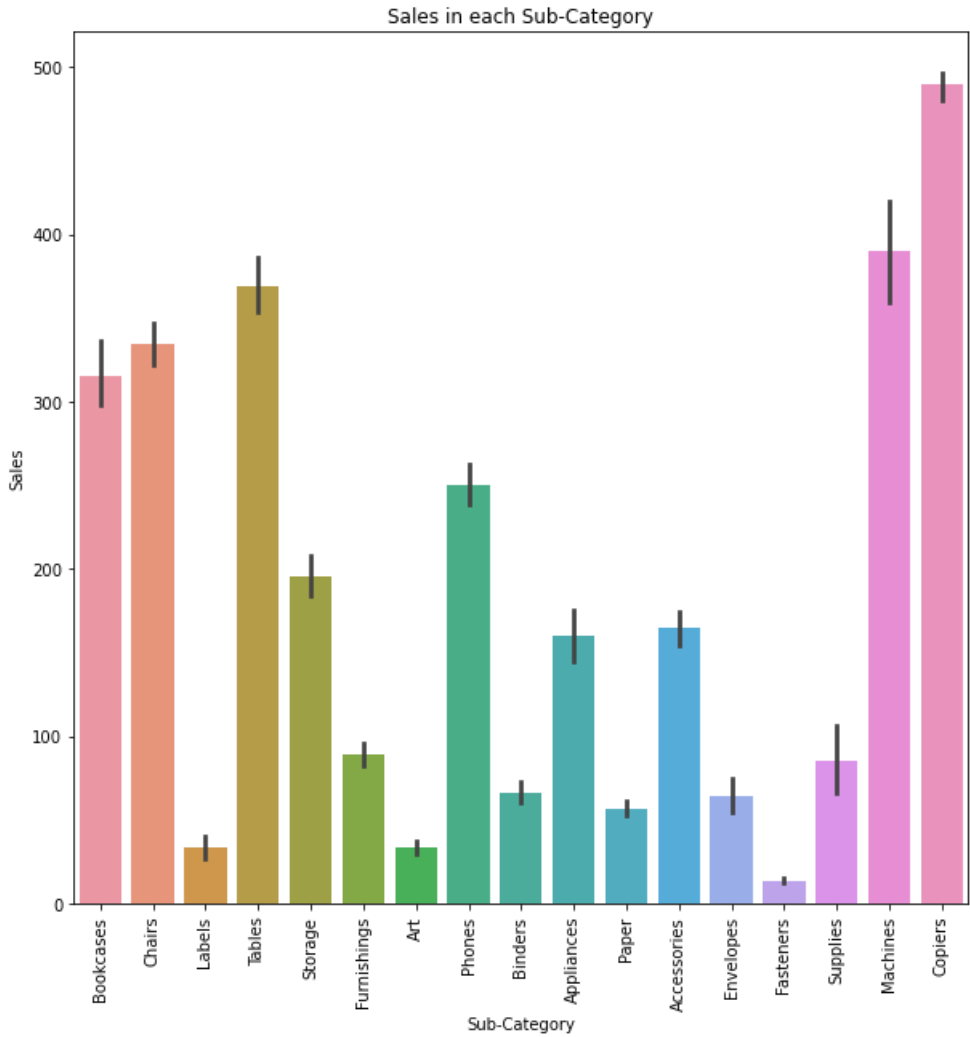
In [217]:

```
chart=sns.barplot(x='Region',y='Profit',data=data)
chart.set_xticklabels(chart.get_xticklabels(), rotation=90)
plt.xlabel("Region")
plt.ylabel("Profit")
plt.title("Profit in each Region by category")
plt.show()
```



In [192]:

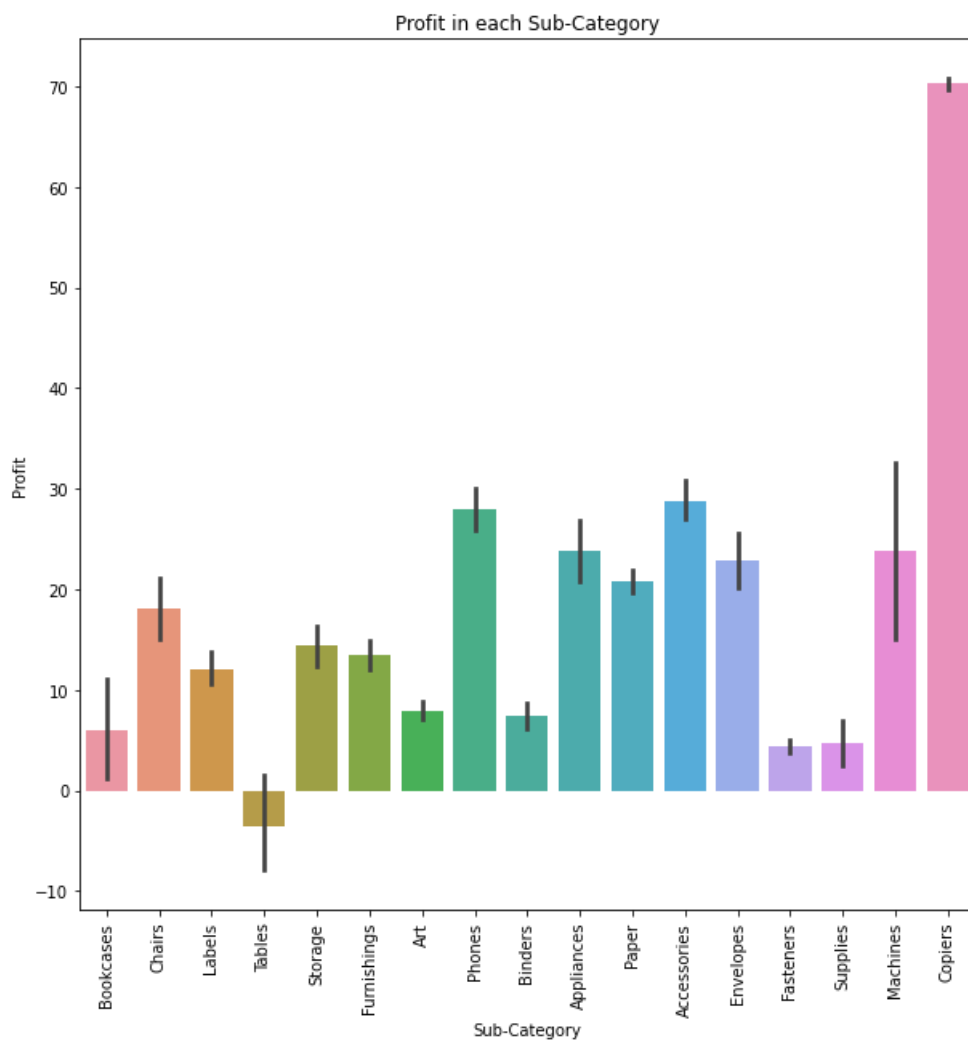
```
fig = plt.figure(figsize = (10, 10))
chart=sns.barplot(x='Sub-Category',y='Sales',data=data)
chart.set_xticklabels(chart.get_xticklabels(), rotation=90)
plt.xlabel("Sub-Category")
plt.ylabel("Sales")
plt.title("Sales in each Sub-Category")
plt.show()
```



In [193]:

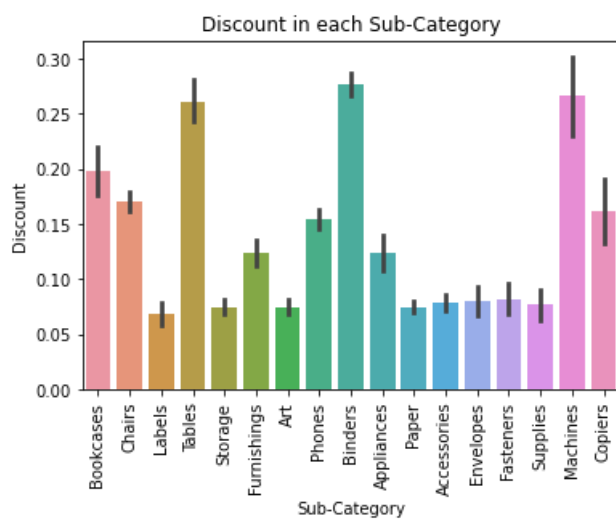
```
fig = plt.figure(figsize = (10, 10))
chart=sns.barplot(x='Sub-Category',y='Profit',data=data)
chart.set_xticklabels(chart.get_xticklabels(), rotation=90)
plt.xlabel("Sub-Category")
plt.ylabel("Profit")
plt.title("Profit in each Sub-Category")
```

```
plt.show()
```



In [194]:

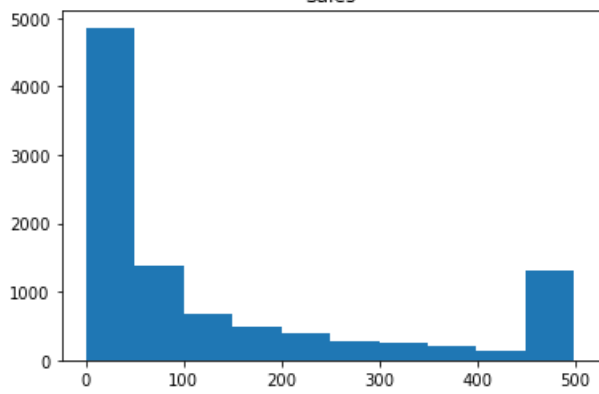
```
chart=sns.barplot(x='Sub-Category',y='Discount',data=data)
chart.set_xticklabels(chart.get_xticklabels(), rotation=90)
plt.xlabel("Sub-Category")
plt.ylabel("Discount")
plt.title("Discount in each Sub-Category")
plt.show()
```



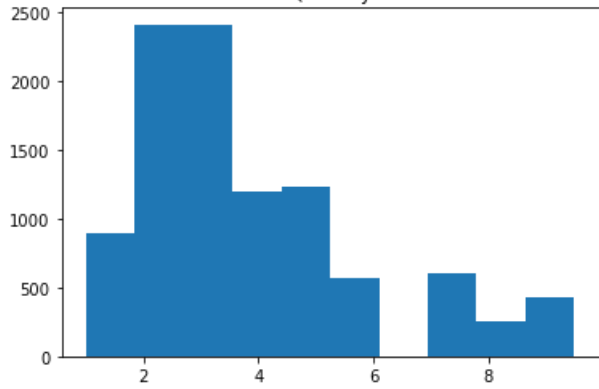
In [195]:

```
for column in df_outlier_removed.columns:
    plt.hist(df_outlier_removed[column])
    plt.title(column)
    plt.show()
```

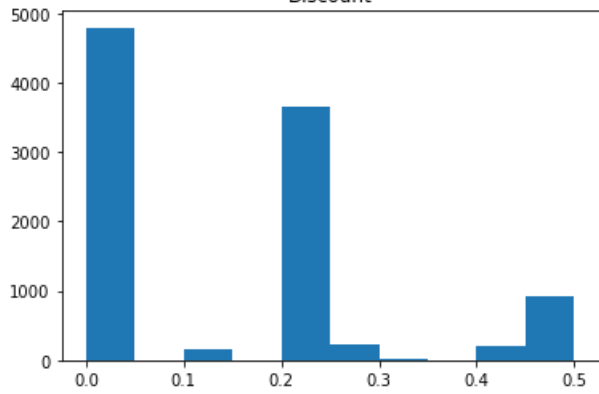
Sales



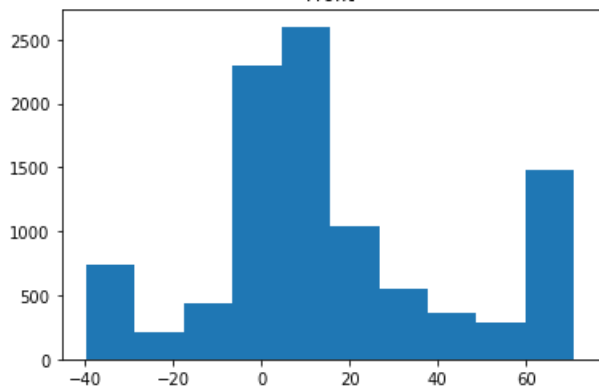
Quantity



Discount



Profit



In [196]:

```
data.groupby(['Region'])['Profit'].mean()
```

Out[196]:

```
Region
Central    10.055174
East       15.421148
South      18.393694
West       19.827769
Name: Profit, dtype: float64
```

In [197]:

```
data.groupby(['Category','Segment'])['Profit'].mean()
```

Out[197]:

```
Category      Segment
Furniture      Consumer    10.956482
               Corporate    11.950637
               Home Office   12.173235
Office Supplies Consumer    12.901006
               Corporate    14.081442
               Home Office   14.450825
Technology      Consumer    29.443440
               Corporate    28.866648
               Home Office   31.308167
Name: Profit, dtype: float64
```

In [198]:

```
data.groupby(['Region','Segment'])['Profit'].mean()
```

Out[198]:

```
Region Segment
Central Consumer    7.850147
              Corporate 11.402589
              Home Office 14.086415
East   Consumer    14.991859
              Corporate 15.405824
              Home Office 16.704143
South  Consumer    19.570062
              Corporate 16.891246
              Home Office 17.586535
West   Consumer    19.497364
              Corporate 20.355286
              Home Office 19.908367
Name: Profit, dtype: float64
```

In [199]:

```
data.groupby(['State','Region'])['Profit'].mean()
```

Out[199]:

State	Region	
Alabama	South	31.459344
Arizona	West	3.534826
Arkansas	South	32.831357
California	West	22.389106
Colorado	West	3.166962
Connecticut	East	25.836225
Delaware	East	29.452662
District of Columbia	East	23.577815
Florida	South	5.146211
Georgia	South	32.394495
Idaho	West	25.098754
Illinois	Central	-2.519093
Indiana	Central	31.166837
Iowa	Central	25.679509
Kansas	Central	27.694375
Kentucky	South	32.396256
Louisiana	South	26.747586
Maine	East	38.273053
Maryland	East	30.640561
Massachusetts	East	28.126380
Michigan	Central	33.228138
Minnesota	Central	28.438335
Mississippi	South	33.568065
Missouri	Central	28.200556
Montana	West	28.132155
Nebraska	Central	23.893307
Nevada	West	30.709326
New Hampshire	East	31.815161
New Jersey	East	28.886825
New Mexico	West	24.656168
New York	East	24.509192
North Carolina	South	7.632957
North Dakota	Central	19.838825
Ohio	East	-1.743206
Oklahoma	Central	34.180472
Oregon	West	4.430817
Pennsylvania	East	-3.642170
Rhode Island	East	27.244355
South Carolina	South	26.162308
South Dakota	Central	24.288821
Tennessee	South	1.882866
Texas	Central	-1.774518
Utah	West	27.081522
Vermont	East	46.523825
Virginia	South	32.266628
Washington	West	24.160078
West Virginia	East	18.631587
Wisconsin	Central	34.608513
Wyoming	West	70.816875
Name: Profit, dtype: float64		

In [200]:

```
data.groupby(['Segment'])['Discount'].mean()
```

Out[200]:

Segment	
Consumer	0.138973
Corporate	0.139354
Home Office	0.129798
Name: Discount, dtype: float64	

In [201]:

```
data.groupby(['Region'])['Discount'].mean()
```

Out[201]:

Region	
Central	0.195669
East	0.132303
South	0.130957
West	0.103091
Name: Discount, dtype: float64	

In [202]:

```
data.groupby(['State'])['Discount'].mean()
```

Out[202]:

```
State
Alabama      0.000000
Arizona      0.266964
Arkansas     0.000000
California   0.072764
Colorado     0.272527
Connecticut  0.007317
Delaware     0.006250
District of Columbia  0.000000
Florida      0.264360
Georgia      0.000000
Idaho        0.085714
Illinois     0.317480
Indiana      0.000000
Iowa         0.000000
Kansas       0.000000
Kentucky     0.000000
Louisiana    0.000000
Maine        0.000000
Maryland     0.005714
Massachusetts 0.015556
Michigan     0.007059
Minnesota    0.000000
Mississippi  0.000000
Missouri     0.000000
Montana      0.066667
Nebraska     0.000000
Nevada       0.061538
New Hampshire 0.011111
New Jersey   0.004615
New Mexico   0.059459
New York     0.055319
North Carolina 0.254618
North Dakota 0.000000
Ohio         0.290405
Oklahoma     0.000000
Oregon       0.258065
Pennsylvania 0.292845
Rhode Island 0.021429
South Carolina 0.000000
South Dakota 0.000000
Tennessee    0.259563
Texas        0.301056
Utah         0.060377
Vermont      0.000000
Virginia     0.000000
Washington   0.064032
West Virginia 0.075000
Wisconsin    0.000000
Wyoming      0.200000
Name: Discount, dtype: float64
```

