In [54]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

DATA INSPECTION

In [14]:
```
df = pd.read_csv(r"C:\Users\Desktop\Sales_Project\Dataset\Raw.csv", encoding='latin
df.head(5)
```

Out[14]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Countr
0	1	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	Unite Stat
1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	Unite Stat
2	3	CA-2016-138688	6/12/2016	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	Unite Stat
3	4	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	Unite Stat
4	5	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	Unite Stat

5 rows × 21 columns

In [17]:
```
df.isnull().sum()
```

Out[17]:
```
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 [18]:
```
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
```

In [23]:
```
print(df.describe(include="all").T)
```

	count	unique	top	freq	mean
Row ID	9994.0	NaN	NaN	NaN	4997.5
Order ID	9994	5009	CA-2017-100111	14	NaN
Order Date	9994	1237	9/5/2016	38	NaN
Ship Date	9994	1334	12/16/2015	35	NaN
Ship Mode	9994	4	Standard Class	5968	NaN
Customer ID	9994	793	WB-21850	37	NaN
Customer Name	9994	793	William Brown	37	NaN
Segment	9994	3	Consumer	5191	NaN
Country	9994	1	United States	9994	NaN
City	9994	531	New York City	915	NaN
State	9994	49	California	2001	NaN
Postal Code	9994.0	NaN	NaN	NaN	55190.379428
Region	9994	4	West	3203	NaN
Product ID	9994	1862	OFF-PA-10001970	19	NaN
Category	9994	3	Office Supplies	6026	NaN
Sub-Category	9994	17	Binders	1523	NaN
Product Name	9994	1850	Staple envelope	48	NaN
Sales	9994.0	NaN	NaN	NaN	229.858001
Quantity	9994.0	NaN	NaN	NaN	3.789574
Discount	9994.0	NaN	NaN	NaN	0.156203
Profit	9994.0	NaN	NaN	NaN	28.656896

	std	min	25%	50%	75%	max
Row ID	2885.163629	1.0	2499.25	4997.5	7495.75	9994.0
Order ID	NaN	NaN	NaN	NaN	NaN	NaN
Order Date	NaN	NaN	NaN	NaN	NaN	NaN
Ship Date	NaN	NaN	NaN	NaN	NaN	NaN
Ship Mode	NaN	NaN	NaN	NaN	NaN	NaN
Customer ID	NaN	NaN	NaN	NaN	NaN	NaN
Customer Name	NaN	NaN	NaN	NaN	NaN	NaN
Segment	NaN	NaN	NaN	NaN	NaN	NaN
Country	NaN	NaN	NaN	NaN	NaN	NaN
City	NaN	NaN	NaN	NaN	NaN	NaN
State	NaN	NaN	NaN	NaN	NaN	NaN
Postal Code	32063.69335	1040.0	23223.0	56430.5	90008.0	99301.0
Region	NaN	NaN	NaN	NaN	NaN	NaN
Product ID	NaN	NaN	NaN	NaN	NaN	NaN
Category	NaN	NaN	NaN	NaN	NaN	NaN
Sub-Category	NaN	NaN	NaN	NaN	NaN	NaN
Product Name	NaN	NaN	NaN	NaN	NaN	NaN
Sales	623.245101	0.444	17.28	54.49	209.94	22638.48
Quantity	2.22511	1.0	2.0	3.0	5.0	14.0
Discount	0.206452	0.0	0.0	0.2	0.2	0.8
Profit	234.260108	-6599.978	1.72875	8.6665	29.364	8399.976

In [28]:
```
df.duplicated().sum()
```

Out[28]: np.int64(0)

In [31]:
```
(df==0).sum()
```

Out[31]:
```
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        4798
Profit            65
dtype: int64
```

In [32]:
```
if "Postal Code" in df.columns:
    df["Postal Code"].fillna(0, inplace=True)
```

```
C:\Users\PMLS\AppData\Local\Temp\ipykernel_7964\3833522378.py:2: FutureWarning: A va
lue is trying to be set on a copy of a DataFrame or Series through chained assignmen
t using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because
the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method
({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform
the operation inplace on the original object.


  df["Postal Code"].fillna(0, inplace=True)
```

In [39]:
```
if "Order Date" in df.columns:
    df['Order Date'] = pd.to_datetime(df['Order Date'])

if "Ship Date" in df.columns:
    df['Ship Date'] = pd.to_datetime(df['Ship Date'])
```

In [44]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 24 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   datetime64[ns]
 3   Ship Date      9994 non-null   datetime64[ns]
 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
 21  Year           9994 non-null   int32
 22  Month          9994 non-null   int32
 23  Quarter        9994 non-null   int32
dtypes: datetime64[ns](2), float64(3), int32(3), int64(3), object(13)
memory usage: 1.7+ MB
```

In [43]:
```
df['Year']=df['Order Date'].dt.year
df['Month']=df['Order Date'].dt.month
df['Quarter']=df['Order Date'].dt.quarter
```

In [46]:
```
df.head()
```

Out[46]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Hende
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Hende
2	3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Ang
3	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Lauder
4	5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Lauder

5 rows × 24 columns

In [47]:
```
if "Customer ID" in df.columns and "Sales" in df.columns:
    clv = df.groupby("Customer ID")["Sales"].sum().reset_index()
    clv.rename(columns={"Sales": "CLV"}, inplace=True)
    df = df.merge(clv, on="Customer ID", how="left")
```

In [48]:
```
if "Order Date" in df.columns and "Customer ID" in df.columns:
    first_purchase = df.groupby("Customer ID")["Order Date"].min().reset_index()
    first_purchase.rename(columns={"Order Date": "First Purchase"}, inplace=True)
    df = df.merge(first_purchase, on="Customer ID", how="left")
    df["Customer Tenure (Days)"] = (df["Order Date"] - df["First Purchase"]).dt.day
```

In [50]:
```
df.head()
```

Out[50]:

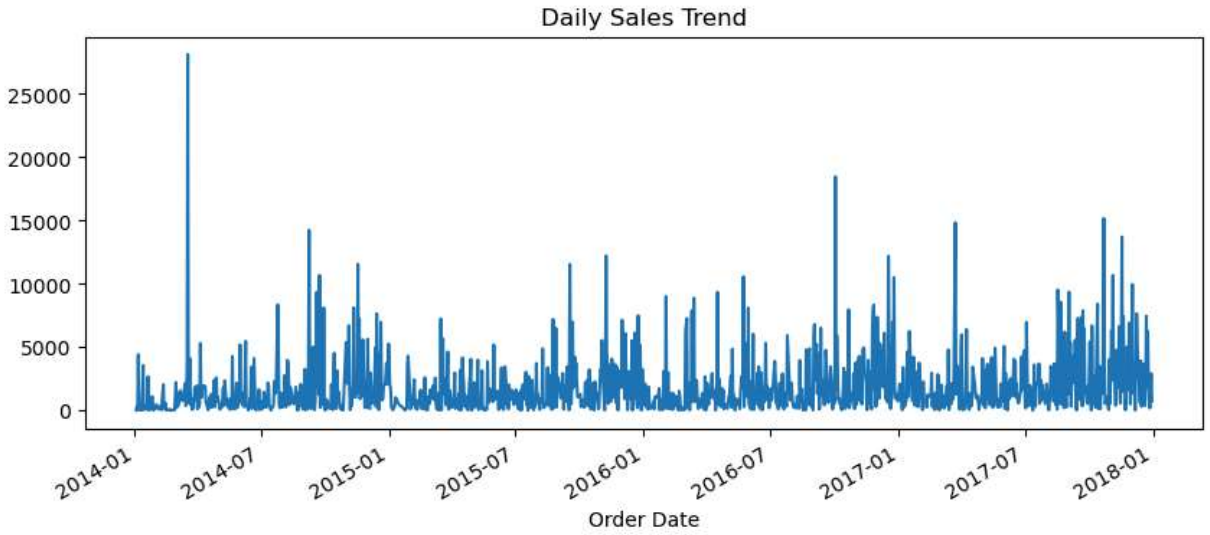
	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Hende
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Hende
2	3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Ang
3	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Lauder
4	5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Lauder

5 rows × 27 columns

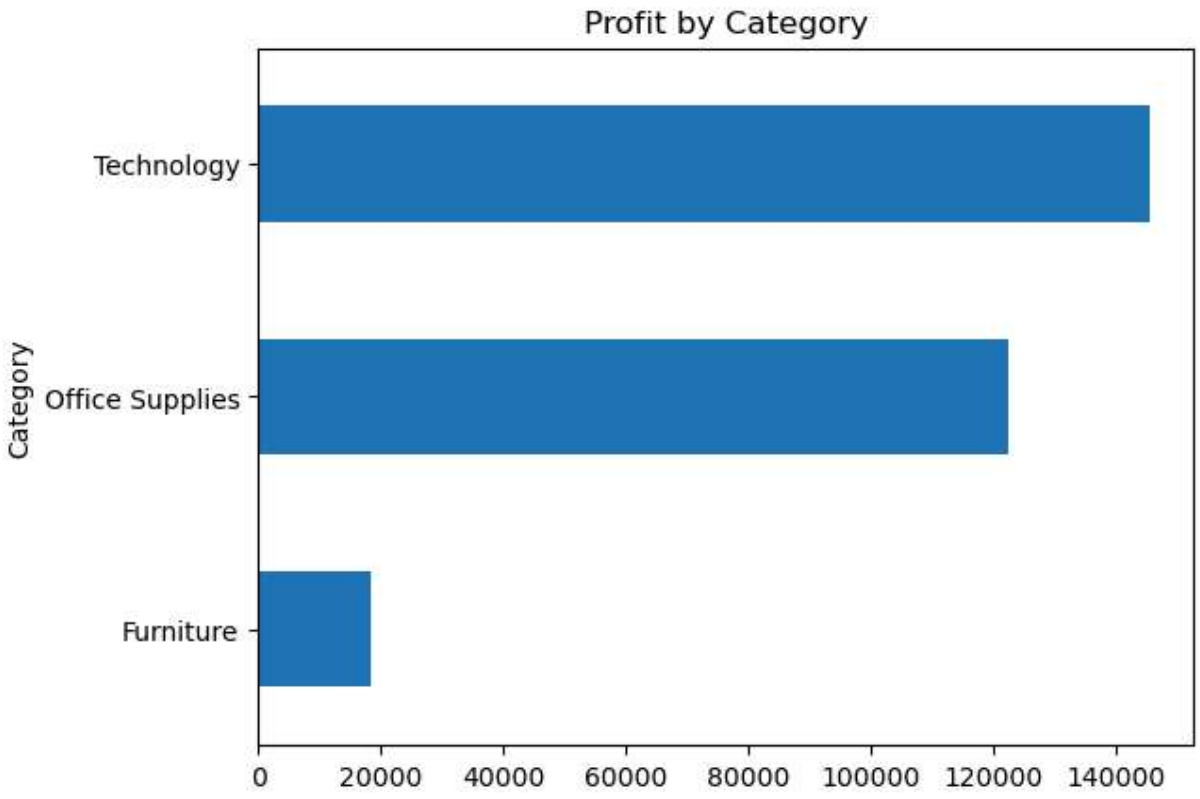
DATA VISUALIZATION

In [51]:
```
sales_trend = df.groupby("Order Date")["Sales"].sum()
sales_trend.plot(figsize=(10,4), title="Daily Sales Trend")
```

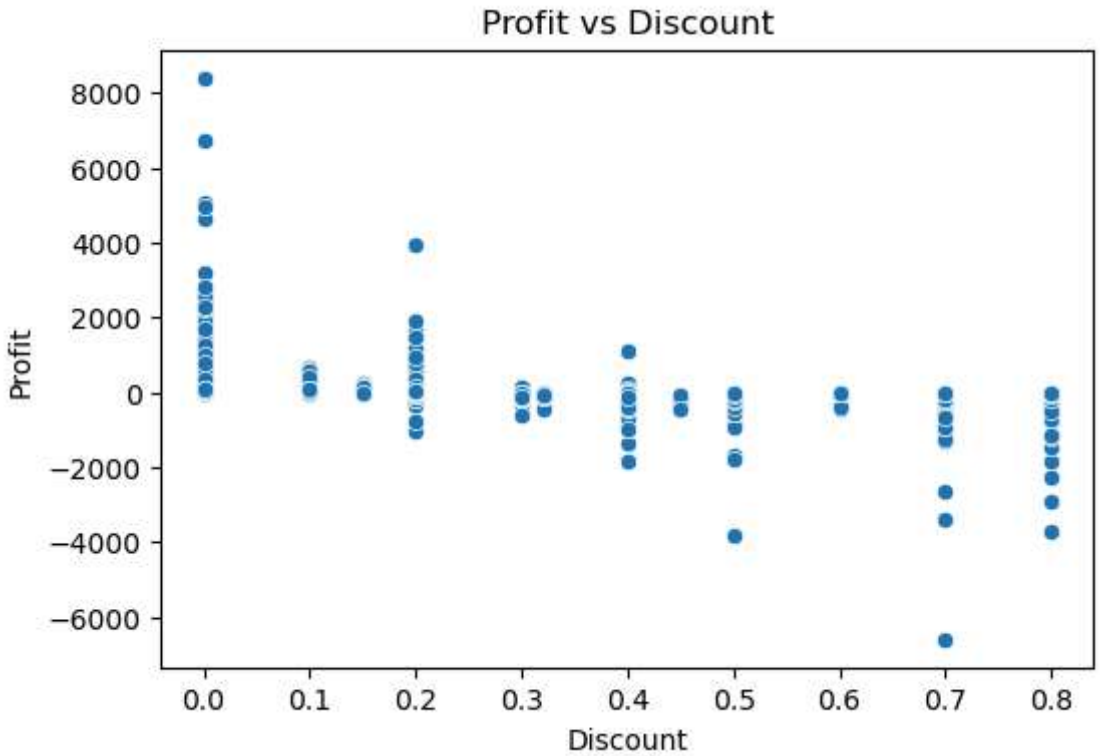
Out[51]:
```
<Axes: title={'center': 'Daily Sales Trend'}, xlabel='Order Date'>
```



In [52]:
```
if "Category" in df.columns:
    profit_by_category = df.groupby("Category")["Profit"].sum().sort_values()
    profit_by_category.plot(kind="barh", title="Profit by Category")
```



In [55]:
```
if "Discount" in df.columns:
    plt.figure(figsize=(6,4))
    sns.scatterplot(data=df, x="Discount", y="Profit")
    plt.title("Profit vs Discount")
```



In [57]:
```
df.to_csv("Cleaned.csv", index=False)
```

In [ ]: