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

data = pd.read_csv("sales.csv", encoding="latin1")

data
```

```
      Row ID        Order ID  Order Date   Ship Date       Ship
Mode  \
0          1  CA-2016-152156  11-08-2016  11-11-2016    Second Class

1          2  CA-2016-152156  11-08-2016  11-11-2016    Second Class

2          3  CA-2016-138688  06-12-2016   6/16/2016    Second Class

3          4  US-2015-108966  10-11-2015  10/18/2015  Standard Class

4          5  US-2015-108966  10-11-2015  10/18/2015  Standard Class

...      ...             ...         ...         ...             ...

9989    9990  CA-2014-110422   1/21/2014   1/23/2014    Second Class

9990    9991  CA-2017-121258   2/26/2017  03-03-2017  Standard Class

9991    9992  CA-2017-121258   2/26/2017  03-03-2017  Standard Class

9992    9993  CA-2017-121258   2/26/2017  03-03-2017  Standard Class

9993    9994  CA-2017-119914  05-04-2017  05-09-2017    Second Class


     Customer ID     Customer Name    Segment        Country
City  \
0       CG-12520       Claire Gute   Consumer  United States
Henderson
1       CG-12520       Claire Gute   Consumer  United States
Henderson
2       DV-13045   Darrin Van Huff  Corporate  United States      Los
Angeles
3       SO-20335    Sean O'Donnell   Consumer  United States  Fort
Lauderdale
4       SO-20335    Sean O'Donnell   Consumer  United States  Fort
Lauderdale
...          ...               ...        ...            ...
...
9989    TB-21400  Tom Boeckenhauer   Consumer  United States
Miami
9990    DB-13060       Dave Brooks   Consumer  United States
```

```
Costa Mesa
9991    DB-13060       Dave Brooks   Consumer  United States
Costa Mesa
9992    DB-13060       Dave Brooks   Consumer  United States
Costa Mesa
9993    CC-12220      Chris Cortes   Consumer  United States
Westminster

      ... Postal Code  Region       Product ID        Category Sub-
Category  \
0     ...       42420   South  FUR-BO-10001798        Furniture
Bookcases
1     ...       42420   South  FUR-CH-10000454        Furniture
Chairs
2     ...       90036    West  OFF-LA-10000240  Office Supplies
Labels
3     ...       33311   South  FUR-TA-10000577        Furniture
Tables
4     ...       33311   South  OFF-ST-10000760  Office Supplies
Storage
...   ...         ...     ...              ...              ...
...
9989  ...       33180   South  FUR-FU-10001889        Furniture
Furnishings
9990  ...       92627    West  FUR-FU-10000747        Furniture
Furnishings
9991  ...       92627    West  TEC-PH-10003645       Technology
Phones
9992  ...       92627    West  OFF-PA-10004041  Office Supplies
Paper
9993  ...       92683    West  OFF-AP-10002684  Office Supplies
Appliances

                                           Product Name     Sales
Quantity  \
0                     Bush Somerset Collection Bookcase  261.9600
2
1     Hon Deluxe Fabric Upholstered Stacking Chairs,...  731.9400
3
2     Self-Adhesive Address Labels for Typewriters b...   14.6200
2
3         Bretford CR4500 Series Slim Rectangular Table  957.5775
5
4                        Eldon Fold 'N Roll Cart System   22.3680
2
...                                                 ...       ...
...
9989                             Ultra Door Pull Handle   25.2480
3
```

```
9990  Tenex B1-RE Series Chair Mats for Low Pile Car...   91.9600
2
9991                                Aastra 57i VoIP phone  258.5760
2
9992  It's Hot Message Books with Stickers, 2 3/4" x 5"   29.6000
4
9993  Acco 7-Outlet Masterpiece Power Center, Wihtou...  243.1600
2

      Discount    Profit
0         0.00   41.9136
1         0.00  219.5820
2         0.00    6.8714
3         0.45 -383.0310
4         0.20    2.5164
...        ...       ...
9989      0.20    4.1028
9990      0.00   15.6332
9991      0.20   19.3932
9992      0.00   13.3200
9993      0.00   72.9480

[9994 rows x 21 columns]
```

```
data.shape
```

```
(9994, 21)
```

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

```
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
```

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

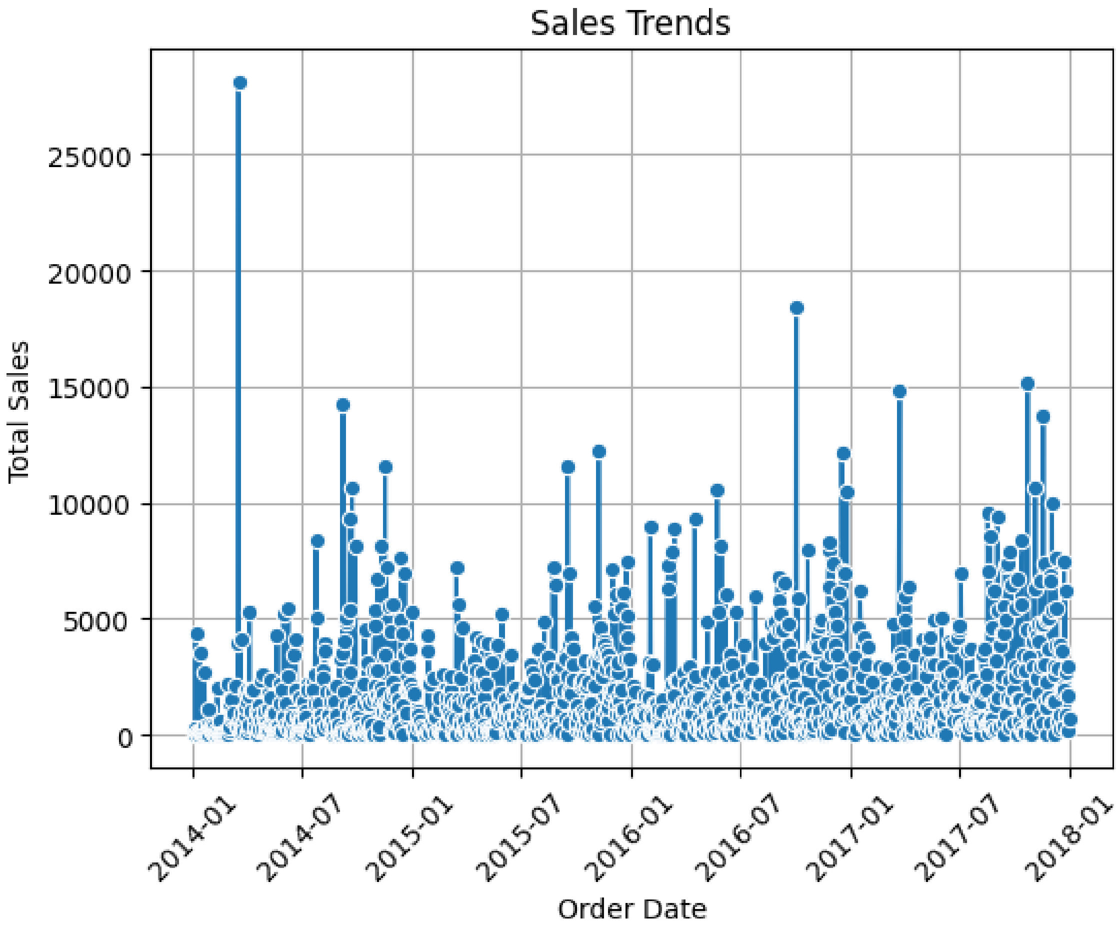
```
np.int64(0)
```

```python
data_cleaned = data.drop_duplicates()
```

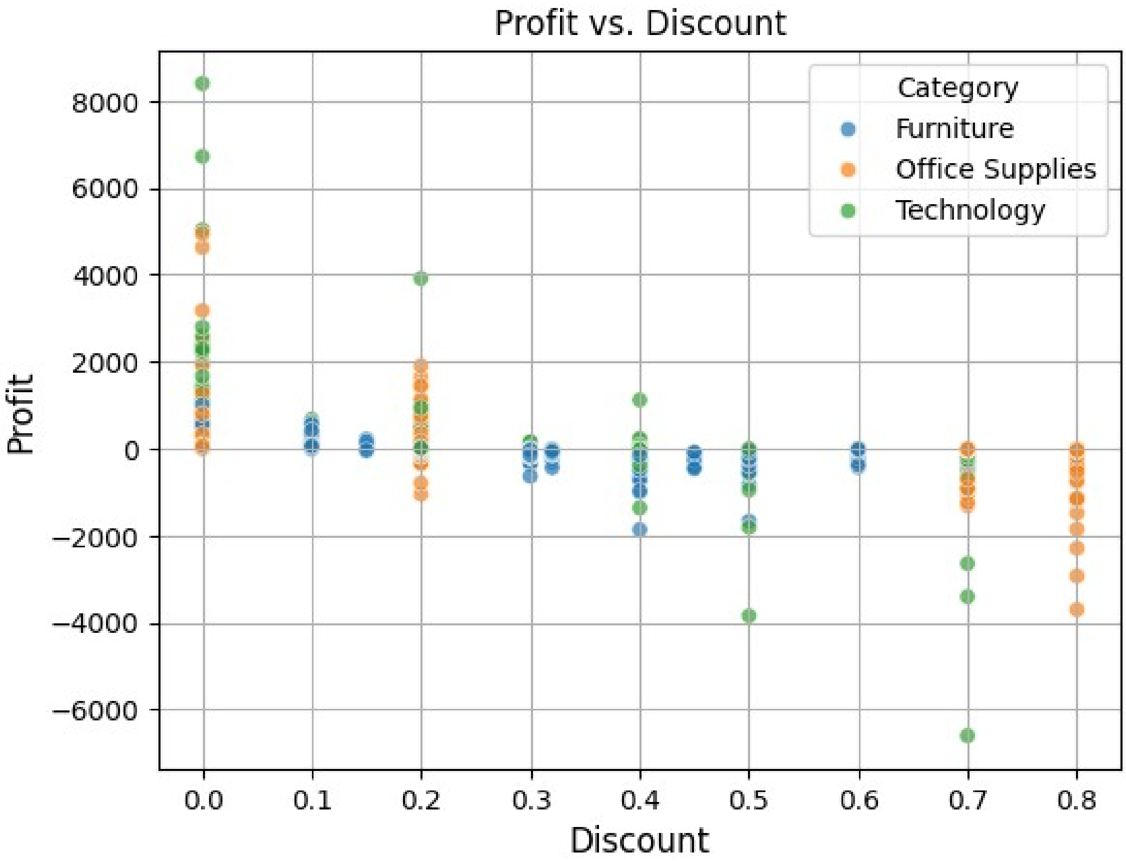
```python
data_cleaned['Order Date'] = pd.to_datetime(data_cleaned['Order Date'], format='mixed', errors='coerce')
data_cleaned['Ship Date'] = pd.to_datetime(data_cleaned['Ship Date'], format='mixed', errors='coerce')
```

```python
sales_trends = data_cleaned.groupby('Order Date')['Sales'].sum().reset_index()
```

```python
sns.lineplot(data=sales_trends, x='Order Date', y='Sales',marker="o")
plt.title('Sales Trends')
plt.xlabel('Order Date')
plt.ylabel('Total Sales')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



```
sns.scatterplot(data=data_cleaned, x='Discount', y='Profit',
hue='Category', alpha=0.7)
plt.title('Profit vs. Discount')
plt.xlabel('Discount', fontsize=12)
plt.ylabel('Profit', fontsize=12)
plt.grid(True)
plt.show()
```



```
region_category_sales = data_cleaned.groupby(['Region', 'Category'])
['Sales'].sum().unstack()

region_category_sales

Category    Furniture  Office Supplies  Technology
Region
Central   163797.1638       167026.415  170416.312
East      208291.2040       205516.055  264973.981
South     117298.6840       125651.313  148771.908
West      252612.7435       220853.249  251991.832

features = data_cleaned[['Profit', 'Discount']]

features

          Profit  Discount
0        41.9136      0.00
1       219.5820      0.00
2         6.8714      0.00
3      -383.0310      0.45
4         2.5164      0.20
...          ...       ...
9989      4.1028      0.20
```

```
9990   15.6332      0.00
9991   19.3932      0.20
9992   13.3200      0.00
9993   72.9480      0.00

[9994 rows x 2 columns]
```

```python
target = data_cleaned['Sales']
```

```python
target
```

```
0       261.9600
1       731.9400
2        14.6200
3       957.5775
4        22.3680
          ...
9989     25.2480
9990     91.9600
9991    258.5760
9992     29.6000
9993    243.1600
Name: Sales, Length: 9994, dtype: float64
```

```python
from sklearn.model_selection import train_test_split
```

```python
X_train, X_test, y_train, y_test = train_test_split(features, target, 
test_size=0.2, random_state=42)
```

```python
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

```python
model = LinearRegression()
model.fit(X_train, y_train)
```

```
LinearRegression()
```

```python
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

```python
mse
```

```
np.float64(700271.8880636955)
```

```python
r2
```

```
-0.18549666591248526
```

```python
# Function to predict sales based on user input
def predict_sales():
    print("Enter values for the features:")
    profit = float(input("Profit: "))
    discount = float(input("Discount: "))

    # Prepare the input as a DataFrame
    user_input = pd.DataFrame([[profit, discount]], columns=['Profit',
'Discount'])

    # Make a prediction
    predicted_sales = model.predict(user_input)[0]
    print(f"Predicted Sales: {predicted_sales:.2f}")

predict_sales()
```

```
Enter values for the features:

Profit:  20
Discount:  0.1

Predicted Sales: 194.02
```