In [203]:

```
data.groupby(['Category','Segment'])['Discount'].mean()
```

Out[203]:

```
Category      Segment
Furniture      Consumer    0.168113
               Corporate    0.166796
               Home Office  0.158094
Office Supplies Consumer    0.130700
               Corporate    0.132857
               Home Office  0.121687
Technology      Consumer    0.132072
               Corporate    0.128700
               Home Office  0.125439
Name: Discount, dtype: float64
```

In [204]:

```
data.groupby(['Region','Segment'])['Discount'].mean()
```

```
Region Segment
Central Consumer    0.204587
      Corporate    0.195245
      Home Office  0.171644
East   Consumer    0.134241
      Corporate    0.130901
      Home Office  0.129084
South  Consumer    0.127804
      Corporate    0.138922
      Home Office  0.125735
West   Consumer    0.101166
      Corporate    0.108125
      Home Office  0.100263
Name: Discount, dtype: float64
```

In [205]:

```
data.groupby(['State','Region'])['Discount'].mean()
```

Out[205]:

```
State      Region
Alabama    South    0.000000
Arizona    West     0.266964
Arkansas   South    0.000000
California West     0.072764
Colorado   West     0.272527
Connecticut East     0.007317
Delaware    East     0.006250
District of Columbia East  0.000000
Florida     South    0.264360
Georgia     South    0.000000
Idaho       West     0.085714
Illinois    Central  0.317480
Indiana     Central  0.000000
Iowa        Central  0.000000
Kansas      Central  0.000000
Kentucky    South    0.000000
Louisiana   South    0.000000
Maine       East     0.000000
Maryland    East     0.005714
Massachusetts East    0.015556
Michigan    Central  0.007059
Minnesota   Central  0.000000
Mississippi South    0.000000
Missouri    Central  0.000000
Montana     West     0.066667
Nebraska    Central  0.000000
Nevada       West     0.061538
New Hampshire East    0.011111
New Jersey  East     0.004615
New Mexico  West     0.059459
New York    East     0.055319
North Carolina South   0.254618
North Dakota Central   0.000000
Ohio        East     0.290405
Oklahoma    Central  0.000000
Oregon      West     0.258065
Pennsylvania East    0.292845
Rhode Island East     0.021429
South Carolina South    0.000000
South Dakota Central   0.000000
Tennessee   South    0.259563
Texas       Central  0.301056
Utah        West     0.060377
Vermont     East     0.000000
Virginia    South    0.000000
Washington  West     0.064032
West Virginia East     0.075000
Wisconsin    Central  0.000000
Wyoming     West     0.200000
Name: Discount, dtype: float64
```

In [206]:

```
data.groupby('Category')['Sub-Category'].nunique()
```

Out[206]:

```
Category
Furniture      4
Office Supplies 9
Technology     4
Name: Sub-Category, dtype: int64
```

In [207]:



data.groupby('Category')['Sub-Category'].unique()

Category  
Furniture [Bookcases, Chairs, Tables, Furnishings]  
Office Supplies [Labels, Storage, Art, Binders, Appliances, Pa...  
Technology [Phones, Accessories, Machines, Copiers]  
Name: Sub-Category, dtype: object

Out[207]:

data.groupby('Ship Mode')['City'].nunique()

In [208]:

Ship Mode  
First Class 236  
Same Day 123  
Second Class 263  
Standard Class 454  
Name: City, dtype: int64

Out[208]:

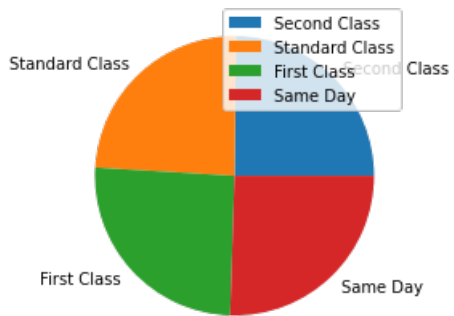
In [209]:

no\_of\_species=data.groupby('Ship Mode')['Quantity'].mean()  
print(no\_of\_species)  
plt.pie(no\_of\_species,labels=[x for x in data['Ship Mode'].unique()])  
plt.legend()

Ship Mode  
First Class 3.681404  
Same Day 3.591160  
Second Class 3.774550  
Standard Class 3.779742  
Name: Quantity, dtype: float64

Out[209]:

<matplotlib.legend.Legend at 0x1c552706bb0>



no\_of\_species=data.groupby('Category')['Sales'].mean()  
plt.pie(no\_of\_species,labels=[x for x in data['Category'].unique()])

In [210]:

Out[210]:

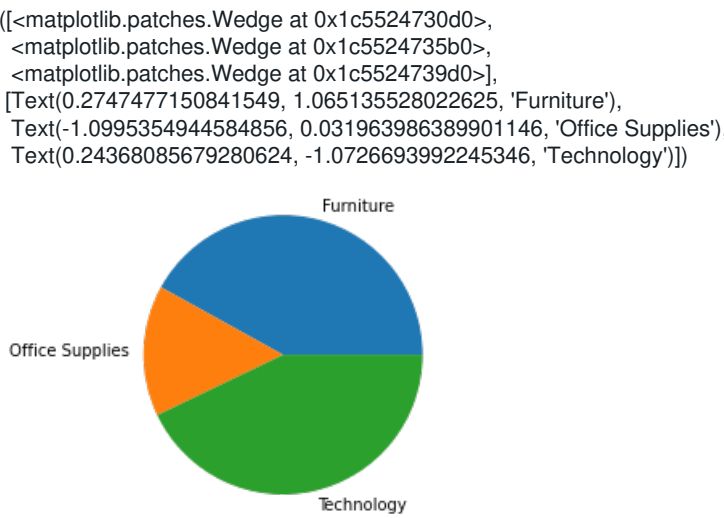
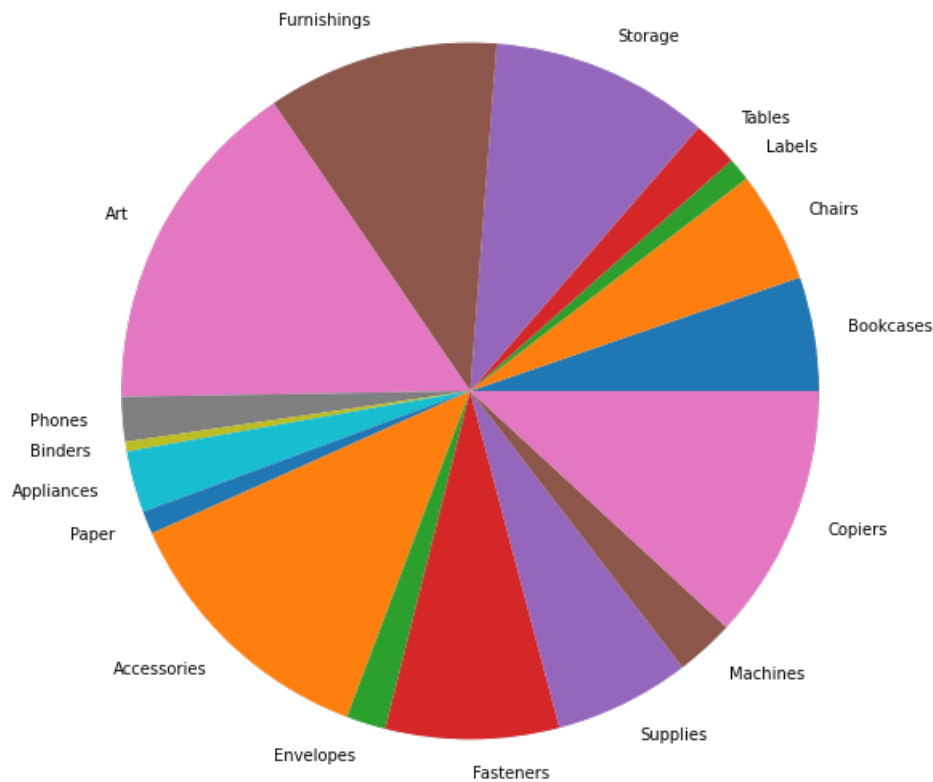


fig = plt.figure(figsize = (10, 10))  
no\_of\_species=data.groupby('Sub-Category')['Sales'].mean()  
plt.pie(no\_of\_species,labels=[x for x in data['Sub-Category'].unique()])

In [211]:

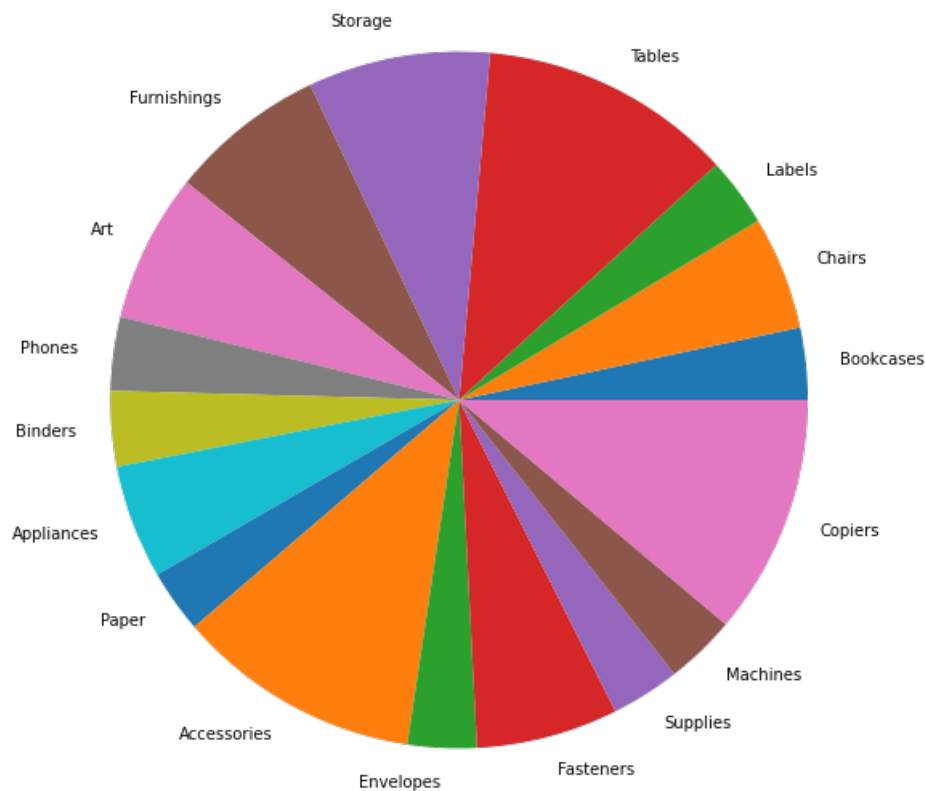
```
[(<matplotlib.patches.Wedge at 0x1c5533c1f10>,
<matplotlib.patches.Wedge at 0x1c553391430>,
<matplotlib.patches.Wedge at 0x1c553391850>,
<matplotlib.patches.Wedge at 0x1c553391c70>,
<matplotlib.patches.Wedge at 0x1c5533b00d0>,
<matplotlib.patches.Wedge at 0x1c5533b04f0>,
<matplotlib.patches.Wedge at 0x1c5533b0910>,
<matplotlib.patches.Wedge at 0x1c5533b0d30>,
<matplotlib.patches.Wedge at 0x1c5533ad190>,
<matplotlib.patches.Wedge at 0x1c5533ad5b0>,
<matplotlib.patches.Wedge at 0x1c552474c70>,
<matplotlib.patches.Wedge at 0x1c5533addc0>,
<matplotlib.patches.Wedge at 0x1c552323220>,
<matplotlib.patches.Wedge at 0x1c552323640>,
<matplotlib.patches.Wedge at 0x1c552323a60>,
<matplotlib.patches.Wedge at 0x1c552323e80>,
<matplotlib.patches.Wedge at 0x1c5523422e0>],
[Text(1.0848643521114285, 0.18184976633982983, 'Bookcases'),
Text(0.9684968349353147, 0.5215494997795301, 'Chairs'),
Text(0.8487690162328316, 0.6997079084040364, 'Labels'),
Text(0.7742439618458892, 0.7813746140905663, 'Tables'),
Text(0.42332844514659945, 1.0152797779576637, 'Storage'),
Text(-0.2841591457864491, 1.0626634367785108, 'Furnishings'),
Text(-0.9768219042391583, 0.5057854954410861, 'Art'),
Text(-1.0963426875392397, -0.08962539528078672, 'Phones'),
Text(-1.0858214086312397, -0.17604507535875724, 'Binders'),
Text(-1.0617353748123413, -0.28760736060138853, 'Appliances'),
Text(-1.0183391431792777, -0.4159151229143931, 'Paper'),
Text(-0.7540600916598542, -0.8008703878693059, 'Accessories'),
Text(-0.3289066416797882, -1.0496763410970658, 'Envelopes'),
Text(0.006973319509409078, -1.0999778965120253, 'Fasteners'),
Text(0.4849562356689632, -0.9873284405332348, 'Supplies'),
Text(0.7416667123045899, -0.8123610575718785, 'Machines'),
Text(1.0245396868906158, -0.4003978396371279, 'Copiers'))]
```



