In [2]:
```
import pandas as pd
df = pd.read_csv('order.csv')
df.head()
```

Out[2]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Prod
0	1	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-I 10001
1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-(10000
2	3	CA-2016-138688	6/12/2016	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036	West	OFF-10000
3	4	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	FUR-10000
4	5	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	OFF-10000

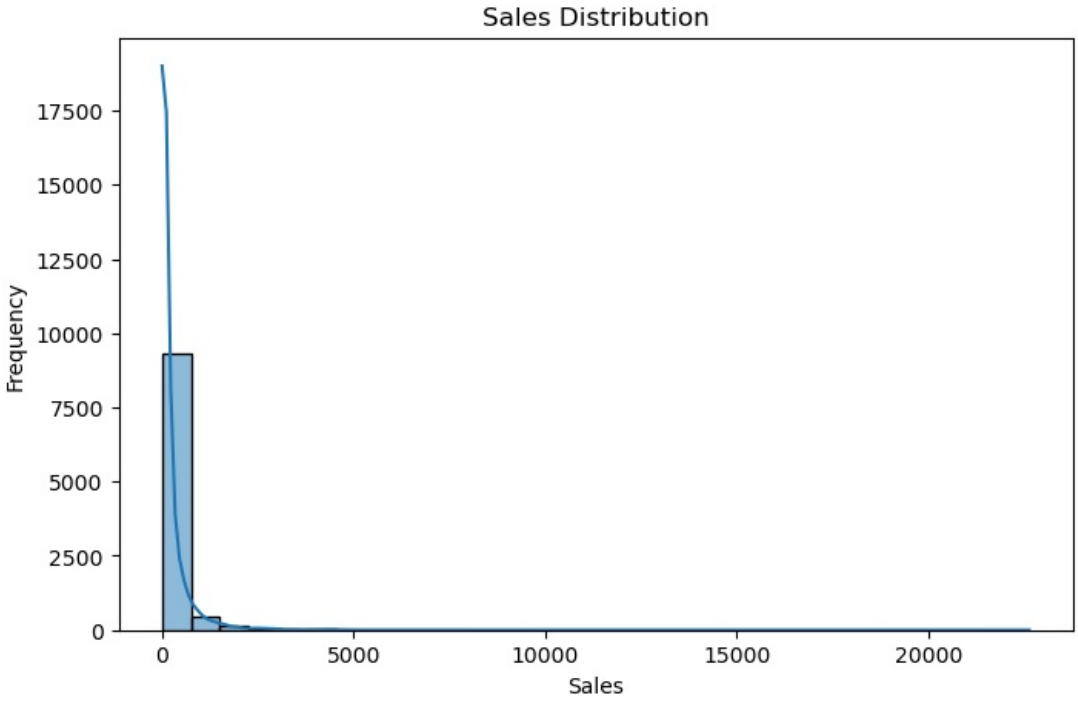
5 rows × 21 columns

In [3]:
```
df.info()
df.describe()
df.isnull().sum()
```

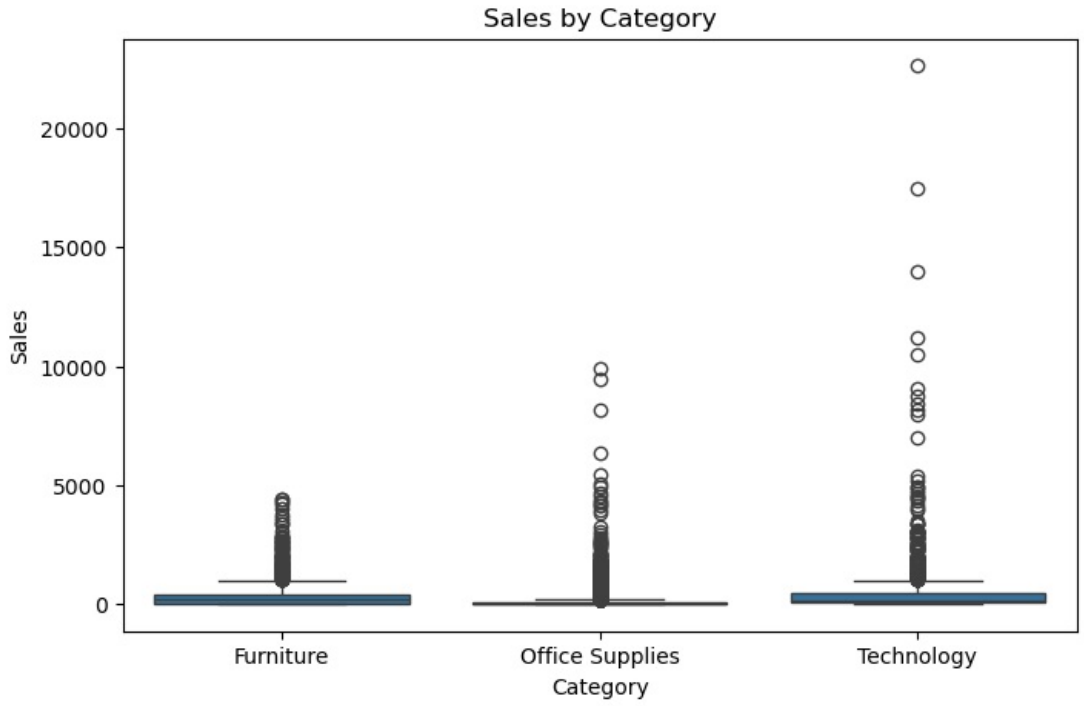
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Row ID         9994 non-null   int64  
 1   Order ID       9994 non-null   object 
 2   Order Date     9994 non-null   object 
 3   Ship Date      9994 non-null   object 
 4   Ship Mode      9994 non-null   object 
 5   Customer ID    9994 non-null   object 
 6   Customer Name  9994 non-null   object 
 7   Segment        9994 non-null   object 
 8   Country        9994 non-null   object 
 9   City           9994 non-null   object 
 10  State          9994 non-null   object 
 11  Postal Code    9994 non-null   int64  
 12  Region         9994 non-null   object 
 13  Product ID     9994 non-null   object 
 14  Category       9994 non-null   object 
 15  Sub-Category   9994 non-null   object 
 16  Product Name   9994 non-null   object 
 17  Sales          9994 non-null   float64
 18  Quantity       9994 non-null   int64  
 19  Discount       9994 non-null   float64
 20  Profit         9994 non-null   float64
dtypes: float64(3), int64(3), object(15)
memory usage: 1.6+ MB
```

Out[3]:
```
Row ID           0
Order ID         0
Order Date       0
Ship Date        0
Ship Mode        0
Customer ID      0
Customer Name    0
Segment          0
Country          0
City             0
State            0
Postal Code      0
Region           0
Product ID       0
Category         0
Sub-Category     0
Product Name     0
Sales            0
Quantity         0
Discount         0
Profit           0
dtype: int64
```

In [6]:
```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(8,5))
sns.histplot(df['Sales'], bins=30, kde=True)
plt.title('Sales Distribution')
plt.xlabel('Sales')
plt.ylabel('Frequency')
plt.show()
```

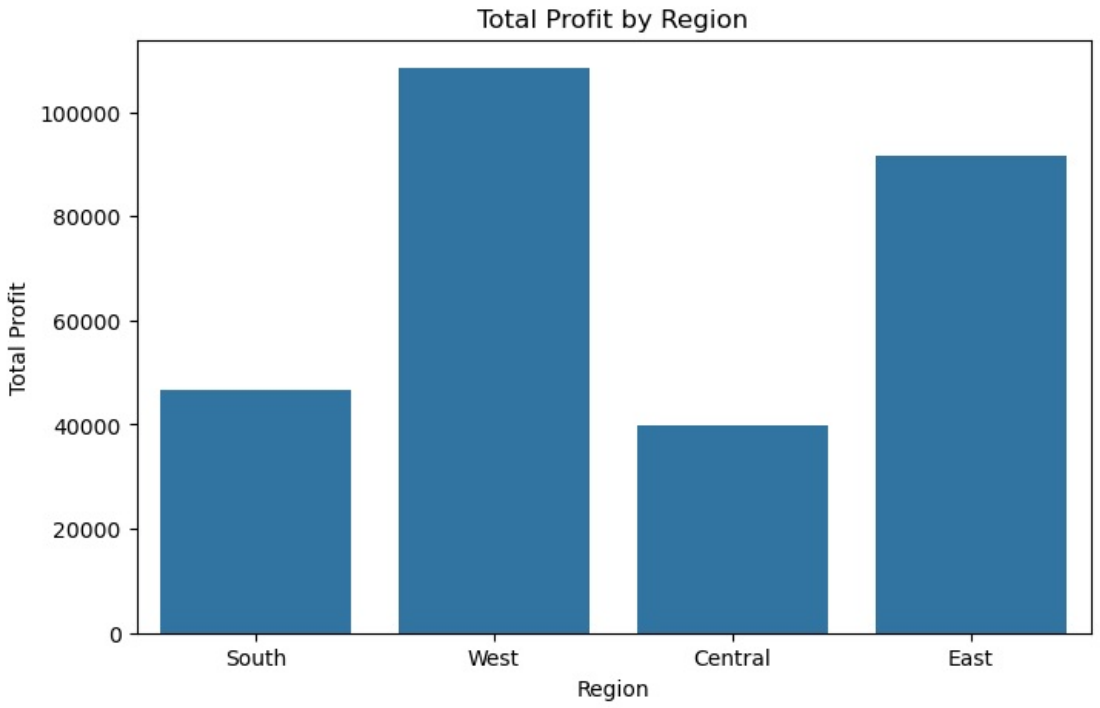


In [7]:
```
plt.figure(figsize=(8,5))
sns.boxplot(x='Category', y='Sales', data=df)
plt.title('Sales by Category')
plt.xlabel('Category')
plt.ylabel('Sales')
plt.show()
```



In [ ]:
```
Sales & Profit are highly skewed – a few large orders drive most revenue.
```

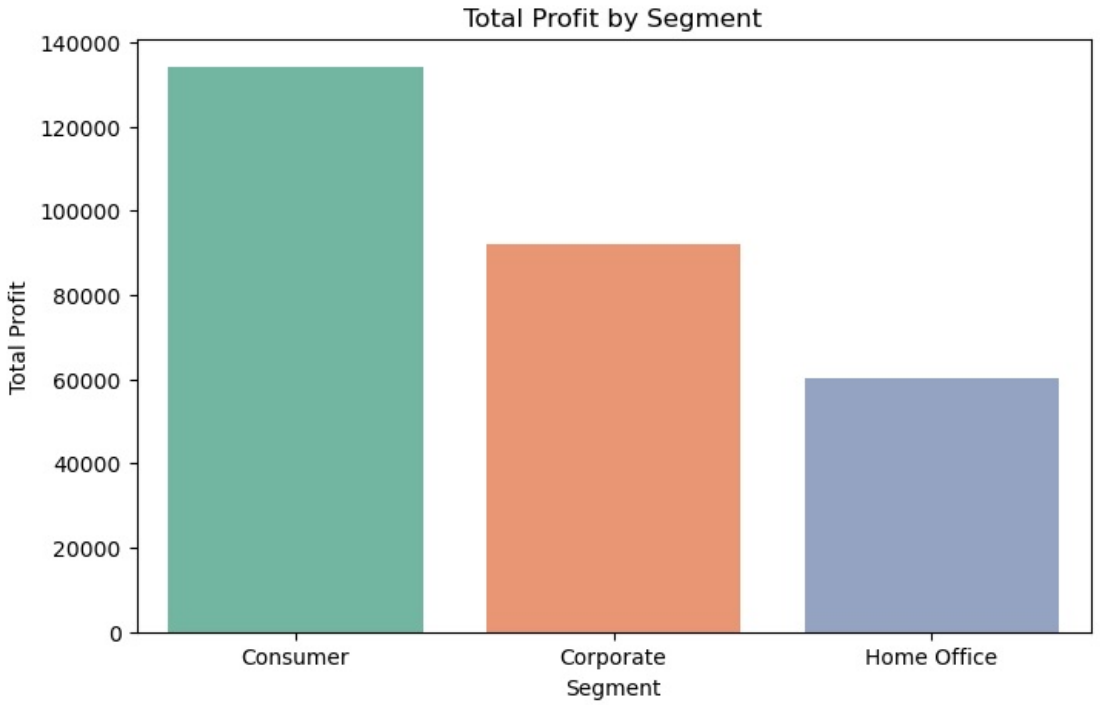
In [9]:
```
plt.figure(figsize=(8,5))
sns.barplot(x='Region', y='Profit', data=df, estimator=sum, errorbar=None)
plt.title('Total Profit by Region')
plt.xlabel('Region')
plt.ylabel('Total Profit')
plt.show()
```



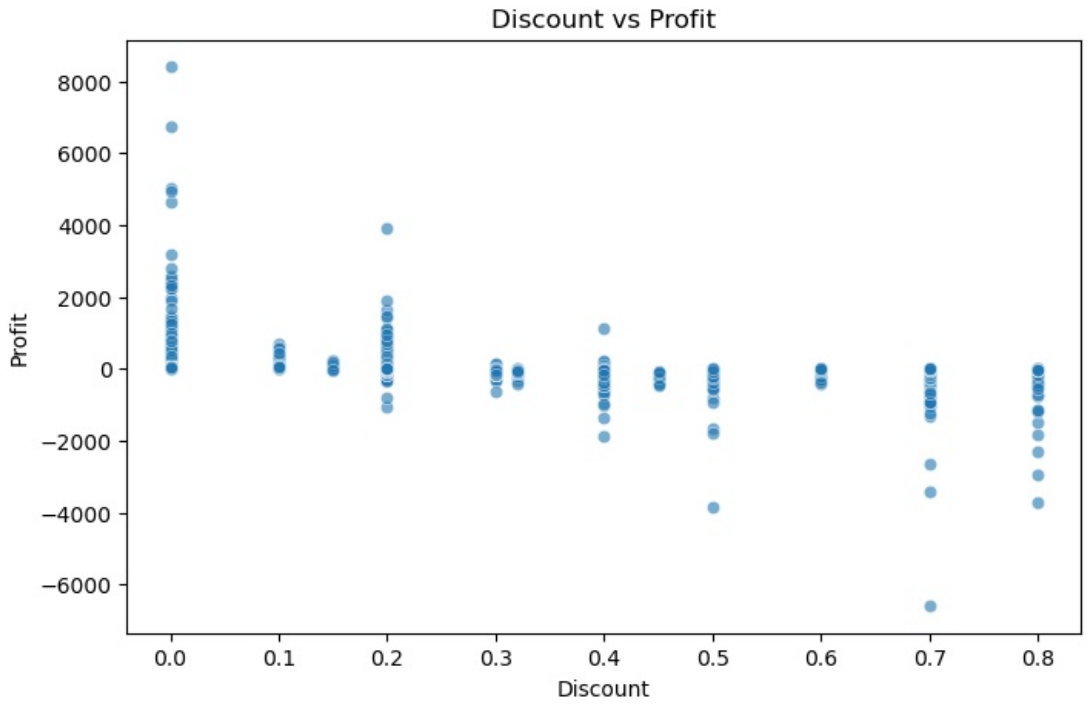
In [ ]:
```
West region and Consumer segment are most profitable.
```

In [10]:
```
plt.figure(figsize=(8,5))
sns.barplot(x='Segment', y='Profit', data=df, estimator=sum, errorbar=None, palette='Set2')
plt.title('Total Profit by Segment')
plt.xlabel('Segment')
plt.ylabel('Total Profit')
plt.show()
```

```
C:\Users\Kiruthika\AppData\Local\Temp\ipykernel_18808\2157180158.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(x='Segment', y='Profit', data=df, estimator=sum, errorbar=None, palette='Set2')
```



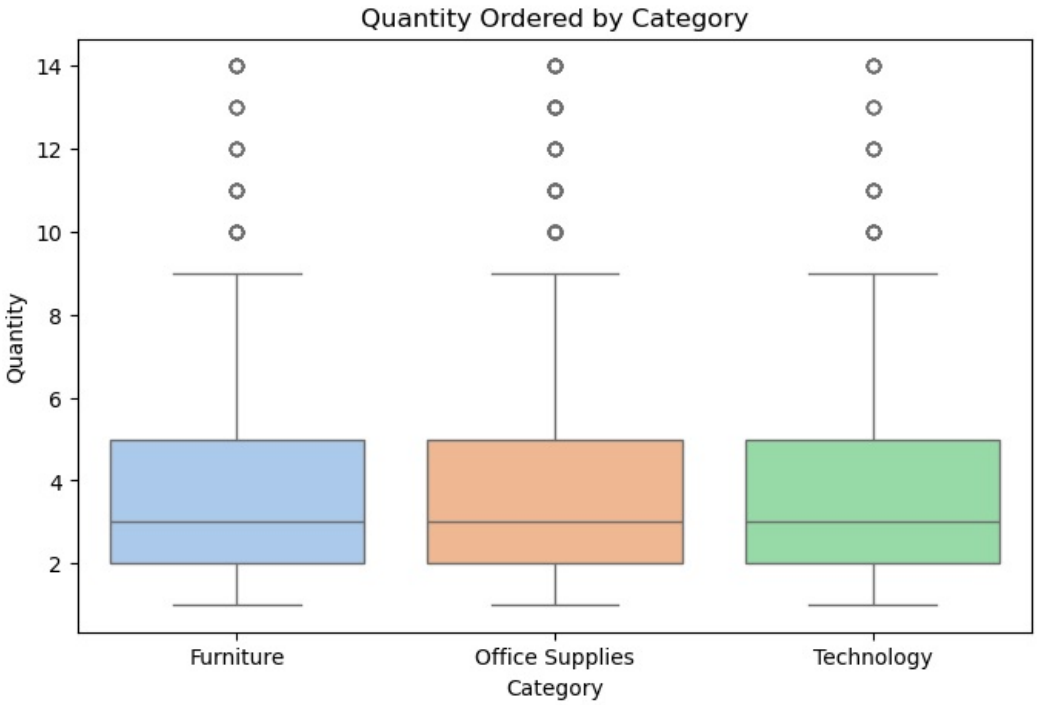
In [11]:
```
plt.figure(figsize=(8,5))
sns.scatterplot(x='Discount', y='Profit', data=df, alpha=0.6)
plt.title('Discount vs Profit')
plt.xlabel('Discount')
plt.ylabel('Profit')
plt.show()
```



In [ ]:
```
```

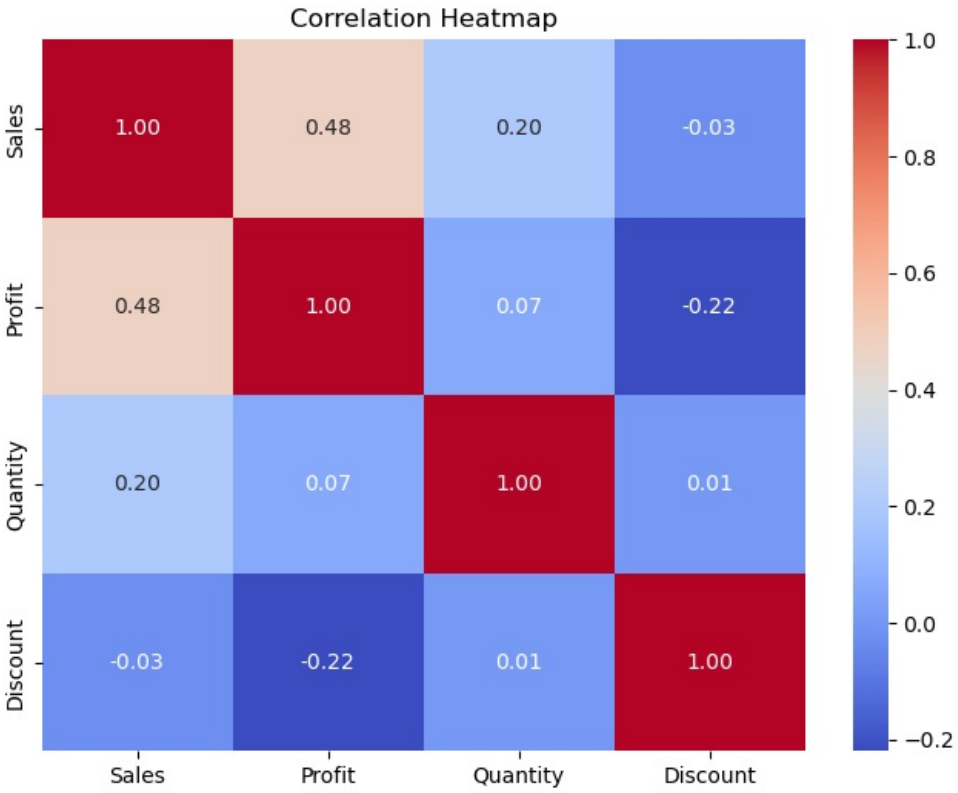
In [12]:
```
plt.figure(figsize=(8,5))
sns.boxplot(x='Category', y='Quantity', data=df, palette='pastel')
plt.title('Quantity Ordered by Category')
plt.xlabel('Category')
plt.ylabel('Quantity')
plt.show()
```

```
C:\Users\Kiruthika\AppData\Local\Temp\ipykernel_18808\1710927016.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable
to `hue` and set `legend=False` for the same effect.

  sns.boxplot(x='Category', y='Quantity', data=df, palette='pastel')
```



In [ ]:
```
South region and Home Office segment underperform.
```

In [13]:
```
plt.figure(figsize=(8,6))
sns.heatmap(df[['Sales', 'Profit', 'Quantity', 'Discount']].corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()
```



In [ ]:
```
Correlation heatmap- shows discount negatively impacts profit.
```

In [ ]:
```
This EDA explored key patterns in Superstore order data, including sales, profit, discount, and customer segmen
Key findings:
- Profit is unevenly distributed and sensitive to discount levels.
- Some categories and regions outperform others in revenue and profitability.
- Segment and Ship Mode choices influence performance and may guide strategy.
```

In [ ]:
```

```