Amazon Sales 2025 EDA REPORT BY VIVEK CHAUHAN

In [1]:
```
# Load the necessary libraries for the visualizations

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
# load the csv file for the analysis

data = pd.read_csv("C:/Users/VIVEK CHAUHAN/Desktop/amazon_sales_data 2025.csv")
data
```

Out[2]:

	Order ID	Date	Product	Category	Price	Quantity	Total Sales	Customer Name	Customer Location	Payment Method	Status
0	ORD0001	14-03-25	Running Shoes	Footwear	60	3	180	Emma Clark	New York	Debit Card	Cancelled
1	ORD0002	20-03-25	Headphones	Electronics	100	4	400	Emily Johnson	San Francisco	Debit Card	Pending
2	ORD0003	15-02-25	Running Shoes	Footwear	60	2	120	John Doe	Denver	Amazon Pay	Cancelled
3	ORD0004	19-02-25	Running Shoes	Footwear	60	3	180	Olivia Wilson	Dallas	Credit Card	Pending
4	ORD0005	10-03-25	Smartwatch	Electronics	150	3	450	Emma Clark	New York	Debit Card	Pending
...	...	...	...	...	...	...	...	...	...	...	...
245	ORD0246	17-03-25	T-Shirt	Clothing	20	2	40	Daniel Harris	Miami	Debit Card	Cancelled
246	ORD0247	30-03-25	Jeans	Clothing	40	1	40	Sophia Miller	Dallas	Debit Card	Cancelled
247	ORD0248	05-03-25	T-Shirt	Clothing	20	2	40	Chris White	Denver	Debit Card	Cancelled
248	ORD0249	08-03-25	Smartwatch	Electronics	150	3	450	Emily Johnson	New York	Debit Card	Cancelled
249	ORD0250	19-02-25	Smartphone	Electronics	500	4	2000	Emily Johnson	Seattle	Amazon Pay	Completed

250 rows × 11 columns

Data Analysis

In [3]:
```
# to check the shape of our dataset
data.shape
```

Out[3]:
```
(250, 11)
```

In [4]:
```
# to check the top 3 rows of the dataset

data.head(3)
```

Out[4]:

	Order ID	Date	Product	Category	Price	Quantity	Total Sales	Customer Name	Customer Location	Payment Method	Status
0	ORD0001	14-03-25	Running Shoes	Footwear	60	3	180	Emma Clark	New York	Debit Card	Cancelled
1	ORD0002	20-03-25	Headphones	Electronics	100	4	400	Emily Johnson	San Francisco	Debit Card	Pending
2	ORD0003	15-02-25	Running Shoes	Footwear	60	2	120	John Doe	Denver	Amazon Pay	Cancelled

In [5]:
```
# to check the last 3 rows of the dataset

data.tail(3)
```

Out[5]:

	Order ID	Date	Product	Category	Price	Quantity	Total Sales	Customer Name	Customer Location	Payment Method	Status
247	ORD0248	05-03-25	T-Shirt	Clothing	20	2	40	Chris White	Denver	Debit Card	Cancelled
248	ORD0249	08-03-25	Smartwatch	Electronics	150	3	450	Emily Johnson	New York	Debit Card	Cancelled
249	ORD0250	19-02-25	Smartphone	Electronics	500	4	2000	Emily Johnson	Seattle	Amazon Pay	Completed

In [6]:
```
# to check the satistics of our dataset
```

```
data.describe().T
```

Out[6]:

	count	mean	std	min	25%	50%	75%	max
Price	250.0	343.580	380.635808	15.0	40.0	150.0	600.0	1200.0
Quantity	250.0	2.856	1.429489	1.0	2.0	3.0	4.0	5.0
Total Sales	250.0	975.380	1252.112254	15.0	100.0	400.0	1500.0	6000.0

In [7]:

```
# to check the information of the dataset

data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 250 entries, 0 to 249
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Order ID           250 non-null    object
 1   Date               250 non-null    object
 2   Product            250 non-null    object
 3   Category           250 non-null    object
 4   Price              250 non-null    int64
 5   Quantity           250 non-null    int64
 6   Total Sales        250 non-null    int64
 7   Customer Name      250 non-null    object
 8   Customer Location  250 non-null    object
 9   Payment Method     250 non-null    object
 10  Status             250 non-null    object
dtypes: int64(3), object(8)
memory usage: 21.6+ KB
```

In [8]:

```
# to check is there any null values are present in ourdatset

data.isnull().sum()
```

Out[8]:
```
Order ID            0
Date                0
Product             0
Category            0
Price               0
Quantity            0
Total Sales         0
Customer Name       0
Customer Location   0
Payment Method      0
Status              0
dtype: int64
```

In [9]:
```
# to check the datatypes of each features

data.dtypes
```

Out[9]:
```
Order ID            object
Date                object
Product             object
Category            object
Price                int64
Quantity             int64
Total Sales          int64
Customer Name       object
Customer Location   object
Payment Method      object
Status              object
dtype: object
```

In [10]:
```
# to check the duplicated values are present in our dataset or not

data.duplicated().sum()
```

Out[10]:
```
0
```

last step for our analysis is to create a new features for the better analysis & change the dtypes whenever is needed.

In [11]:
```
# to change the dtype of the Date column

data.Date = pd.to_datetime(data.Date)
```

In [12]:
```
# to check the dtype is change or not

data.Date
```

Out[12]:
```
0     2025-03-14
1     2025-03-20
2     2025-02-15
3     2025-02-19
4     2025-10-03
         ...
245   2025-03-17
246   2025-03-30
247   2025-05-03
248   2025-08-03
249   2025-02-19
Name: Date, Length: 250, dtype: datetime64[ns]
```

In [13]:
```
# create a day column for the analysis

data["Day"] = data.Date.dt.day
```

In [14]:
```
# create a month column for the analysis

data["Month"] = data.Date.dt.month
```

In [15]:
```
# create a year column for the analysis but we know that here is only 2025 data but here i created for the practice purpose on

data["Year"] = data.Date.dt.year
```

In [16]:
```
# to check is there new columns is created or not

data.head(3)
```

Out[16]:

	Order ID	Date	Product	Category	Price	Quantity	Total Sales	Customer Name	Customer Location	Payment Method	Status	Day	Month	Year
0	ORD0001	2025-03-14	Running Shoes	Footwear	60	3	180	Emma Clark	New York	Debit Card	Cancelled	14	3	2025
1	ORD0002	2025-03-20	Headphones	Electronics	100	4	400	Emily Johnson	San Francisco	Debit Card	Pending	20	3	2025
2	ORD0003	2025-02-15	Running Shoes	Footwear	60	2	120	John Doe	Denver	Amazon Pay	Cancelled	15	2	2025

In [17]:

```
# print all the column names for the idea

data.columns
```

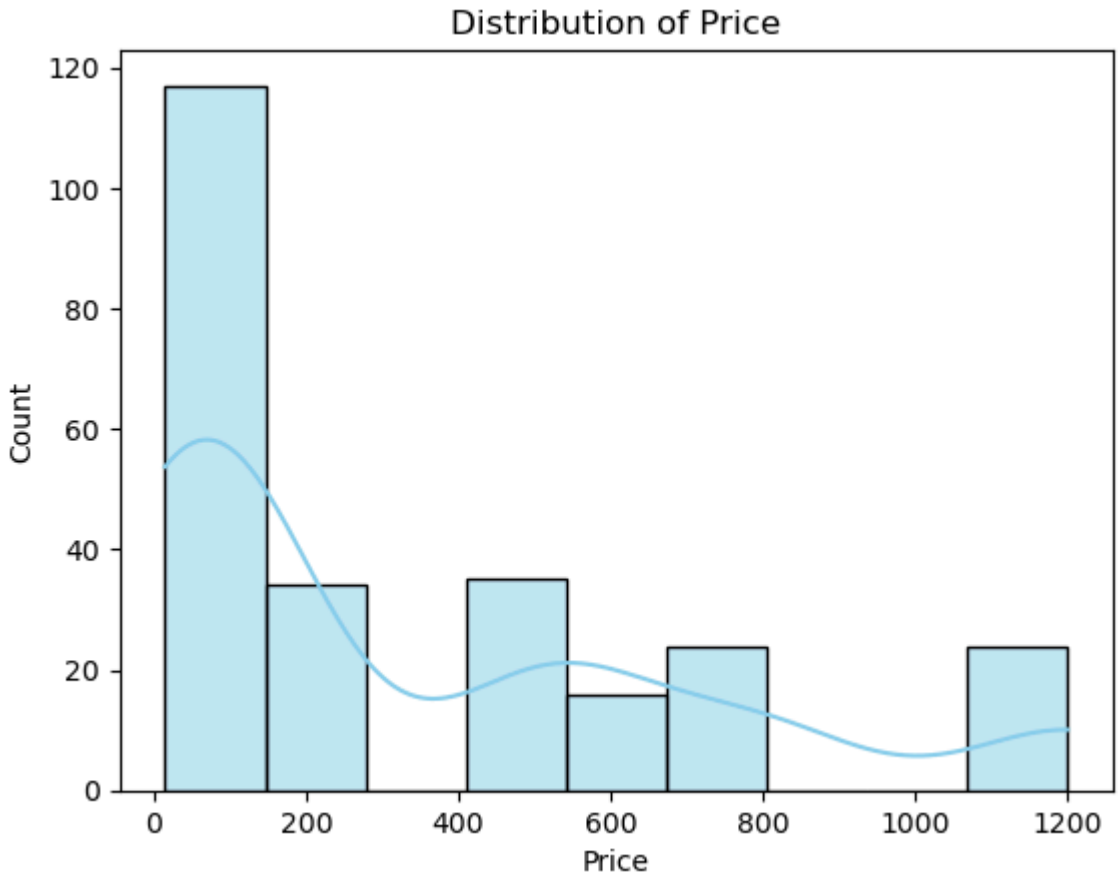
Out[17]:

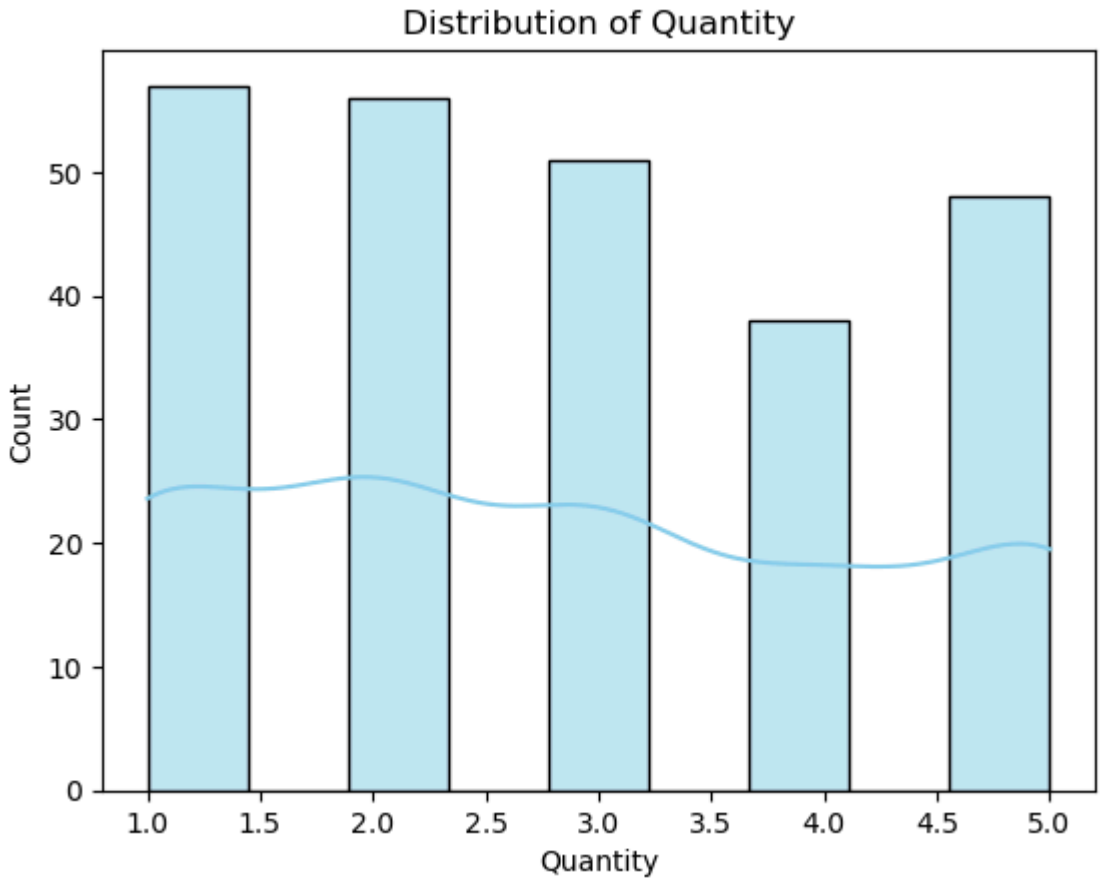
```
Index(['Order ID', 'Date', 'Product', 'Category', 'Price', 'Quantity',
       'Total Sales', 'Customer Name', 'Customer Location', 'Payment Method',
       'Status', 'Day', 'Month', 'Year'],
      dtype='object')
```

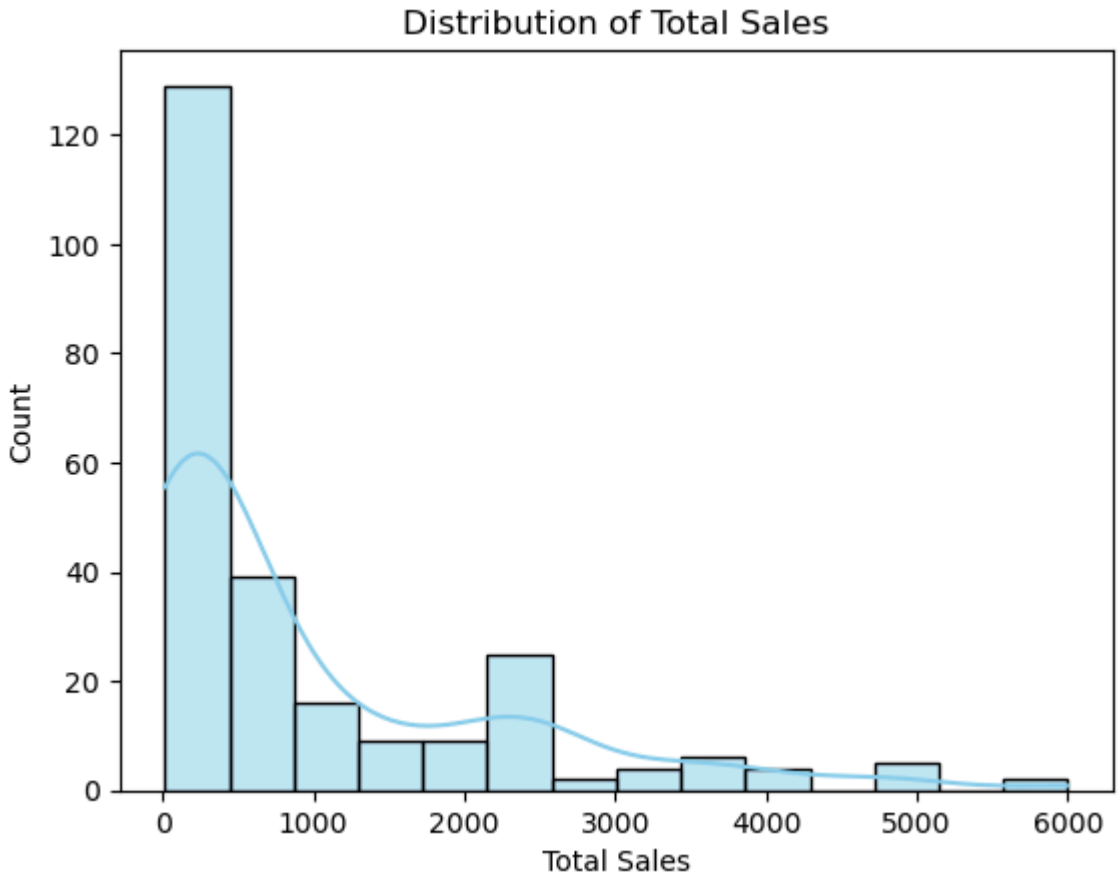
Uni-Variate Analysis

In [46]:

```
# 📈 Distribution for numeric columns
numeric_cols = data.select_dtypes(include=['int64', 'float64']).columns
for col in numeric_cols:
    plt.figure()
    sns.histplot(data[col], kde=True, color='skyblue')
    plt.title(f"Distribution of {col}")
    plt.show()
```


Distribution of Price
Count
Price
0
20
40
60
80
100
120
0
200
400
600
800
1000
1200


Distribution of Quantity
Count
0
10
20
30
40
50
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Quantity



Price and Total Sales show a positively skewed (Right Schewed) pattern, where the majority of transactions fall in the lower range, while fewer high-value transactions stretch the tail towards to the end.

In [50]:
```
# 🔎 Highest, Lowest & Average for Price, Quantity, Total Sales
metrics = ['Price', 'Quantity', 'Total Sales']

for col in metrics:
```

```
    highest = data[col].max()
    lowest = data[col].min()
    average = data[col].mean()

    print(f"\n📊 {col} Analysis:")
    print(f"    🔺 Highest {col}: {highest}")
    print(f"    🔻 Lowest {col}: {lowest}")
    print(f"    📉 Average {col}: {average:.2f}")
```

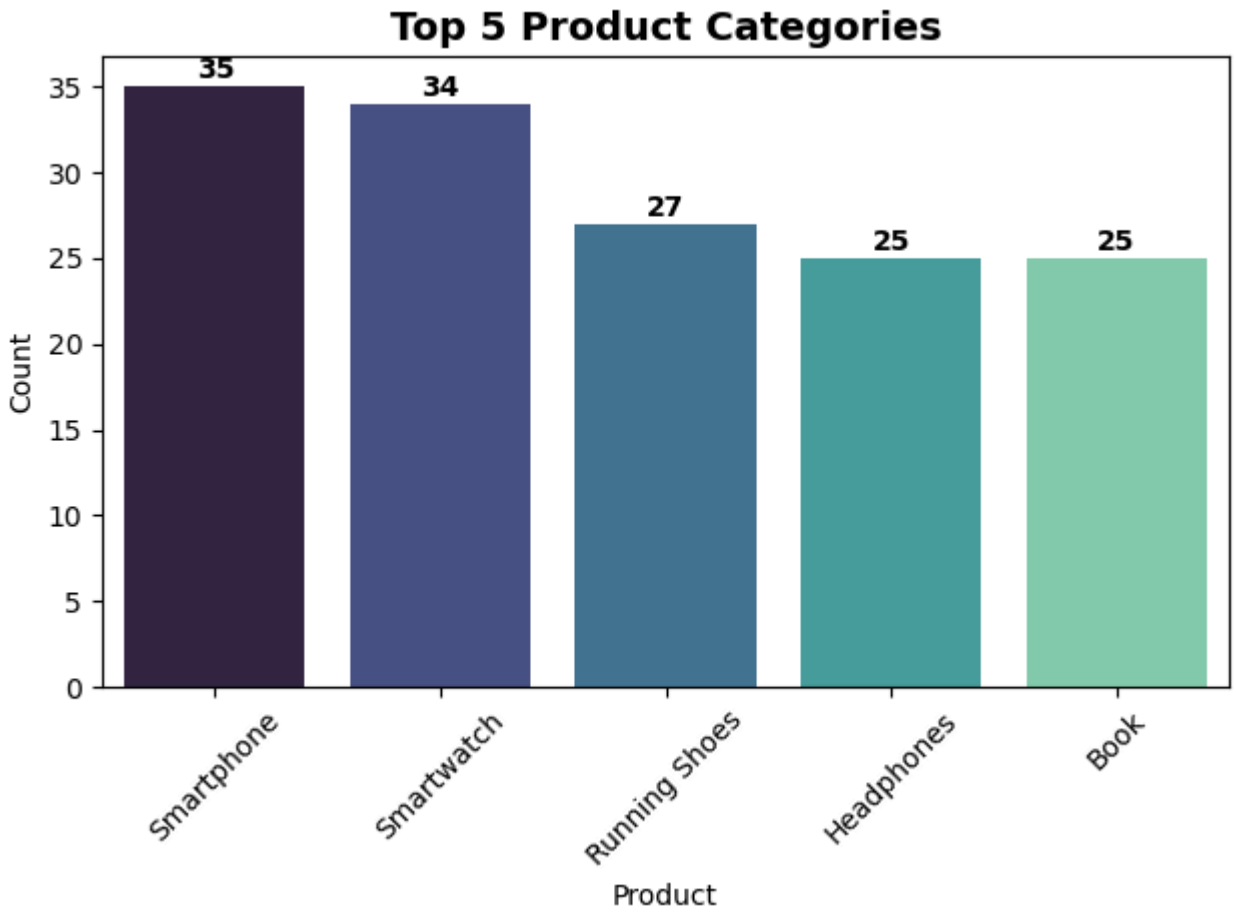
```
📊 Price Analysis:
    🔺 Highest Price: 1200
    🔻 Lowest Price: 15
    📉 Average Price: 343.58

📊 Quantity Analysis:
    🔺 Highest Quantity: 5
    🔻 Lowest Quantity: 1
    📉 Average Quantity: 2.86

📊 Total Sales Analysis:
    🔺 Highest Total Sales: 6000
    🔻 Lowest Total Sales: 15
    📉 Average Total Sales: 975.38
```

In [19]:

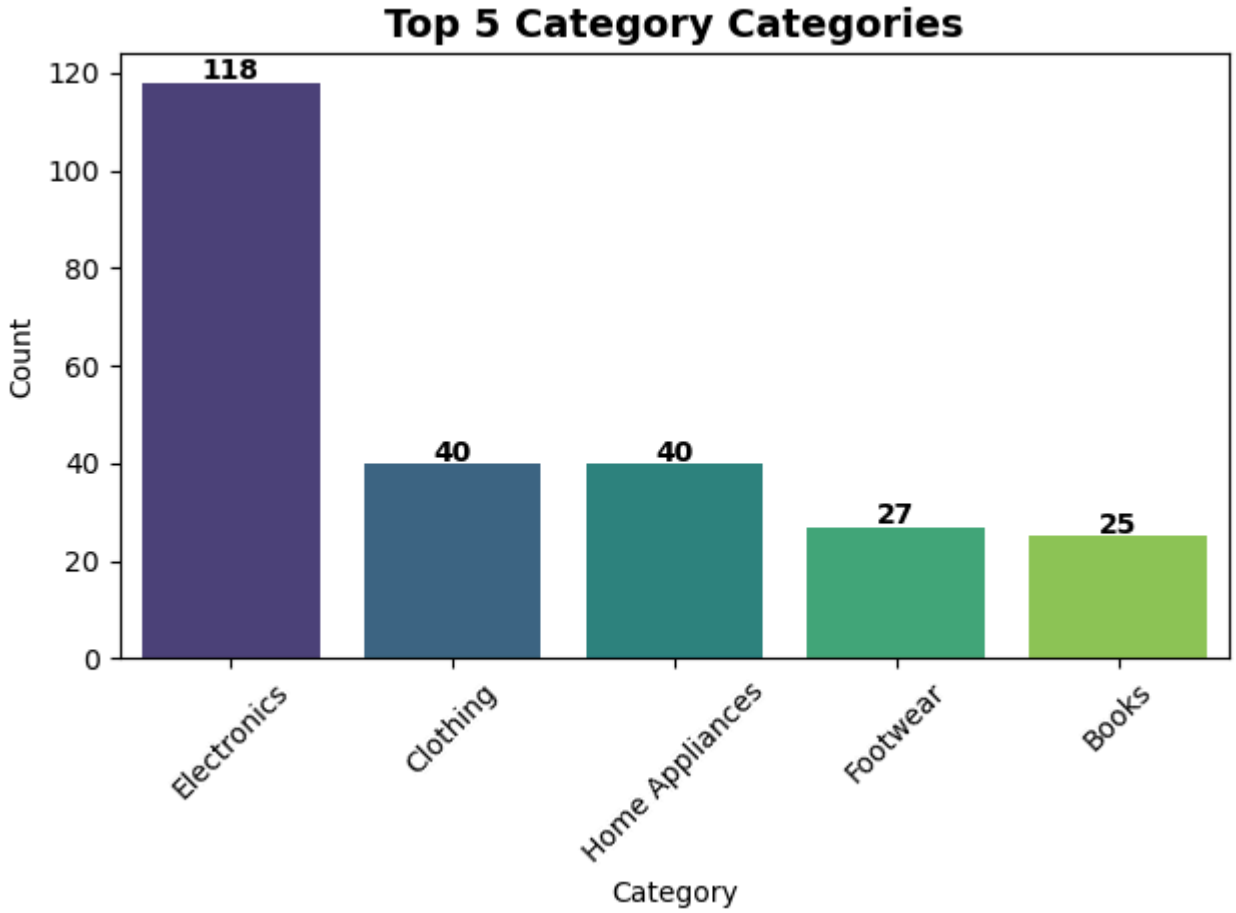
```
# Product
top5 = data['Product'].value_counts().nlargest(5)
ax = sns.barplot(x=top5.index, y=top5.values, palette='mako')
for i, v in enumerate(top5.values):
    ax.text(i, v + 0.5, str(v), ha='center', fontsize=10, fontweight='bold')
plt.title("Top 5 Product Categories", fontsize=14, fontweight='bold')
plt.xlabel("Product")
plt.ylabel("Count")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Smartphones rank among the top five product categories across all categories.

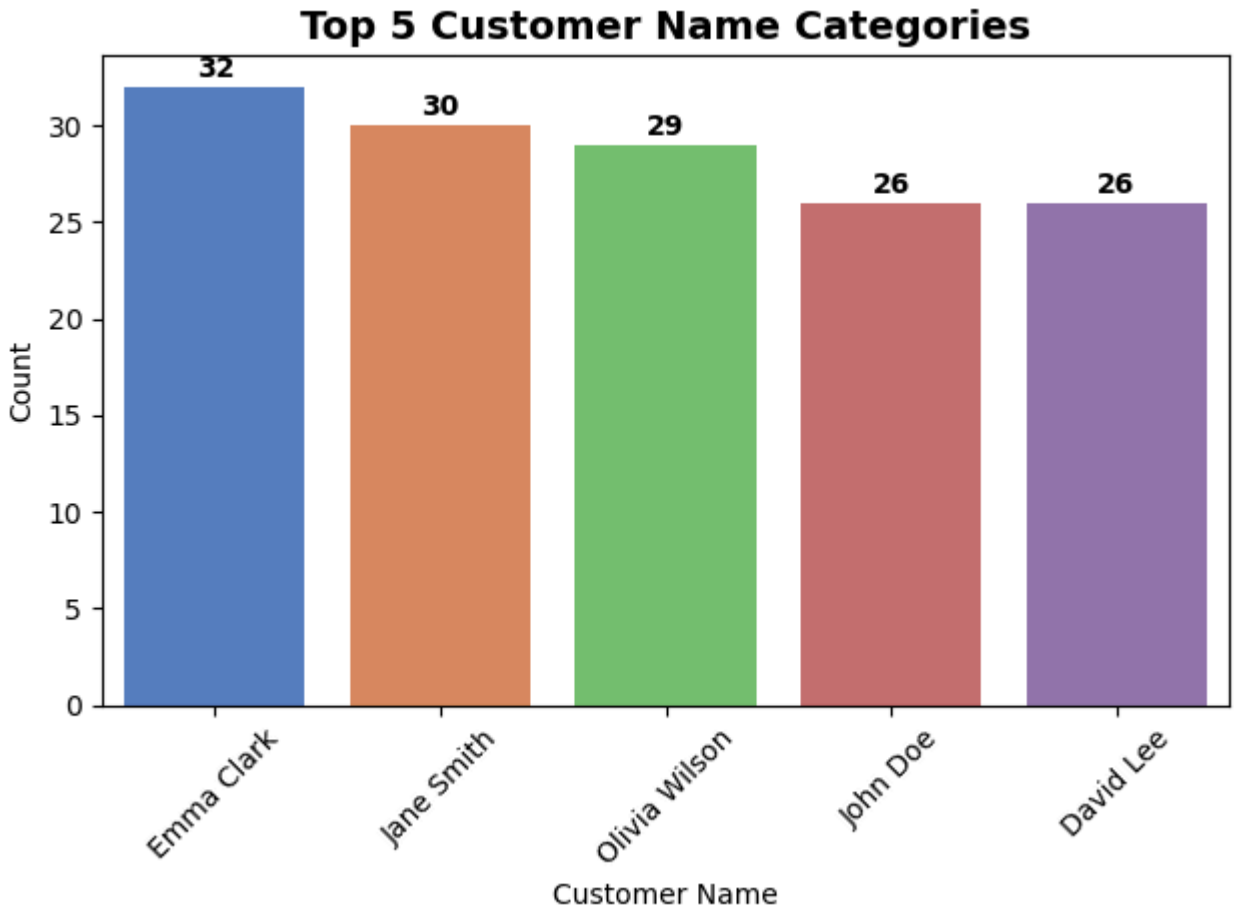
In [31]:
```
# Category
top5 = data['Category'].value_counts().nlargest(5)
ax = sns.barplot(x=top5.index, y=top5.values, palette='viridis')
for i, v in enumerate(top5.values):
    ax.text(i, v + 0.5, str(v), ha='center', fontsize=10, fontweight='bold')
plt.title("Top 5 Category Categories", fontsize=14, fontweight='bold')
```

```
plt.xlabel("Category")
plt.ylabel("Count")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



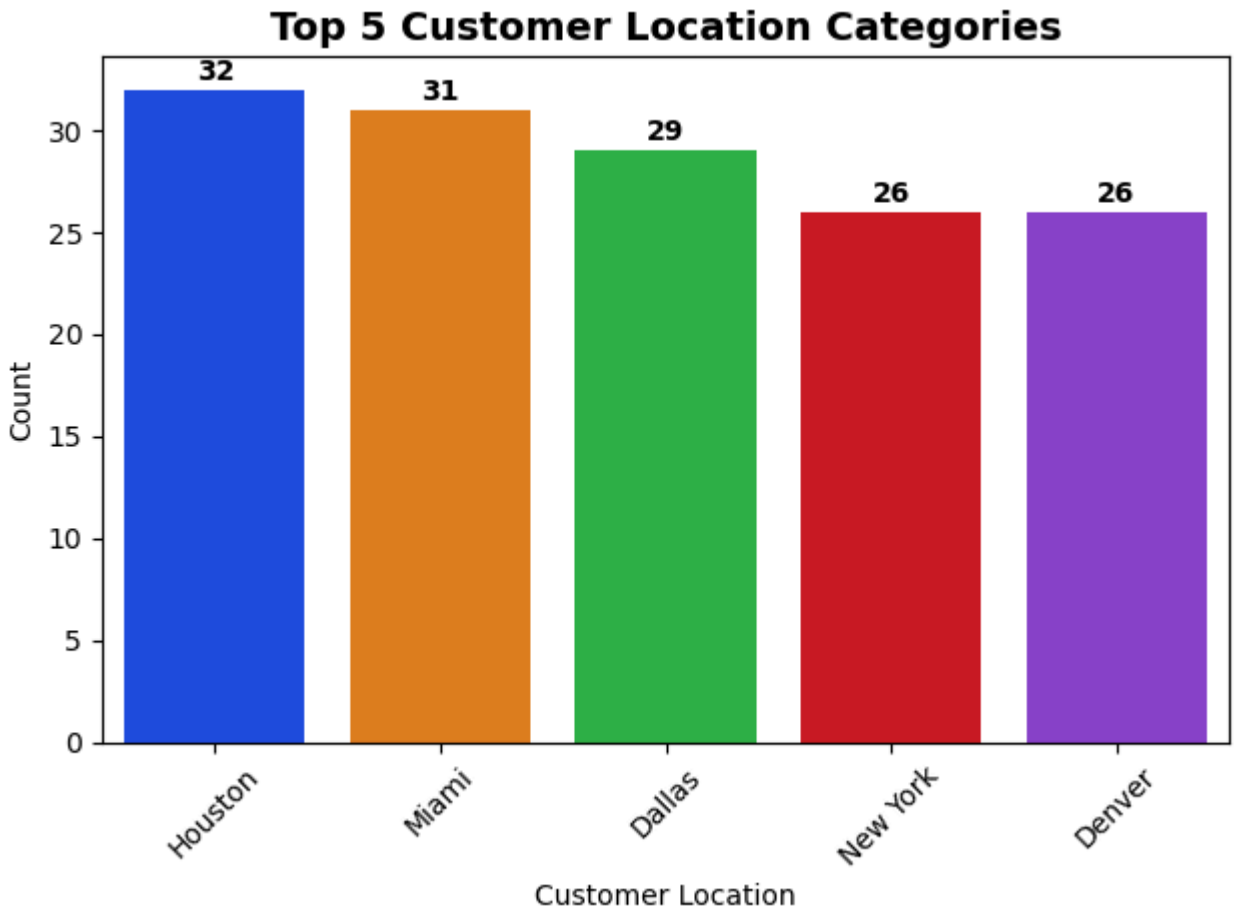
The Electronics category ranks first among all product categories, while Smartphones fall within the top five.

In [32]:
```
# Customer Name
top5 = data['Customer Name'].value_counts().nlargest(5)
ax = sns.barplot(x=top5.index, y=top5.values, palette='muted')
for i, v in enumerate(top5.values):
    ax.text(i, v + 0.5, str(v), ha='center', fontsize=10, fontweight='bold')
plt.title("Top 5 Customer Name Categories", fontsize=14, fontweight='bold')
plt.xlabel("Customer Name")
plt.ylabel("Count")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Among all customers, Emma Clark ranks first in the top five Customer Categories.

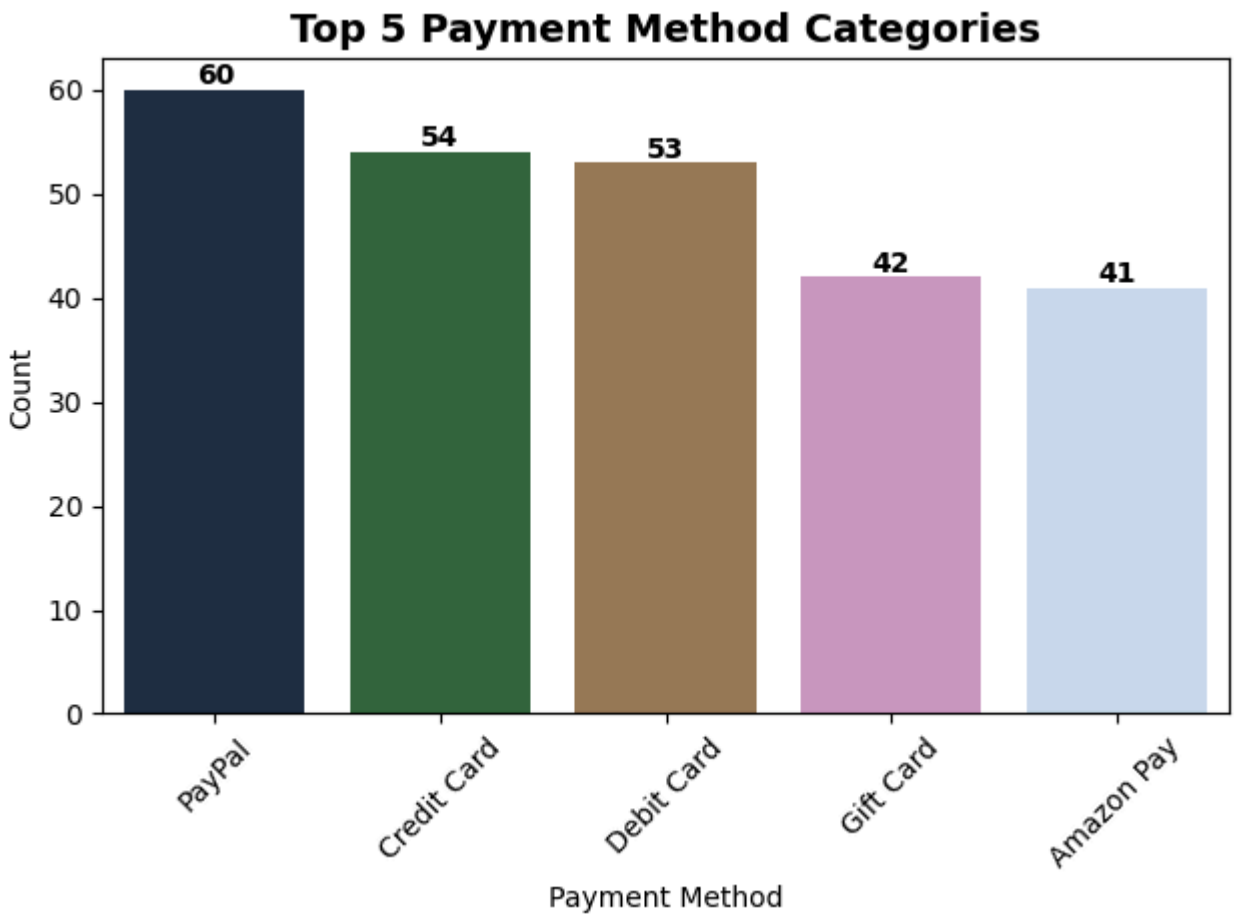
In [33]:
```
# Customer Location
top5 = data['Customer Location'].value_counts().nlargest(5)
ax = sns.barplot(x=top5.index, y=top5.values, palette='bright')
for i, v in enumerate(top5.values):
    ax.text(i, v + 0.5, str(v), ha='center', fontsize=10, fontweight='bold')
plt.title("Top 5 Customer Location Categories", fontsize=14, fontweight='bold')
plt.xlabel("Customer Location")
plt.ylabel("Count")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Houston ranks first among the top five customer locations.

In [34]:
```
# Payment Method
top5 = data['Payment Method'].value_counts().nlargest(5)
ax = sns.barplot(x=top5.index, y=top5.values, palette='cubehelix')
for i, v in enumerate(top5.values):
    ax.text(i, v + 0.5, str(v), ha='center', fontsize=10, fontweight='bold')
plt.title("Top 5 Payment Method Categories", fontsize=14, fontweight='bold')
plt.xlabel("Payment Method")
plt.ylabel("Count")
```

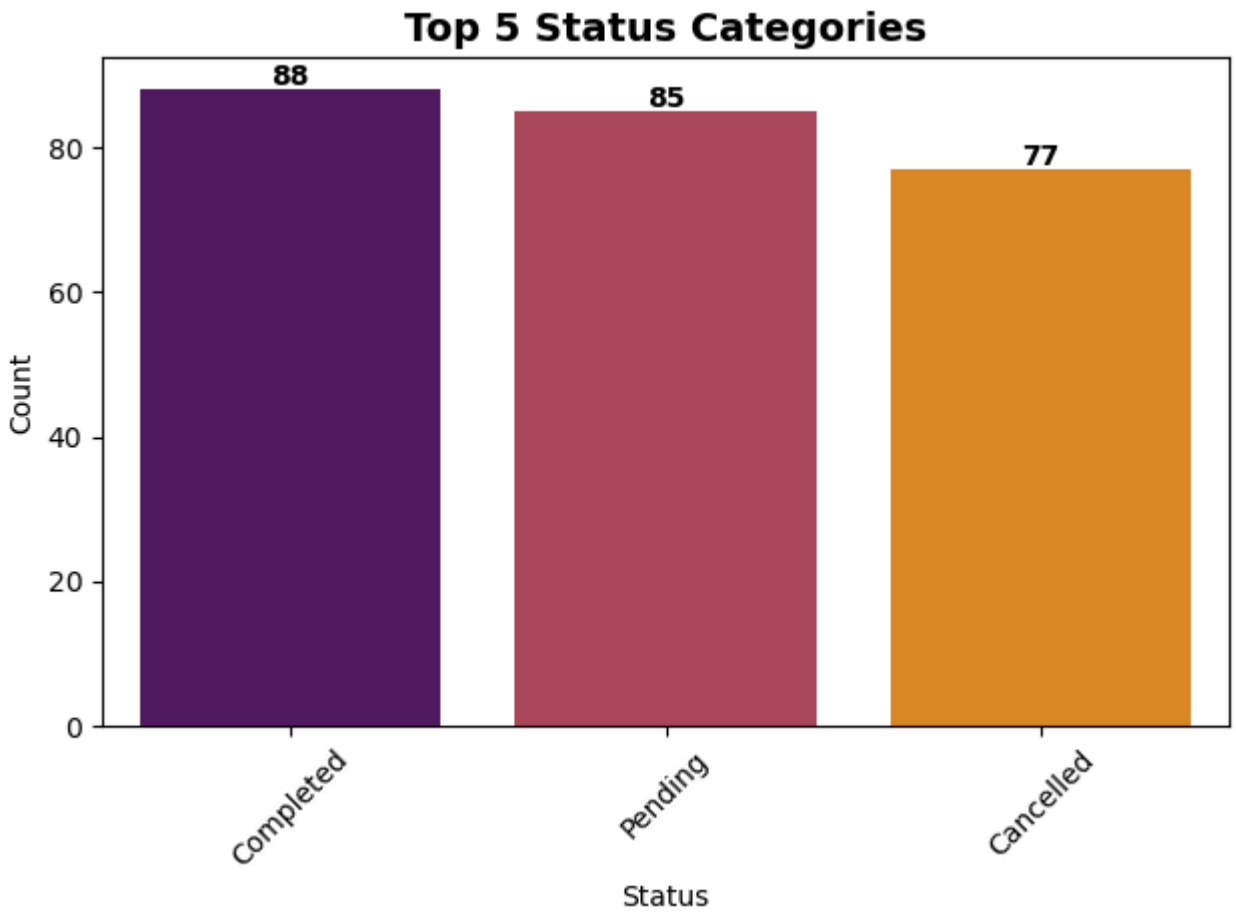
```
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Among the top five payment methods, PayPal holds the leading position.

In [37]:
```
# Status
top5 = data['Status'].value_counts().nlargest(5)
```

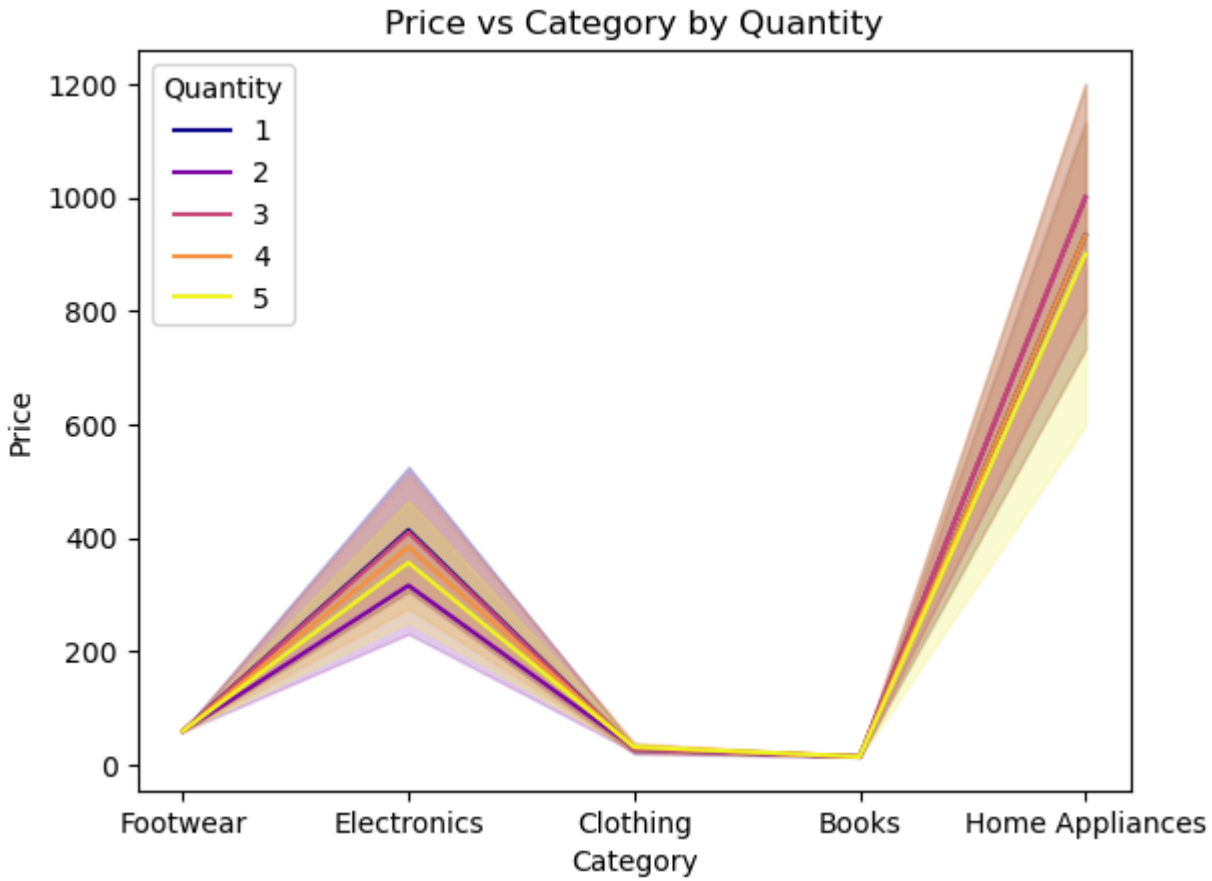
```python
ax = sns.barplot(x=top5.index, y=top5.values, palette='inferno')
for i, v in enumerate(top5.values):
    ax.text(i, v + 0.5, str(v), ha='center', fontsize=10, fontweight='bold')
plt.title("Top 5 Status Categories", fontsize=14, fontweight='bold')
plt.xlabel("Status")
plt.ylabel("Count")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Among the top five status categories, Completed holds the leading position.

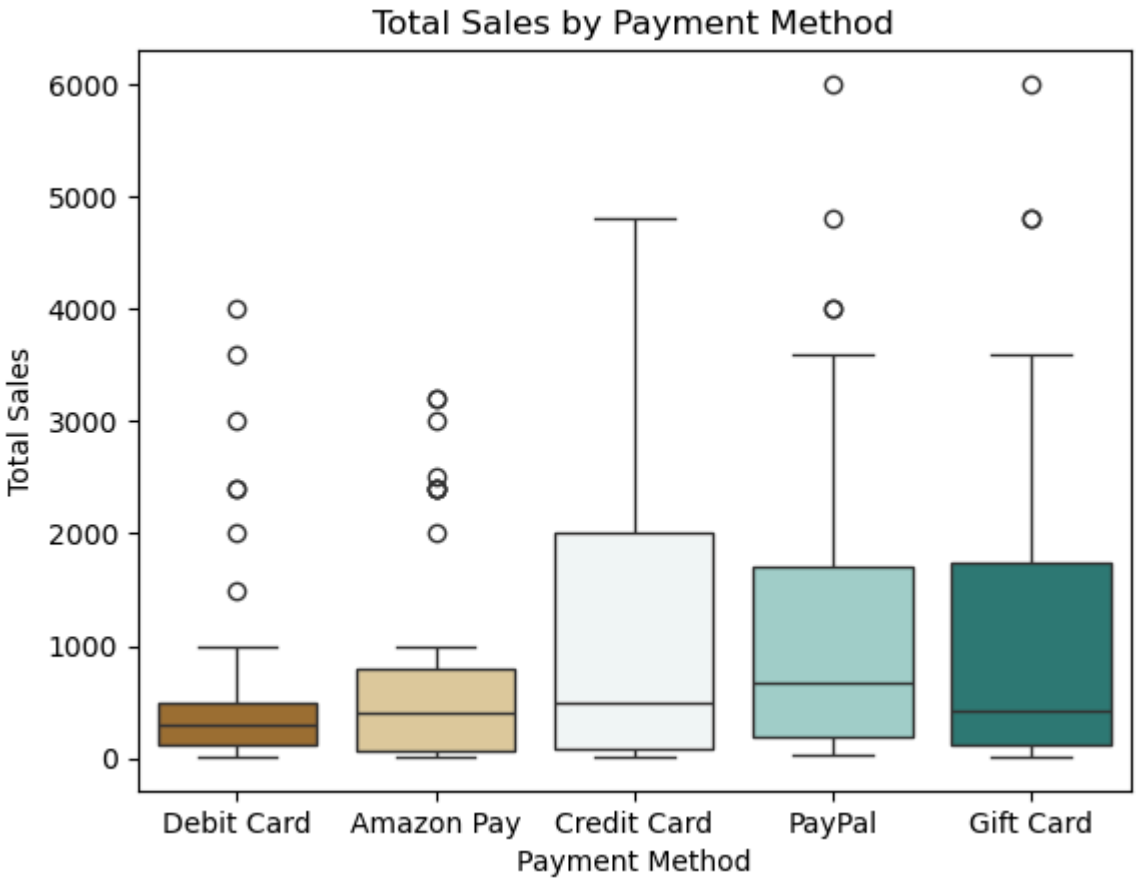
Bi-Variate Analysis

In [49]:
```
# $ Price vs Quantity
plt.figure()
sns.lineplot(data=data, x='Category', y='Price', hue='Quantity', palette='plasma')
plt.title("Price vs Category by Quantity")
plt.show()
```



Compared to other categories, Home Appliances exhibit the highest pricing as well as the highest sales volume.

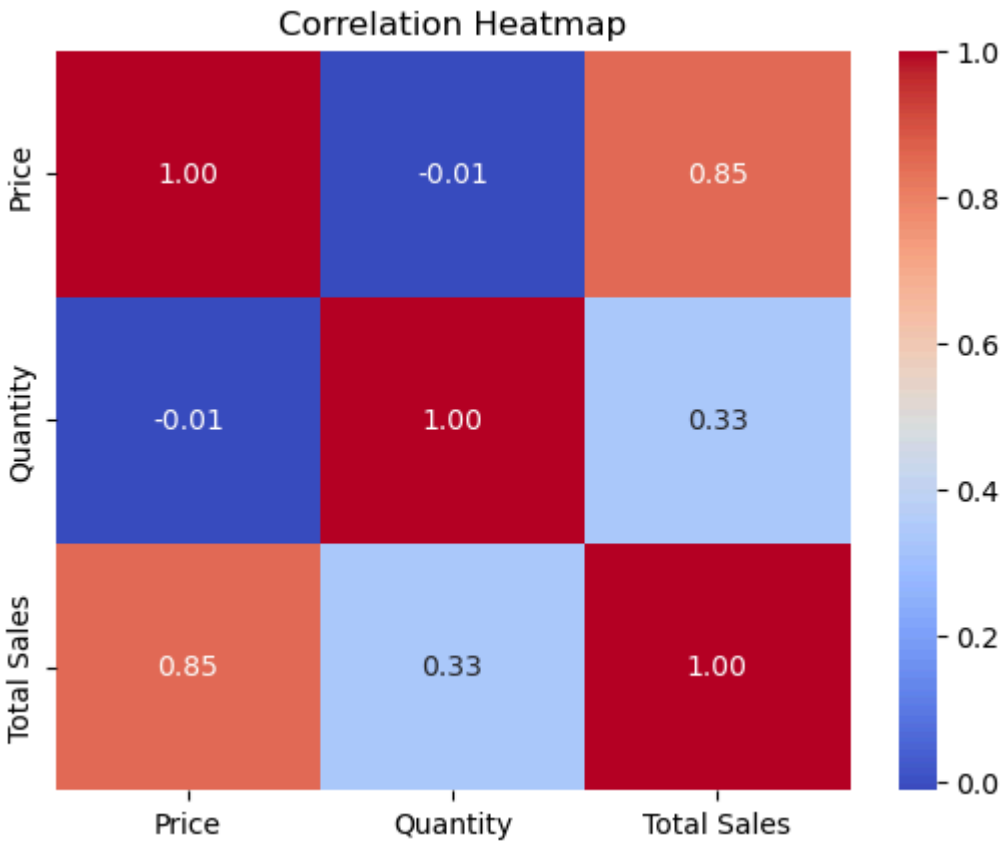
In [40]:
```
# 💳 Total Sales by Payment Method (Boxplot)
plt.figure()
sns.boxplot(data=data, x='Payment Method', y='Total Sales', palette='BrBG')
plt.title("Total Sales by Payment Method")
plt.show()
```



In the boxplot of Total Sales by Payment Method, Debit Card transactions show a higher number of outliers, while Credit Card transactions have none. From an amount perspective, PayPal and Gift Card transactions exhibit relatively higher outliers.

Importantly, for business analysis, these outliers were not removed as they may represent genuine transactions.

In [27]:
```
# 🔥 Correlation Heatmap
plt.figure()
corr = data[numeric_cols].corr()
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```

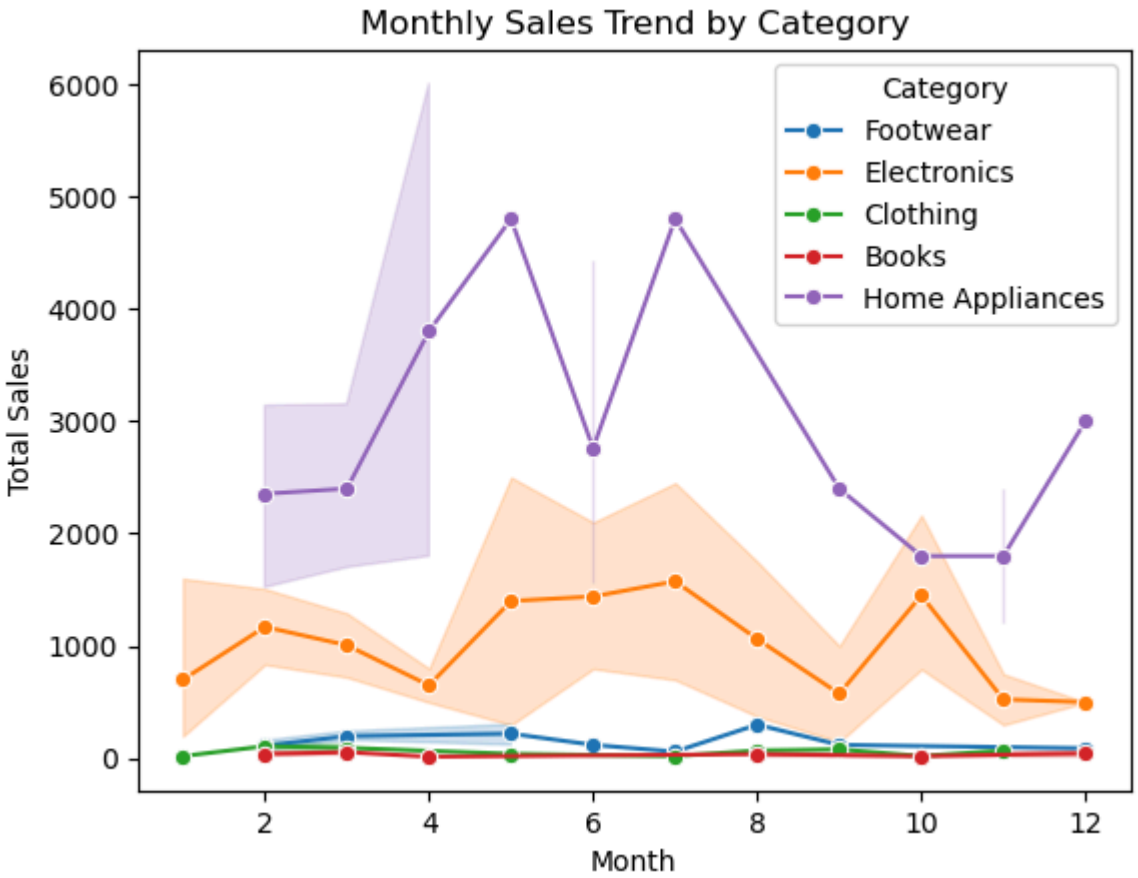


The correlation analysis shows that Price and Quantity have almost no relationship (–0.01), indicating that higher prices do not influence purchased quantities. Price and Total Sales are strongly

positively correlated (0.85), which is expected as sales directly depend on price. Quantity and Total Sales have a moderate positive correlation (0.33), suggesting that an increase in quantity contributes to sales but not as strongly as price.

Multi-Variate Analysis

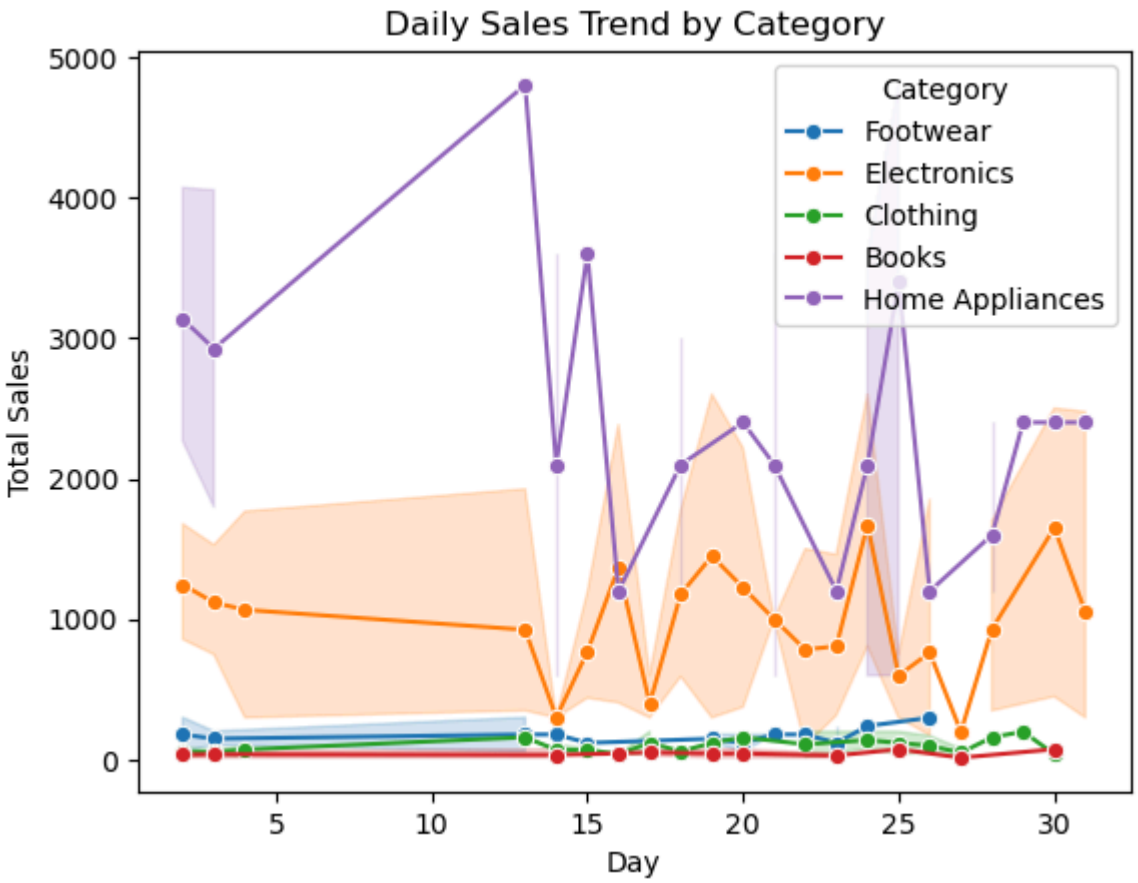
In [28]:
```
# 📅 Monthly Sales Trend by Category
plt.figure()
sns.lineplot(data=data, x='Month', y='Total Sales', hue='Category', marker='o')
plt.title("Monthly Sales Trend by Category")
plt.show()
```



The monthly sales analysis by category reveals that Home Appliances consistently generate the highest total sales, while Electronics maintain an average level of sales compared to other categories.

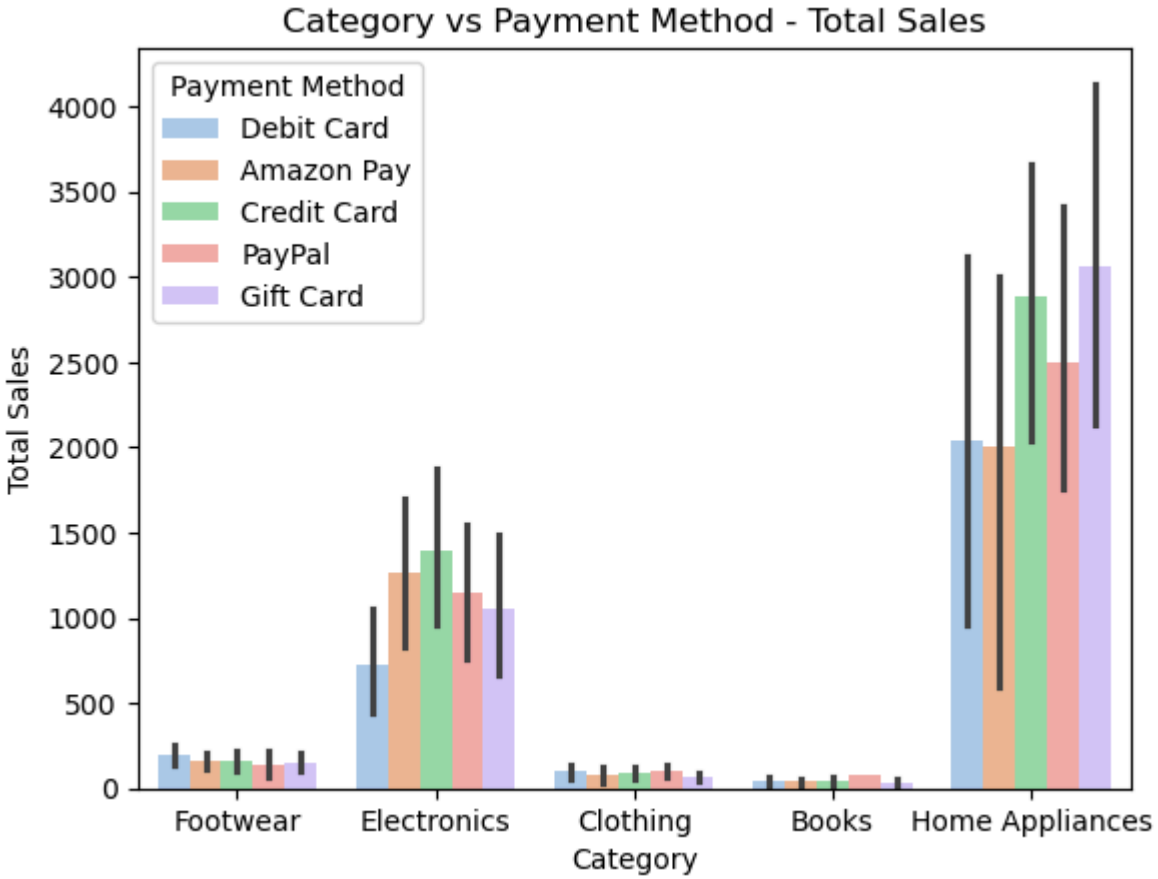
In [41]:
```
# 📅 Daily Sales Trend by Category
plt.figure()
sns.lineplot(data=data, x='Day', y='Total Sales', hue='Category', marker='o')
```

```
plt.title("Daily Sales Trend by Category")
plt.show()
```



The day-wise analysis of total sales by category shows a similar trend, where Home Appliances continue to dominate with the highest sales, while Electronics remain at an average level compared to other categories.

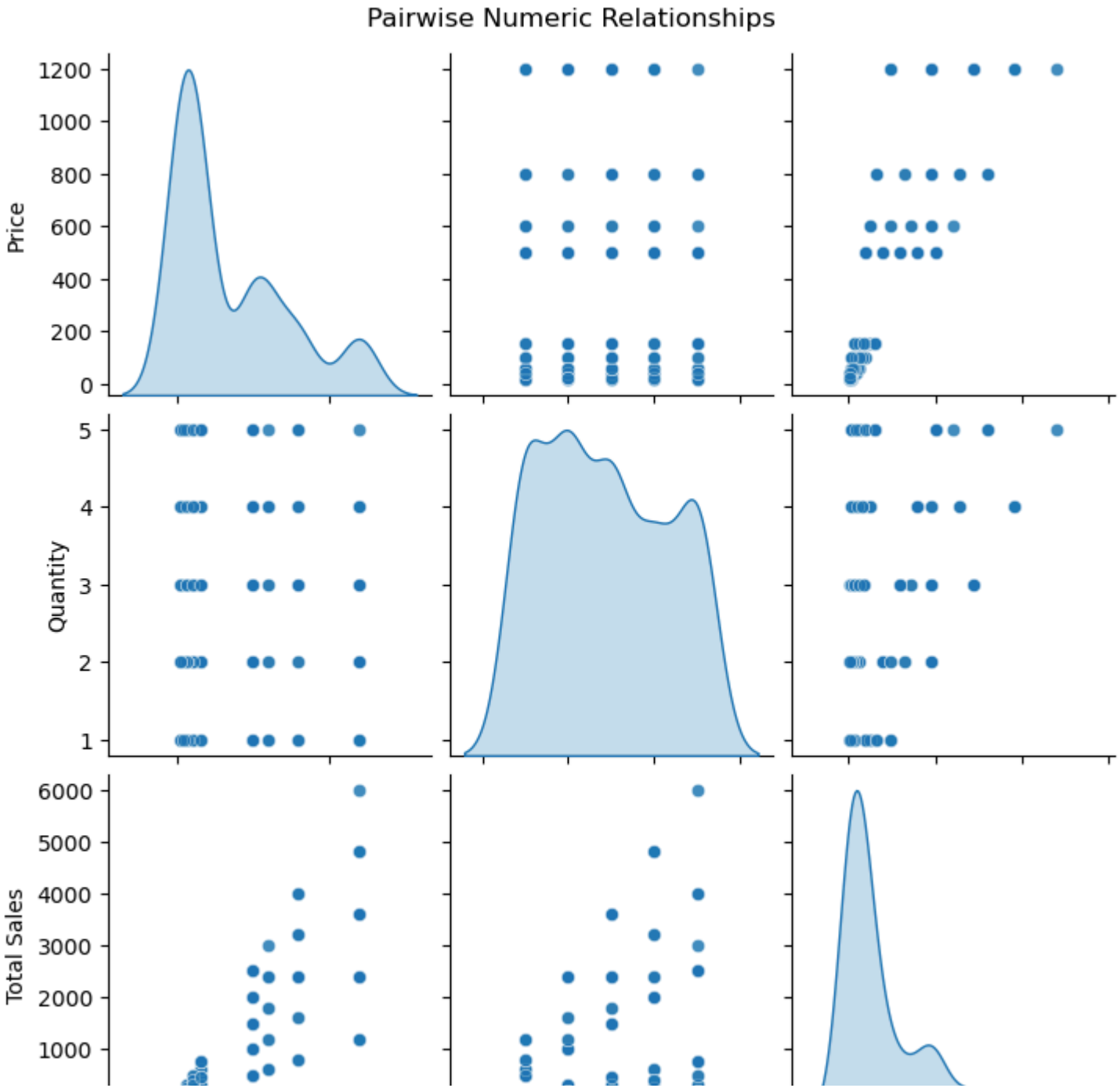
In [29]:
```
# Category vs Payment Method vs Total Sales
plt.figure()
sns.barplot(data=data, x='Category', y='Total Sales', hue='Payment Method', palette='pastel')
plt.title("Category vs Payment Method - Total Sales")
plt.show()
```

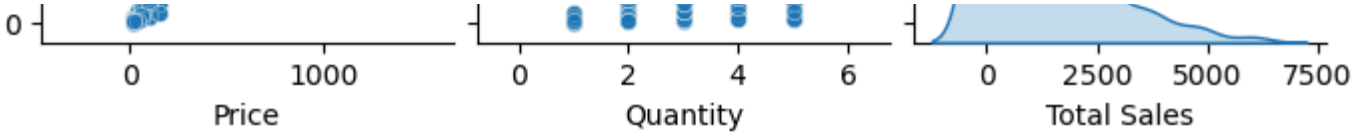


In the analysis of categories, total sales, and payment methods, Credit Card emerges as the most frequently used payment option

for Electronics purchases, whereas Gift Cards are predominantly used for Home Appliances transactions.

In [30]:
```
# 🔍 Pairplot for numeric relationships
sns.pairplot(data[numeric_cols], diag_kind='kde', plot_kws={'alpha':0.6})
plt.suptitle("Pairwise Numeric Relationships", y=1.02)
plt.show()
```


Pairwise Numeric Relationships
Price
Quantity
Total Sales
1200
1000
800
600
400
200
0
5
4
3
2
1
6000
5000
4000
3000
2000
1000



🚀 Sales Data Analysis – Insights & Findings

📌 Univariate Analysis

- 🛒 **Product Category** → *Smartphones* are the **top product line**.
- 💻 **Category** → *Electronics* dominate, followed by *Home Appliances*.
- 👧 **Customer Name** → *Emma Clark* ranks **first among frequent buyers**.
- 🌎 **Customer Location** → *Houston* has the **highest customer presence**.
- 💳 **Payment Method** → *PayPal* is the **most preferred payment option**.
- ✅ **Order Status** → *Completed* orders dominate overall.
- 📈 **Price & Quantity** → *Home Appliances* show **higher price + higher quantity** → strong revenue driver.
- 📊 **Distribution** → *Price* & *Total Sales* are **right-skewed** (most values low, few very high).

🔗 Bivariate Analysis

- 📦 **Boxplot – Total Sales vs Payment Method**
  - *Debit Card* → highest **number of outliers**.
  - *Credit Card* → **no outliers** observed.
  - *PayPal & Gift Card* → largest **high-value outliers**.
  - ⚠️ Outliers **not removed** (they may represent real business opportunities).
- 🔥 **Correlation Heatmap**
  - *Price vs Total Sales* → **Strong Positive (0.85)** ✅

- *Quantity vs Total Sales* → **Moderate Positive (0.33)**
- *Price vs Quantity* → **No correlation (-0.01)** ❌

🎯 Multivariate Analysis

- 🗓️ **Monthly & Daily Sales by Category**
  - *Home Appliances* → consistently **highest sales**.
  - *Electronics* → maintain **average sales trend**.
- 💳 **Category × Payment Method × Sales**
  - *Electronics* → **Credit Card** dominates.
  - *Home Appliances* → **Gift Card** most used.

💡 Business Takeaways

✨ *Home Appliances* = **key revenue driver** → high prices + high quantities.
📱 *Smartphones* = **top product line** → must be prioritized.
💳 *Payment Preferences* vary: *PayPal overall, Credit Card for Electronics, Gift Card for Home Appliances*.
📈 Outliers highlight **big-ticket sales** → should be leveraged, not discarded.