In [212]:

```
fig = plt.figure(figsize = (10, 10))
no_of_species=data.groupby('Sub-Category')['Discount'].mean()
plt.pie(no_of_species,labels=[x for x in data['Sub-Category'].unique()])
```

```
[<matplotlib.patches.Wedge at 0x1c5526f7eb0>,
<matplotlib.patches.Wedge at 0x1c552706af0>,
<matplotlib.patches.Wedge at 0x1c552706dc0>,
<matplotlib.patches.Wedge at 0x1c552704e80>,
<matplotlib.patches.Wedge at 0x1c552704400>,
<matplotlib.patches.Wedge at 0x1c5527049a0>,
<matplotlib.patches.Wedge at 0x1c5533db370>,
<matplotlib.patches.Wedge at 0x1c5533dbd90>,
<matplotlib.patches.Wedge at 0x1c5533db760>,
<matplotlib.patches.Wedge at 0x1c5533db190>,
<matplotlib.patches.Wedge at 0x1c5533fb490>,
<matplotlib.patches.Wedge at 0x1c5533fb2b0>,
<matplotlib.patches.Wedge at 0x1c5533fb280>,
<matplotlib.patches.Wedge at 0x1c5533d82b0>,
<matplotlib.patches.Wedge at 0x1c5533d8b20>,
<matplotlib.patches.Wedge at 0x1c5533d8e80>,
<matplotlib.patches.Wedge at 0x1c5526f6280>],
[Text(1.0939376237955085, 0.11532768637511329, 'Bookcases'),
Text(1.023446952708239, 0.4031827563180496, 'Chairs'),
Text(0.8818274467142084, 0.6575563506054064, 'Labels'),
Text(0.48798167001989584, 0.9858366445434018, 'Tables'),
Text(-0.1921988295527702, 1.0830787644112247, 'Storage'),
Text(-0.6818306456547755, 0.8631957892888451, 'Furnishings'),
Text(-0.9867190917490882, 0.48619485186276357, 'Art'),
Text(-1.0901816819391679, 0.14664208251483382, 'Phones'),
Text(-1.096154051174169, -0.09190373275584306, 'Binders'),
Text(-1.029720705309429, -0.38687888163746076, 'Appliances'),
Text(-0.8970046770647812, -0.6366966383796193, 'Paper'),
Text(-0.5324872897186954, -0.962526511992287, 'Accessories'),
Text(-0.053644151457198974, -1.098691178181766, 'Envelopes'),
Text(0.281099616706148, -1.0634768476500356, 'Fasteners'),
Text(0.5892924576130009, -0.9288349688725277, 'Supplies'),
Text(0.764374825285323, -0.7910316848711132, 'Machines'),
Text(1.0333753664351273, -0.37700842436882853, 'Copiers')]]
```



In [213]:

```
sns.heatmap(df_outlier_removed.corr(),xticklabels=df_outlier_removed.corr().columns, yticklabels=df_outlier_removed.corr().columns,annot=True)
```

<AxesSubplot:>



In [ ]: