erce-shipping-analysis-eda-project

September 20, 2024

[3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

[14]:
```
df = pd.read_csv("H:\\Data Aalystics\\Projects\\Python\\E-Commerce Shipping␣
↪analysis\\E-Commerce Shipping analysis.csv")
```

[15]:
```
print(df)
```

```
          ID Warehouse_block Mode_of_Shipment  Customer_care_calls  \
0          1               D           Flight                    4
1          2               F           Flight                    4
2          3               A           Flight                    2
3          4               B           Flight                    3
4          5               C           Flight                    2
…        …               …               …                    …
10994  10995               A             Ship                    4
10995  10996               B             Ship                    4
10996  10997               C             Ship                    5
10997  10998               F             Ship                    5
10998  10999               D             Ship                    2

       Customer_rating  Cost_of_the_Product  Profit  Prior_purchases  \
0                    2                  177    8.85                3
1                    5                  216   10.80                2
2                    2                  183    9.15                4
3                    3                  176    8.80                4
4                    2                  184    9.20                3
…                  …                  …      …                 …
10994                1                  252   12.60                5
10995                1                  232   11.60                5
10996                4                  242   12.10                5
10997                2                  223   11.15                6
10998                5                  155    7.75                5

      Product_importance  Order_date   Ship_Date Gender  Discount_offered  \
0                    low  2022-03-17  2022-03-25      F                44
```

```
1                    low  2023-09-12  2023-09-16      M                59
2                    low  2022-09-19  2022-09-26      M                48
3          Not Specified  2020-03-09  2020-03-12      M                10
4          Not Specified  2021-07-12  2021-07-16      F                46
…                    …          …          …    …                …
10994             medium  2022-01-31  2022-02-09      F                 1
10995             medium  2021-04-28  2021-05-05      F                 6
10996                low  2022-03-17  2022-03-22      F                 4
10997             medium  2021-08-30  2021-09-03      M                 2
10998                low  2020-02-21  2020-02-25      F                 6

       Weight_in_gms  Reached.on.Time_Y.N Product Category  \
0               1233                    1  Office Supplies
1               3088                    1  Office Supplies
2               3374                    1  Office Supplies
3               1177                    1       Technology
4               2484                    1       Technology
…               …                     …                …
10994           1538                    1       Technology
10995           1247                    0       Technology
10996           1155                    0  Office Supplies
10997           1210                    0       Technology
10998           1639                    0  Office Supplies

      Product sub Category
0                   Labels
1      Pens & Art Supplies
2                   Papers
3          Office Machines
4                   Tables
…                    …
10994   Mobile Accessories
10995   Mobile Accessories
10996   Mobile Accessories
10997   Mobile Accessories
10998   Mobile Accessories

[10999 rows x 17 columns]
```

[16]:
```
df.head(2)
```

[16]:
```
   ID Warehouse_block Mode_of_Shipment  Customer_care_calls  Customer_rating  \
0   1               D           Flight                    4                2
1   2               F           Flight                    4                5

   Cost_of_the_Product  Profit  Prior_purchases Product_importance  \
0                  177    8.85                3                low
```

```
1                  216   10.80                2                 low

   Order_date   Ship_Date Gender  Discount_offered  Weight_in_gms  \
0  2022-03-17  2022-03-25      F                44           1233
1  2023-09-12  2023-09-16      M                59           3088

   Reached.on.Time_Y.N Product Category Product sub Category
0                    1  Office Supplies                Labels
1                    1  Office Supplies  Pens & Art Supplies
```

[8]:
```
df.shape
```

[8]:
```
(10999, 17)
```

1 Findign Missing Values

[101]:
```
df.isnull().sum()
```

[101]:
```
ID                     0
Warehouse_block        0
Mode_of_Shipment       0
Customer_care_calls    0
Customer_rating        0
Cost_of_the_Product    0
Prior_purchases        0
Product_importance     0
Gender                 0
Discount_offered       0
Weight_in_gms          0
Reached.on.Time_Y.N    0
Product Category       426
dtype: int64
```

2 Filling Mean Value in Null Places

[102]:
```
# Calculate the mean of the column
mean_cost = df['Cost_of_the_Product'].mean()
```

[103]:
```
# Fill NaN values with the calculated mean
df['Cost_of_the_Product'] = df['Cost_of_the_Product'].fillna(mean_cost)
```

[67]:
```
df.isnull().sum()
```

[67]:
```
ID                     0
Warehouse_block        0
```

```
Mode_of_Shipment       0
Customer_care_calls    0
Customer_rating        0
Cost_of_the_Product    0
Prior_purchases        0
Product_importance     0
Gender                 0
Discount_offered       0
Weight_in_gms          0
Reached.on.Time_Y.N    0
dtype: int64
```

3 Finding Product_Importance Column how much the product priority is needed.

[132]:
```
df['Product_importance'].value_counts()
```

[132]:
```
Product_importance
low              5297
medium           4076
high              688
Not Specified     427
Critical          260
Critical          251
Name: count, dtype: int64
```

[133]:
```
df['Product_importance'].unique()
```

[133]:
```
array(['low', 'Not Specified', 'Critical', 'Critical ', 'medium', 'high'],
      dtype=object)
```

4 Here we Having Two Priority Criteria Under Critical Now we need to Rectify it

[134]:
```
df["Product_importance"].replace("Critical ","Critical")
```

[134]:
```
0                 low
1                 low
2                 low
3        Not Specified
4        Not Specified
             …
10994           medium
10995           medium
```

```
10996              low
10997           medium
10998              low
Name: Product_importance, Length: 10999, dtype: object
```

5 Now reload the replace value into the data set

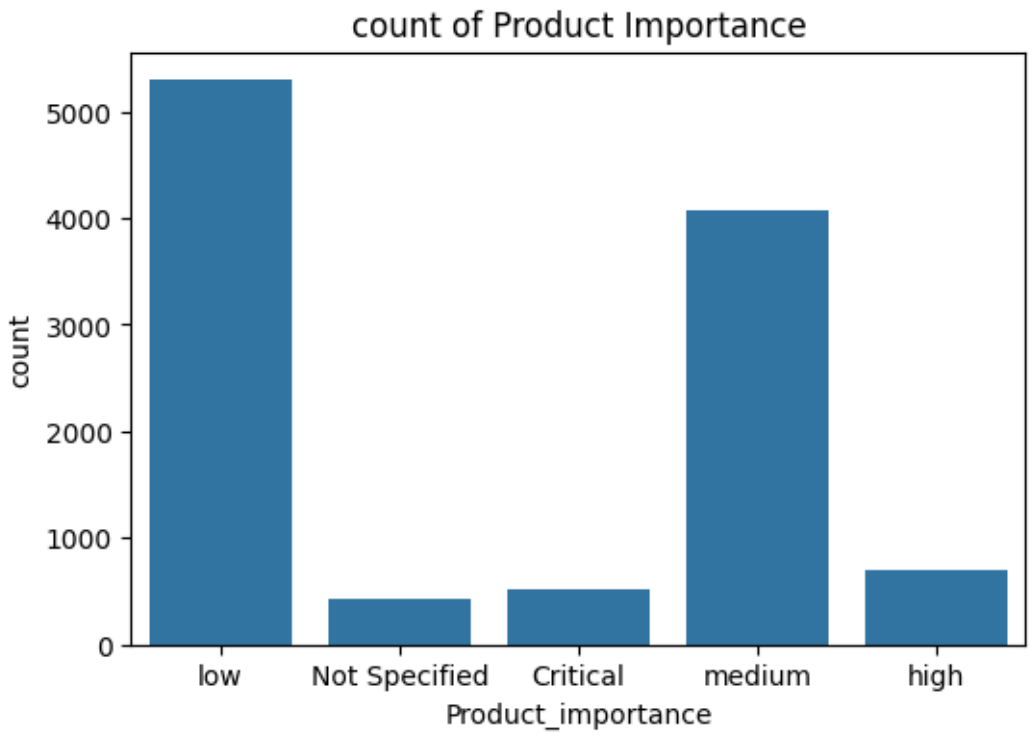
[135]:
```
df["Product_importance"]=df["Product_importance"].replace("Critical␣
 ↪","Critical")
```

[136]:
```
df['Product_importance'].value_counts()
```

[136]:
```
Product_importance
low              5297
medium           4076
high              688
Critical          511
Not Specified     427
Name: count, dtype: int64
```

6 Product Importance

[137]:
```
plt.figure(figsize=(6,4))
sns.countplot(x="Product_importance",data=df)
plt.title('count of Product Importance')
plt.savefig("Count of Order Priority.jpg")
plt.show()
```



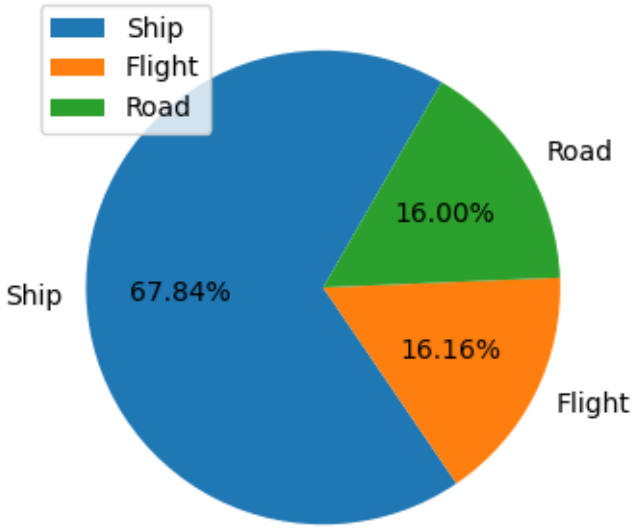
7 Mode_of_Shipment Analysis

[138]:
```
df['Mode_of_Shipment'].value_counts()
```

[138]:
```
Mode_of_Shipment
Ship      7462
Flight    1777
Road      1760
Name: count, dtype: int64
```

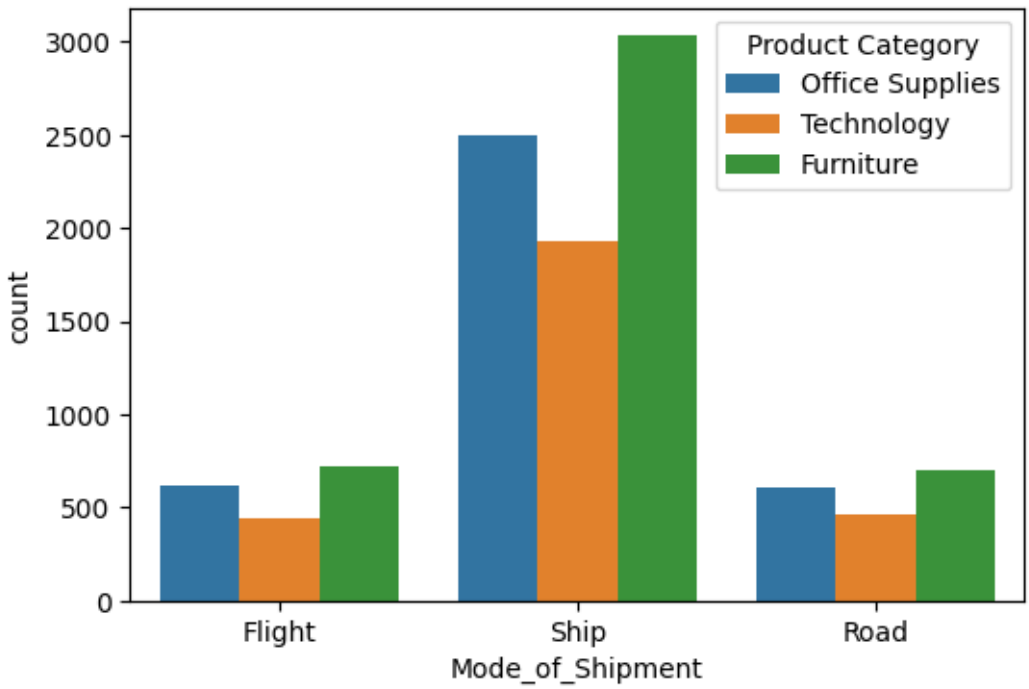
[139]:
```
x = df['Mode_of_Shipment'].value_counts().index
y = df['Mode_of_Shipment'].value_counts().values
```

[140]:
```
plt.figure(figsize=(5,4))
plt.pie(y, labels=x, startangle=60, autopct="%0.2f%%") # "%0.2f%%" this is␣
 ↪formula
plt.legend(loc=2) # Here loc=1 refer to right top, loc=2 refer to left top, loc␣
 ↪= 3 refer to bottom left, loc 4 refer to bottom right
plt.show()
```



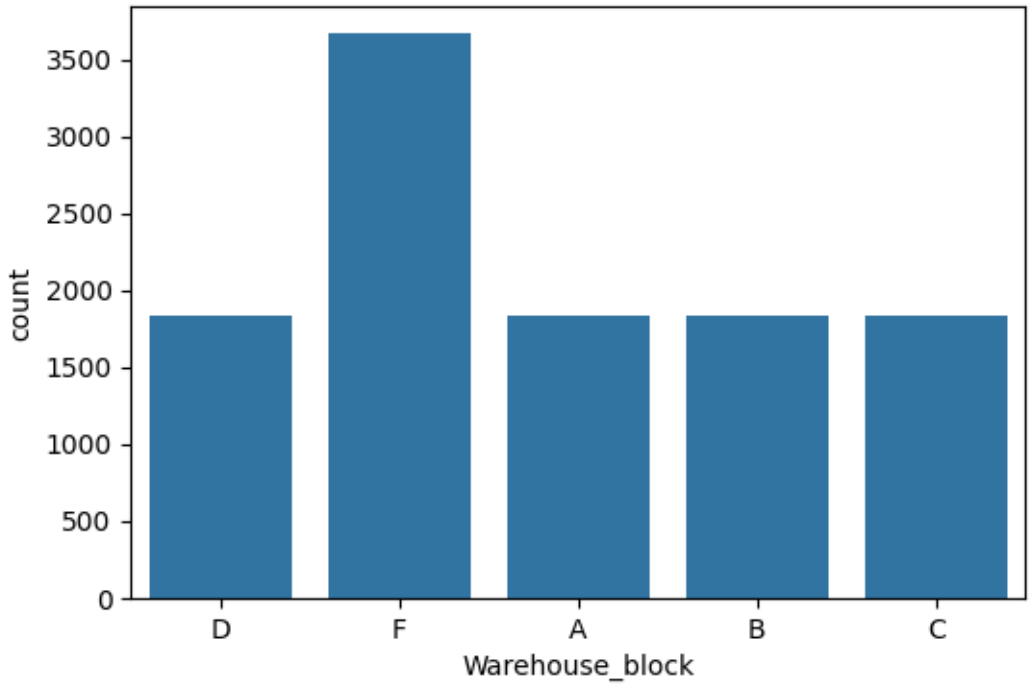
8 Bivariate analysis B/W Mode of Shipment & Product Category

[141]:
```
plt.figure(figsize=(6,4))
sns.countplot(x='Mode_of_Shipment', data=df, hue='Product Category')
plt.show()
```



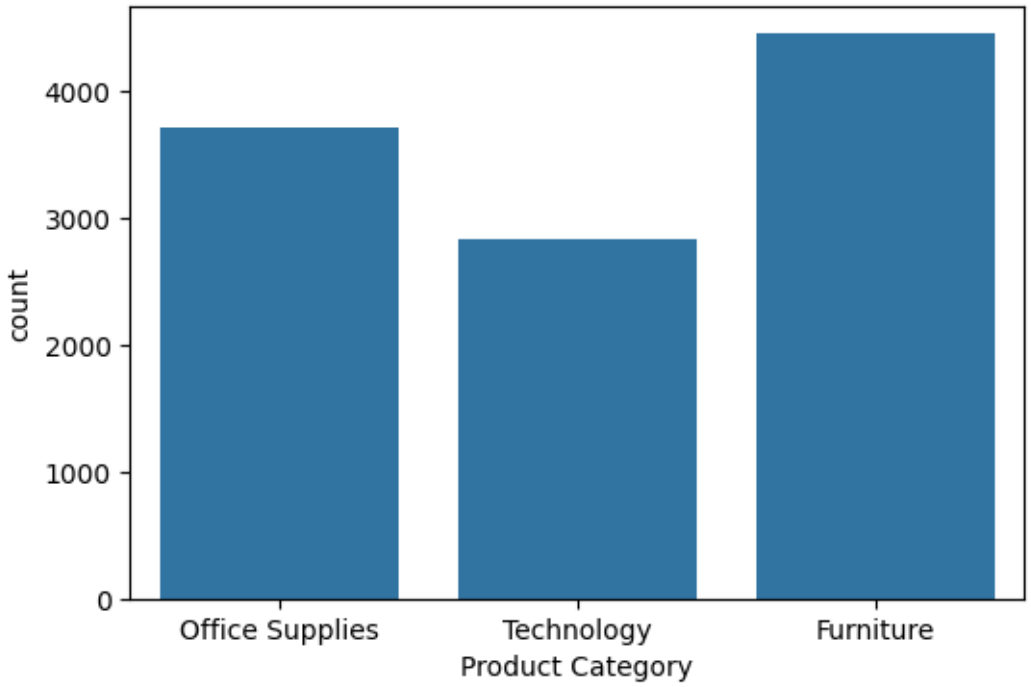
9 Warehouse_block

[142]:
```
plt.figure(figsize=(6,4))
sns.countplot(x='Warehouse_block', data=df)
plt.show()
```

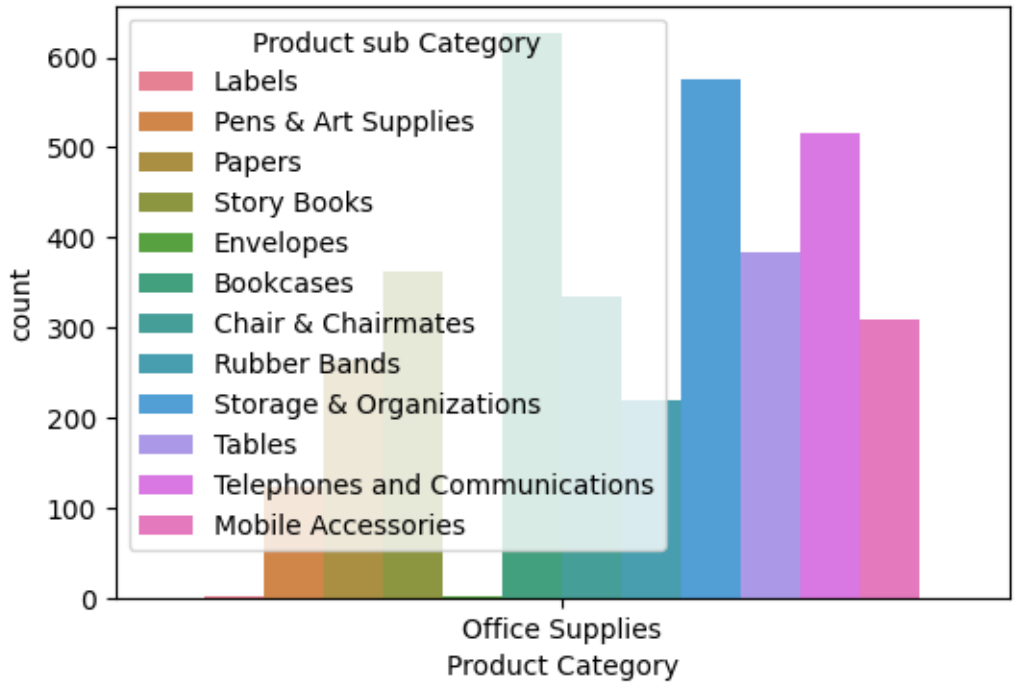


10 Product Category

[143]:
```
plt.figure(figsize=(6,4))
sns.countplot(x='Product Category', data=df)
plt.show()
```



[149]:
```
plt.figure(figsize=(6,4))
sns.countplot(x='Product Category', data=df[df['Product Category']=="Office␣
↪Supplies"],hue="Product sub Category")
plt.show()
```



