US Retail SuperStore Sales Data Analysis

Objective:

Analyze retail sales data to derive insights into customer behavior, popular products, and sales trends.

Dataset:

Superstore Sales Datase

About DataSet:

Row ID: An identifier for each row in the dataset.

Order ID: Unique identifier for each order.

Order Date: Date when the order was placed.

Ship Date: Date when the order was shipped.

Ship Mode: Method used for shipping the order.

Customer ID: Unique identifier for each customer.

Customer Name: Name of the customer.

Segment: Market segment to which the customer belongs (e.g., consumer, corporate, etc.).

Country: Country where the order was placed.

City: City where the order was placed.

State: State where the order was placed.

Postal Code: Postal code of the location where the order was placed.

Region: Geographic region where the order was placed.

Product ID: Unique identifier for each product.

Category: Category of the product (e.g., electronics, furniture, etc.).

Sub-Category: Sub-category of the product (e.g., laptops, chairs, etc.).

Product Name: Name of the product.

Sales: Sales amount for the product.

This dataset can be used for various analyses such as sales performance, customer segmentation, shipping analysis, product popularity, and more.

Data Exploration:

Load the dataset into your preferred data analysis tool (e.g., Python with Pandas, Jupyter Notebook, Tableau, etc.). Explore the structure of the dataset, check for missing values, and understand the types of data available.

Data Cleaning:

Handle missing values, duplicates, and any inconsistencies in the data. Convert data types if necessary

In [1]:
```
# importing Libraries

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

In [3]:
```
# loading dataset
```

```
data = pd.read_csv(r"C:\Users\HP\Downloads\Rupal_Data\InternCareer internship\SampleSuperstore.csv")
```

In [4]:
```
data
```

Out[4]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00	41.9136
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	0.00	219.5820
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	2	0.00	6.8714
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	0.45	-383.0310
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	2	0.20	2.5164
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9989	Second Class	Consumer	United States	Miami	Florida	33180	South	Furniture	Furnishings	25.2480	3	0.20	4.1028
9990	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Furniture	Furnishings	91.9600	2	0.00	15.6332
9991	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Technology	Phones	258.5760	2	0.20	19.3932
9992	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Office Supplies	Paper	29.6000	4	0.00	13.3200
9993	Second Class	Consumer	United States	Westminster	California	92683	West	Office Supplies	Appliances	243.1600	2	0.00	72.9480

9994 rows × 13 columns

In [5]:
```
#Display the first few rows of the dataset

data.head(5)
```

Out[5]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00	41.9136
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	0.00	219.5820
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	2	0.00	6.8714
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	0.45	-383.0310
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	2	0.20	2.5164

In [6]:
```
# Display the last few rows of the dataset

data.tail(5)
```

Out[6]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
9989	Second Class	Consumer	United States	Miami	Florida	33180	South	Furniture	Furnishings	25.248	3	0.2	4.1028
9990	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Furniture	Furnishings	91.960	2	0.0	15.6332
9991	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Technology	Phones	258.576	2	0.2	19.3932
9992	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Office Supplies	Paper	29.600	4	0.0	13.3200
9993	Second Class	Consumer	United States	Westminster	California	92683	West	Office Supplies	Appliances	243.160	2	0.0	72.9480

In [7]:
```
data.isnull()   # Check for missing values
```

Out[7]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
0	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9989	False	False	False	False	False	False	False	False	False	False	False	False	False
9990	False	False	False	False	False	False	False	False	False	False	False	False	False
9991	False	False	False	False	False	False	False	False	False	False	False	False	False
9992	False	False	False	False	False	False	False	False	False	False	False	False	False
9993	False	False	False	False	False	False	False	False	False	False	False	False	False

9994 rows × 13 columns

In [10]:
```
print(data.isnull().sum())  # Printing Check for missing values
```

```
Ship Mode       0
Segment         0
Country         0
City            0
State           0
Postal Code     0
Region          0
Category        0
Sub-Category    0
Sales           0
Quantity        0
Discount        0
Profit          0
dtype: int64
```

there is No missing value or error in this dataset therefore data is clean

In [12]:
```
# Get information about the data types and number of non-null values

print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Ship Mode     9994 non-null   object 
 1   Segment       9994 non-null   object 
 2   Country       9994 non-null   object 
 3   City          9994 non-null   object 
 4   State         9994 non-null   object 
 5   Postal Code   9994 non-null   int64  
 6   Region        9994 non-null   object 
 7   Category      9994 non-null   object 
 8   Sub-Category  9994 non-null   object 
 9   Sales         9994 non-null   float64
 10  Quantity      9994 non-null   int64  
 11  Discount      9994 non-null   float64
 12  Profit        9994 non-null   float64
dtypes: float64(3), int64(2), object(8)
memory usage: 1015.1+ KB
None
```

In [14]:
```
# Summary statistics

print(data.describe())
```

```
        Postal Code         Sales     Quantity     Discount       Profit
count   9994.000000   9994.000000  9994.000000  9994.000000  9994.000000
mean   55190.379428    229.858001     3.789574     0.156203    28.656896
std    32063.693350    623.245101     2.225110     0.206452   234.260108
min     1040.000000      0.444000     1.000000     0.000000 -6599.978000
25%    23223.000000     17.280000     2.000000     0.000000     1.728750
50%    56430.500000     54.490000     3.000000     0.200000     8.666500
75%    90008.000000    209.940000     5.000000     0.200000    29.364000
max    99301.000000  22638.480000    14.000000     0.800000  8399.976000
```

In [15]:
```
# Handle missing values
# Drop rows with missing values
data.dropna(inplace=True)
```

In [16]:
```
# Remove duplicates
data.drop_duplicates(inplace=True)
```

In [24]:
```
data.dtypes
```

Out[24]:
```
Ship Mode        object
Segment          object
Country          object
City             object
State            object
Postal Code       int64
Region           object
Category         object
Sub-Category     object
Sales           float64
Quantity          int64
Discount        float64
Profit          float64
dtype: object
```

In [26]:
```
# Convert data types if necessary
# Example: Convert 'Sales' column to float
data['Sales'] = data['Sales'].astype(float)
```

In [27]:
```
data.dtypes
```

Out[27]:
```
Ship Mode        object
Segment          object
Country          object
City             object
State            object
Postal Code       int64
Region           object
Category         object
Sub-Category     object
Sales           float64
Quantity          int64
Discount        float64
Profit          float64
dtype: object
```

Descriptive statistics

In [29]:
```
# Descriptive statistics
# Total sales
total_sales = data['Sales'].sum()
print("Total Sales:", total_sales)
```

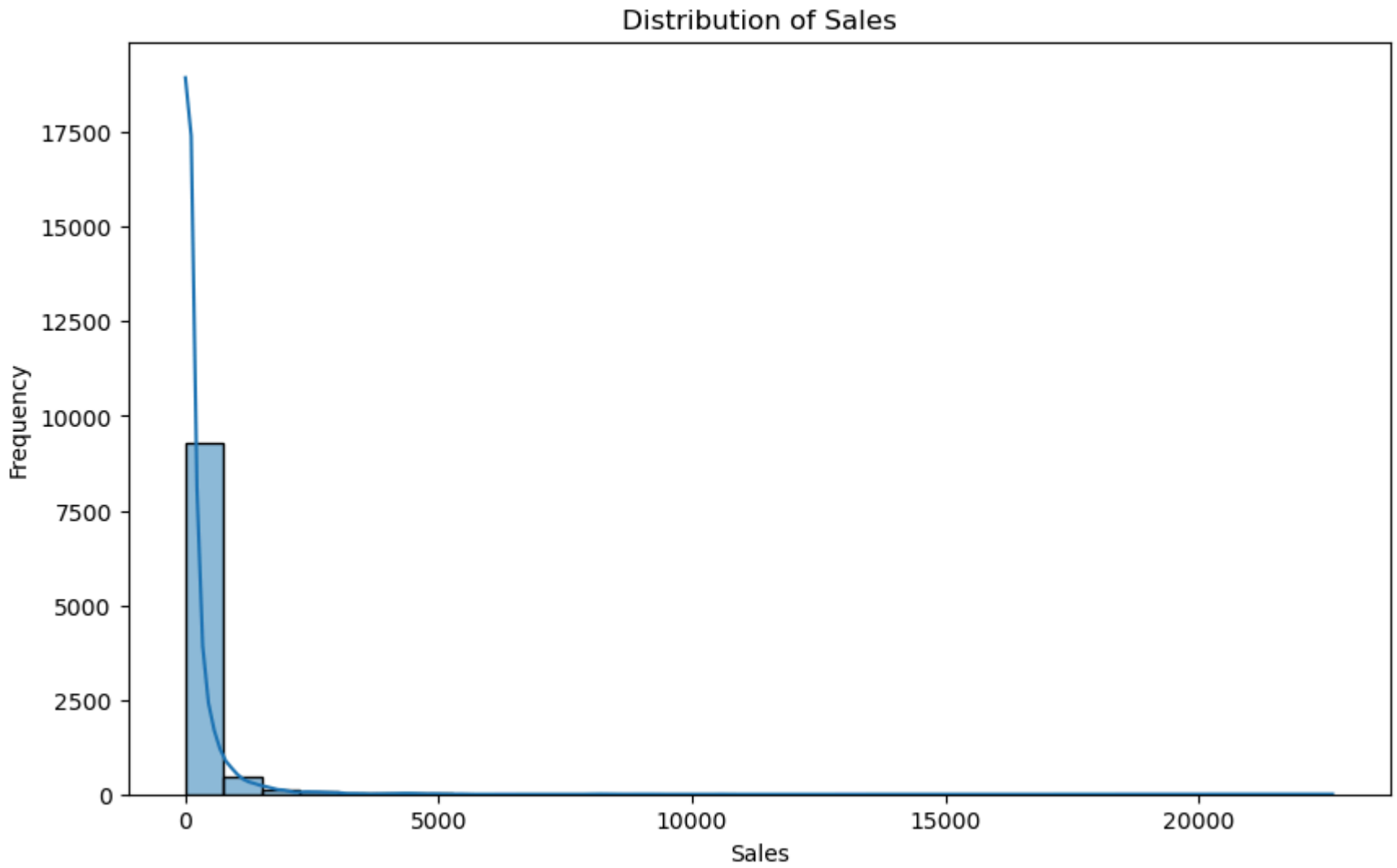
```
Total Sales: 2296195.5903
```

In [30]:
```
# Average order value
average_order_value = data['Sales'].mean()
print("Average Order Value:", average_order_value)
```

```
Average Order Value: 230.14890150345792
```

Data visualization

In [32]:
```
# Visualize the distribution of sales
plt.figure(figsize=(10, 6))
sns.histplot(data['Sales'], bins=30, kde=True)
plt.title('Distribution of Sales')
plt.xlabel('Sales')
plt.ylabel('Frequency')
plt.show()
```



Descriptive Statistics:

Calculate basic descriptive statistics for key metrics such as total sales, average order value, etc.

Visualize the distribution of sales, order quantity, and other relevant metrics.

In [36]:
```
# Calculate total sales
total_sales = data['Sales'].sum()
total_sales
```

Out[36]:
```
2296195.5903
```

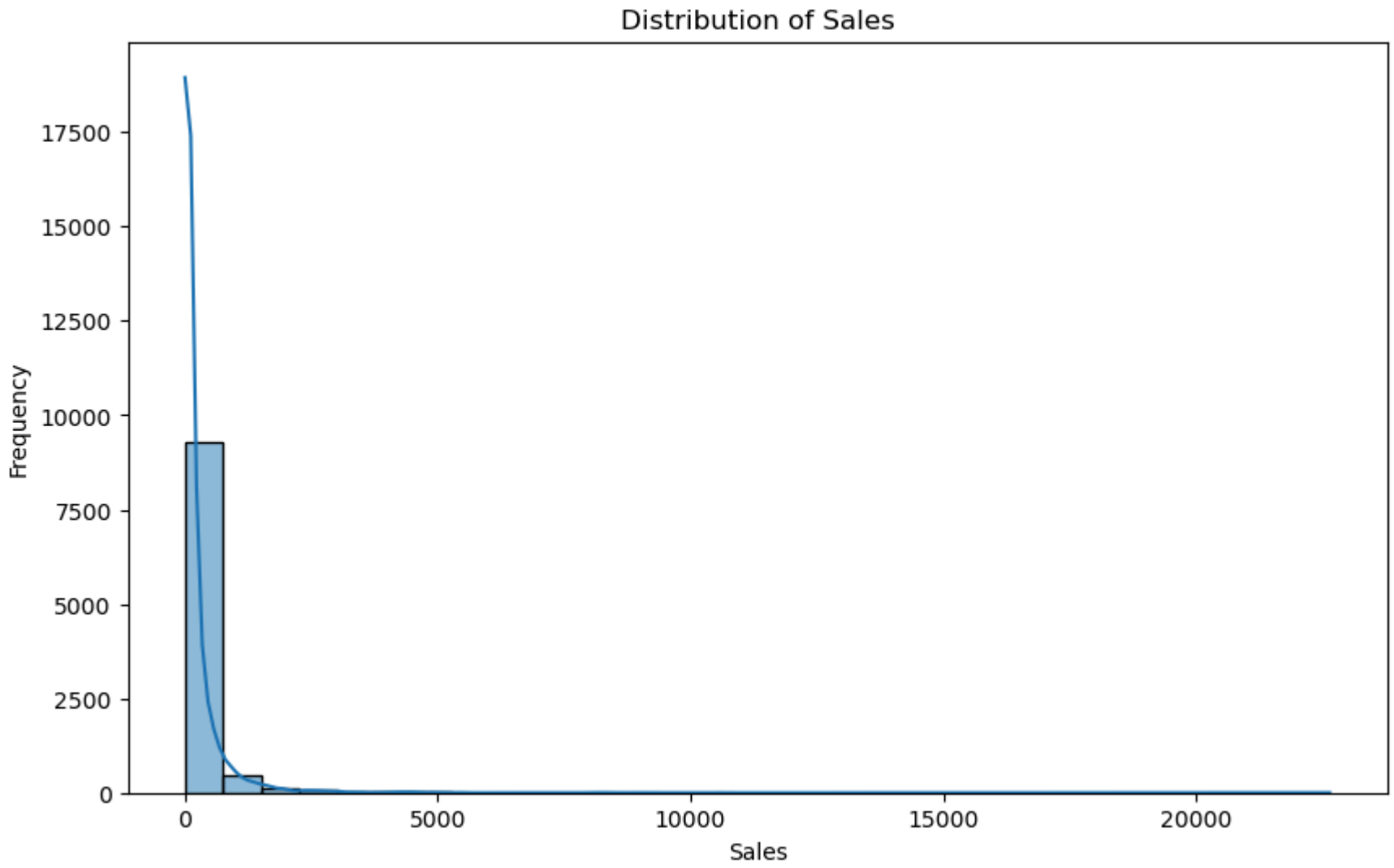
In [37]:
```
# Calculate average order value
average_order_value = data['Sales'].mean()
average_order_value
```

Out[37]:
```
230.14890150345792
```

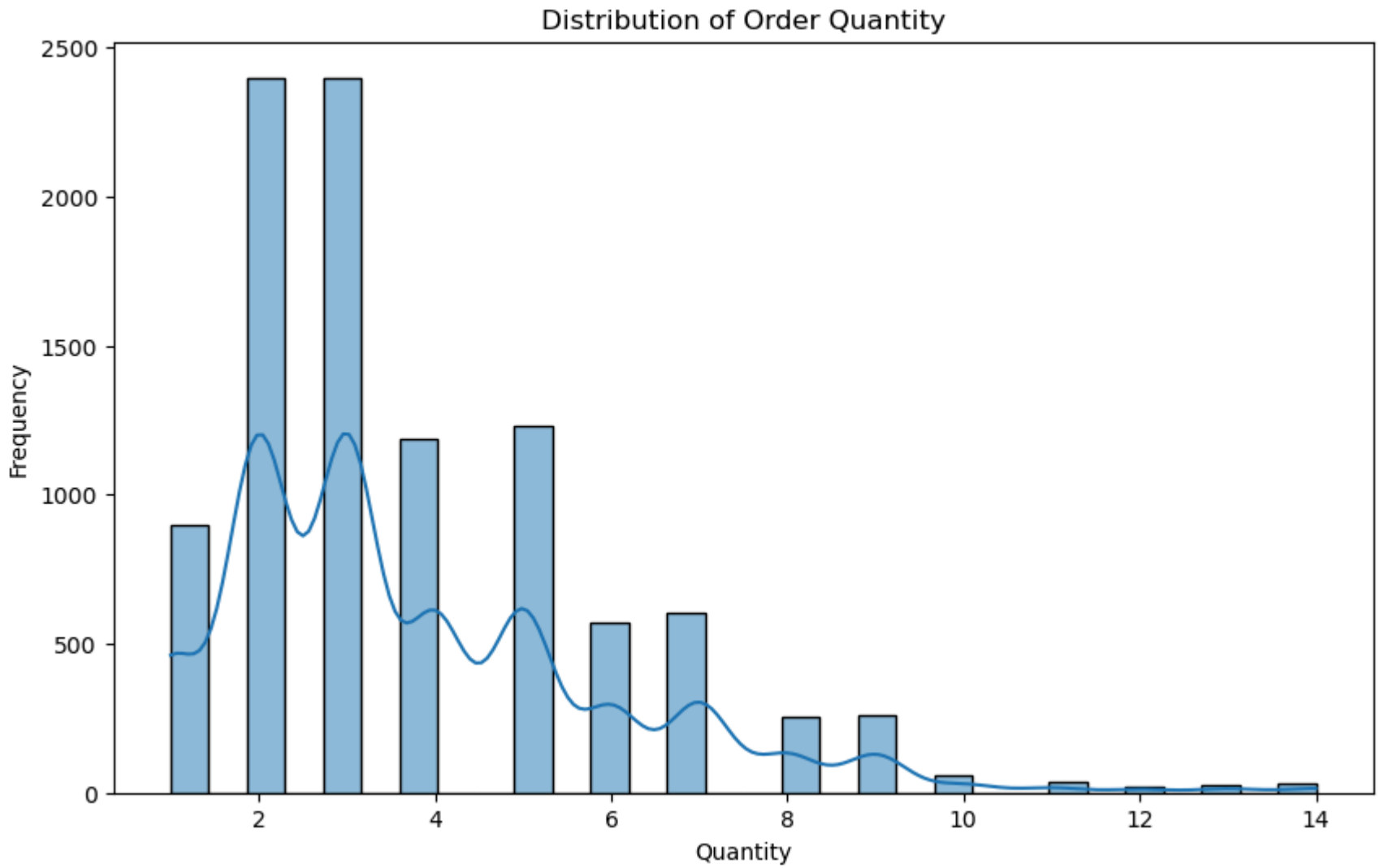
In [38]:
```
# Display descriptive statistics
print("Descriptive Statistics:")
print("Total Sales:", total_sales)
print("Average Order Value:", average_order_value)
```

```
Descriptive Statistics:
Total Sales: 2296195.5903
Average Order Value: 230.14890150345792
```

In [39]:
```
# Visualize the distribution of sales
plt.figure(figsize=(10, 6))
sns.histplot(data['Sales'], bins=30, kde=True)
plt.title('Distribution of Sales')
plt.xlabel('Sales')
plt.ylabel('Frequency')
plt.show()
```



In [40]:
```
# Visualize the distribution of order quantity
plt.figure(figsize=(10, 6))
sns.histplot(data['Quantity'], bins=30, kde=True)
plt.title('Distribution of Order Quantity')
plt.xlabel('Quantity')
plt.ylabel('Frequency')
plt.show()
```



In [43]:
```
# Identify top-selling categories
top_selling_categories = data.groupby('Category')['Quantity'].sum().nlargest(5)
print("\nTop Selling Categories:")
print(top_selling_categories)
```

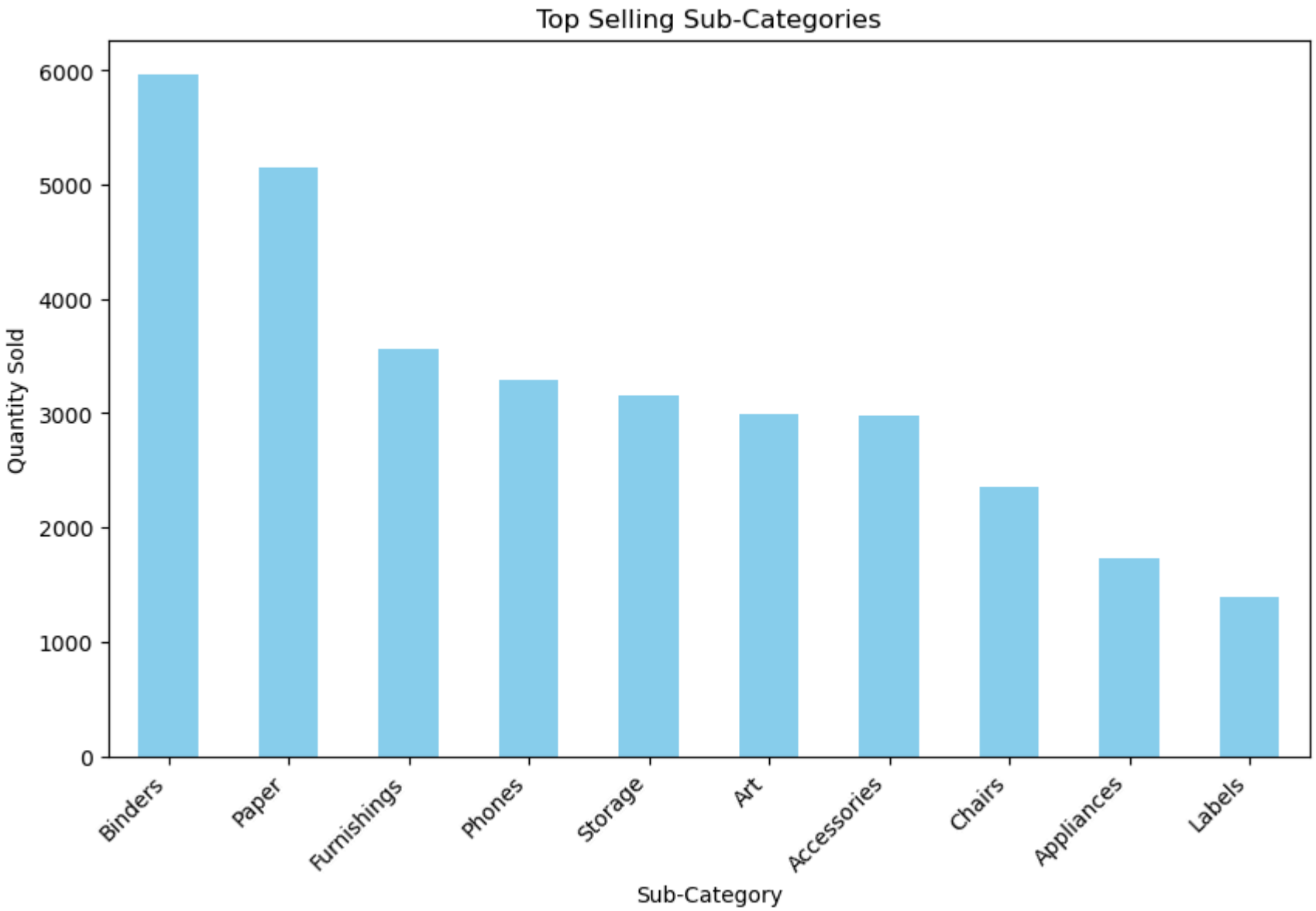
```
Top Selling Categories:
Category
Office Supplies    22861
Furniture           8020
Technology          6939
Name: Quantity, dtype: int64
```

Visualization:

Create visualizations (charts, graphs, dashboards) to present key findings effectively.

Top Selling Sub-Categories Bar Chart:

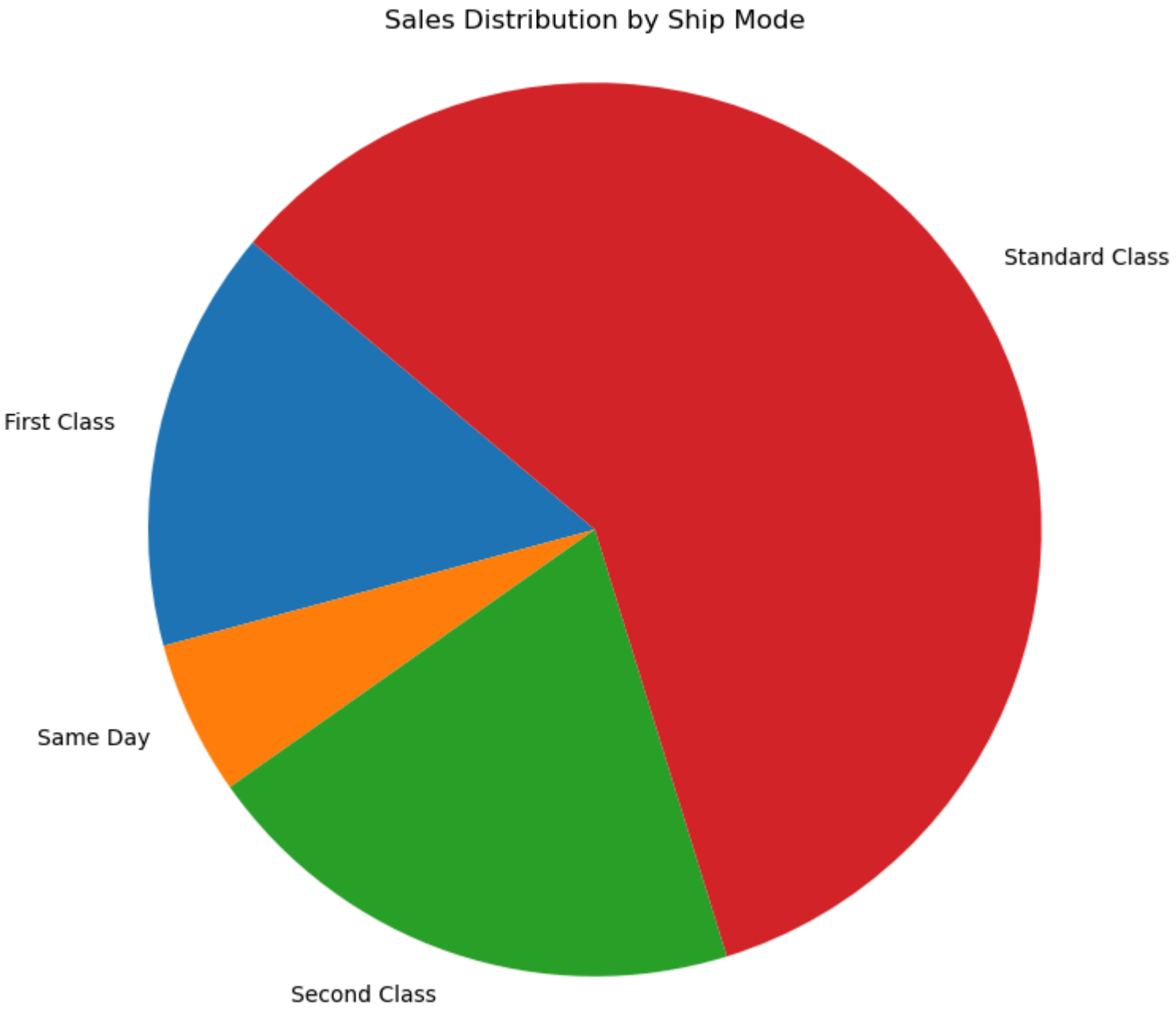
In [46]:
```
plt.figure(figsize=(10, 6))
top_selling_subcategories = data.groupby('Sub-Category')['Quantity'].sum().nlargest(10)
top_selling_subcategories.plot(kind='bar', color='skyblue')
plt.title('Top Selling Sub-Categories')
plt.xlabel('Sub-Category')
plt.ylabel('Quantity Sold')
plt.xticks(rotation=45, ha='right')
plt.show()
```



Sales by Ship Mode Pie Chart:

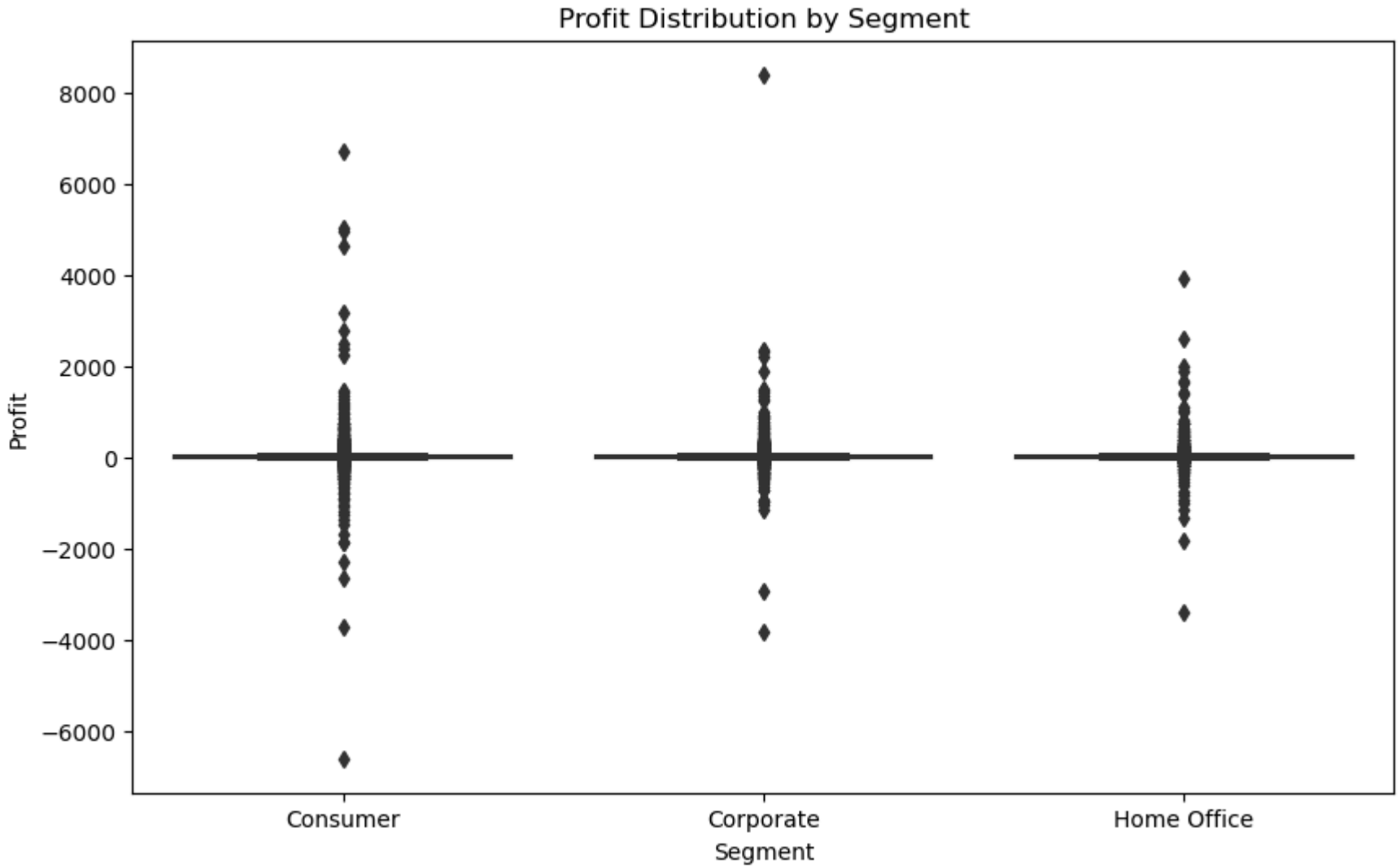
In [48]:
```
plt.figure(figsize=(8, 8))
sales_by_ship_mode = data.groupby('Ship Mode')['Sales'].sum()
```

```
plt.pie(sales_by_ship_mode, labels=sales_by_ship_mode.index,startangle=140)
plt.title('Sales Distribution by Ship Mode')
plt.axis('equal')
plt.show()
```


Sales Distribution by Ship Mode
Standard Class
First Class
Same Day
Second Class

Profit Distribution by Segment Box Plot:

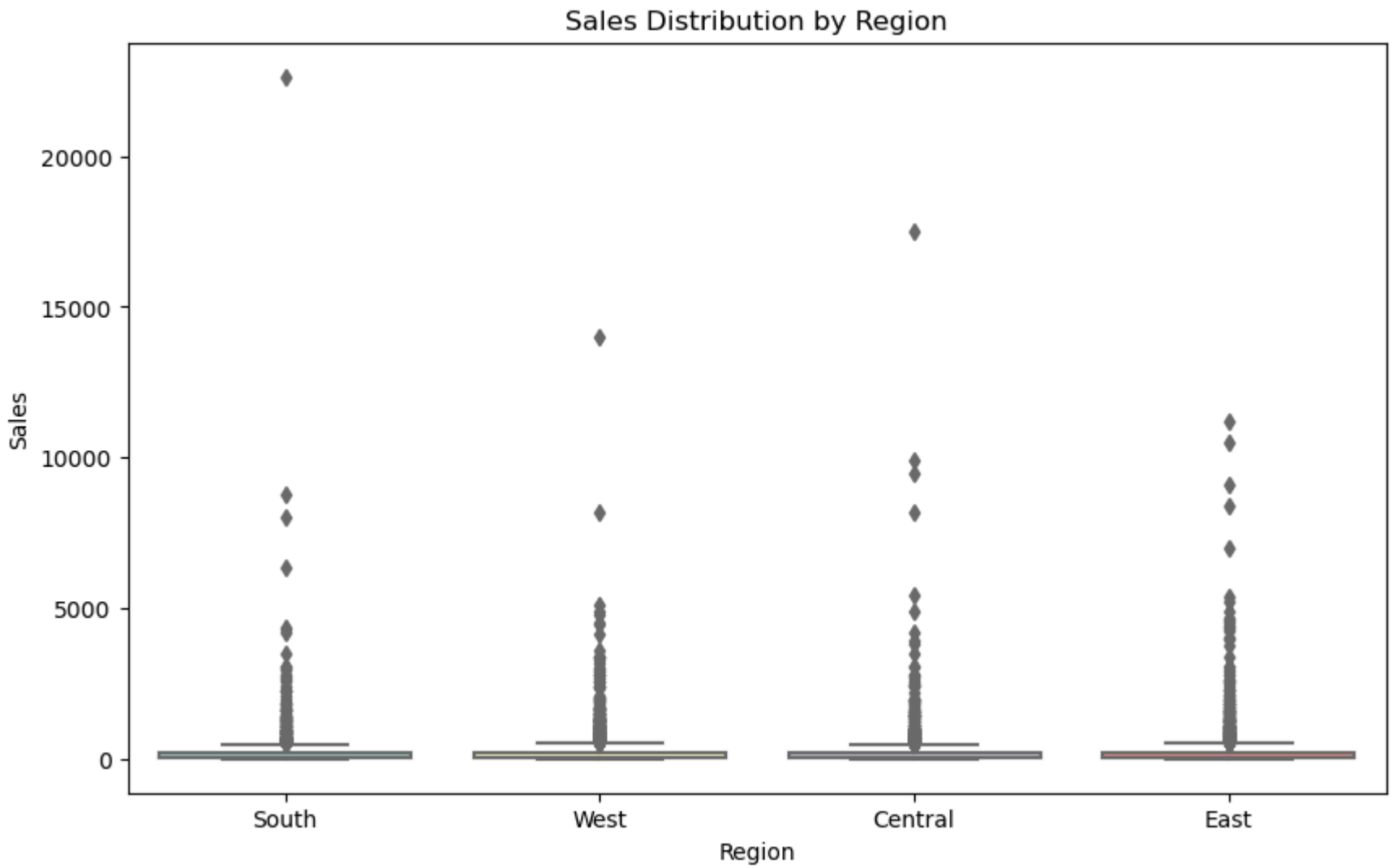
In [50]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(x='Segment', y='Profit', data=data, palette='viridis')
plt.title('Profit Distribution by Segment')
plt.xlabel('Segment')
plt.ylabel('Profit')
plt.show()
```



Sales Distribution by Region Box Plot:

In [51]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(x='Region', y='Sales', data=data, palette='Set3')
plt.title('Sales Distribution by Region')
plt.xlabel('Region')
```

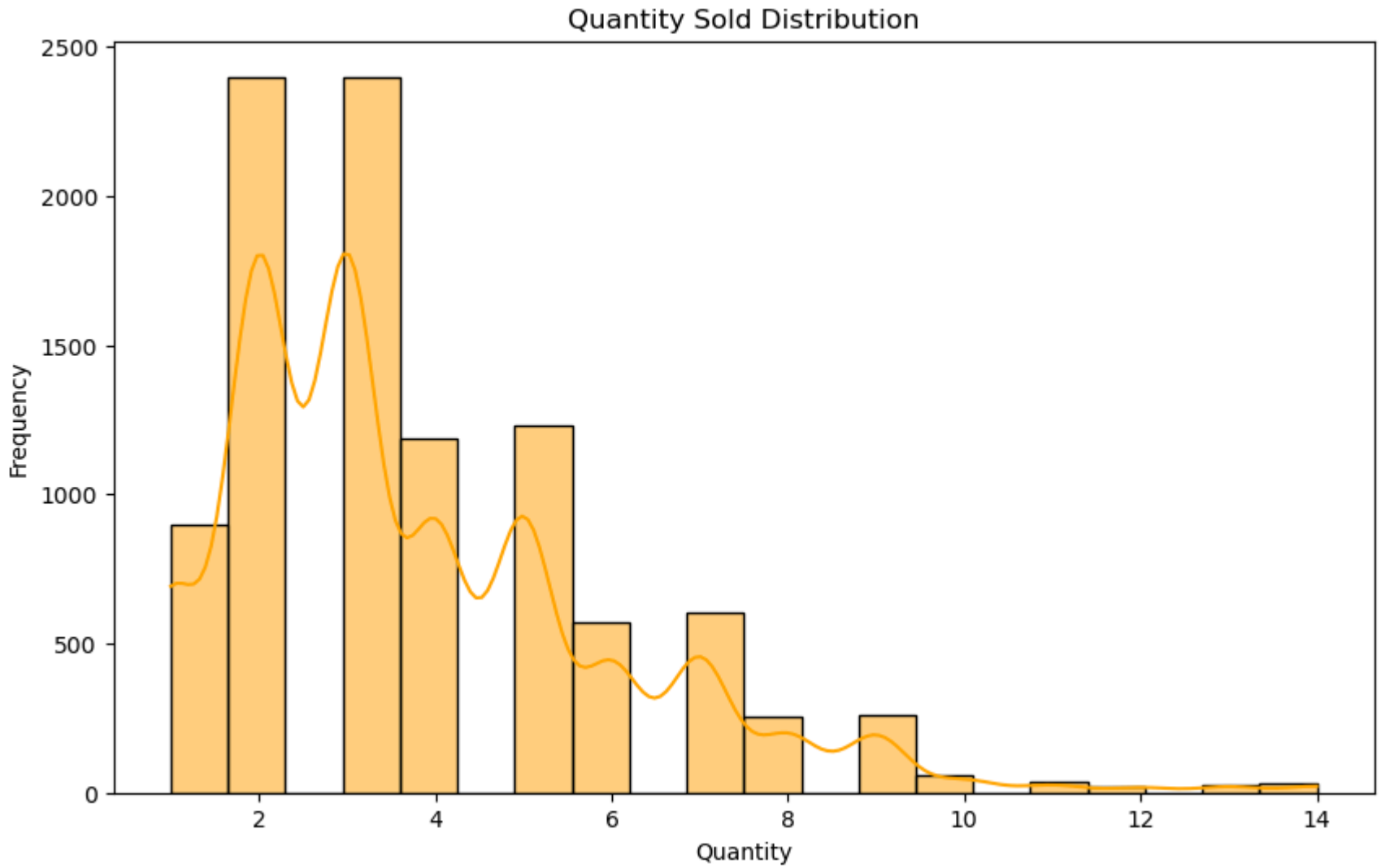
```
plt.ylabel('Sales')
plt.show()
```



Quantity Sold Distribution Histogram:

In [52]:
```
plt.figure(figsize=(10, 6))
sns.histplot(data['Quantity'], bins=20, kde=True, color='orange')
plt.title('Quantity Sold Distribution')
```

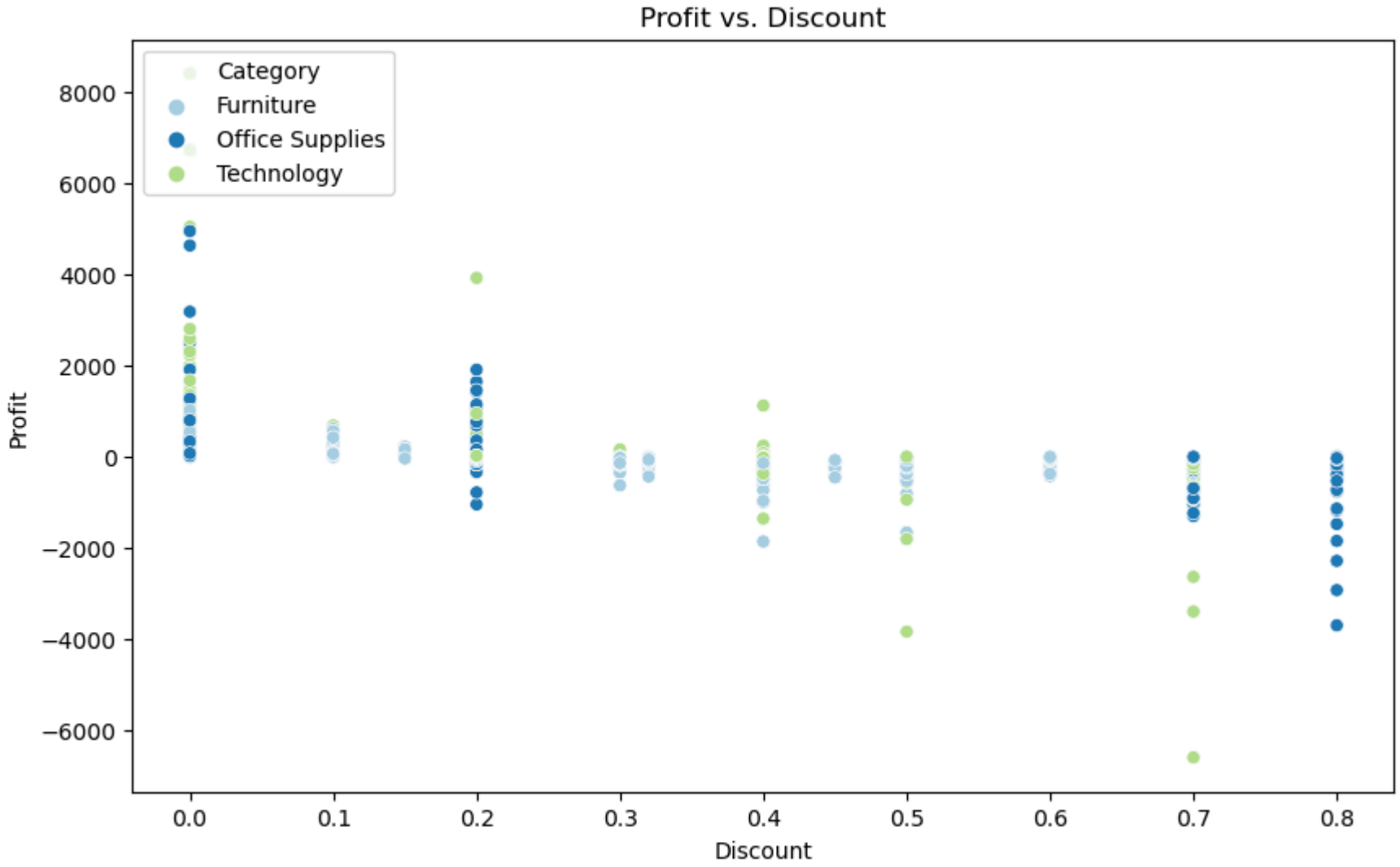
```
plt.xlabel('Quantity')
plt.ylabel('Frequency')
plt.show()
```



Profit vs. Discount Scatter Plot:

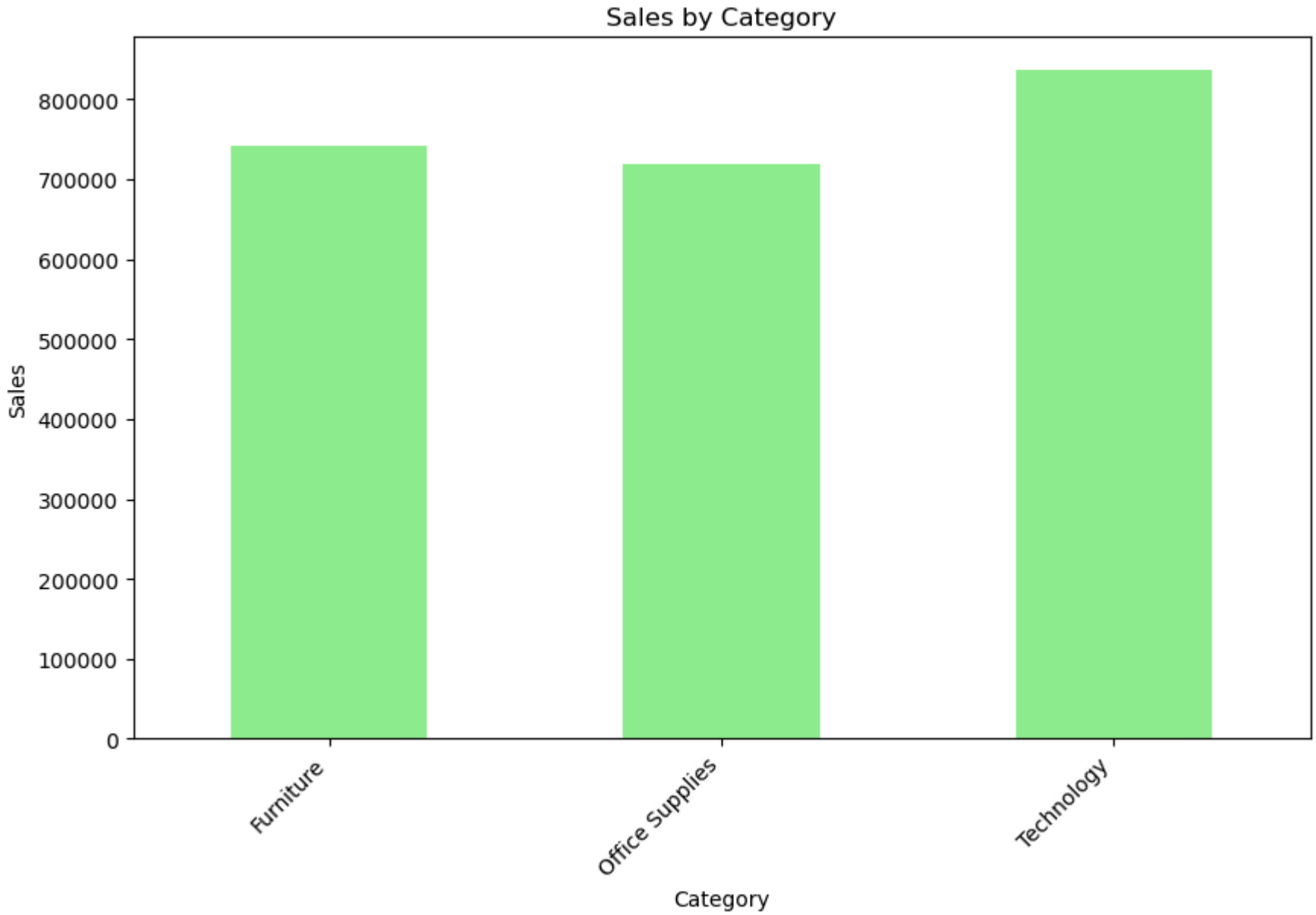
In [53]:
```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Discount', y='Profit', data=data, hue='Category', palette='Paired')
```

```
plt.title('Profit vs. Discount')
plt.xlabel('Discount')
plt.ylabel('Profit')
plt.legend(title='Category', loc='upper left')
plt.show()
```



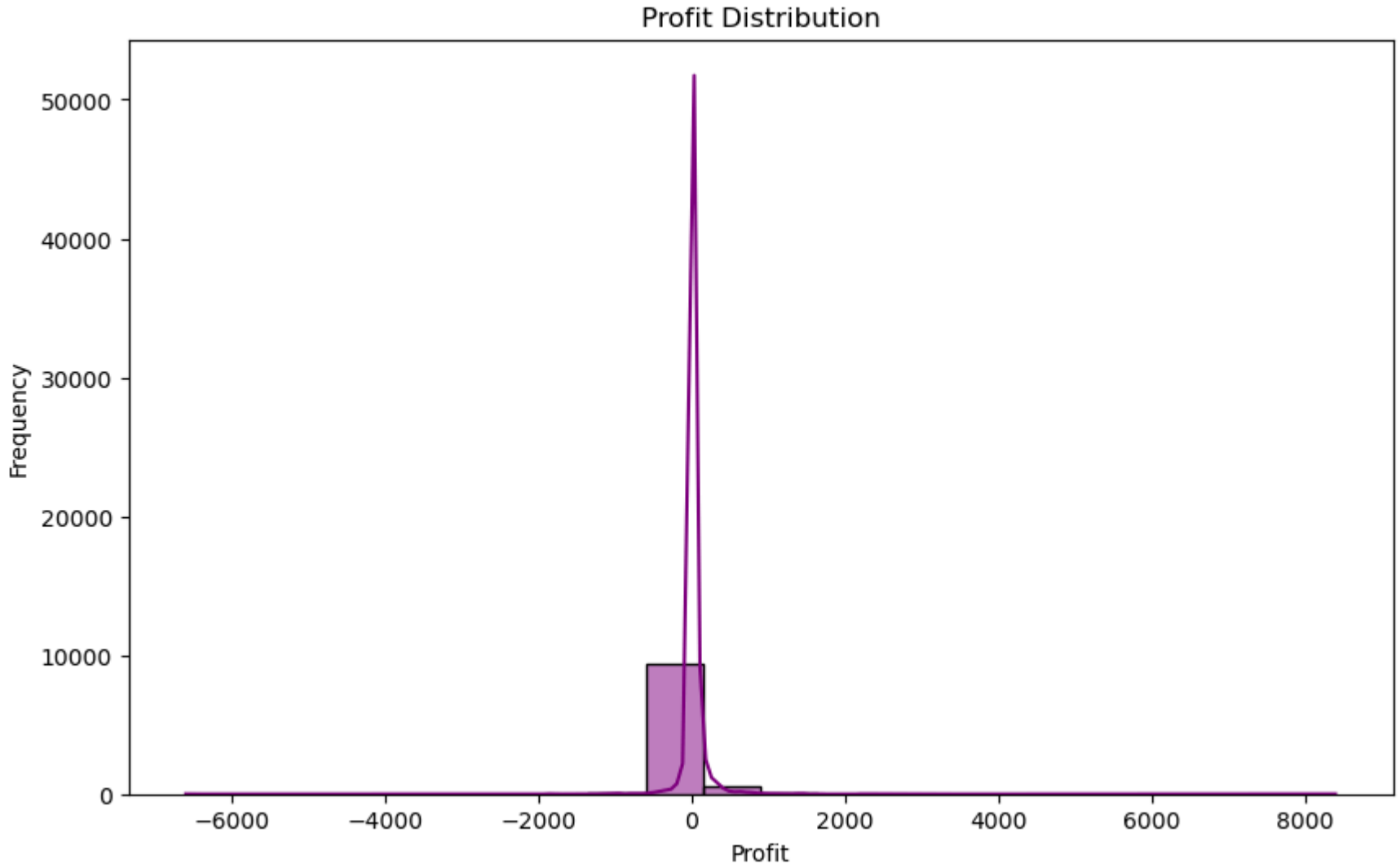
Sales by Category Bar Chart:

In [54]:
```
plt.figure(figsize=(10, 6))
sales_by_category = data.groupby('Category')['Sales'].sum()
sales_by_category.plot(kind='bar', color='lightgreen')
plt.title('Sales by Category')
plt.xlabel('Category')
plt.ylabel('Sales')
plt.xticks(rotation=45, ha='right')
plt.show()
```



Profit Distribution Histogram:

In [55]:
```
plt.figure(figsize=(10, 6))
sns.histplot(data['Profit'], bins=20, kde=True, color='purple')
plt.title('Profit Distribution')
plt.xlabel('Profit')
plt.ylabel('Frequency')
plt.show()
```



==========