

E_Commerce_Sales_Analysis_Python

February 10, 2026

```
[1]: !pip install pandas openpyxl
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pandas in
c:\users\annap\appdata\roaming\python\python313\site-packages (2.3.2)
Collecting openpyxl
    Downloading openpyxl-3.1.5-py2.py3-none-any.whl.metadata (2.5 kB)
Requirement already satisfied: numpy>=1.26.0 in
c:\users\annap\appdata\roaming\python\python313\site-packages (from pandas)
(2.3.3)
Requirement already satisfied: python-dateutil>=2.8.2 in
c:\users\annap\appdata\roaming\python\python313\site-packages (from pandas)
(2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
c:\users\annap\appdata\roaming\python\python313\site-packages (from pandas)
(2025.2)
Requirement already satisfied: tzdata>=2022.7 in
c:\users\annap\appdata\roaming\python\python313\site-packages (from pandas)
(2025.2)
Collecting et-xmlfile (from openpyxl)
    Downloading et_xmlfile-2.0.0-py3-none-any.whl.metadata (2.7 kB)
Requirement already satisfied: six>=1.5 in
c:\users\annap\appdata\roaming\python\python313\site-packages (from python-
dateutil>=2.8.2->pandas) (1.17.0)
Downloading openpyxl-3.1.5-py2.py3-none-any.whl (250 kB)
Downloading et_xmlfile-2.0.0-py3-none-any.whl (18 kB)
Installing collected packages: et-xmlfile, openpyxl
```

```
----- 1/2 [openpyxl]
```

```
----- 1/2 [openpyxl]
----- 2/2 [openpyxl]
```

Successfully installed et-xmlfile-2.0.0 openpyxl-3.1.5

```
[notice] A new release of pip is available: 25.2 -> 26.0
[notice] To update, run: C:\Program Files\Python313\python.exe -m pip install
--upgrade pip
```

[2]: `import pandas as pd`

[4]: `df = pd.read_csv(r"C:\Users\annap\OneDrive\Desktop\Excel\E-commerce Dataset.
˓→csv")`

[5]: `import pandas as pd`

```
df = pd.read_csv(r"C:\Users\annap\OneDrive\Desktop\Excel\E-commerce Dataset.
˓→csv")
```

```
df.head()
```

[5]:

	User_ID	Product_ID	Category	Price (Rs.)	Discount (%)	\
0	337c166f	f414122f-e	Sports	36.53	15	
1	d38a19bf	fde50f9c-5	Clothing	232.79	20	
2	d7f5f0b0	0d96fc90-3	Sports	317.02	25	
3	395d4994	964fc44b-d	Toys	173.19	25	
4	a83c145c	d70e2fc6-e	Beauty	244.80	20	

```
Final_Price(Rs.) Payment_Method Purchase_Date
0           31.05    Net Banking   12-11-2024
1          186.23    Net Banking   09-02-2024
```

```
2           237.76   Credit Card    01-09-2024
3           129.89      UPI        01-04-2024
4           195.84   Net Banking  27-09-2024
```

```
[6]: df.info()
df.columns
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3660 entries, 0 to 3659
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype  
--- 
 0   User_ID          3660 non-null    object  
 1   Product_ID       3660 non-null    object  
 2   Category         3660 non-null    object  
 3   Price (Rs.)     3660 non-null    float64 
 4   Discount (%)    3660 non-null    int64  
 5   Final_Price(Rs.) 3660 non-null    float64 
 6   Payment_Method   3660 non-null    object  
 7   Purchase_Date    3660 non-null    object  
dtypes: float64(2), int64(1), object(5)
memory usage: 228.9+ KB
```

```
[6]: Index(['User_ID', 'Product_ID', 'Category', 'Price (Rs.)', 'Discount (%)',
       'Final_Price(Rs.)', 'Payment_Method', 'Purchase_Date'],
       dtype='object')
```

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

```
[9]: #What is the total revenue generated?
df["Final_Price(Rs.)"].sum()
```

```
[9]: np.float64(757278.08)
```

```
[10]: #Which category generates the highest revenue?
df.groupby("Category")["Final_Price(Rs.)"].sum().sort_values(ascending=False)
```

```
[10]: Category
Clothing           115314.84
Books              111149.35
Home & Kitchen    110328.08
Sports              108518.79
Toys                107289.69
Beauty              104215.10
Electronics         100462.23
```

```
Name: Final_Price(Rs.), dtype: float64
```

```
[11]: #What is the average discount percentage?  
df["Discount (%)"].mean()
```

```
[11]: np.float64(18.825136612021858)
```

```
[12]: #Which payment method is used most frequently?  
df["Payment_Method"].value_counts()
```

```
[12]: Payment_Method  
Credit Card      760  
UPI             757  
Debit Card       731  
Net Banking      716  
Cash on Delivery 696  
Name: count, dtype: int64
```

```
[14]: #Which category offers the highest average discount?  
df.groupby("Category")["Discount (%)"].mean().sort_values(ascending=False)
```

```
[14]: Category  
Home & Kitchen    19.608379  
Sports            19.326923  
Electronics        19.267068  
Books              19.035581  
Beauty             18.475248  
Toys               18.126195  
Clothing           17.919021  
Name: Discount (%), dtype: float64
```

```
[15]: #What is the Average Order Value (AOV)?  
df["Final_Price(Rs.)"].mean()
```

```
[15]: np.float64(206.90657923497267)
```

```
[16]: #How much revenue is lost due to discounts?  
(df["Price (Rs.)"] - df["Final_Price(Rs.)"]).sum()
```

```
[16]: np.float64(175292.39)
```

```
[17]: #Who are the top 10 high-value customers?  
df.groupby("User_ID")["Final_Price(Rs.)"].sum().sort_values(ascending=False).  
head(10)
```

```
[17]: User_ID  
8b885340      496.82  
20797b76      495.02
```

```
d8970dd2    493.04
da7bc76a    492.41
68722b9b    491.70
05e6557c    487.06
67abda0a    486.79
d646700c    484.56
edb89577    480.49
f9e89622    479.63
Name: Final_Price(Rs.), dtype: float64
```

```
[20]: df["Purchase_Date"] = pd.to_datetime(df["Purchase_Date"], errors="coerce")
```

```
[21]: df["Purchase_Date"].dtype
```