[9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10999 entries, 0 to 10998
Data columns (total 17 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   ID                    10999 non-null  int64
 1   Warehouse_block       10999 non-null  object
 2   Mode_of_Shipment      10999 non-null  object
 3   Customer_care_calls   10999 non-null  int64
 4   Customer_rating       10999 non-null  int64
 5   Cost_of_the_Product   10999 non-null  int64
 6   Profit                10999 non-null  float64
 7   Prior_purchases       10999 non-null  int64
 8   Product_importance    10999 non-null  object
 9   Order_date            10999 non-null  object
 10  Ship_Date             10999 non-null  object
 11  Gender                10999 non-null  object
 12  Discount_offered      10999 non-null  int64
 13  Weight_in_gms         10999 non-null  int64
 14  Reached.on.Time_Y.N   10999 non-null  int64
 15  Product Category      10999 non-null  object
 16  Product sub Category  10999 non-null  object
dtypes: float64(1), int64(8), object(8)
memory usage: 1.4+ MB
```

11 Number of shipments By Year

[23]:
```
df['Order_Year'] = df['Order_date'].dt.year
```

[24]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10999 entries, 0 to 10998
Data columns (total 19 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   ID                    10999 non-null  int64
 1   Warehouse_block       10999 non-null  object
 2   Mode_of_Shipment      10999 non-null  object
 3   Customer_care_calls   10999 non-null  int64
 4   Customer_rating       10999 non-null  int64
 5   Cost_of_the_Product   10999 non-null  int64
```

```
 6   Profit                10999 non-null  float64
 7   Prior_purchases       10999 non-null  int64
 8   Product_importance    10999 non-null  object
 9   Order_date            10999 non-null  datetime64[ns]
 10  Ship_Date             10999 non-null  object
 11  Gender                10999 non-null  object
 12  Discount_offered      10999 non-null  int64
 13  Weight_in_gms         10999 non-null  int64
 14  Reached.on.Time_Y.N   10999 non-null  int64
 15  Product Category      10999 non-null  object
 16  Product sub Category  10999 non-null  object
 17  Order_year            10999 non-null  int32
 18  Order_Year            10999 non-null  int32
dtypes: datetime64[ns](1), float64(1), int32(2), int64(8), object(7)
memory usage: 1.5+ MB
```

[25]:
```
# Remove a column by name
df = df.drop('Order_year', axis=1)
```

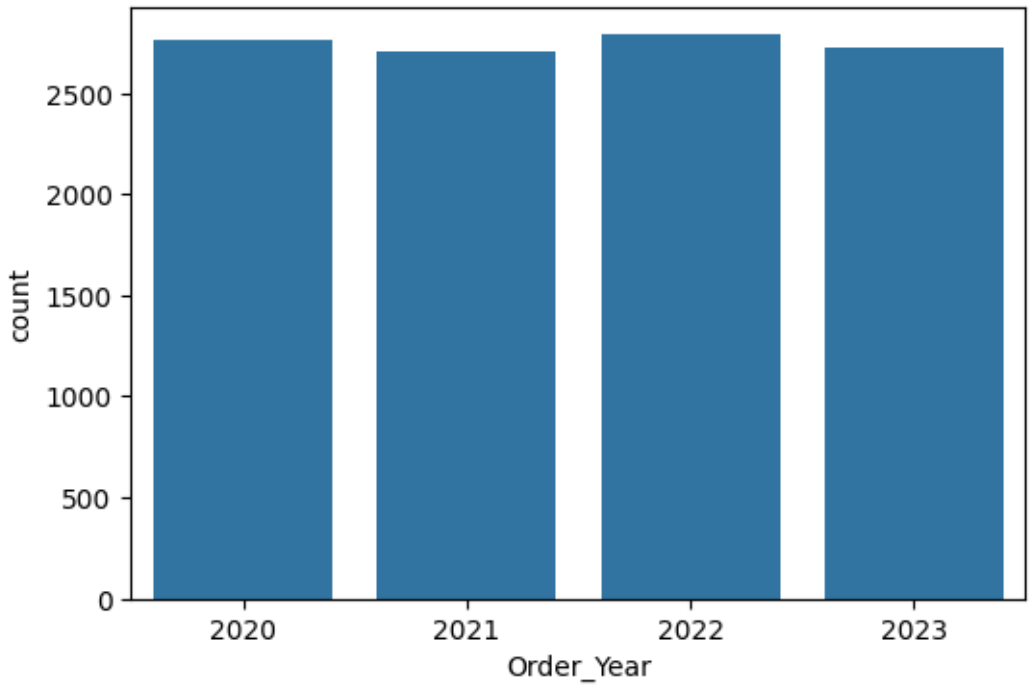
[26]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10999 entries, 0 to 10998
Data columns (total 18 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   ID                    10999 non-null  int64
 1   Warehouse_block       10999 non-null  object
 2   Mode_of_Shipment      10999 non-null  object
 3   Customer_care_calls   10999 non-null  int64
 4   Customer_rating       10999 non-null  int64
 5   Cost_of_the_Product   10999 non-null  int64
 6   Profit                10999 non-null  float64
 7   Prior_purchases       10999 non-null  int64
 8   Product_importance    10999 non-null  object
 9   Order_date            10999 non-null  datetime64[ns]
 10  Ship_Date             10999 non-null  object
 11  Gender                10999 non-null  object
 12  Discount_offered      10999 non-null  int64
 13  Weight_in_gms         10999 non-null  int64
 14  Reached.on.Time_Y.N   10999 non-null  int64
 15  Product Category      10999 non-null  object
 16  Product sub Category  10999 non-null  object
 17  Order_Year            10999 non-null  int32
dtypes: datetime64[ns](1), float64(1), int32(1), int64(8), object(7)
memory usage: 1.5+ MB
```

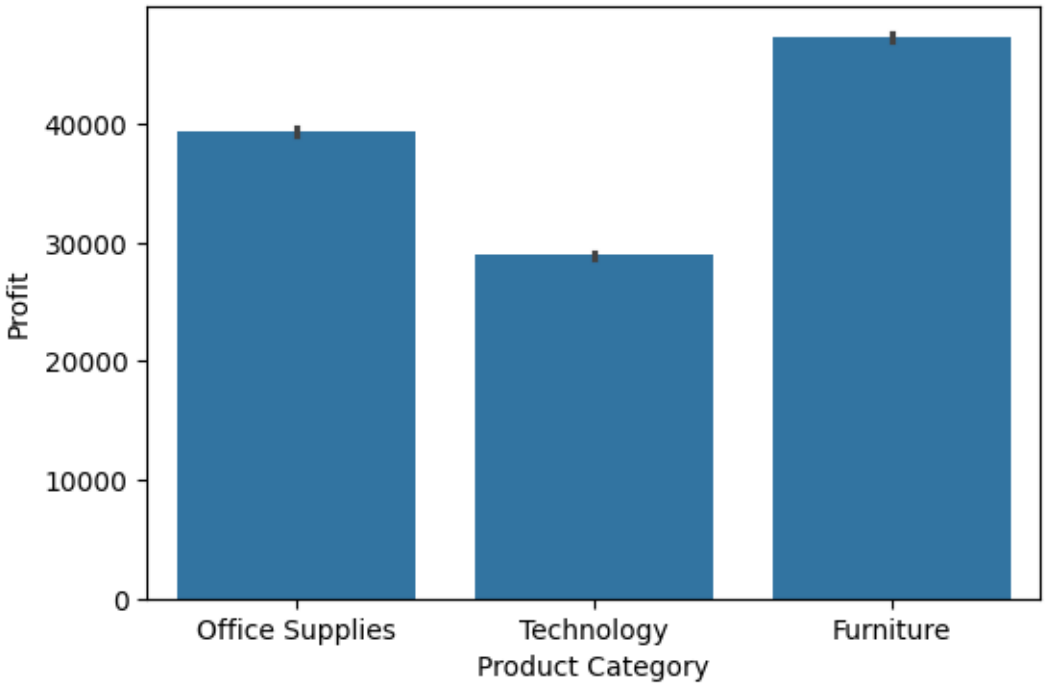
[27]:
```
df['Order_Year'].value_counts()
```

[27]:
```
Order_Year
2022    2791
2020    2767
2023    2731
2021    2710
Name: count, dtype: int64
```

[28]:
```
plt.figure(figsize=(6,4))
sns.countplot(x='Order_Year', data=df)
plt.show()
```



[30]:
```
plt.figure(figsize=(6,4))
sns.barplot(x='Product Category', y='Profit', data=df,estimator='sum')
plt.show()
```



12 Sales As per Warehouse Block wise

[31]:
```
df['Warehouse_block'].value_counts()
```

[31]:
```
Warehouse_block
F    3666
D    1834
A    1833
B    1833
C    1833
Name: count, dtype: int64
```

[32]:
```
df['Warehouse_block'].value_counts()[:3] # top 3
```

[32]:
```
Warehouse_block
F    3666
D    1834
A    1833
Name: count, dtype: int64
```