Questions: e-comm-sales-analysis

Here are some insightful questions you can explore using Python for data analysis:

General Overview

What is the total number of transactions?

```
df["Transaction_ID"].count()

✓ [27] 14ms

 np.int64(50000)
```

What is the average purchase amount?

```
1   df["Purchase_Amount"].mean()
    ✓ [28] 27ms

      np.float64(503.15979300000004)
```

What are the most and least common payment methods?

```
1   df["Payment_Method"].value_counts()            df
2
    ✓ [35] 21ms
```

Payment_Method	count
UPI	8477
Cash on Delivery	8434
Debit Card	8355
Credit Card	8310
PayPal	8250
Net Banking	8174

What is the distribution of transactions across different countries?

```
df["Country"].value_counts()

✓ [43] 30ms
```

Country	count
Canada	5082
Mexico	5059
Germany	5047
India	4996
France	4993
Australia	4985
USA	4979
Japan	4960
UK	4951
Brazil	4948

Customer Insights

What is the average age of customers making transactions?

```
1  df["Age"].mean()
   ✓ [44] 31ms

   np.float64(43.96868)
```

Which age group spends the most on purchases?

```
df.groupby("Age_Group")["Purchase_Amount"].sum().sort_values(ascending=False)

✓ [54] 57ms
```

Age_Group	Purchase_Amount
Adult	11424012.31
Young Adult	8541038.11
Senior	5192939.23

3 rows ∨ Length: 3, dtype: float64

How many unique users are there in the dataset?

```
df["User_Name"].nunique()
✓ [43] < 10 ms

100
```

What is the most common product category purchased by different age groups?

```
df.groupby("Age_Group")["Product_Category"].value_counts().groupby(level=0).head(1)
✓ [58] 17ms
```

3 rows ∨ Length: 3, dtype: int64

Age_Group	Product_Category	count
Adult	Clothing	2934
Senior	Electronics	1351
Young Adult	Toys	2173

Sales & Revenue Insights

Which product category generates the highest revenue?

```
1  df.groupby("Product_Category")["Purchase_Amount"].sum().sort_values(ascending=False).head(5)
   ✓ [61] < 10 ms
```

5 rows ∨ Length: 5, dtype: float64

Product_Category	Purchase_Amount
Sports	3195335.9
Toys	3185652.36
Books	3181897.3
Clothing	3171225.96
Electronics	3133965.04

What is the total revenue by country?

```
df.groupby("Country")["Purchase_Amount"].sum().sort_values(ascending=False)
✓ [63] < 10 ms
```

Country	Purchase_Amount
France	2545739.19
Canada	2544335.12
USA	2541220.22
Mexico	2534475.67
Australia	2514911.65
Brazil	2507287.54
India	2503542.71
Germany	2502442.2
Japan	2492312.2
UK	2471723.15

10 rows   Length: 10, dtype: float64

Which country has the highest number of transactions?

```
1  df.groupby("Country")["Transaction_ID"].sum().sort_values(ascending=False).head(1)
✓ [65] < 10 ms
```

Country	Transaction_ID
Mexico	126317929

1 row   Length: 1, dtype: int64

What are the top 5 highest-value transactions?

```
df.groupby("Transaction_ID")["Purchase_Amount"].sum().sort_values(ascending=False).head(5)
✓ [67] 14ms
```

Transaction_ID	Purchase_Amount
47133	999.98
41015	999.98
36985	999.95
19236	999.94
20014	999.93

5 rows   Length: 5, dtype: float64

What is the revenue trend over time (daily, weekly, monthly)?

```
daily_revenue = df.resample('D', on="Transaction_Date")["Purchase_Amount"].sum()    daily_revenue
print(daily_revenue.head())

✓ [81] 33ms
Transaction_Date
2023-03-09    34211.92
2023-03-10    35190.87
2023-03-11    33670.82
2023-03-12    40101.44
2023-03-13    32031.16
Freq: D, Name: Purchase_Amount, dtype: float64
```

```
weekly_revenue = df.resample('W', on="Transaction_Date")["Purchase_Amount"].sum()
print(weekly_revenue.head())

✓ [76] 34ms
Transaction_Date
2023-03-12    143175.05
2023-03-19    231590.20
2023-03-26    258201.44
2023-04-02    235633.97
2023-04-09    231284.95
Freq: W-SUN, Name: Purchase_Amount, dtype: float64
```

```
monthly_revenue = df.resample('M', on="Transaction_Date")["Purchase_Amount"].sum()
print(monthly_revenue.head())
✓ [78] 34ms

 Transaction_Date
 2023-03-31     802390.44
 2023-04-30    1046921.48
 2023-05-31    1050553.40
 2023-06-30    1011635.06
 2023-07-31    1105402.88
 Freq: ME, Name: Purchase_Amount, dtype: float64
```

Payment Method Analysis

```
1  df["Payment_Method"].value_counts()        df
   ✓ [87] < 10 ms

   ▦ ∿  6 rows ∨  Length: 6, dtype: int64

   Payment_Method        ⇕   123 count        ⇕
   UPI                              8477
   Cash on Delivery                 8434
   Debit Card                       8355
   Credit Card                      8310
   PayPal                           8250
   Net Banking                      8174
```
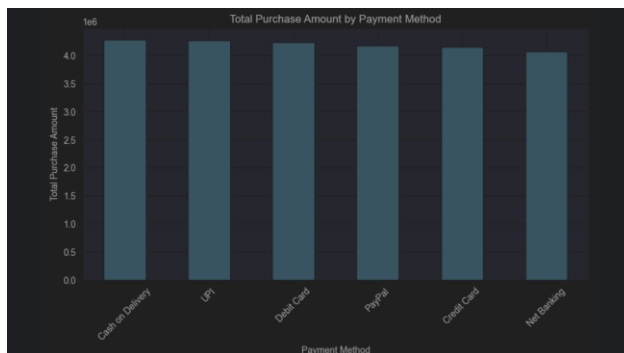
What is the most commonly used payment method?

```
1  df["Payment_Method"].value_counts().head(3)
   ✓ [88] 10ms

   ▦ ∿  3 rows ∨  Length: 3, dtype: int64

   Payment_Method        ⇕   123 count        ⇕
   UPI                              8477
   Cash on Delivery                 8434
   Debit Card                       8355
```

Is there a significant difference in purchase amounts based on payment method?

Total Purchase Amount by Payment Method

```
df.groupby("Payment_Method")["Purchase_Amount"].sum().sort_values(ascending=False)
✓ [90] < 10 ms

▦ ∿  6 rows ∨  Length: 6, dtype: float64

Payment_Method        ⇕   123 Purchase_Amount        ⇕
Cash on Delivery                 4276269.38
UPI                              4259355.65
Debit Card                       4234776.21
PayPal                           4173529.85
Credit Card                      4150547.15
Net Banking                      4063511.41
```

Answer-   Not really .

Time-Based Insights

Which month has the highest number of transactions?

```python
transactions = df.resample('M', on="Transaction_Date")["Transaction_ID"].count().sort_values(ascending=False)
print(transactions.head(1))
```

✓ [98] 35ms

```
Transaction_Date
2024-01-31    2190
Name: Transaction_ID, dtype: int64
```

How does revenue fluctuate over different time periods (weekday vs. weekend)?

```python
revenue_by_day_type = df.groupby("Day_Type")["Purchase_Amount"].sum()
print(revenue_by_day_type)
```

✓ [105] 12ms

```
Day_Type
Weekday    17865111.70
Weekend     7292877.95
Name: Purchase_Amount, dtype: float64
```