new_sales_project

December 6, 2025

[3]:
```
import pandas as pd
```

[ ]:
```
-----BASIC EDA------
```

[5]:
```
df = pd.read_csv('new_small_sales_dataset.csv')
```

[ ]:
```
---------Load the dataset and show the first 10 rows.
```

[7]:
```
df.head(10)
```

[7]:

	Invoice_ID	Date	Branch	Customer_Type	Product_Line	Unit_Price	Quantity	Total
0	1	2024-01-01 00:00:00	A	Normal	Household	96	6	576
1	2	2024-01-02 00:00:00	B	Normal	Household	62	4	248
2	3	2024-01-03 00:00:00	A	Member	Electronics	75	2	150
3	4	2024-01-04 00:00:00	B	Normal	Electronics	24	8	192
4	5	2024-01-05 00:00:00	B	Normal	Food	39	4	156
5	6	2024-01-06 00:00:00	C	Normal	Household	55	9	495
6	7	2024-01-07 00:00:00	A	Member	Health	25	6	150
7	8	2024-01-08 00:00:00	C	Member	Food	48	6	288
8	9	2024-01-09 00:00:00	A	Member	Household	5	2	10
9	10	2024-01-10 00:00:00	A	Normal	Health	71	8	568

[ ]:
```
----Check for missing values and data types----
```

[9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 8 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Invoice_ID     300 non-null    int64
 1   Date           300 non-null    object
 2   Branch         300 non-null    object
 3   Customer_Type  300 non-null    object
 4   Product_Line   300 non-null    object
 5   Unit_Price     300 non-null    int64
 6   Quantity       300 non-null    int64
 7   Total          300 non-null    int64
dtypes: int64(4), object(4)
memory usage: 18.9+ KB
```

[ ]:
```
---CHANGE DATETYPE----
```

[11]:
```
import datetime
```

[13]:
```
df['Date']=pd.to_datetime(df['Date'])
```

[ ]:
```
----Summary statistics for numeric columns.----
```

[15]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 8 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Invoice_ID     300 non-null    int64
 1   Date           300 non-null    datetime64[ns]
 2   Branch         300 non-null    object
 3   Customer_Type  300 non-null    object
 4   Product_Line   300 non-null    object
 5   Unit_Price     300 non-null    int64
 6   Quantity       300 non-null    int64
 7   Total          300 non-null    int64
dtypes: datetime64[ns](1), int64(4), object(3)
memory usage: 18.9+ KB
```

[ ]:
```
----Summary statistics for numeric columns.----
```

[23]:
```
df[['Unit_Price','Quantity','Total']].describe()
```

[23]:

	Unit_Price	Quantity	Total
count	300.000000	300.000000	300.000000

```
mean     50.130000    4.633333  234.540000
std      29.534285    2.588673  204.301703
min       5.000000    1.000000    6.000000
25%      22.750000    2.000000   69.000000
50%      48.000000    4.000000  176.000000
75%      78.000000    7.000000  360.000000
max      99.000000    9.000000  891.000000
```

[ ]:
```
----Sales Analysis----
```

[ ]:
```
-----Calculate total revenue generated.----
```

[27]:
```
Total_revenue_generated=df['Total'].sum()
```

[29]:
```
Total_revenue_generated
```

[29]:
```
70362
```

[ ]:
```
----Find the top-selling products by quantity----
```

[35]:
```
Top_selling_product=(df.groupby('Product_Line')['Quantity'].sum().
 ↪sort_values(ascending=False))
```

[41]:
```
Top_selling_product.head(1)
```

[41]:
```
Product_Line
Health    404
Name: Quantity, dtype: int64
```

[ ]:
```
----Find the highest revenue-generating customers---
```

[43]:
```
Highest_revenue_generating_customer=(df.groupby('Customer_Type')['Total'].sum().
 ↪sort_values(ascending=False))
```

[45]:
```
Highest_revenue_generating_customer
```

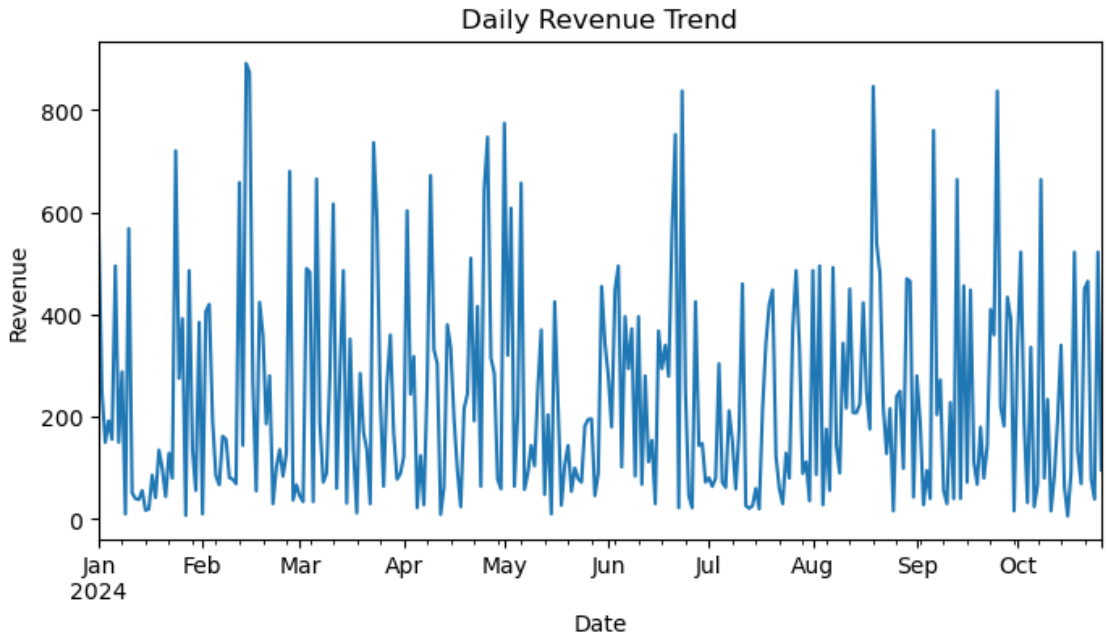
[45]:
```
Customer_Type
Member    36959
Normal    33403
Name: Total, dtype: int64
```

[ ]:
```
----Plot daily sales revenue trend.----
```

[47]:
```
import matplotlib.pyplot as plt
```

[117]:
```
daily = df.groupby("Date")["Total"].sum()
daily.plot(figsize=(8,4))
plt.title("Daily Revenue Trend")
```

```
plt.ylabel("Revenue")
plt.show()
```



[55]:
```
df['Day']=df['Date'].dt.day
```

[57]:
```
df.head()
```

[57]:

	Invoice_ID	Date	Branch	Customer_Type	Product_Line	Unit_Price	Quantity	Total	Day
0	1	2024-01-01	A	Normal	Household	96	6	576	1
1	2	2024-01-02	B	Normal	Household	62	4	248	2
2	3	2024-01-03	A	Member	Electronics	75	2	150	3
3	4	2024-01-04	B	Normal	Electronics	24	8	192	4
4	5	2024-01-05	B	Normal	Food	39	4	156	5

[ ]:
```
----Monthly revenue analysis (group by month).---
```

[59]:
```
df['Month'] = df['Date'].dt.month
```

[61]:
```
df.head()
```

[61]:
```
   Invoice_ID       Date Branch Customer_Type Product_Line  Unit_Price  \
0           1 2024-01-01      A        Normal    Household          96
1           2 2024-01-02      B        Normal    Household          62
2           3 2024-01-03      A        Member  Electronics          75
3           4 2024-01-04      B        Normal  Electronics          24
4           5 2024-01-05      B        Normal         Food          39

   Quantity  Total  Day  Month
0         6    576    1      1
1         4    248    2      1
2         2    150    3      1
3         8    192    4      1
4         4    156    5      1
```

[63]:
```
Monthly_revenue =df.groupby('Month')['Total'].sum()
```

[65]:
```
Monthly_revenue
```

[65]:
```
Month
1     6127
2     7128
3     7578
4     7911
5     6658
6     8256
7     5153
8     8565
9     7336
10    5650
Name: Total, dtype: int64
```

[ ]:
```
----Product-wise average selling price and quantity.---
```

[73]:
```
pro_wise_avg_sell_price_quantity=df.
  ↪groupby('Product_Line')[['Unit_Price','Quantity']].mean()
```

[75]:
```
pro_wise_avg_sell_price_quantity
```

[75]:
```
              Unit_Price  Quantity
Product_Line
Electronics    54.118644  4.610169
Food           49.513889  4.583333
Health         47.219780  4.439560
Household      51.076923  4.923077
```

[ ]:
```
----Customer Behavior---
```

[ ]:
```
----How many unique customers are there?---
```

[95]:
```
no_of_unique_customer=df['Customer_Type'].nunique()
```

[97]:
```
no_of_unique_customer
```

[97]:
```
2
```

[ ]:
```
----Which customer places the most orders?---
```

[103]:
```
customer_wise_order=df.groupby('Customer_Type')['Customer_Type'].count()
```

[105]:
```
customer_wise_order
```

[105]:
```
Customer_Type
Member    148
Normal    152
Name: Customer_Type, dtype: int64
```

[ ]:
```
----Revenue per customer.---
```

[109]:
```
Revenue_per_customer=df.groupby('Customer_Type')['Total'].sum()
```

[111]:
```
Revenue_per_customer
```

[111]:
```
Customer_Type
Member    36959
Normal    33403
Name: Total, dtype: int64
```

[ ]:
```

```