In [ ]: STEP 01 : DATA COLLECTION

In [2]:
```python
import pandas as pd
```

In [3]:
```python
s1 = pd.read_csv(r"C:\Users\Sinchana\Downloads\ecommerce_1000_records.csv")
s1
```

Out[3]:

	Order_ID	Order_Date	Customer_ID	Product	Category	Quantity	Price	Payment_Method	Delivery_Status	Total_Amount
0	ORD10000	46:14.4	CUST1223	Speaker	Accessories	4	20415	Cash on Delivery	Processing	81660
1	ORD10001	46:14.4	CUST1510	Headphones	Electronics	5	40415	Credit Card	Cancelled	202075
2	ORD10002	46:14.4	CUST1208	Smartphone	Gadgets	3	37719	Credit Card	Cancelled	113157
3	ORD10003	46:14.4	CUST1744	Tablet	Gadgets	5	15843	Cash on Delivery	Shipped	79215
4	ORD10004	46:14.4	CUST1399	Smartwatch	Gadgets	5	46903	UPI	Cancelled	234515
...	...	...	...	...	...	...	...	...	...	...
995	ORD10995	46:14.4	CUST1571	Speaker	Electronics	2	27484	Debit Card	Shipped	54968
996	ORD10996	46:14.4	CUST1355	Smartphone	Accessories	1	45133	Debit Card	Delivered	45133
997	ORD10997	46:14.4	CUST1432	Smartphone	Accessories	1	8625	UPI	Delivered	8625
998	ORD10998	46:14.4	CUST1594	Keyboard	Gadgets	4	36559	UPI	Cancelled	146236
999	ORD10999	46:14.4	CUST1551	Monitor	Electronics	3	4533	Credit Card	Shipped	13599

1000 rows × 10 columns

In [4]:
```python
s1.shape
```

Out[4]: (1000, 10)

In [5]:
```python
s1.info
```

Out[5]:
```
<bound method DataFrame.info of      Order_ID Order_Date Customer_ID     Product     Category  Quantity  \
0    ORD10000    46:14.4    CUST1223     Speaker  Accessories         4
1    ORD10001    46:14.4    CUST1510  Headphones  Electronics         5
2    ORD10002    46:14.4    CUST1208  Smartphone      Gadgets         3
3    ORD10003    46:14.4    CUST1744      Tablet      Gadgets         5
4    ORD10004    46:14.4    CUST1399  Smartwatch      Gadgets         5
..        ...        ...         ...         ...          ...       ...
995  ORD10995    46:14.4    CUST1571     Speaker  Electronics         2
996  ORD10996    46:14.4    CUST1355  Smartphone  Accessories         1
997  ORD10997    46:14.4    CUST1432  Smartphone  Accessories         1
998  ORD10998    46:14.4    CUST1594    Keyboard      Gadgets         4
999  ORD10999    46:14.4    CUST1551     Monitor  Electronics         3

     Price    Payment_Method Delivery_Status  Total_Amount
0    20415  Cash on Delivery      Processing         81660
1    40415       Credit Card       Cancelled        202075
2    37719       Credit Card       Cancelled        113157
3    15843  Cash on Delivery         Shipped         79215
4    46903               UPI       Cancelled        234515
..     ...               ...             ...           ...
995  27484        Debit Card         Shipped         54968
996  45133        Debit Card       Delivered         45133
997   8625               UPI       Delivered          8625
998  36559               UPI       Cancelled        146236
999   4533       Credit Card         Shipped         13599

[1000 rows x 10 columns]>
```

In [6]:
```python
s1.tail
```

Out[6]:
```
<bound method NDFrame.tail of      Order_ID Order_Date Customer_ID     Product     Category  Quantity  \
0    ORD10000    46:14.4    CUST1223     Speaker  Accessories         4
1    ORD10001    46:14.4    CUST1510  Headphones  Electronics         5
2    ORD10002    46:14.4    CUST1208  Smartphone      Gadgets         3
3    ORD10003    46:14.4    CUST1744      Tablet      Gadgets         5
4    ORD10004    46:14.4    CUST1399  Smartwatch      Gadgets         5
..        ...        ...         ...         ...          ...       ...
995  ORD10995    46:14.4    CUST1571     Speaker  Electronics         2
996  ORD10996    46:14.4    CUST1355  Smartphone  Accessories         1
997  ORD10997    46:14.4    CUST1432  Smartphone  Accessories         1
998  ORD10998    46:14.4    CUST1594    Keyboard      Gadgets         4
999  ORD10999    46:14.4    CUST1551     Monitor  Electronics         3

     Price    Payment_Method Delivery_Status  Total_Amount
0    20415  Cash on Delivery      Processing         81660
1    40415       Credit Card       Cancelled        202075
2    37719       Credit Card       Cancelled        113157
3    15843  Cash on Delivery         Shipped         79215
4    46903               UPI       Cancelled        234515
..     ...               ...             ...           ...
995  27484        Debit Card         Shipped         54968
996  45133        Debit Card       Delivered         45133
997   8625               UPI       Delivered          8625
998  36559               UPI       Cancelled        146236
999   4533       Credit Card         Shipped         13599

[1000 rows x 10 columns]>
```

In [7]:
```python
s1.columns
```

Out[7]:
```
Index(['Order_ID', 'Order_Date', 'Customer_ID', 'Product', 'Category',
       'Quantity', 'Price', 'Payment_Method', 'Delivery_Status',
       'Total_Amount'],
      dtype='object')
```

In [ ]: STEP 02 : DATA CLEANING

In [8]:
```python
s1 = pd.read_csv(r"C:\Users\Sinchana\Downloads\ecommerce_1000_records.csv")
s1
```

Out[8]:

	Order_ID	Order_Date	Customer_ID	Product	Category	Quantity	Price	Payment_Method	Delivery_Status	Total_Amount
0	ORD10000	46:14.4	CUST1223	Speaker	Accessories	4	20415	Cash on Delivery	Processing	81660
1	ORD10001	46:14.4	CUST1510	Headphones	Electronics	5	40415	Credit Card	Cancelled	202075
2	ORD10002	46:14.4	CUST1208	Smartphone	Gadgets	3	37719	Credit Card	Cancelled	113157
3	ORD10003	46:14.4	CUST1744	Tablet	Gadgets	5	15843	Cash on Delivery	Shipped	79215
4	ORD10004	46:14.4	CUST1399	Smartwatch	Gadgets	5	46903	UPI	Cancelled	234515
...	...	...	...	...	...	...	...	...	...	...
995	ORD10995	46:14.4	CUST1571	Speaker	Electronics	2	27484	Debit Card	Shipped	54968
996	ORD10996	46:14.4	CUST1355	Smartphone	Accessories	1	45133	Debit Card	Delivered	45133
997	ORD10997	46:14.4	CUST1432	Smartphone	Accessories	1	8625	UPI	Delivered	8625
998	ORD10998	46:14.4	CUST1594	Keyboard	Gadgets	4	36559	UPI	Cancelled	146236
999	ORD10999	46:14.4	CUST1551	Monitor	Electronics	3	4533	Credit Card	Shipped	13599

1000 rows × 10 columns

In [9]:
```python
s1 = s1.drop_duplicates()
s1
```

Out[9]:

	Order_ID	Order_Date	Customer_ID	Product	Category	Quantity	Price	Payment_Method	Delivery_Status	Total_Amount
0	ORD10000	46:14.4	CUST1223	Speaker	Accessories	4	20415	Cash on Delivery	Processing	81660
1	ORD10001	46:14.4	CUST1510	Headphones	Electronics	5	40415	Credit Card	Cancelled	202075
2	ORD10002	46:14.4	CUST1208	Smartphone	Gadgets	3	37719	Credit Card	Cancelled	113157
3	ORD10003	46:14.4	CUST1744	Tablet	Gadgets	5	15843	Cash on Delivery	Shipped	79215
4	ORD10004	46:14.4	CUST1399	Smartwatch	Gadgets	5	46903	UPI	Cancelled	234515
...	...	...	...	...	...	...	...	...	...	...
995	ORD10995	46:14.4	CUST1571	Speaker	Electronics	2	27484	Debit Card	Shipped	54968
996	ORD10996	46:14.4	CUST1355	Smartphone	Accessories	1	45133	Debit Card	Delivered	45133
997	ORD10997	46:14.4	CUST1432	Smartphone	Accessories	1	8625	UPI	Delivered	8625
998	ORD10998	46:14.4	CUST1594	Keyboard	Gadgets	4	36559	UPI	Cancelled	146236
999	ORD10999	46:14.4	CUST1551	Monitor	Electronics	3	4533	Credit Card	Shipped	13599

1000 rows × 10 columns

In [10]:
```python
s1.dropna(inplace=True)
s1
```

Out[10]:

	Order_ID	Order_Date	Customer_ID	Product	Category	Quantity	Price	Payment_Method	Delivery_Status	Total_Amount
0	ORD10000	46:14.4	CUST1223	Speaker	Accessories	4	20415	Cash on Delivery	Processing	81660
1	ORD10001	46:14.4	CUST1510	Headphones	Electronics	5	40415	Credit Card	Cancelled	202075
2	ORD10002	46:14.4	CUST1208	Smartphone	Gadgets	3	37719	Credit Card	Cancelled	113157
3	ORD10003	46:14.4	CUST1744	Tablet	Gadgets	5	15843	Cash on Delivery	Shipped	79215
4	ORD10004	46:14.4	CUST1399	Smartwatch	Gadgets	5	46903	UPI	Cancelled	234515
...	...	...	...	...	...	...	...	...	...	...
995	ORD10995	46:14.4	CUST1571	Speaker	Electronics	2	27484	Debit Card	Shipped	54968
996	ORD10996	46:14.4	CUST1355	Smartphone	Accessories	1	45133	Debit Card	Delivered	45133
997	ORD10997	46:14.4	CUST1432	Smartphone	Accessories	1	8625	UPI	Delivered	8625
998	ORD10998	46:14.4	CUST1594	Keyboard	Gadgets	4	36559	UPI	Cancelled	146236
999	ORD10999	46:14.4	CUST1551	Monitor	Electronics	3	4533	Credit Card	Shipped	13599

1000 rows × 10 columns

In [11]:
```python
s1['Product'].value_counts()
```

Out[11]:
```
Product
Keyboard      122
Smartphone    111
Mouse         105
Monitor       104
Tablet        100
Speaker        96
Camera         96
Smartwatch     96
Laptop         91
Headphones     79
Name: count, dtype: int64
```

In [12]:
```python
s1.dtypes
```

Out[12]:
```
Order_ID           object
Order_Date         object
Customer_ID        object
Product            object
Category           object
Quantity            int64
Price               int64
Payment_Method     object
Delivery_Status    object
Total_Amount        int64
dtype: object
```

In [13]:
```python
s1.reset_index(drop=True, inplace=True)
s1
```

Out[13]:

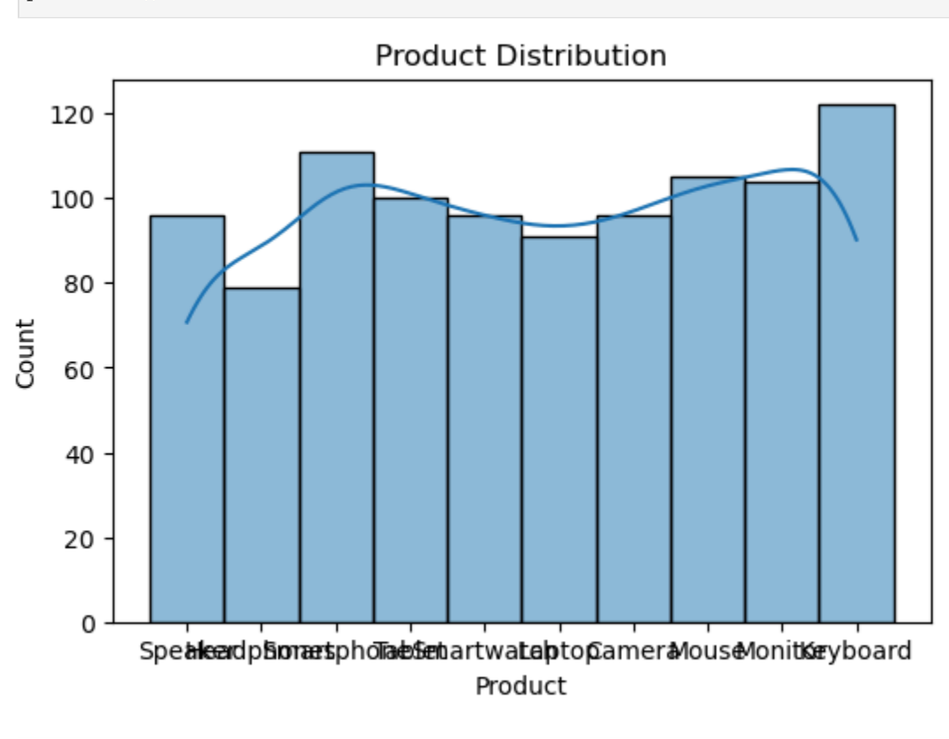
	Order_ID	Order_Date	Customer_ID	Product	Category	Quantity	Price	Payment_Method	Delivery_Status	Total_Amount
0	ORD10000	46:14.4	CUST1223	Speaker	Accessories	4	20415	Cash on Delivery	Processing	81660
1	ORD10001	46:14.4	CUST1510	Headphones	Electronics	5	40415	Credit Card	Cancelled	202075
2	ORD10002	46:14.4	CUST1208	Smartphone	Gadgets	3	37719	Credit Card	Cancelled	113157
3	ORD10003	46:14.4	CUST1744	Tablet	Gadgets	5	15843	Cash on Delivery	Shipped	79215
4	ORD10004	46:14.4	CUST1399	Smartwatch	Gadgets	5	46903	UPI	Cancelled	234515
...	...	...	...	...	...	...	...	...	...	...
995	ORD10995	46:14.4	CUST1571	Speaker	Electronics	2	27484	Debit Card	Shipped	54968
996	ORD10996	46:14.4	CUST1355	Smartphone	Accessories	1	45133	Debit Card	Delivered	45133
997	ORD10997	46:14.4	CUST1432	Smartphone	Accessories	1	8625	UPI	Delivered	8625
998	ORD10998	46:14.4	CUST1594	Keyboard	Gadgets	4	36559	UPI	Cancelled	146236
999	ORD10999	46:14.4	CUST1551	Monitor	Electronics	3	4533	Credit Card	Shipped	13599

1000 rows × 10 columns

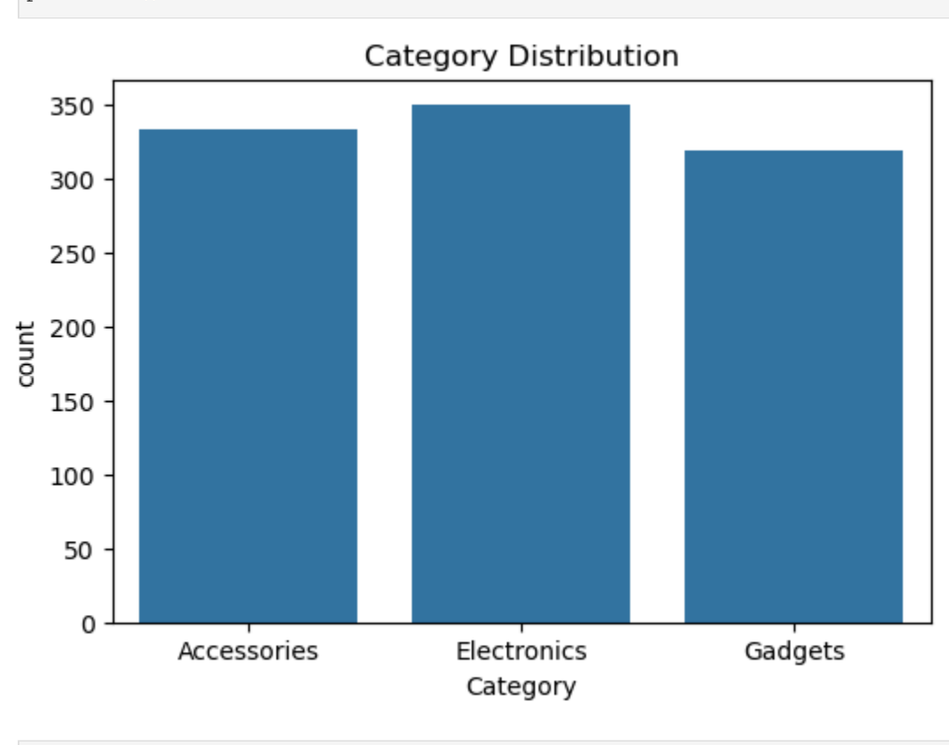
In [ ]: STEP 03 : EXPLORATORY DATA ANALYSIS

In [14]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

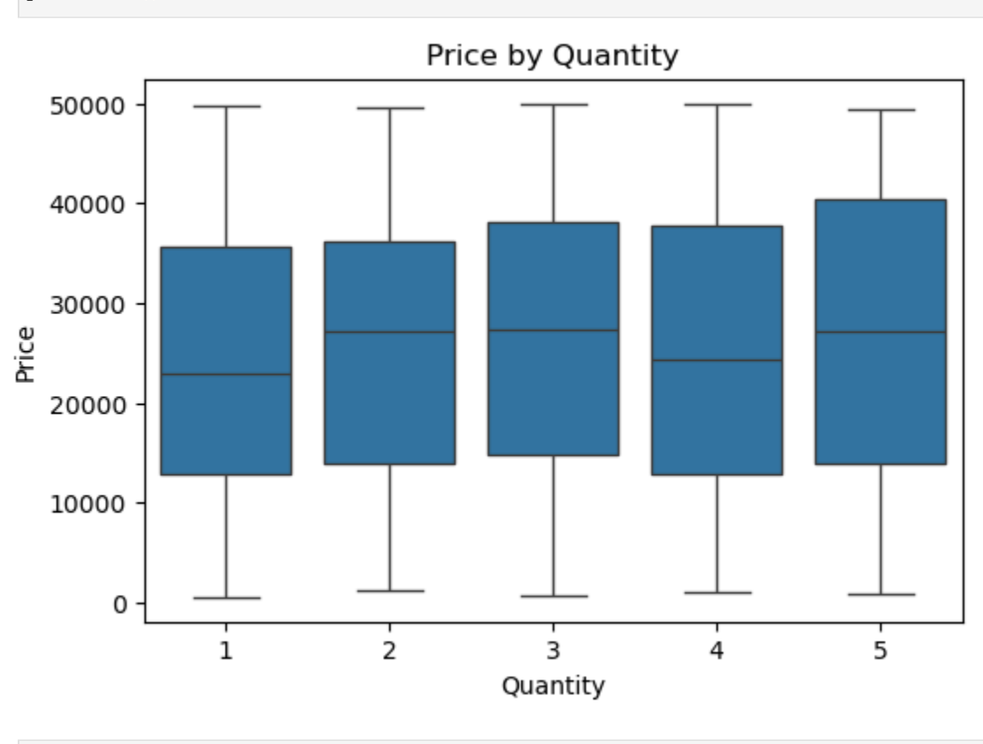
In [15]:
```python
plt.figure(figsize=(6,4))
sns.histplot(s1['Product'], kde=True)
plt.title("Product Distribution")
plt.show()
```



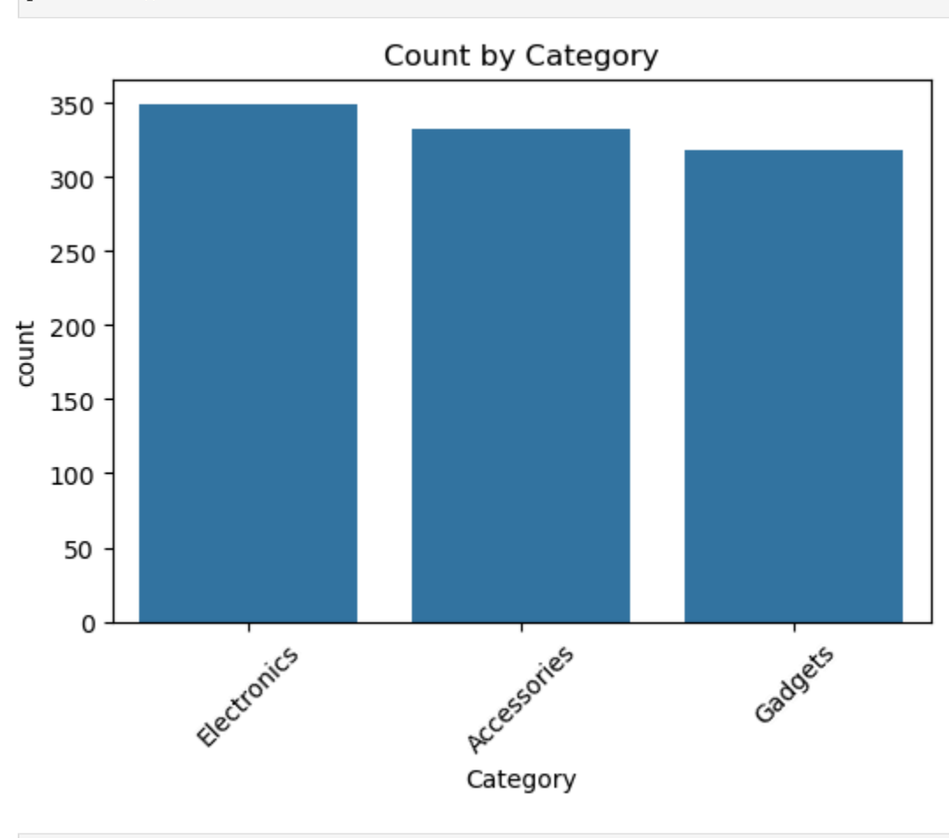
In [16]:
```python
plt.figure(figsize=(6,4))
sns.countplot(x='Category', data=s1)
plt.title("Category Distribution")
plt.show()
```



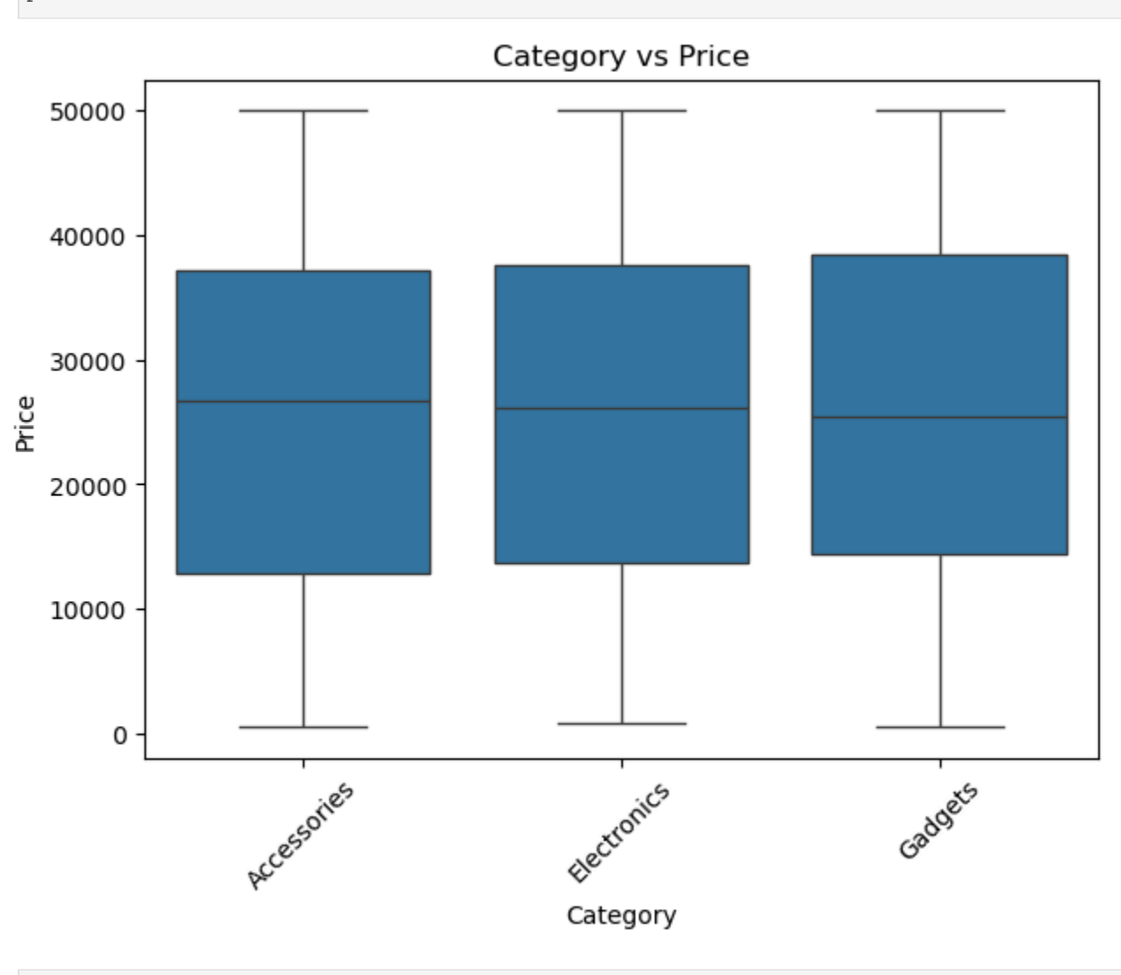
In [17]:
```python
plt.figure(figsize=(6,4))
sns.boxplot(x='Quantity', y='Price', data=s1)
plt.title("Price by Quantity")
plt.show()
```



In [18]:
```python
plt.figure(figsize=(6,4))
sns.countplot(x='Category', data=s1, order=s1['Category'].value_counts().index)
plt.title("Count by Category")
plt.xticks(rotation=45)
plt.show()
```



In [19]:
```python
plt.figure(figsize=(7,5))
sns.boxplot(x='Category', y='Price', data=s1)
plt.title("Category vs Price")
plt.xticks(rotation=45)
plt.show()
```



In [28]: STEP 04: VISUALIZATION

In [20]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

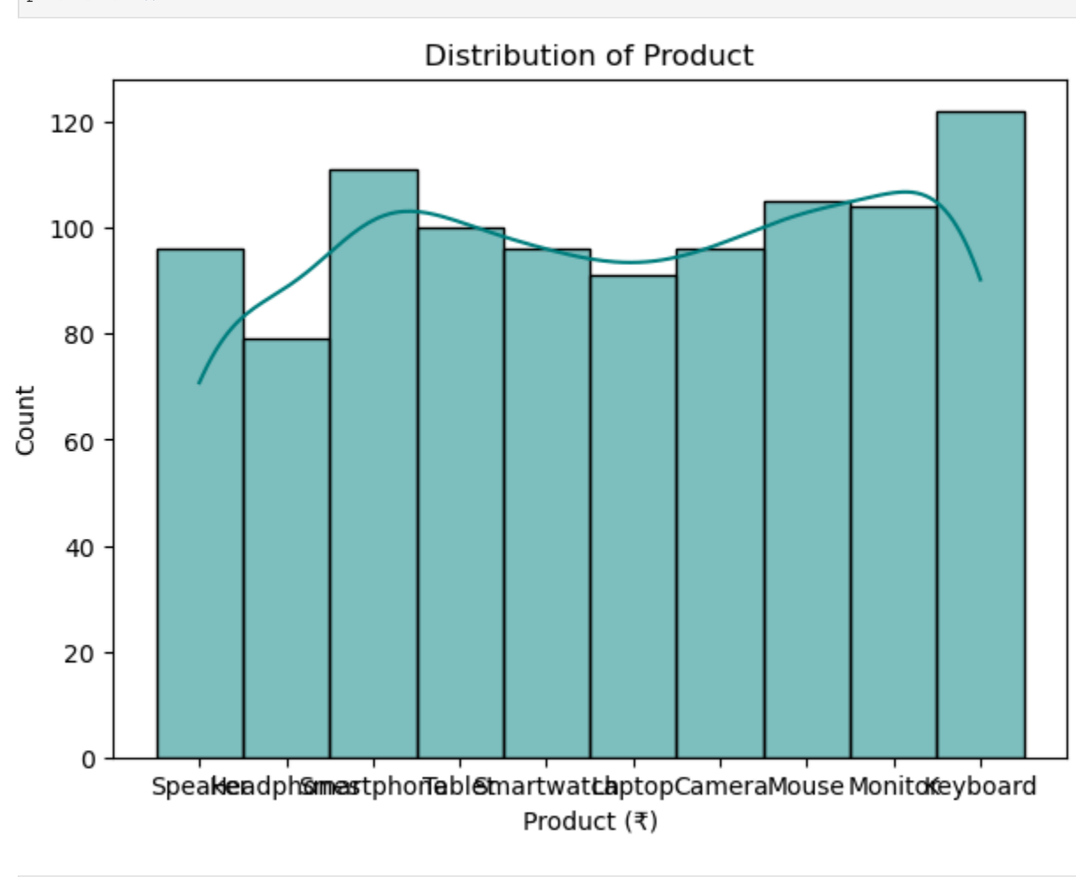
In [21]:
```python
s1 = pd.read_csv(r"C:\Users\Sinchana\Downloads\ecommerce_1000_records.csv")
s1
```

Out[21]:

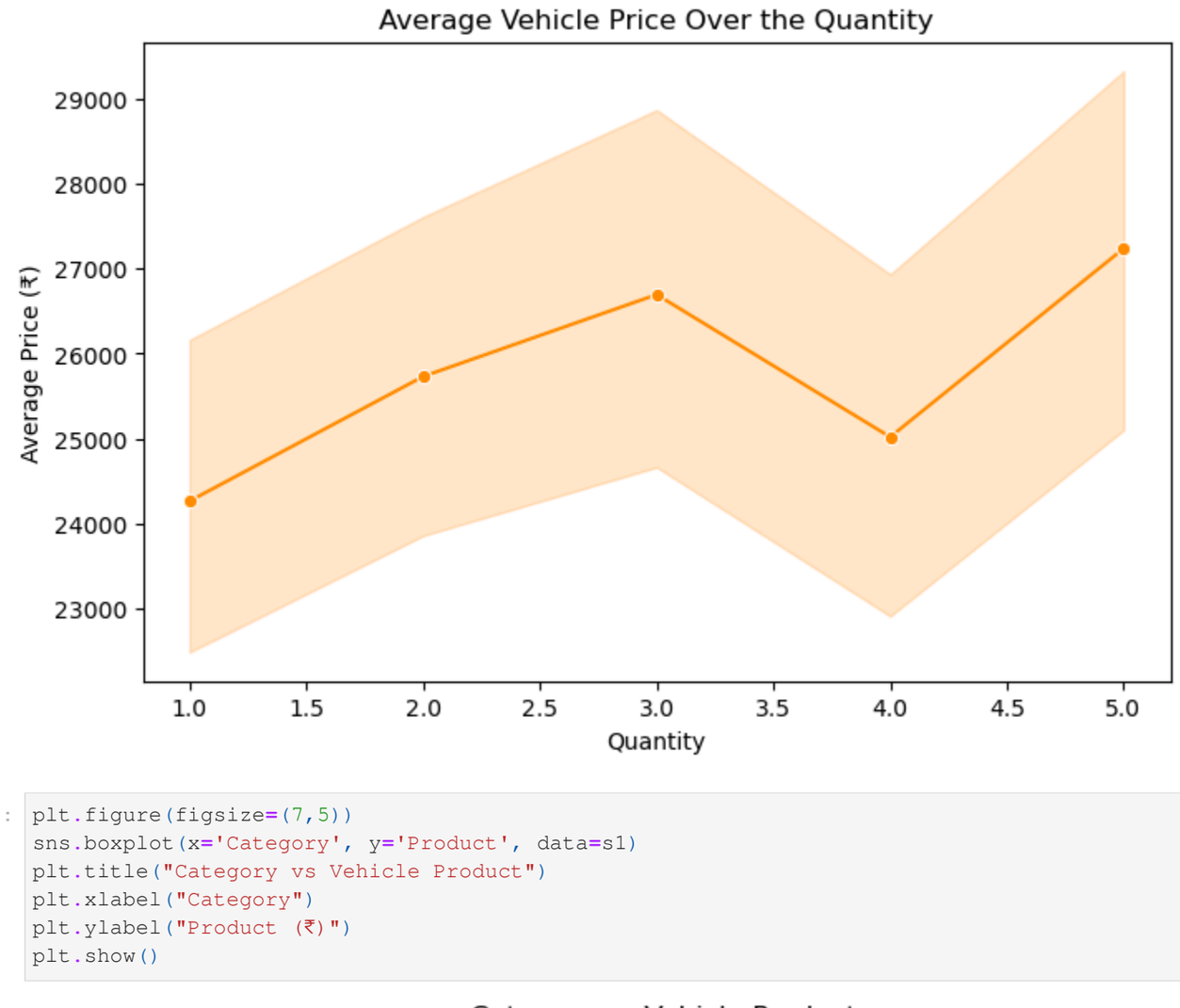
	Order_ID	Order_Date	Customer_ID	Product	Category	Quantity	Price	Payment_Method	Delivery_Status	Total_Amount
0	ORD10000	46:14.4	CUST1223	Speaker	Accessories	4	20415	Cash on Delivery	Processing	81660
1	ORD10001	46:14.4	CUST1510	Headphones	Electronics	5	40415	Credit Card	Cancelled	202075
2	ORD10002	46:14.4	CUST1208	Smartphone	Gadgets	3	37719	Credit Card	Cancelled	113157
3	ORD10003	46:14.4	CUST1744	Tablet	Gadgets	5	15843	Cash on Delivery	Shipped	79215
4	ORD10004	46:14.4	CUST1399	Smartwatch	Gadgets	5	46903	UPI	Cancelled	234515
...	...	...	...	...	...	...	...	...	...	...
995	ORD10995	46:14.4	CUST1571	Speaker	Electronics	2	27484	Debit Card	Shipped	54968
996	ORD10996	46:14.4	CUST1355	Smartphone	Accessories	1	45133	Debit Card	Delivered	45133
997	ORD10997	46:14.4	CUST1432	Smartphone	Accessories	1	8625	UPI	Delivered	8625
998	ORD10998	46:14.4	CUST1594	Keyboard	Gadgets	4	36559	UPI	Cancelled	146236
999	ORD10999	46:14.4	CUST1551	Monitor	Electronics	3	4533	Credit Card	Shipped	13599

1000 rows × 10 columns

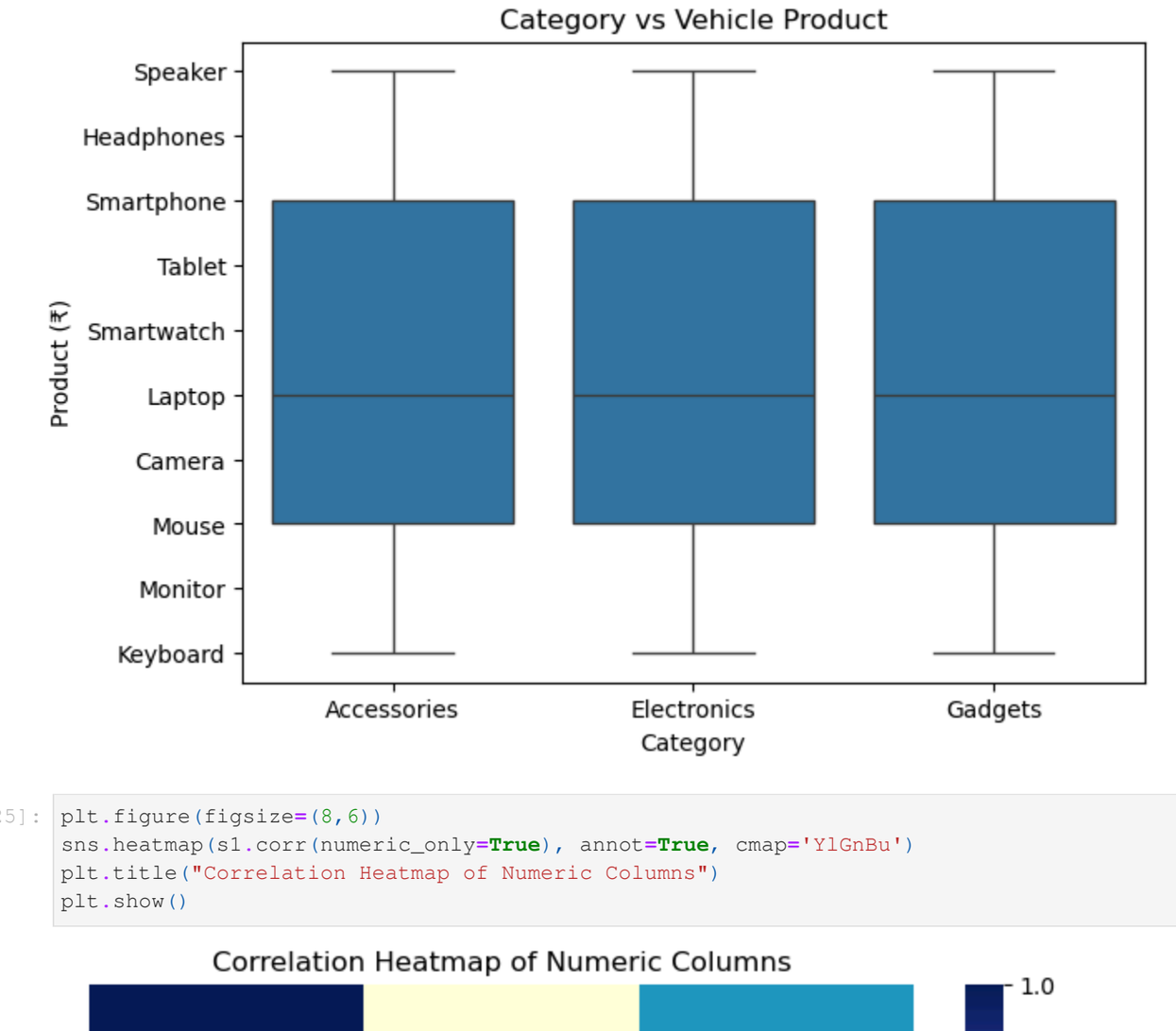
In [22]:
```python
plt.figure(figsize=(7,5))
sns.histplot(s1['Product'], kde=True, color='teal')
plt.title("Distribution of Product")
plt.xlabel("Product (₹)")
plt.ylabel("Count")
plt.show()
```



