superstore-usa

February 4, 2025

[5]:
```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
```

1 Import daset

[23]:
```
dataset = pd.read_excel("Superstore_USA.xlsx")
dataset.head()
```

[23]:

	Row ID	Order Priority	Discount	Unit Price	Shipping Cost	Customer ID	\
0	18606	Not Specified	0.01	2.88	0.50	2	
1	20847	High	0.01	2.84	0.93	3	
2	23086	Not Specified	0.03	6.68	6.15	3	
3	23087	Not Specified	0.01	5.68	3.60	3	
4	23088	Not Specified	0.00	205.99	2.50	3	

	Customer Name	Ship Mode	Customer Segment	Product Category	…	\
0	Janice Fletcher	Regular Air	Corporate	Office Supplies	…	
1	Bonnie Potter	Express Air	Corporate	Office Supplies	…	
2	Bonnie Potter	Express Air	Corporate	Office Supplies	…	
3	Bonnie Potter	Regular Air	Corporate	Office Supplies	…	
4	Bonnie Potter	Express Air	Corporate	Technology	…	

	Region	State or Province	City	Postal Code	Order Date	Ship Date	\
0	Central	Illinois	Addison	60101	2012-05-28	2012-05-30	
1	West	Washington	Anacortes	98221	2010-07-07	2010-07-08	
2	West	Washington	Anacortes	98221	2011-07-27	2011-07-28	
3	West	Washington	Anacortes	98221	2011-07-27	2011-07-28	
4	West	Washington	Anacortes	98221	2011-07-27	2011-07-27	

	Profit	Quantity ordered new	Sales	Order ID
0	1.3200	2	5.90	88525
1	4.5600	4	13.01	88522
2	-47.6400	7	49.92	88523

```
3  -30.5100                    7    41.64    88523
4  998.2023                    8  1446.67    88523

[5 rows x 24 columns]
```

[26]:
```
dataset.isnull().sum()
```

[26]:
```
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     0
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
```

[43]:
```
dataset.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    9426 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
```

[25]:
```
dataset["Product Base Margin"].fillna(dataset["Product Base Margin"].
↪mean(),inplace = True)
```

```
C:\Users\admin\AppData\Local\Temp\ipykernel_4928\1514211421.py:1: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained
assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


  dataset["Product Base Margin"].fillna(dataset["Product Base
Margin"].mean(),inplace = True)
```

[27]:
```
dataset["Order Priority"].value_counts()
```

[27]:
```
Order Priority
High             1970
Low              1926
Not Specified    1881
Medium           1844
Critical         1804
Critical            1
```

```
Name: count, dtype: int64
```

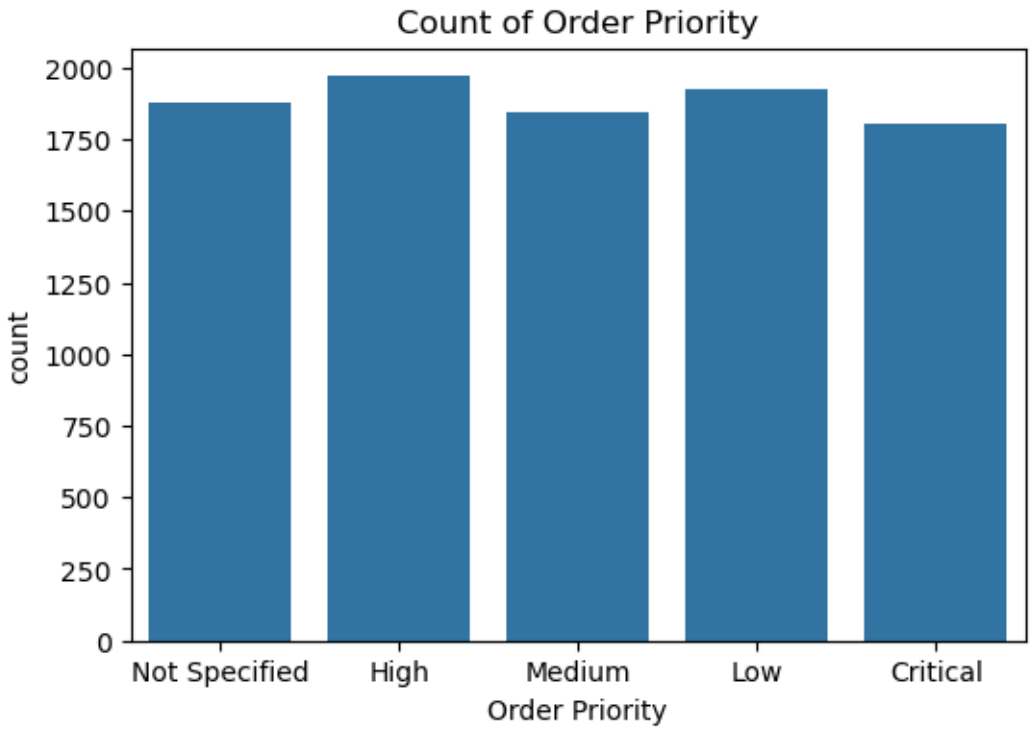
[32]:
```
dataset["Order Priority"].unique()
```

[32]:
```
array(['Not Specified', 'High', 'Medium', 'Low', 'Critical'], dtype=object)
```

[31]:
```
dataset["Order Priority"] = dataset["Order Priority"].replace('Critical␣
↪','Critical')
```

2 Order Priority

[37]:
```
plt.figure(figsize = (6,4))
sns.countplot(x = "Order Priority",data = dataset)

plt.title("Count of Order Priority")
plt.show()
```

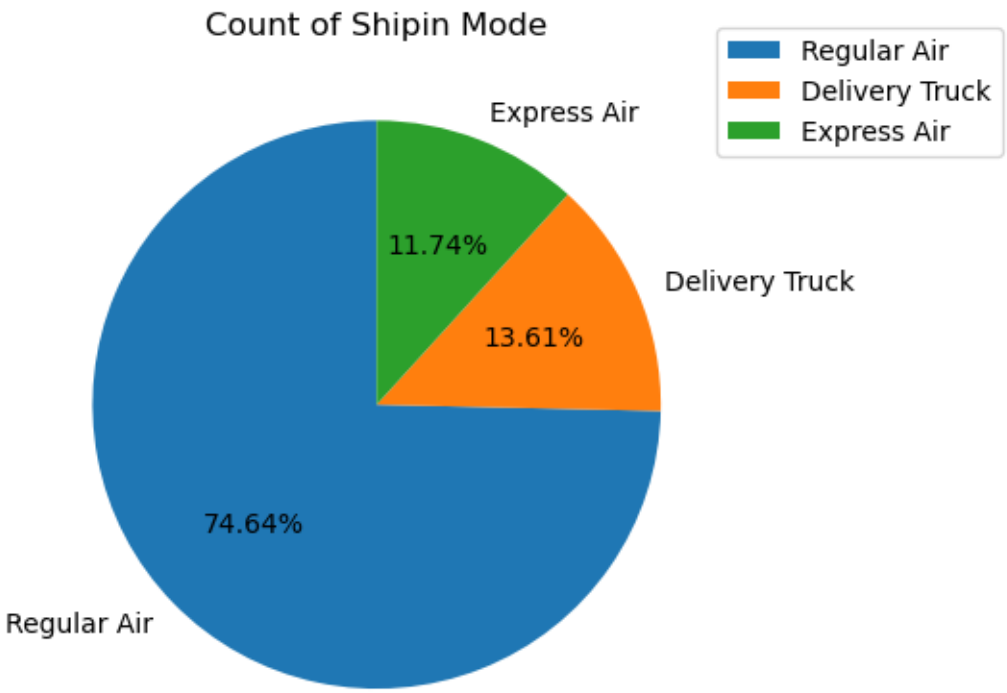


2.0.1 Shiping Mode

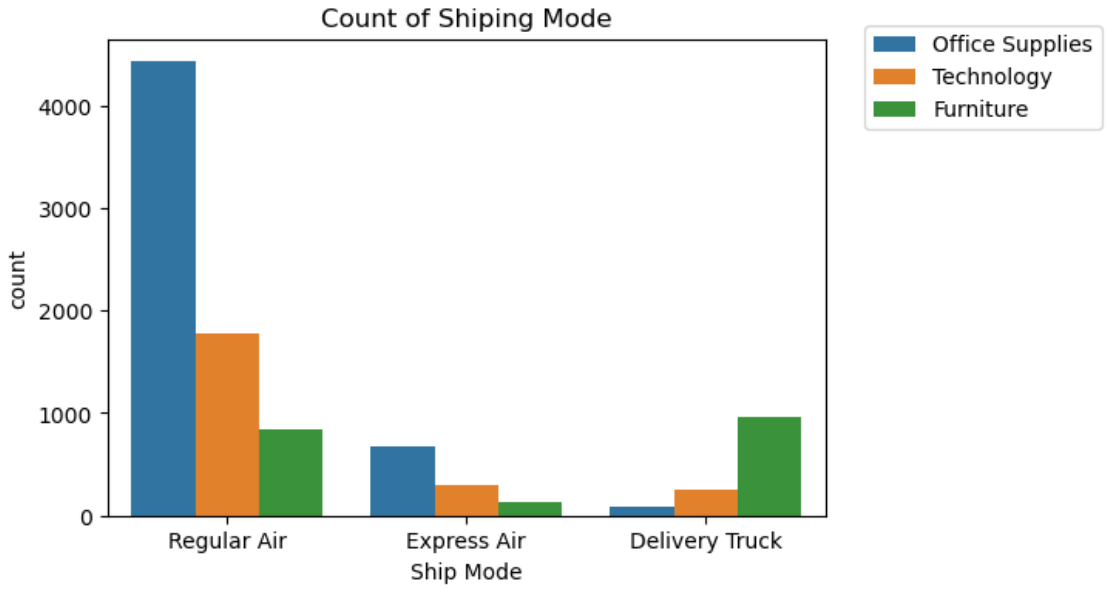
[40]:
```
dataset['Ship Mode'].value_counts()
```

[40]:
```
Ship Mode
Regular Air        7036
Delivery Truck     1283
Express Air        1107
Name: count, dtype: int64
```

[66]:
```
x = dataset["Ship Mode"].value_counts().index
y = dataset["Ship Mode"].value_counts().values
plt.pie(y,labels = x,startangle=90,autopct = "%0.2f%%")

plt.title("Count of Shipin Mode")
plt.legend(loc = "upper right",bbox_to_anchor=(1.4, 1.05))
plt.show()
```

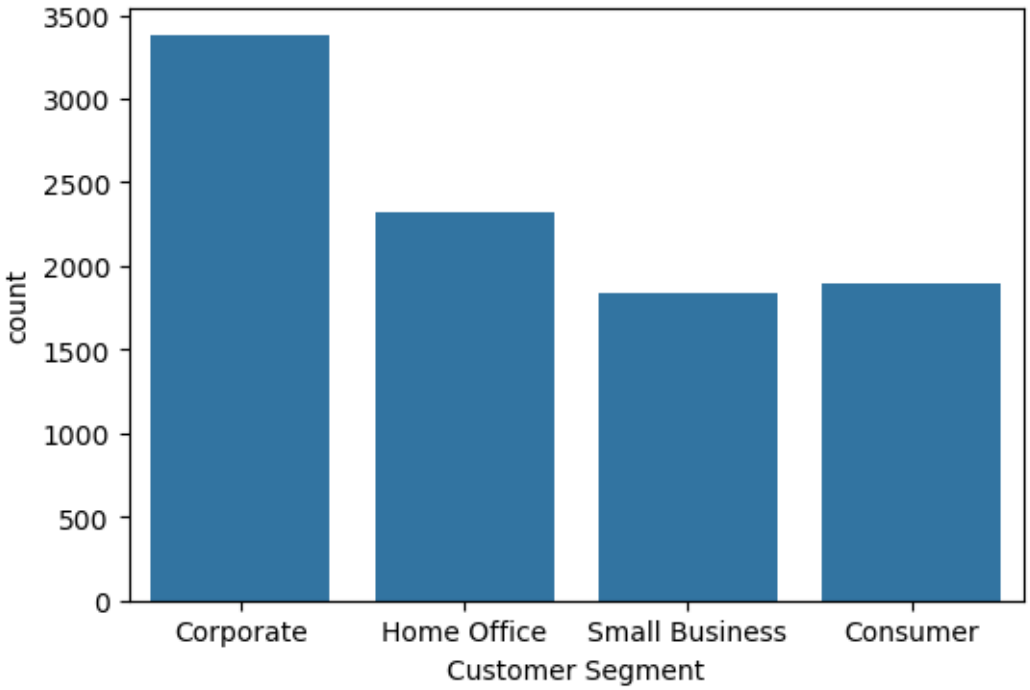


[80]:
```
plt.figure(figsize = (6,4))
sns.countplot(x = "Ship Mode",data = dataset,hue = "Product Category")

plt.title("Count of Shiping Mode")
plt.legend(loc = "upper right",bbox_to_anchor=(1.4, 1.05))
plt.show()
```



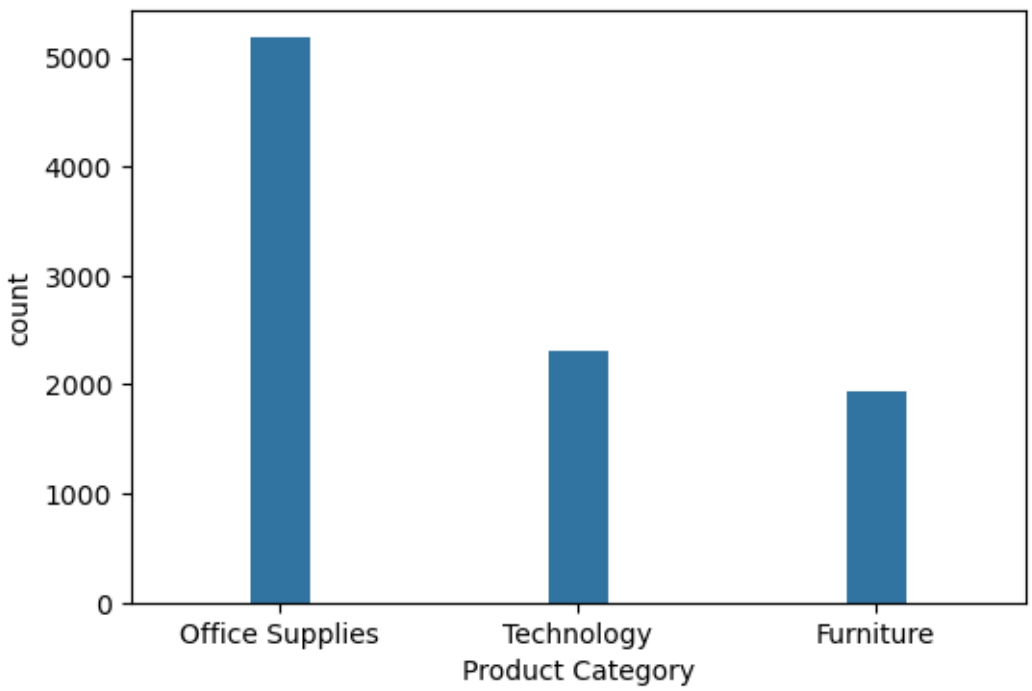
2.0.2 Customer Segment

[78]:
```
plt.figure(figsize = (6,4))
sns.countplot(x = "Customer Segment",data = dataset)

# plt.title("Count of Shiping Mode")
# plt.legend(loc = "upper right",bbox_to_anchor=(1.4, 1.05))
plt.show()
```



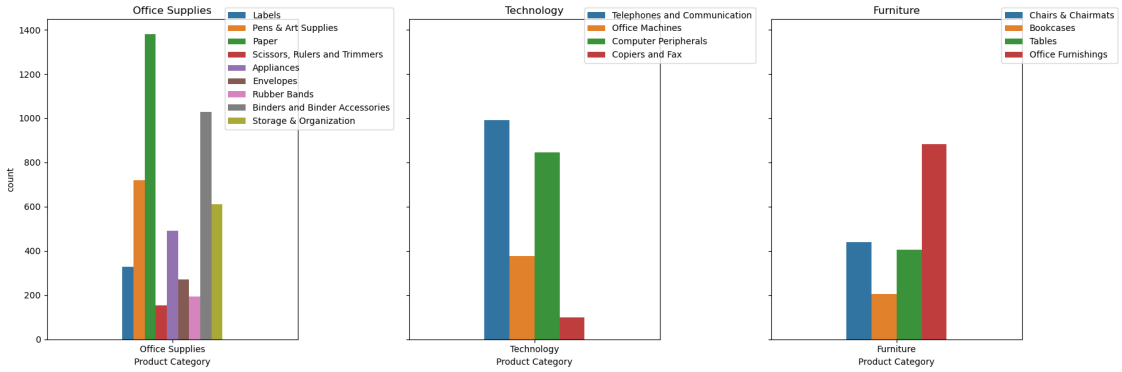
2.0.3 Product Category

[85]:
```
plt.figure(figsize = (6,4))
sns.countplot(x = "Product Category",data = dataset,width = 0.2)

# plt.title("Count of Shiping Mode")
# plt.legend(loc = "upper right",bbox_to_anchor=(1.4, 1.05))
plt.show()
```



[97]:
```
# Set up a single row with 3 subplots
fig, axes = plt.subplots(1, 3, figsize=(18, 6), sharey=True)  # Sharey  =
 ↪Diffrent scale for  each axis

# Office Supplies
sns.countplot(x="Product Category",data=dataset[dataset["Product Category"] ==
 ↪"Office Supplies"],width=0.4,hue="Product Sub-Category",
              ax=axes[0])
axes[0].set_title("Office Supplies")
axes[0].legend(loc="upper right", bbox_to_anchor=(1.4, 1.05))

# Technology
sns.countplot(x="Product Category",
              data=dataset[dataset["Product Category"] == "Technology"],
              width=0.4,
              hue="Product Sub-Category",
              ax=axes[1])
axes[1].set_title("Technology")
axes[1].legend(loc="upper right", bbox_to_anchor=(1.4, 1.05))

# Furniture
sns.countplot(x="Product Category",
              data=dataset[dataset["Product Category"] == "Furniture"],
              width=0.4,
```

```
              hue="Product Sub-Category",
              ax=axes[2])
axes[2].set_title("Furniture")
axes[2].legend(loc="upper right", bbox_to_anchor=(1.4, 1.05))

# Adjust layout and show plot
plt.tight_layout()
plt.show()
```



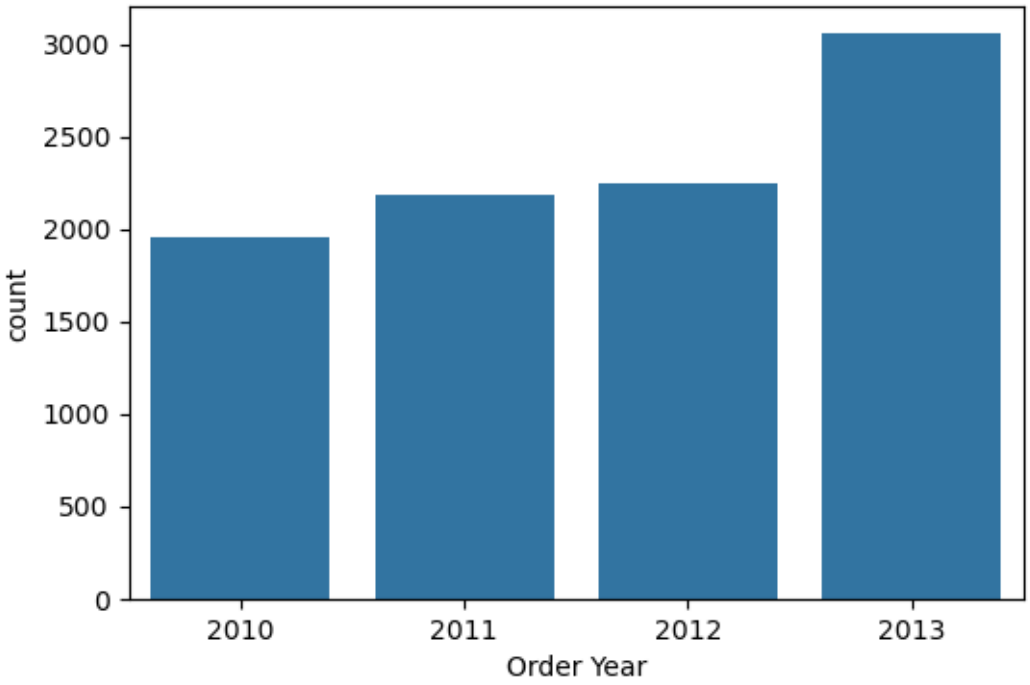
2.0.4 Order Data

[106]:
```
dataset["Order Year"] = dataset["Order Date"].dt.year
```

[109]:
```
dataset["Order Year"].value_counts()
```

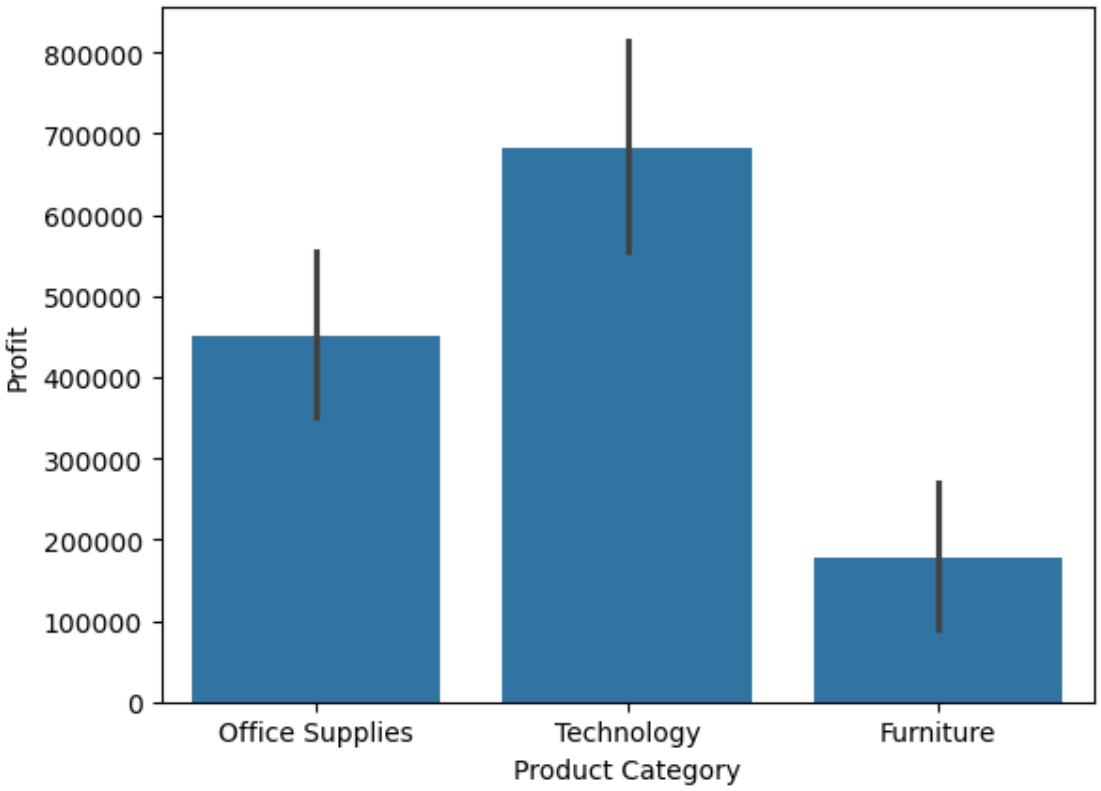
[109]:
```
Order Year
2013    3054
2012    2241
2011    2179
2010    1952
Name: count, dtype: int64
```

[110]:
```
plt.figure(figsize = (6,4))
sns.countplot(x = "Order Year",data = dataset)
plt.show()
```



2.0.5 Profit

[113]:
```
sns.barplot(x = "Product Category",y = "Profit",data = dataset,estimator=sum)
plt.show()
```



