

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

df = pd.read_excel(r"C:\Users\rishv\OneDrive\Desktop\Projects\
Supertore_Usa_Python_DA.xls\Superstore_USA.xlsx")

pip install openpyxl

Requirement already satisfied: openpyxl in c:\users\rishv\appdata\
local\programs\python\python313\lib\site-packages (3.1.5)
Requirement already satisfied: et-xmlfile in c:\users\rishv\appdata\
local\programs\python\python313\lib\site-packages (from openpyxl)
(2.0.0)
Note: you may need to restart the kernel to use updated packages.
```

```
df
```

```
df.shape
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 9426 entries, 0 to 9425
```

```
Data columns (total 24 columns):
```

#	Column	Non-Null Count	Dtype
0	Row ID	9426 non-null	int64
1	Order Priority	9426 non-null	object
2	Discount	9426 non-null	float64
3	Unit Price	9426 non-null	float64
4	Shipping Cost	9426 non-null	float64
5	Customer ID	9426 non-null	int64
6	Customer Name	9426 non-null	object
7	Ship Mode	9426 non-null	object
8	Customer Segment	9426 non-null	object
9	Product Category	9426 non-null	object
10	Product Sub-Category	9426 non-null	object
11	Product Container	9426 non-null	object
12	Product Name	9426 non-null	object
13	Product Base Margin	9354 non-null	float64
14	Region	9426 non-null	object
15	State or Province	9426 non-null	object
16	City	9426 non-null	object
17	Postal Code	9426 non-null	int64
18	Order Date	9426 non-null	datetime64[ns]
19	Ship Date	9426 non-null	datetime64[ns]

```

20 Profit 9426 non-null float64
21 Quantity ordered new 9426 non-null int64
22 Sales 9426 non-null float64
23 Order ID 9426 non-null int64
dtypes: datetime64[ns](2), float64(6), int64(5), object(11)
memory usage: 1.7+ MB

df.head()

```

Null_Values_Check

```

df.isnull().sum()

Row ID 0
Order Priority 0
Discount 0
Unit Price 0
Shipping Cost 0
Customer ID 0
Customer Name 0
Ship Mode 0
Customer Segment 0
Product Category 0
Product Sub-Category 0
Product Container 0
Product Name 0
Product Base Margin 72
Region 0
State or Province 0
City 0
Postal Code 0
Order Date 0
Ship Date 0
Profit 0
Quantity ordered new 0
Sales 0
Order ID 0
dtype: int64

#df['Product Base Margin'].fillna(df['Product Base Margin'].mean(),
inplace=True)

```

Typing_Error/Order priority count

```

df['Order Priority'].value_counts()

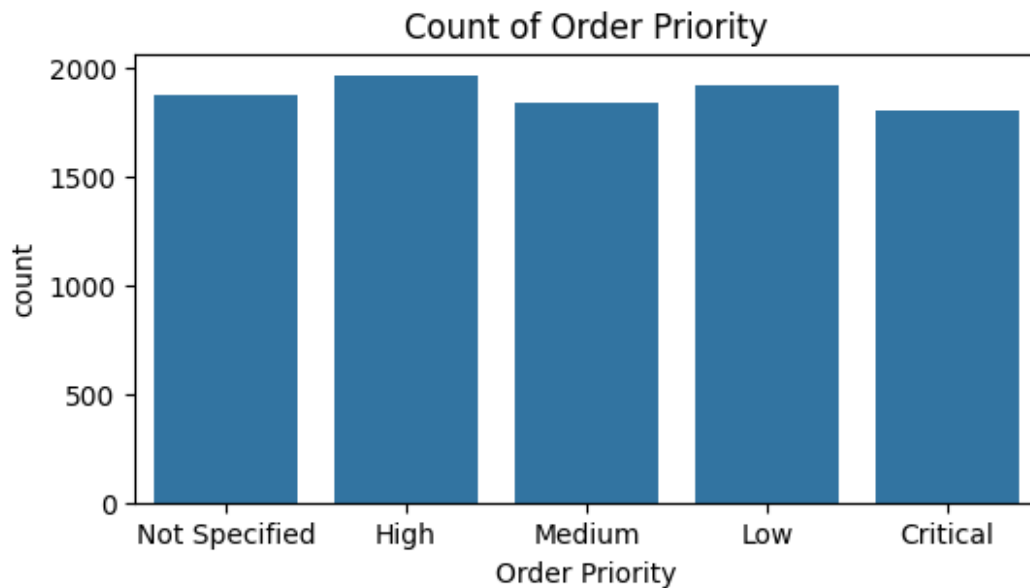
df['Order Priority'].unique()

```

```
df["Order Priority"] = df['Order Priority'].replace( 'Critical ',  
'Critical')
```

```
plt.figure(figsize=(6,3))  
plt.title('Count of Order Priority')  
sns.countplot(x='Order Priority',data = df)
```

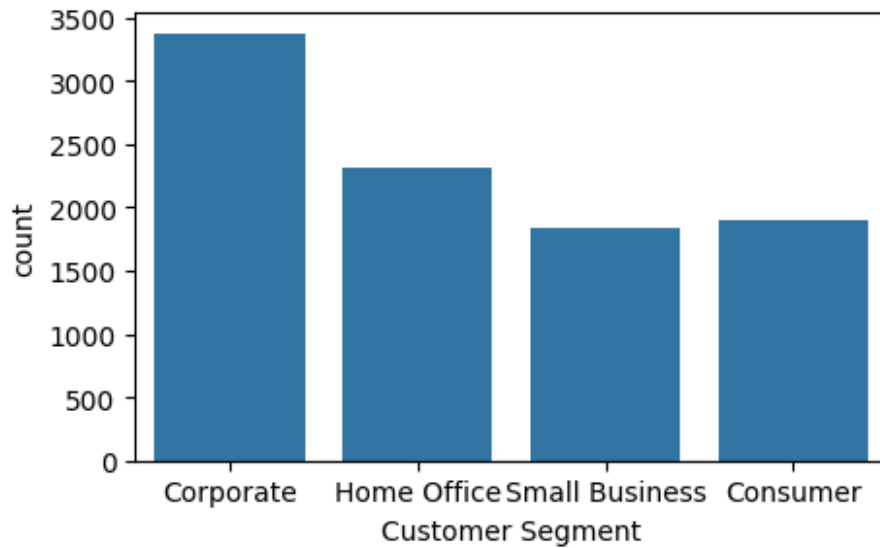
```
<Axes: title={'center': 'Count of Order Priority'}, xlabel='Order  
Priority', ylabel='count'>
```



Customer Segements

```
plt.figure(figsize=(5,3))  
sns.countplot(x='Customer Segment',data=df)
```

```
<Axes: xlabel='Customer Segment', ylabel='count'>
```



Shipping_Mode

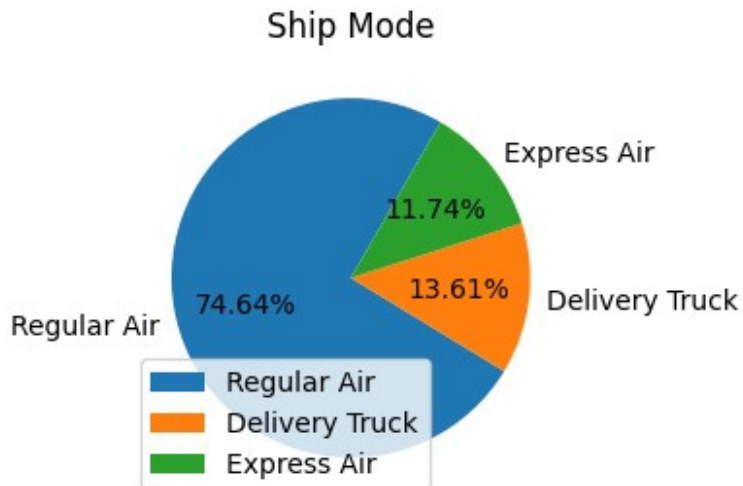
```
df['Ship Mode'].value_counts()
```

```
Ship Mode
Regular Air      7036
Delivery Truck   1283
Express Air      1107
Name: count, dtype: int64
```

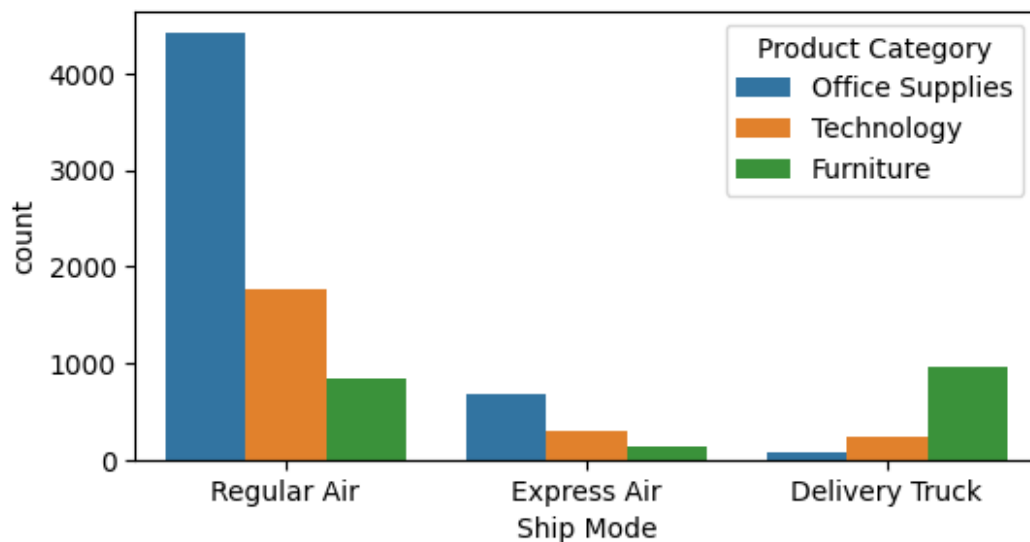
```
x = df['Ship Mode'].value_counts().index
y = df['Ship Mode'].value_counts().values
```

```
plt.figure(figsize=(6,3))
plt.title('Ship Mode')
plt.pie(y,labels=x,startangle = 60, autopct='%0.2f%')
plt.legend(loc=3)
```

```
<matplotlib.legend.Legend at 0x25f77885590>
```

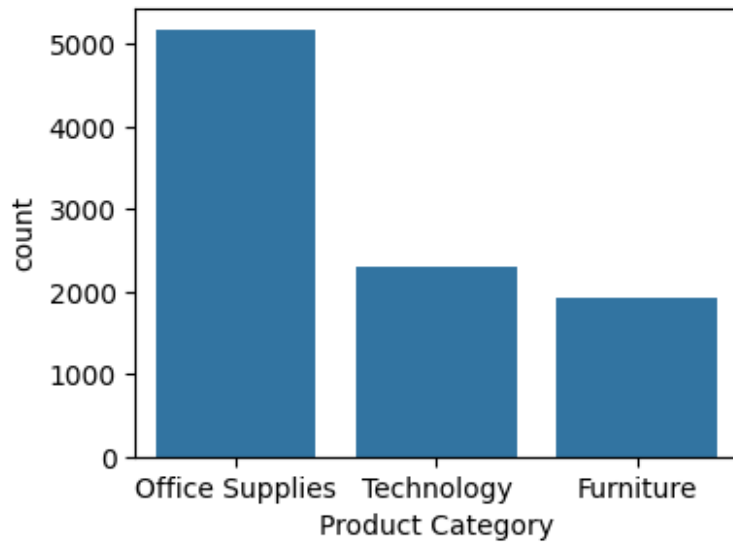


```
plt.figure(figsize=(6,3))  
sns.countplot(x='Ship Mode', data=df, hue='Product Category')  
<Axes: xlabel='Ship Mode', ylabel='count'>
```

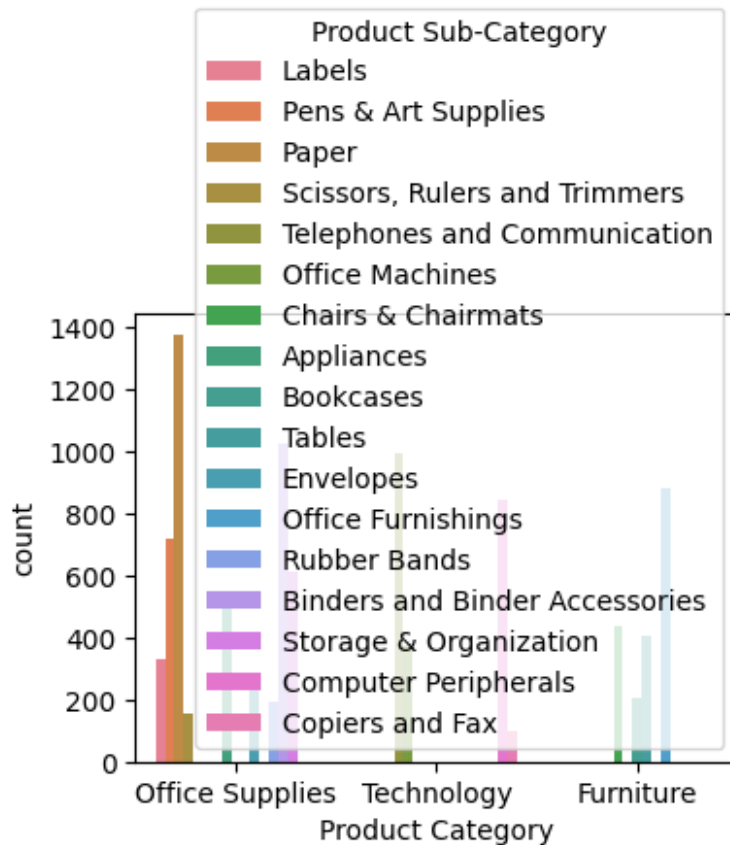


Product Category Insight

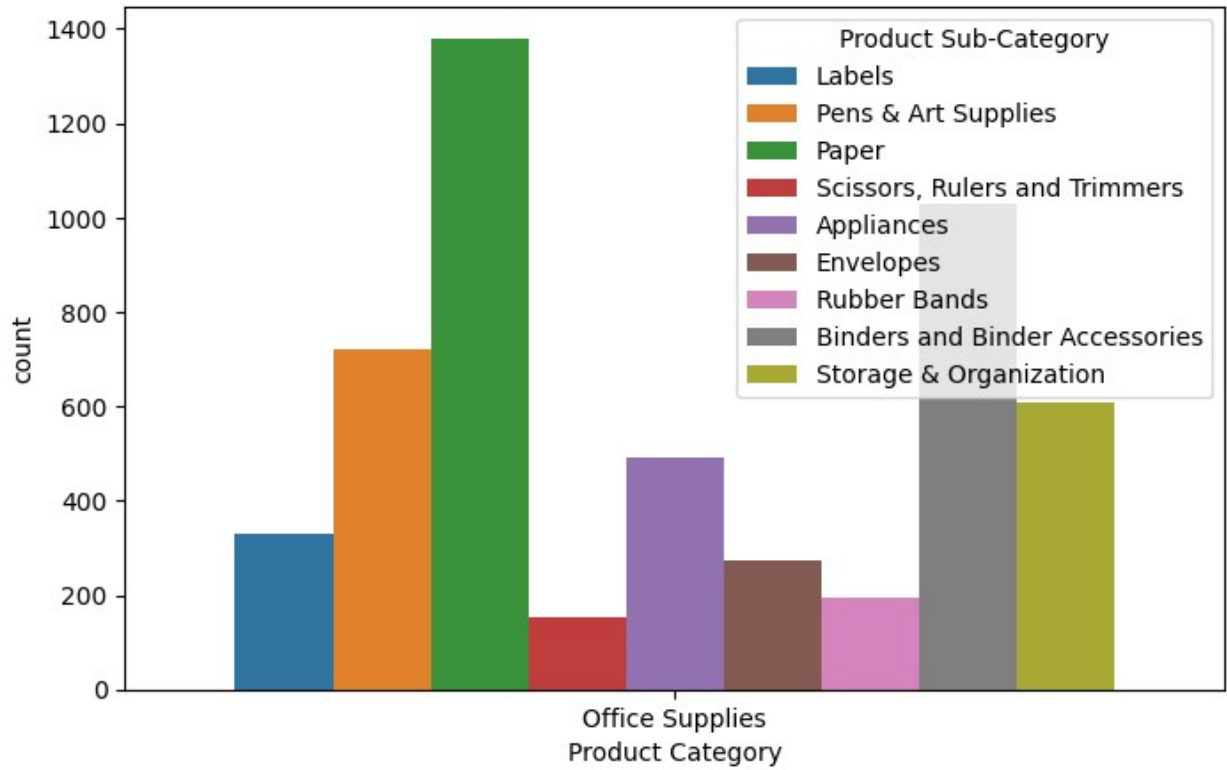
```
plt.figure(figsize=(4,3))  
sns.countplot(x='Product Category', data=df)  
<Axes: xlabel='Product Category', ylabel='count'>
```



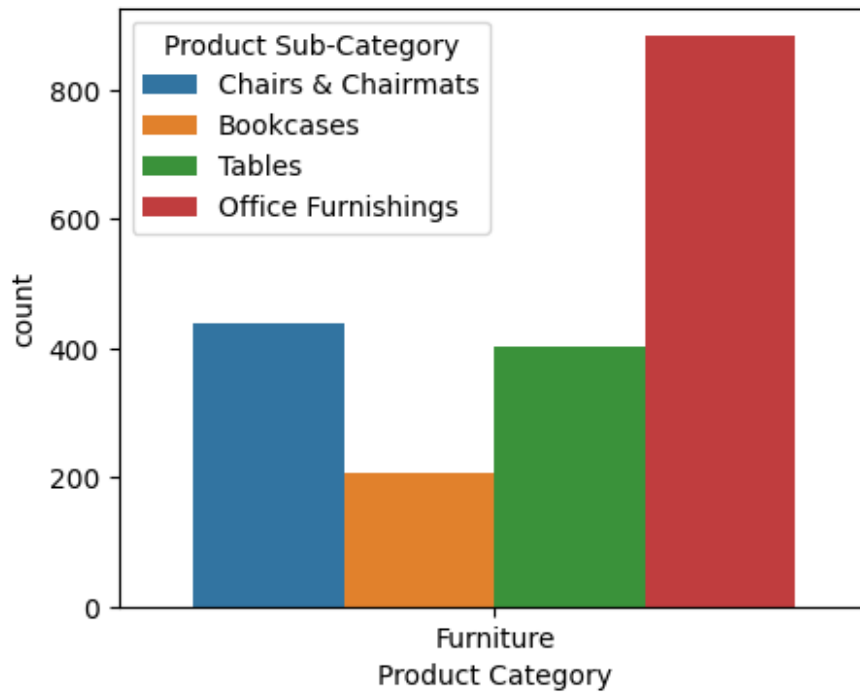
```
plt.figure(figsize=(4,3))
sns.countplot(x='Product Category',data=df, hue = 'Product Sub-
Category')
#here no. of subcategories are quite high that is why we will
partially vizualise these subcategories according to the Product
categories
<Axes: xlabel='Product Category', ylabel='count'>
```



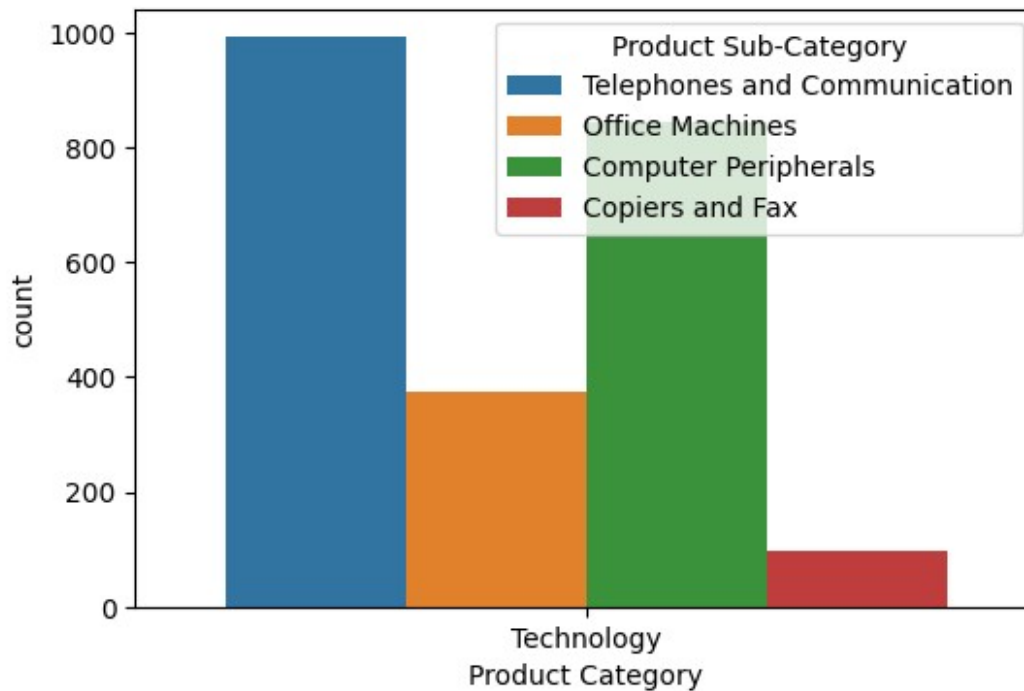
```
plt.figure(figsize=(8,5))
sns.countplot(x='Product Category',data=df[df['Product
Category']=='Office Supplies'], hue = 'Product Sub-Category')
<Axes: xlabel='Product Category', ylabel='count'>
```



```
plt.figure(figsize=(5,4))
sns.countplot(x='Product Category',data=df[df['Product
Category']=='Furniture'], hue = 'Product Sub-Category')
<Axes: xlabel='Product Category', ylabel='count'>
```

```
plt.figure(figsize=(6,4))
sns.countplot(x='Product Category',data=df[df['Product
Category']=='Technology'], hue = 'Product Sub-Category')
<Axes: xlabel='Product Category', ylabel='count'>
```



Year wise Order Insight

```
df['Order Year'] = df['Order Date'].dt.year
df['Order Year'].value_counts()

Order Year
2013      3054
2012      2241
2011      2179
2010      1952
Name: count, dtype: int64

#df.info()
```

Profit

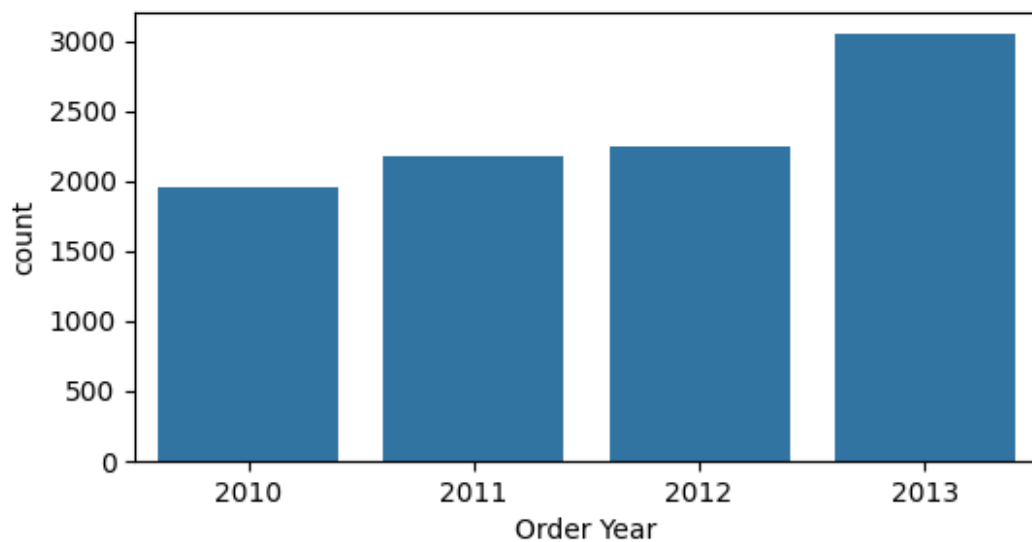
```
df['Profit'].value_counts()

Profit
0      109
-2      66
5       60
-5      59
2       58
...
652      1
598      1
-699      1
-340      1
462      1
Name: count, Length: 2201, dtype: int64

df['Profit'] = df['Profit'].astype(float)

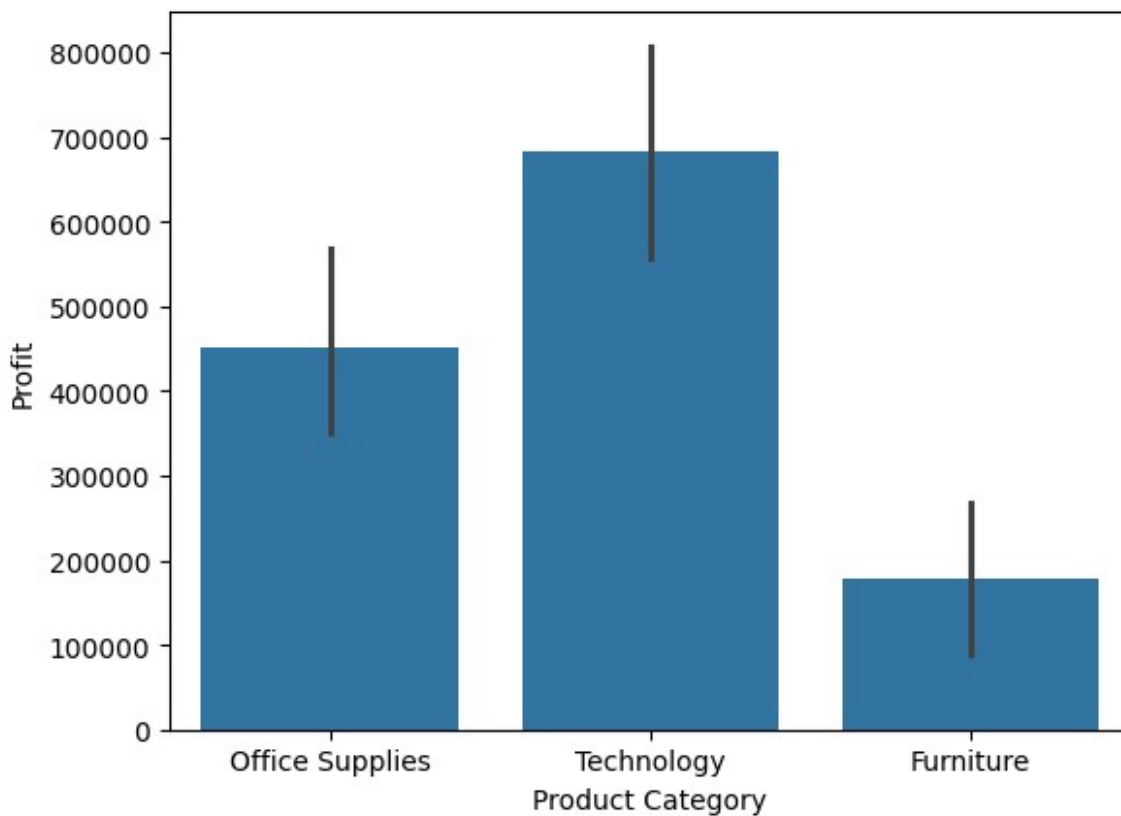
plt.figure(figsize=(6,3))
sns.countplot(x='Order Year', data=df)

<Axes: xlabel='Order Year', ylabel='count'>
```



```
sns.barplot(x= 'Product Category', y= 'Profit', data = df, estimator= 'sum')
```

```
<Axes: xlabel='Product Category', ylabel='Profit'>
```



Analysing profit from different localities

```
df['State or Province'].value_counts()[:5] #top 5 states of profit making
```

```
State or Province
California      1021
Texas           646
Illinois        584
New York        574
Florida         522
Name: count, dtype: int64
```

```
df['State or Province'].value_counts()[-5:] #bottom 5 states in profit making list
```

```
State or Province
North Dakota     34
South Dakota     28
Wyoming          21
Rhode Island     20
Delaware         15
Name: count, dtype: int64
```

```
plt.figure(figsize=(4,3))
sns.barplot(x='Product Category',y= 'Product Base Margin',
data=df,estimator= np.sum) #Base Margin of Product Categories
```

```
<Axes: xlabel='Product Category', ylabel='Product Base Margin'>
```

