```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
df = pd.read_csv('/content/Sample - Superstore.csv', encoding='latin-1')
display(df.head())
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Product ID	Cate
0	1	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-BO-10001798	Furr
1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-CH-10000454	Furr
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-LA-10000240	C Sup
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-TA-10000577	Furr
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-ST-10000760	C Sup

5 rows × 21 columns

```
print(df.describe(include='all'))
```

```
             Row ID        Order ID Order Date   Ship Date       Ship Mode  \
count   9994.000000            9994       9994        9994            9994
unique          NaN            5009       1237        1334               4
top             NaN  CA-2017-100111   9/5/2016  12/16/2015  Standard Class
freq            NaN              14         38          35            5968
mean    4997.500000             NaN        NaN         NaN             NaN
std     2885.163629             NaN        NaN         NaN             NaN
min        1.000000             NaN        NaN         NaN             NaN
25%     2499.250000             NaN        NaN         NaN             NaN
50%     4997.500000             NaN        NaN         NaN             NaN
75%     7495.750000             NaN        NaN         NaN             NaN
max     9994.000000             NaN        NaN         NaN             NaN

       Customer ID  Customer Name   Segment        Country           City  \
count         9994           9994      9994           9994           9994
unique         793            793         3              1            531
top       WB-21850  William Brown  Consumer  United States  New York City
freq            37             37      5191           9994            915
mean           NaN            NaN       NaN            NaN            NaN
std            NaN            NaN       NaN            NaN            NaN
min            NaN            NaN       NaN            NaN            NaN
25%            NaN            NaN       NaN            NaN            NaN
50%            NaN            NaN       NaN            NaN            NaN
75%            NaN            NaN       NaN            NaN            NaN
max            NaN            NaN       NaN            NaN            NaN

        ...   Postal Code  Region       Product ID         Category  \
count   ...   9994.000000    9994             9994             9994
unique  ...           NaN       4             1862                3
top     ...           NaN    West  OFF-PA-10001970  Office Supplies
freq    ...           NaN    3203               19             6026
mean    ...  55190.379428     NaN              NaN              NaN
std     ...  32063.693350     NaN              NaN              NaN
min     ...   1040.000000     NaN              NaN              NaN
25%     ...  23223.000000     NaN              NaN              NaN
50%     ...  56430.500000     NaN              NaN              NaN
75%     ...  90008.000000     NaN              NaN              NaN
max     ...  99301.000000     NaN              NaN              NaN

       Sub-Category     Product Name         Sales     Quantity     Discount  \
count          9994             9994   9994.000000  9994.000000  9994.000000
unique           17             1850           NaN          NaN          NaN
top         Binders  Staple envelope           NaN          NaN          NaN
freq           1523               48           NaN          NaN          NaN
mean            NaN              NaN    229.858001     3.789574     0.156203
```

```
std             NaN              NaN    623.245101     2.225110     0.206452
min             NaN              NaN      0.444000     1.000000     0.000000
25%             NaN              NaN     17.280000     2.000000     0.000000
50%             NaN              NaN     54.490000     3.000000     0.200000
75%             NaN              NaN    209.940000     5.000000     0.200000
max             NaN              NaN  22638.480000    14.000000     0.800000

             Profit
count   9994.000000
unique          NaN
top             NaN
freq            NaN
mean      28.656896
```

```
print(df.info())
```

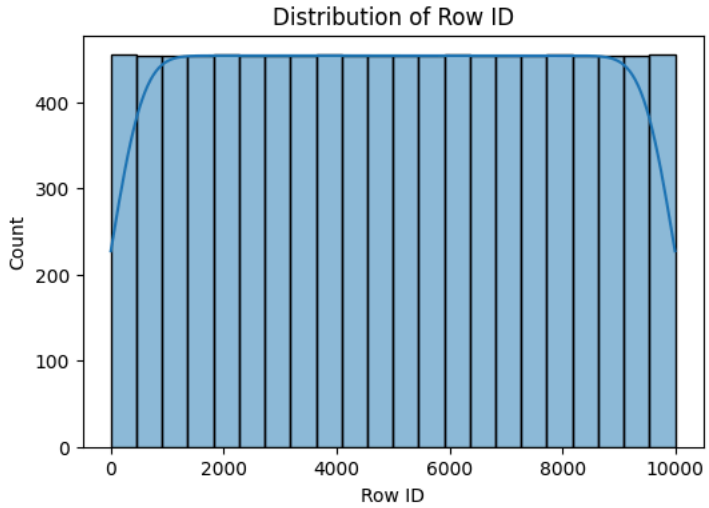
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Row ID         9994 non-null   int64
 1   Order ID       9994 non-null   object
 2   Order Date     9994 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
None
```

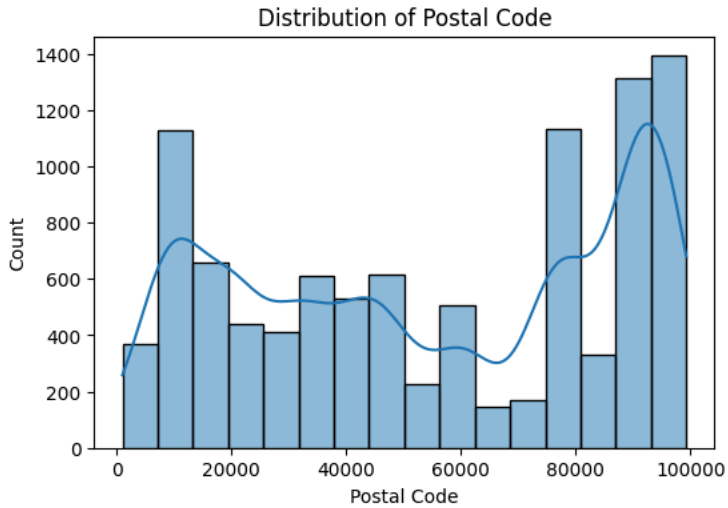
```
print(df.isnull().sum())
print("Duplicate rows:", df.duplicated().sum())
categorical_cols = df.select_dtypes(include='object').columns
for col in categorical_cols:
    print(f"\nValue counts for {col}:\n", df[col].value_counts())
```

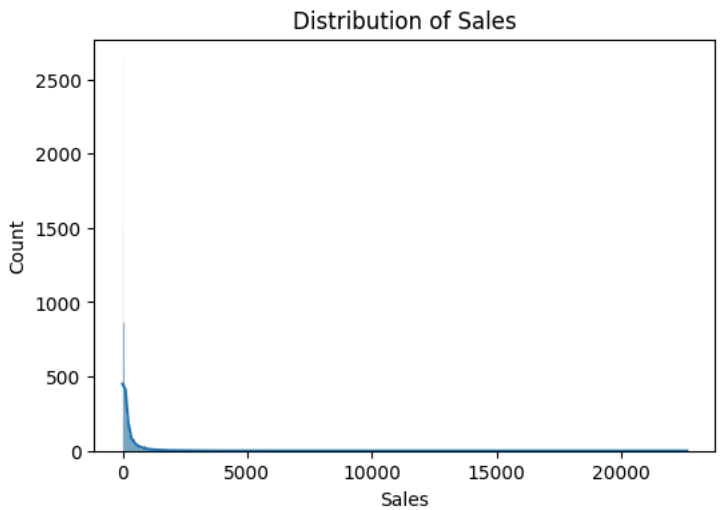
```
Row ID           0
Order ID         0
Order Date       0
Ship Date        0
Ship Mode        0
Customer ID      0
Customer Name    0
Segment          0
Country          0
City             0
State            0
Postal Code      0
Region           0
Product ID       0
Category         0
Sub-Category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

Value counts for Order ID:
 Order ID
CA-2017-100111    14
CA-2017-157987    12
CA-2016-165330    11
US-2016-108504    11
US-2015-126977    10
                  ..
CA-2014-110422     1
CA-2016-125794     1
CA-2017-163566     1
US-2015-151435     1
CA-2016-130225     1
```

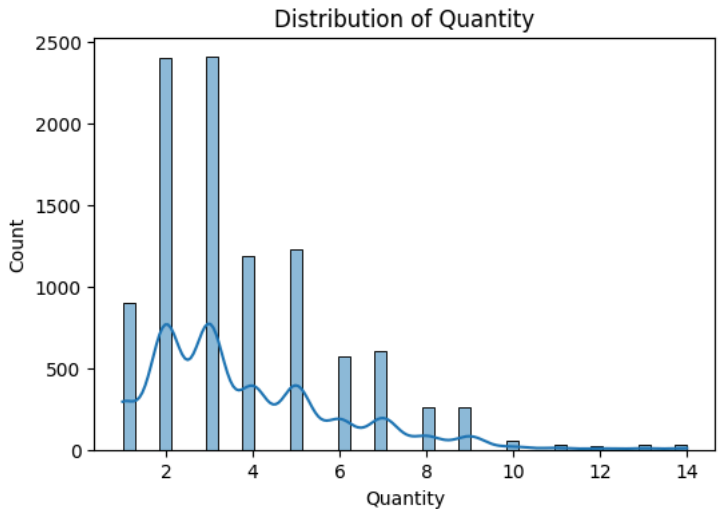
```
Name: count, Length: 5009, dtype: int64

Value counts for Order Date:
 Order Date
9/5/2016      38
9/2/2017      36
11/10/2016    35
12/1/2017     34
12/2/2017     34
              ..
1/28/2016      1
11/9/2016      1
6/3/2016       1
4/12/2015      1
1/21/2014      1
Name: count, Length: 1237, dtype: int64

Value counts for Ship Date:
 Ship Date
12/16/2015    35
9/26/2017     34
```

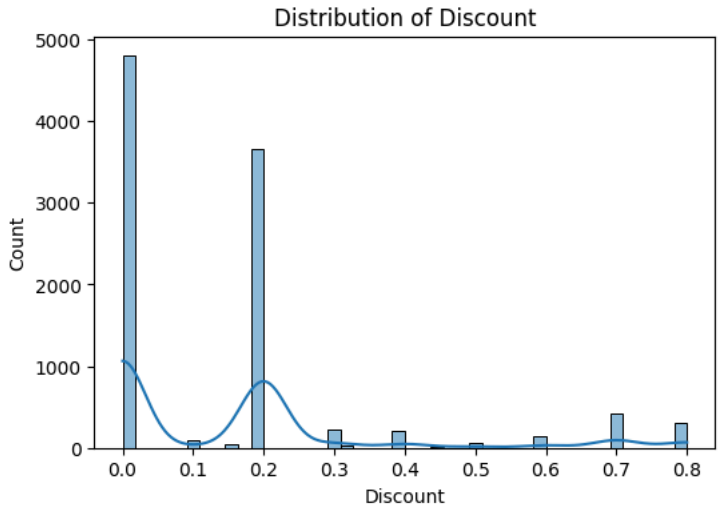
```
numeric_cols = df.select_dtypes(include=['float64', 'int64']).columns
for col in numeric_cols:
    plt.figure(figsize=(6, 4))
    sns.histplot(df[col], kde=True)
    plt.title(f'Distribution of {col}')
    plt.show()
```

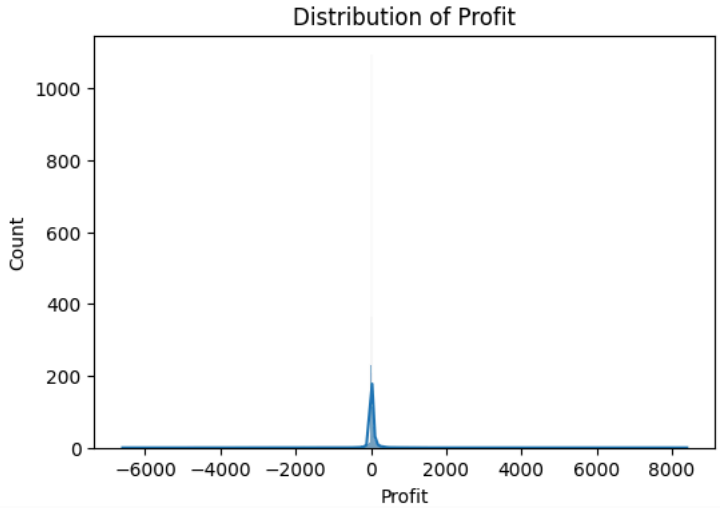

Distribution of Row ID
Count
Row ID
0
100
200
300
400
0
2000
4000
6000
8000
10000


Distribution of Postal Code
Count
Postal Code
0
200
400
600
800
1000
1200
1400
0
20000
40000
60000
80000
100000

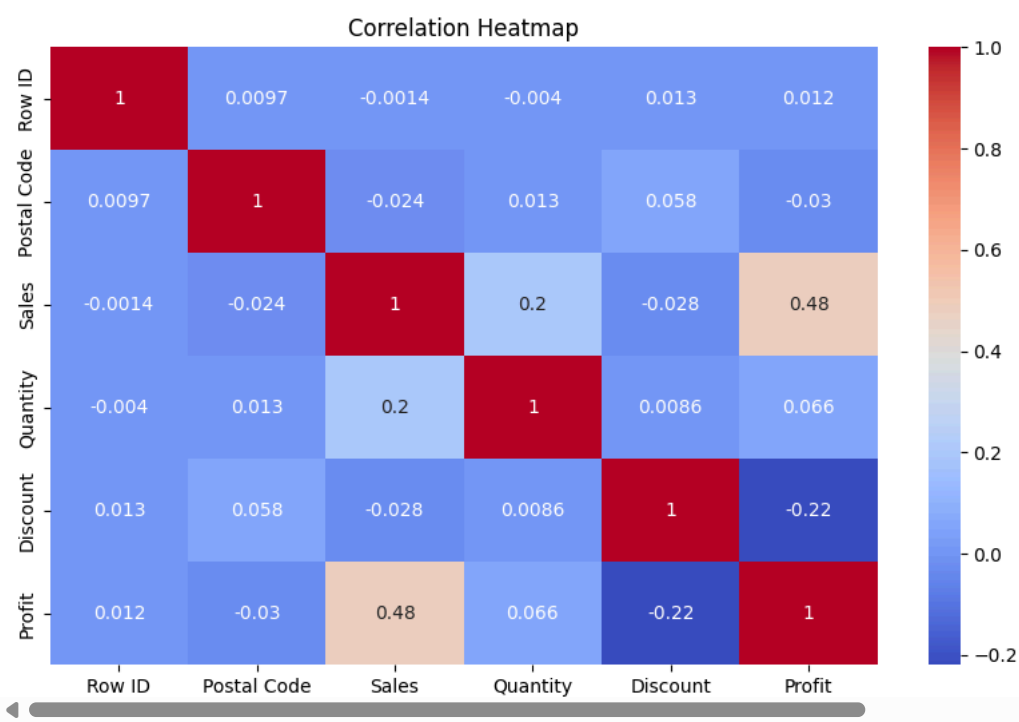

Distribution of Sales
Count
Sales
0
500
1000
1500
2000
2500
0
5000
10000
15000
20000


Distribution of Quantity
Count
Quantity
0
500
1000
1500
2000
2500
2
4
6
8
10
12
14

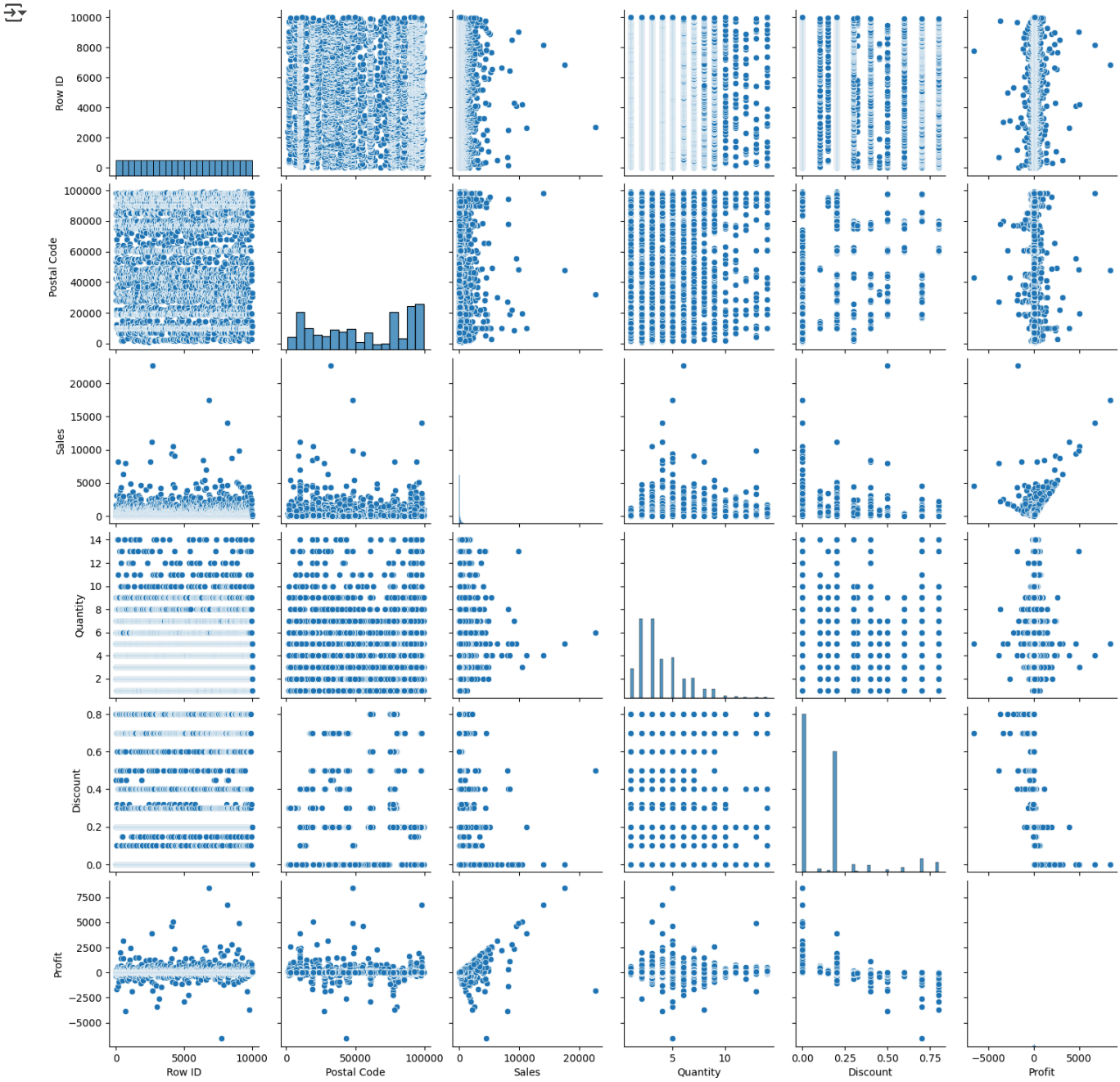

Distribution of Discount
Count
Discount
0
1000
2000
3000
4000
5000
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8


Distribution of Profit
Count
Profit
0
200
400
600
800
1000
−6000
−4000
−2000
0
2000
4000
6000
8000

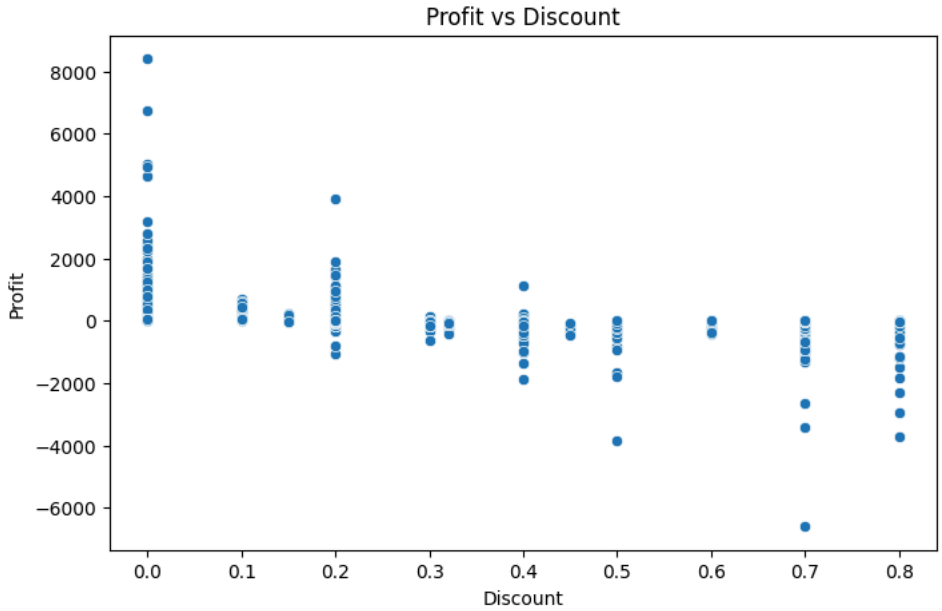
```
plt.figure(figsize=(10, 6))
sns.heatmap(df[numeric_cols].corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



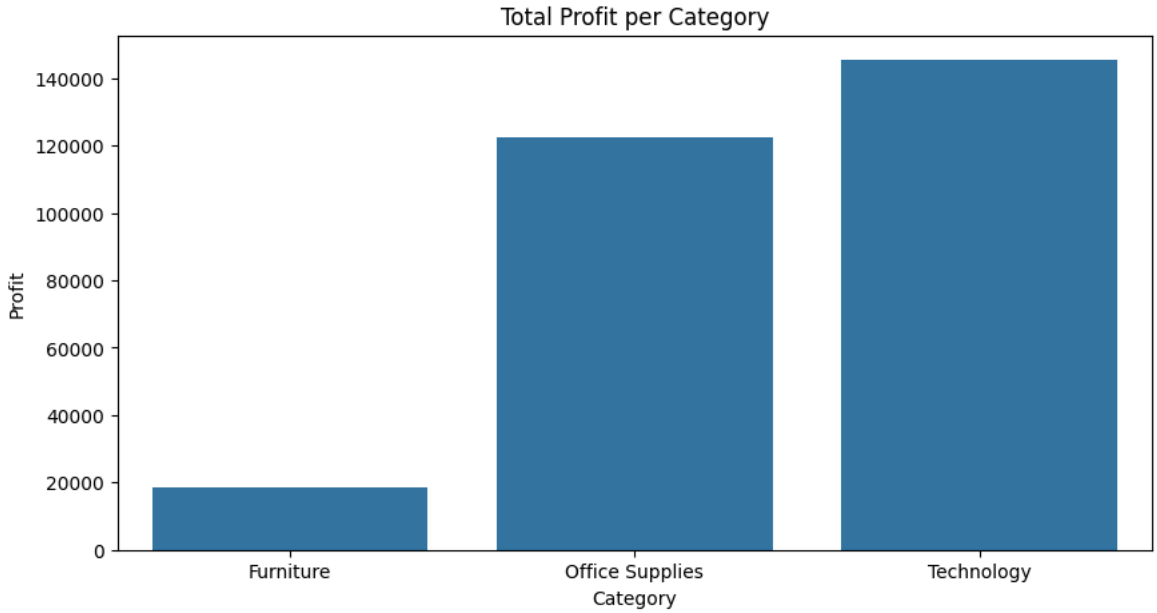

```
sns.pairplot(df[numeric_cols])
plt.show()
```

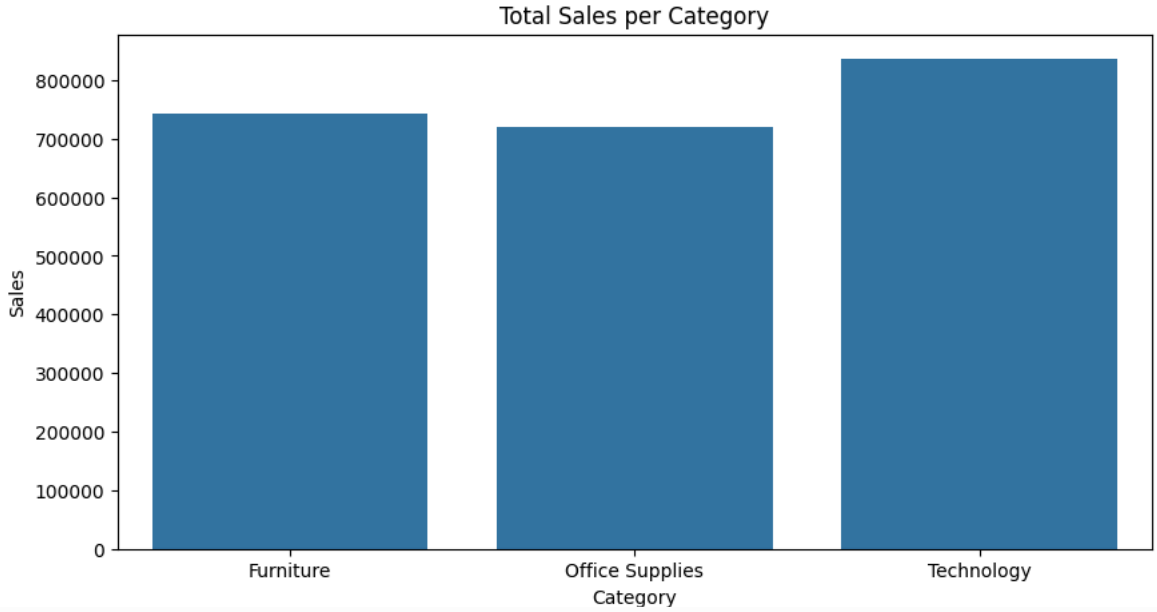



```
plt.figure(figsize=(8, 5))
sns.scatterplot(x='Discount', y='Profit', data=df)
plt.title('Profit vs Discount')
plt.show()
```



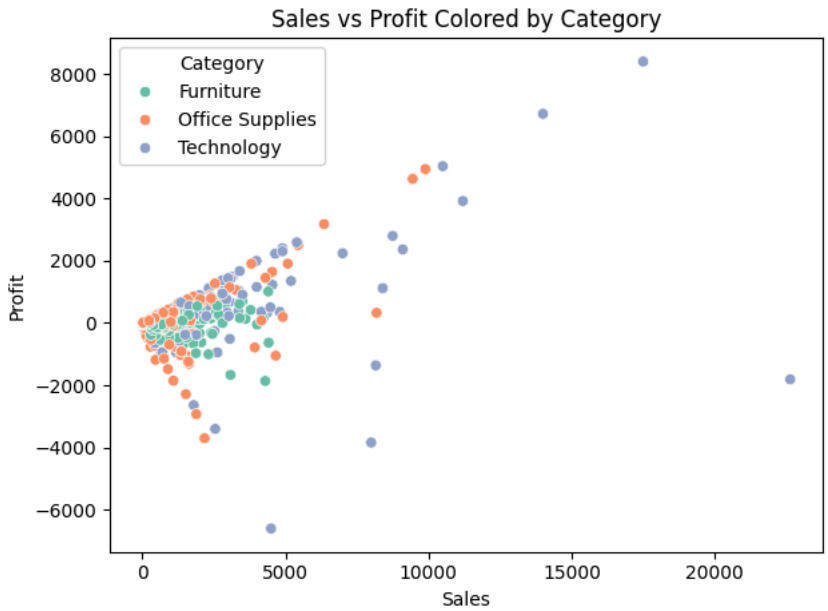

```
plt.figure(figsize=(10, 5))
sns.barplot(x='Category', y='Profit', data=df, estimator=sum, ci=None)
plt.title('Total Profit per Category')
plt.show()

plt.figure(figsize=(10, 5))
sns.barplot(x='Category', y='Sales', data=df, estimator=sum, ci=None)
plt.title('Total Sales per Category')
plt.show()
```






```
plt.figure(figsize=(10, 5))
sns.boxplot(x='Segment', y='Profit', data=df)
plt.title('Profit Distribution by Customer Segment')
plt.show()
```

```
sns.scatterplot(data=df, x='Sales', y='Profit', hue='Category', palette='Set2')
plt.title('Sales vs Profit Colored by Category')
plt.xlabel('Sales')
plt.ylabel('Profit')
plt.legend(title='Category')
plt.tight_layout()
plt.show()
```




```
sns.regplot(data=df, x='Discount', y='Profit', scatter_kws={'alpha':0.4}, line_kws={'color':'red'})
plt.title('Discount vs Profit with Regression Line')
plt.xlabel('Discount')
plt.ylabel('Profit')
plt.tight_layout()
plt.show()
```