2.0.6 State or Provience

[119]:
```
dataset["State or Province"].value_counts()
```

[119]:
```
State or Province
California              1021
Texas                    646
Illinois                 584
New York                 574
Florida                  522
Ohio                     396
Washington               327
Michigan                 327
Pennsylvania             271
North Carolina           251
Indiana                  241
Minnesota                239
Massachusetts            222
Georgia                  214
Virginia                 198
Maryland                 178
```

```
Colorado                177
New Jersey              177
Wisconsin               169
Oregon                  168
Tennessee               166
Missouri                161
Iowa                    156
Utah                    146
Arizona                 134
Kansas                  133
Maine                   128
Alabama                 125
Arkansas                123
Idaho                   114
South Carolina          105
Oklahoma                104
Louisiana                89
New Mexico               84
Kentucky                 83
Connecticut              82
Mississippi              78
Nebraska                 77
District of Columbia     68
Vermont                  61
New Hampshire            54
Montana                  49
West Virginia            43
Nevada                   43
North Dakota             34
South Dakota             28
Wyoming                  21
Rhode Island             20
Delaware                 15
Name: count, dtype: int64
```

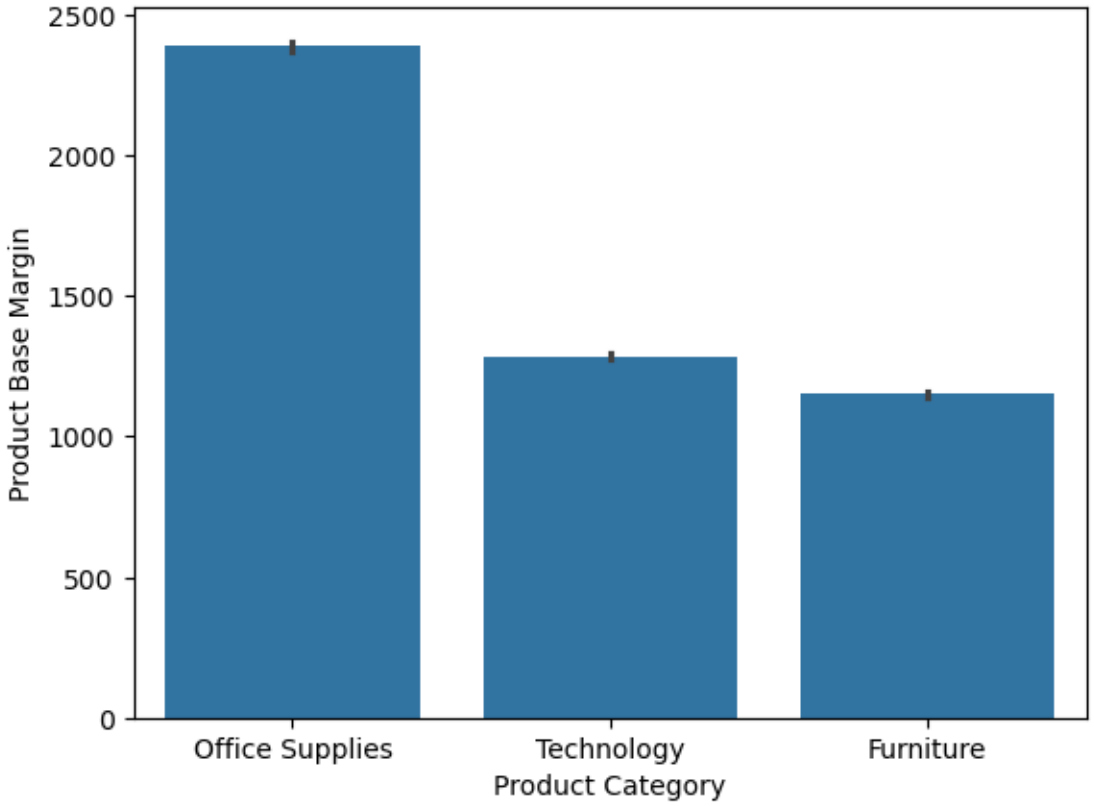
[124]:
```
dataset["State or Province"].value_counts()[:-5:-1]
```

[124]:
```
State or Province
Delaware        15
Rhode Island    20
Wyoming         21
South Dakota    28
Name: count, dtype: int64
```

[ ]:
```

```

2.0.7 Product Base Margin

[118]:
```
sns.barplot(x = "Product Category",y = "Product Base Margin",data =␣
↪dataset,estimator=sum)
plt.show()
```



[ ]:
```

```