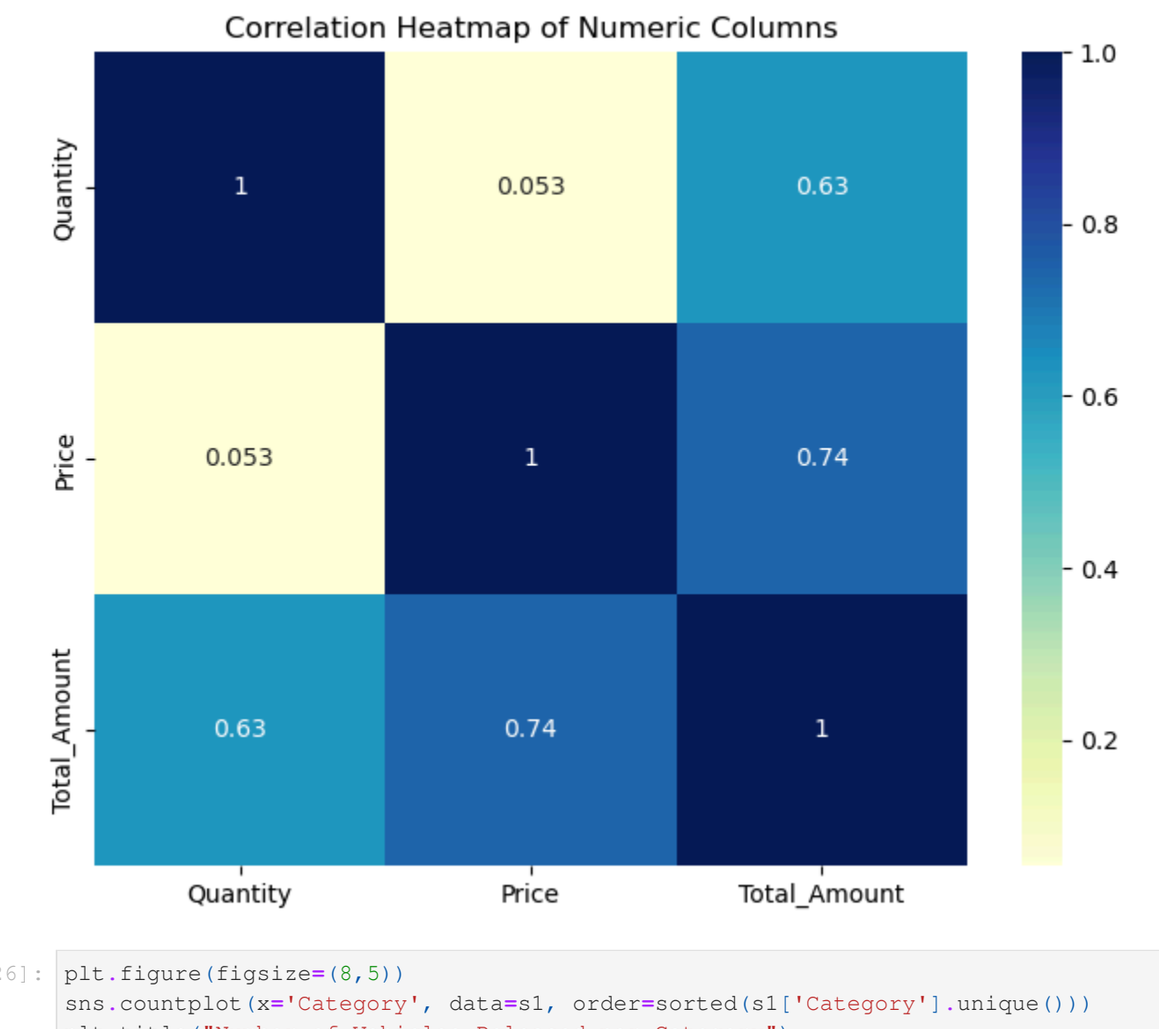
In [23]:
```python
plt.figure(figsize=(8,5))
sns.lineplot(x='Quantity', y='Price', data=s1, estimator='mean', marker='o', color='darkorange')
plt.title("Average Vehicle Price Over the Quantity")
plt.xlabel("Quantity")
plt.ylabel("Average Price (₹)")
plt.show()
```



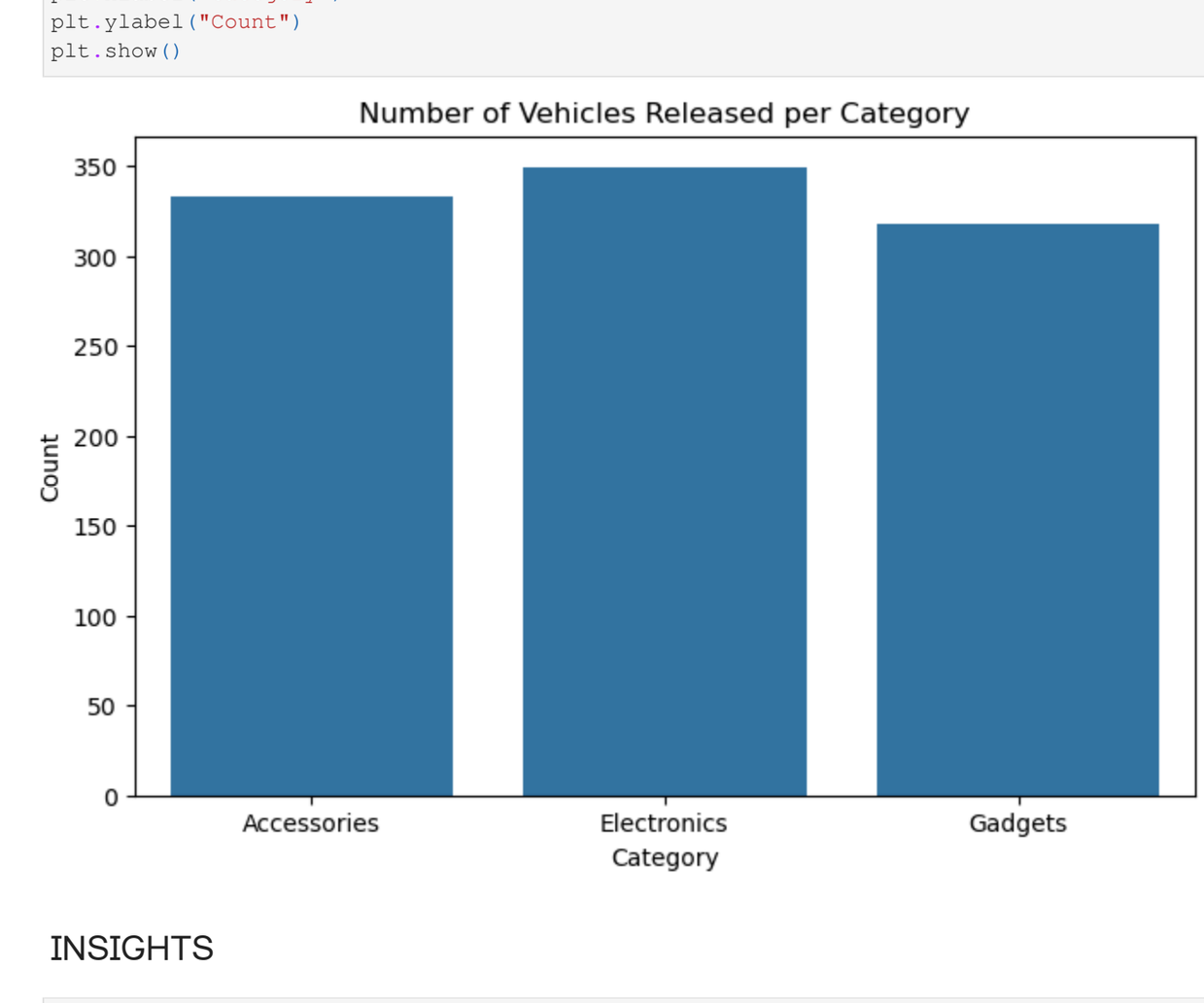
In [24]:
```python
plt.figure(figsize=(7,5))
sns.boxplot(x='Category', y='Product', data=s1)
plt.title("Category vs Vehicle Product")
plt.xlabel("Category")
plt.ylabel("Product (₹)")
plt.show()
```



In [25]:
```python
plt.figure(figsize=(8,6))
sns.heatmap(s1.corr(numeric_only=True), annot=True, cmap='YlGnBu')
plt.title("Correlation Heatmap of Numeric Columns")
plt.show()
```



In [26]:
```python
plt.figure(figsize=(8,5))
sns.countplot(x='Category', data=s1, order=sorted(s1['Category'].unique()))
plt.title("Number of Vehicles Released per Category")
plt.xlabel("Category")
plt.ylabel("Count")
plt.show()
```



INSIGHTS

In [ ]:
```
1️⃣ Product Distribution (Histogram + KDE)
Insights
*The distribution shows how frequently different product values appear in the dataset.
*If the histogram is skewed, it indicates certain products dominate in the dataset.
*KDE curve helps understand whether product frequency is uniform, clustered, or peaked.
*Peaks in the graph indicate products that appear significantly more often.
```

In [ ]:
```
2️⃣ Category Distribution (Countplot)
Insights
*Shows how many records belong to each category.
*Helps identify most common categories and those that appear least.
*If one category is much higher, your dataset is imbalanced in terms of category representation.
*Useful for understanding product segmentation and business focus.
```

In [ ]:
```
3️⃣ Price by Quantity (Boxplot)
Insights
*Shows how Price varies across different quantities.
*Helps identify:
*Which quantity levels have higher or lower prices.
*Whether higher quantity purchases lead to higher pricing or discounts.
*Outliers highlight unusual price points for specific quantities.
*If boxes widen, quantity has high price variability.
```

In [ ]:
```
4️⃣ Count by Category (Ordered Countplot)
Insights
*Displays categories in descending order of frequency.
*Makes it easy to visually detect top-performing or most frequent categories.
*Helps prioritize which categories contribute most to sales or availability.
*Useful for feature importance and categorical analysis.
```

In [ ]:
```
5️⃣Category vs Price (Boxplot)
Insights
*Compares price distribution for each category.
*Helps identify:
    *Which categories have higher average prices.
    *Which categories show price consistency (tight box) or variability (wide box).
    *Categories with many outliers, indicating premium or discounted items.
*Useful for understanding pricing strategy across categor
```

INSIGHTS

In [ ]:
```
1️⃣ Distribution of Product (Histogram + KDE)
Insights
*The distribution shows how frequently different product values occur.
*If the curve shows a single strong peak, the product range has a common value or cluster.
*A right-skew indicates a few high-value products that pull the distribution upward.
*A left-skew indicates more products are high-priced with fewer low-value items.
*KDE line helps identify smooth patterns and hidden clusters in product distribution.
```

In [ ]:
```
2️⃣ Average Price by Quantity (Lineplot)
Insights
*Shows how average price changes with quantity.
*Helps identify whether price increases, decreases, or remains stable as quantity increases.
*If the line trends upward:
*Higher quantity purchases relate to higher prices.
*If the line trends downward:
*Discounts or bulk pricing may be occurring.
*If the pattern is zig-zag, quantity and price do not have a strong linear relationship.
```

In [ ]:
```
3️⃣ Category vs Product (Boxplot)
Insights
*Compares product value distribution across all categories.
*Helps identify:
*Categories with high median product values.
*Categories with large variability, indicating diverse product pricing.
*Categories with many outliers, indicating premium or discounted products.
*If boxes differ significantly:
*Different categories may follow distinct pricing/product segments.
```

In [ ]:
```
4️⃣ Correlation Heatmap (Numeric Columns)
Insights
*Shows relationships between all numeric features.
*Strong positive correlation indicates variables increase together.
*Strong negative correlation indicates variables move in opposite directions.
*Helps identify:
*Whether quantity drives price.
*Whether product value is related to quantity or price.
*Which numeric variables are closely linked and may cause multicollinearity.
*Useful for feature selection and regression model preparation.
```

In [ ]:
```
5️⃣ Count of Category (Ordered Countplot)
Insights
*Shows the number of records in each category.
*Categories with higher counts dominate the dataset.
*Categories with lower counts may be niche or under-represented.
*Highlights data imbalance, which may affect:
*Modeling
```

```
*Statistical conclusions
*Classification results
*Helps understand overall category distribution in the business/product inventory.
```