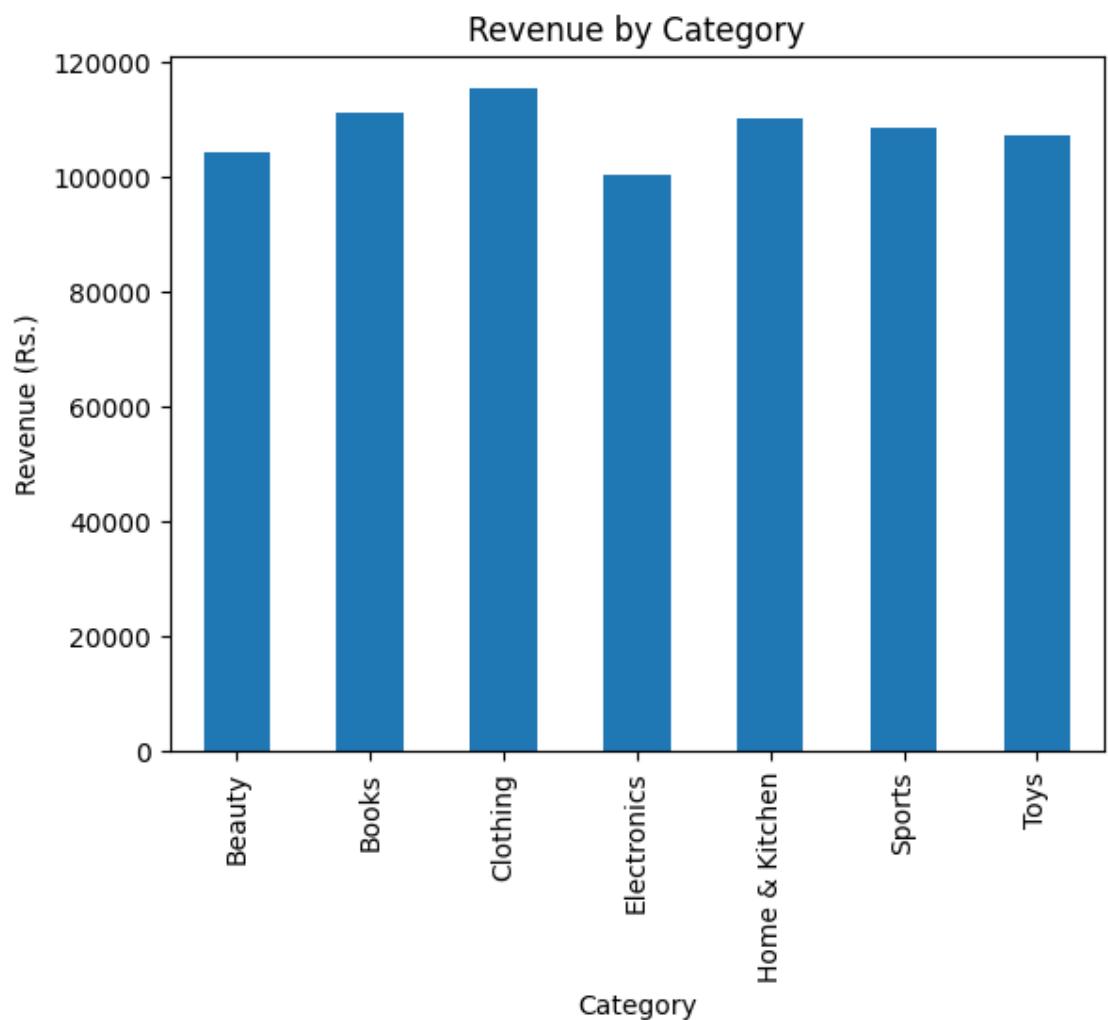
```
[21]: dtype('M8[ns]')
```

```
[24]: #Which day of the week has the highest sales?
weekday_sales = (
    df.groupby(df["Purchase_Date"].dt.day_name())["Final_Price(Rs.)"]
    .sum()
    .sort_values(ascending=False)
)

weekday_sales
```

```
Purchase_Date
Saturday      51273.82
Thursday      47670.47
Monday        43337.35
Wednesday     43108.44
Sunday         41588.57
Friday         38726.10
Tuesday        36754.25
Name: Final_Price(Rs.), dtype: float64
```

```
[25]: #Category-wise Revenue Chart
df.groupby("Category")["Final_Price(Rs.)"].sum().plot(kind="bar")
plt.title("Revenue by Category")
plt.ylabel("Revenue (Rs.)")
plt.show()
```



[]: