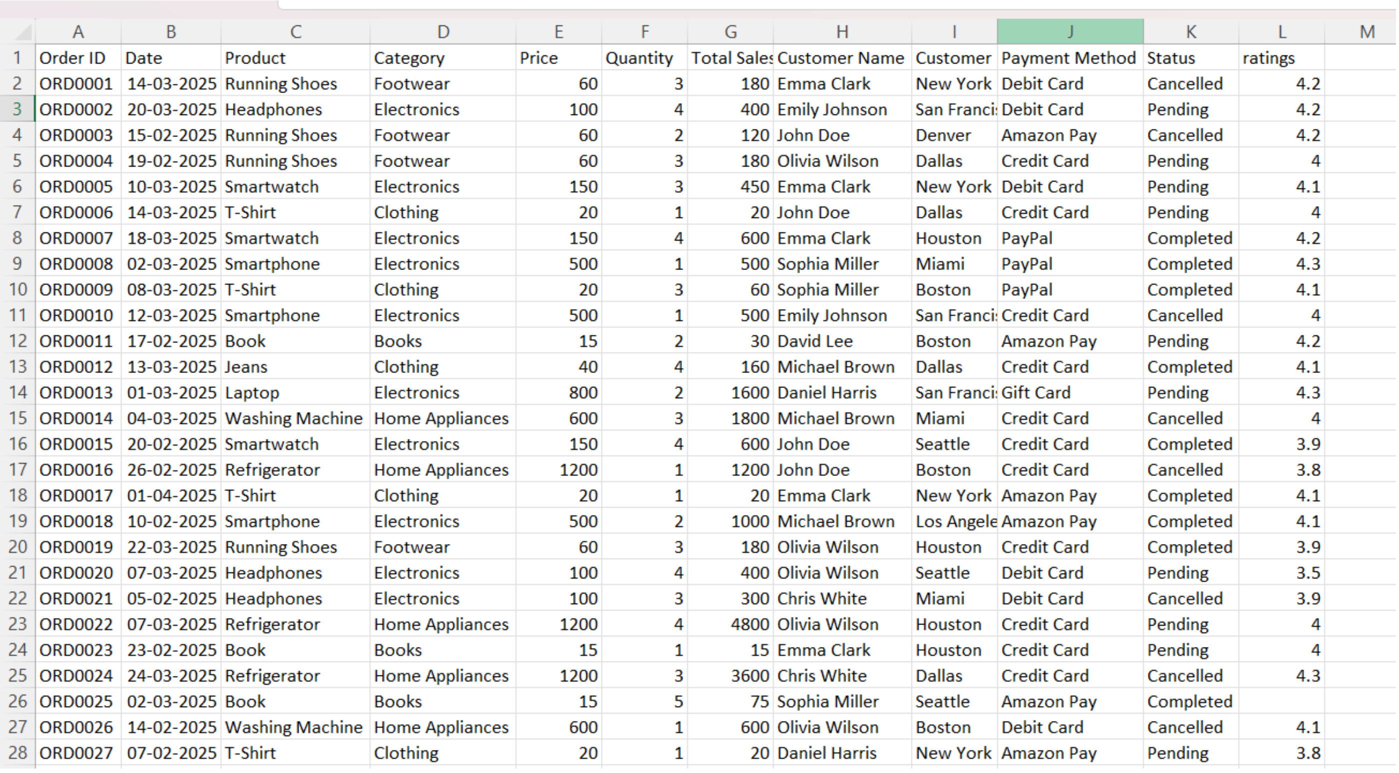
NAME : MANMATH MAROTI KORNULE
ROLL N0 : CS7-59
PRN : 202401110045

SUBJECT : EDS

ASSIGNMENT : THEORY ACTIVITY NO.1

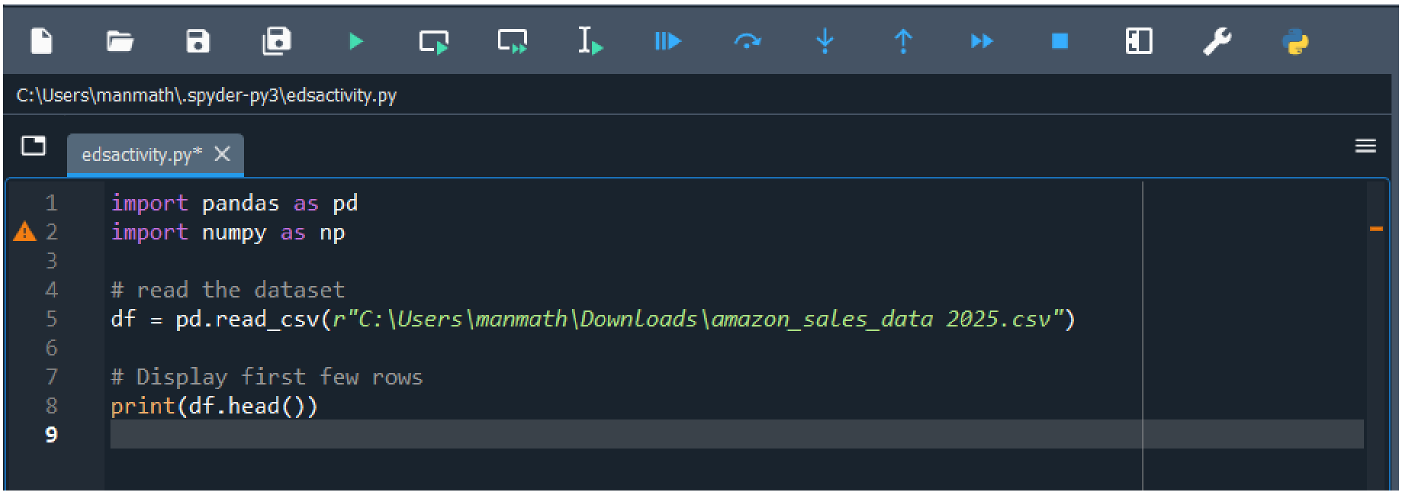
DATASET: AMAZON PRODUCT DATASET

URL: https://www.kaggle.com/datasets/zahidmughal2343/amazon-sales-2025

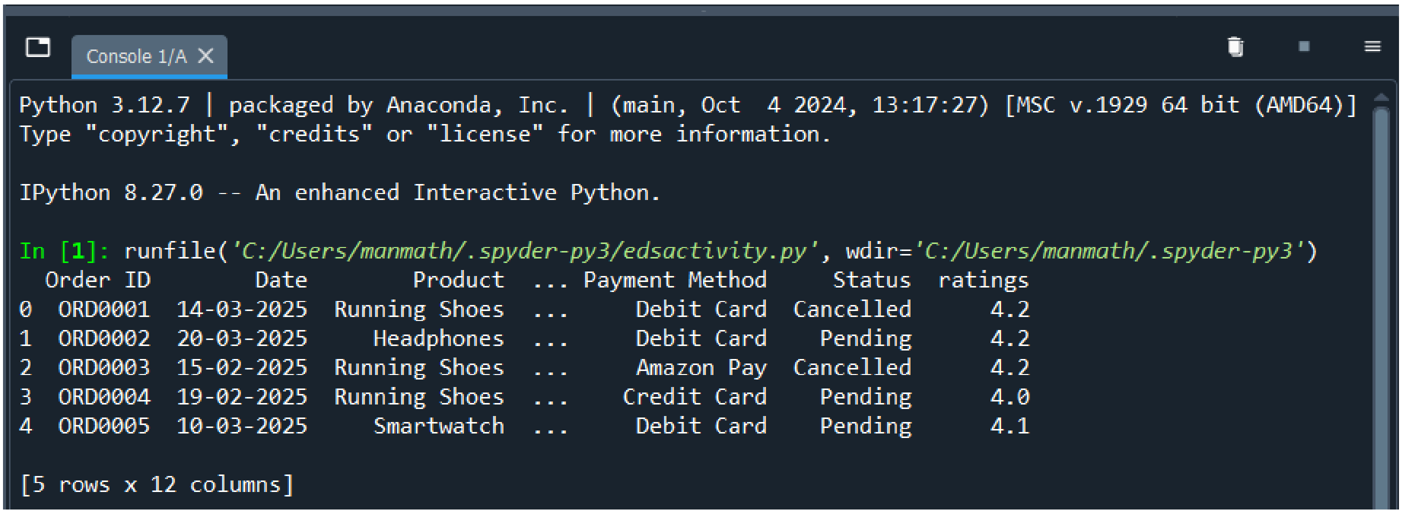


	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Order ID	Date	Product	Category	Price	Quantity	Total Sales	Customer Name	Customer	Payment Method	Status	ratings	
2	ORD0001	14-03-2025	Running Shoes	Footwear	60	3	180	Emma Clark	New York	Debit Card	Cancelled	4.2	
3	ORD0002	20-03-2025	Headphones	Electronics	100	4	400	Emily Johnson	San Franci	Debit Card	Pending	4.2	
4	ORD0003	15-02-2025	Running Shoes	Footwear	60	2	120	John Doe	Denver	Amazon Pay	Cancelled	4.2	
5	ORD0004	19-02-2025	Running Shoes	Footwear	60	3	180	Olivia Wilson	Dallas	Credit Card	Pending	4	
6	ORD0005	10-03-2025	Smartwatch	Electronics	150	3	450	Emma Clark	New York	Debit Card	Pending	4.1	
7	ORD0006	14-03-2025	T-Shirt	Clothing	20	1	20	John Doe	Dallas	Credit Card	Pending	4	
8	ORD0007	18-03-2025	Smartwatch	Electronics	150	4	600	Emma Clark	Houston	PayPal	Completed	4.2	
9	ORD0008	02-03-2025	Smartphone	Electronics	500	1	500	Sophia Miller	Miami	PayPal	Completed	4.3	
10	ORD0009	08-03-2025	T-Shirt	Clothing	20	3	60	Sophia Miller	Boston	PayPal	Completed	4.1	
11	ORD0010	12-03-2025	Smartphone	Electronics	500	1	500	Emily Johnson	San Franci	Credit Card	Cancelled	4	
12	ORD0011	17-02-2025	Book	Books	15	2	30	David Lee	Boston	Amazon Pay	Pending	4.2	
13	ORD0012	13-03-2025	Jeans	Clothing	40	4	160	Michael Brown	Dallas	Credit Card	Completed	4.1	
14	ORD0013	01-03-2025	Laptop	Electronics	800	2	1600	Daniel Harris	San Franci	Gift Card	Pending	4.3	
15	ORD0014	04-03-2025	Washing Machine	Home Appliances	600	3	1800	Michael Brown	Miami	Credit Card	Cancelled	4	
16	ORD0015	20-02-2025	Smartwatch	Electronics	150	4	600	John Doe	Seattle	Credit Card	Completed	3.9	
17	ORD0016	26-02-2025	Refrigerator	Home Appliances	1200	1	1200	John Doe	Boston	Credit Card	Cancelled	3.8	
18	ORD0017	01-04-2025	T-Shirt	Clothing	20	1	20	Emma Clark	New York	Amazon Pay	Completed	4.1	
19	ORD0018	10-02-2025	Smartphone	Electronics	500	2	1000	Michael Brown	Los Angele	Amazon Pay	Completed	4.1	
20	ORD0019	22-03-2025	Running Shoes	Footwear	60	3	180	Olivia Wilson	Houston	Credit Card	Completed	3.9	
21	ORD0020	07-03-2025	Headphones	Electronics	100	4	400	Olivia Wilson	Seattle	Debit Card	Pending	3.5	
22	ORD0021	05-02-2025	Headphones	Electronics	100	3	300	Chris White	Miami	Debit Card	Cancelled	3.9	
23	ORD0022	07-03-2025	Refrigerator	Home Appliances	1200	4	4800	Olivia Wilson	Houston	Credit Card	Pending	4	
24	ORD0023	23-02-2025	Book	Books	15	1	15	Emma Clark	Houston	Credit Card	Pending	4	
25	ORD0024	24-03-2025	Refrigerator	Home Appliances	1200	3	3600	Chris White	Dallas	Credit Card	Cancelled	4.3	
26	ORD0025	02-03-2025	Book	Books	15	5	75	Sophia Miller	Seattle	Amazon Pay	Completed		
27	ORD0026	14-02-2025	Washing Machine	Home Appliances	600	1	600	Olivia Wilson	Boston	Debit Card	Cancelled	4.1	
28	ORD0027	07-02-2025	T-Shirt	Clothing	20	1	20	Daniel Harris	New York	Amazon Pay	Pending	3.8	

20 PROBLEM STATEMENTS BASED ON DATASET :

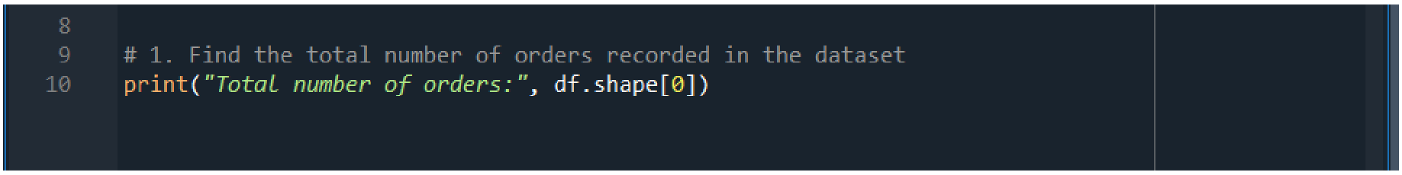


```python
import pandas as pd
import numpy as np

# read the dataset
df = pd.read_csv(r"C:\Users\manmath\Downloads\amazon_sales_data 2025.csv")

# Display first few rows
print(df.head())
```

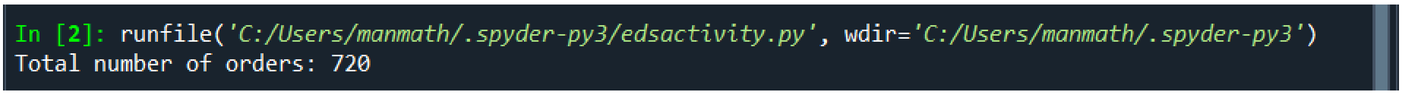


```
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct  4 2024, 13:17:27) [MSC v.1929 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.27.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/manmath/.spyder-py3/edsactivity.py', wdir='C:/Users/manmath/.spyder-py3')
  Order ID        Date       Product  ... Payment Method     Status  ratings
0  ORD0001  14-03-2025  Running Shoes  ...     Debit Card  Cancelled      4.2
1  ORD0002  20-03-2025     Headphones  ...     Debit Card    Pending      4.2
2  ORD0003  15-02-2025  Running Shoes  ...     Amazon Pay  Cancelled      4.2
3  ORD0004  19-02-2025  Running Shoes  ...    Credit Card    Pending      4.0
4  ORD0005  10-03-2025     Smartwatch  ...     Debit Card    Pending      4.1

[5 rows x 12 columns]
```

1.

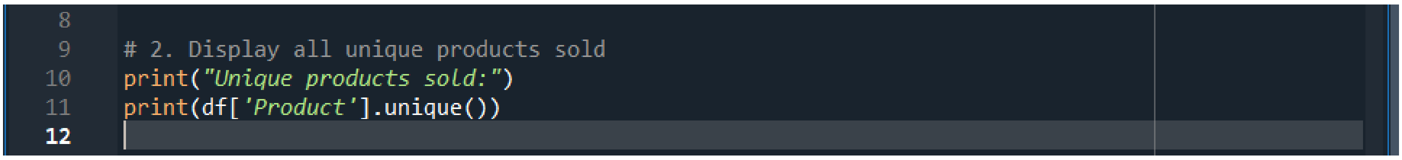


```python

# 1. Find the total number of orders recorded in the dataset
print("Total number of orders:", df.shape[0])
```

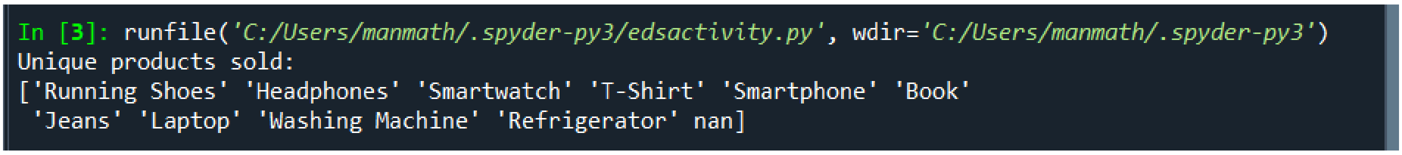


```
In [2]: runfile('C:/Users/manmath/.spyder-py3/edsactivity.py', wdir='C:/Users/manmath/.spyder-py3')
Total number of orders: 720
```

2.

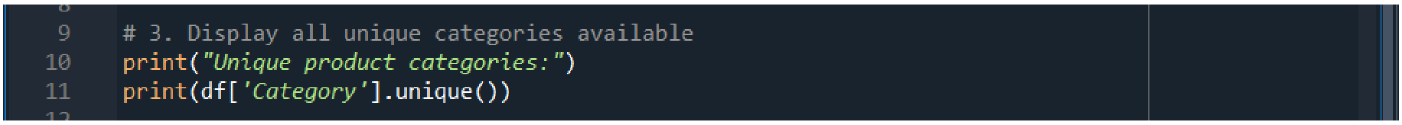


```python

# 2. Display all unique products sold
print("Unique products sold:")
print(df['Product'].unique())
```

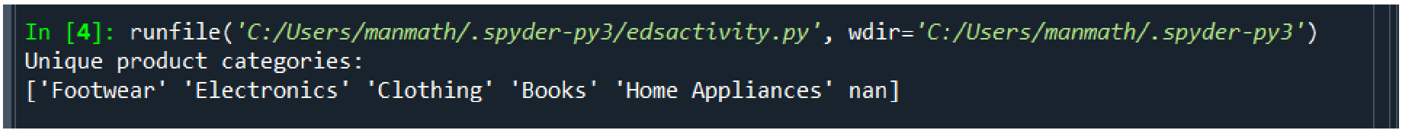


```
In [3]: runfile('C:/Users/manmath/.spyder-py3/edsactivity.py', wdir='C:/Users/manmath/.spyder-py3')
Unique products sold:
['Running Shoes' 'Headphones' 'Smartwatch' 'T-Shirt' 'Smartphone' 'Book'
 'Jeans' 'Laptop' 'Washing Machine' 'Refrigerator' nan]
```

3.

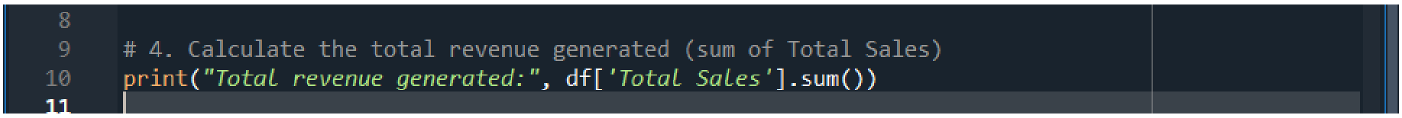


```
# 3. Display all unique categories available
print("Unique product categories:")
print(df['Category'].unique())
```

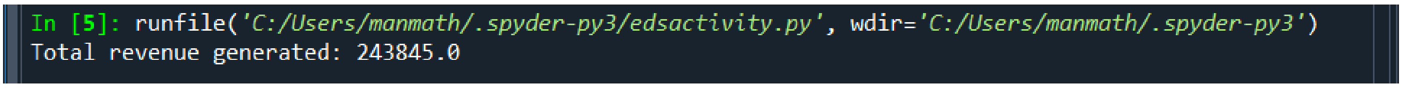


```
In [4]: runfile('C:/Users/manmath/.spyder-py3/edsactivity.py', wdir='C:/Users/manmath/.spyder-py3')
Unique product categories:
['Footwear' 'Electronics' 'Clothing' 'Books' 'Home Appliances' nan]
```

4.

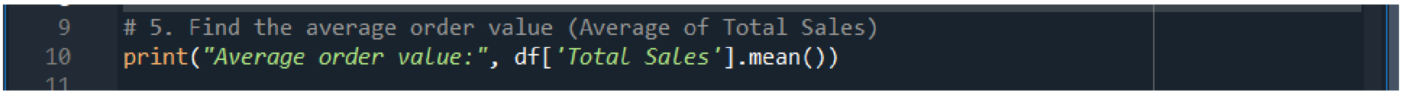


```
# 4. Calculate the total revenue generated (sum of Total Sales)
print("Total revenue generated:", df['Total Sales'].sum())
```

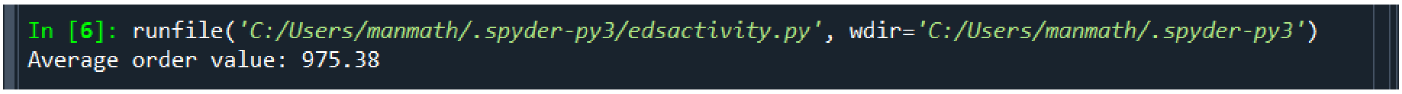


```
In [5]: runfile('C:/Users/manmath/.spyder-py3/edsactivity.py', wdir='C:/Users/manmath/.spyder-py3')
Total revenue generated: 243845.0
```

5.

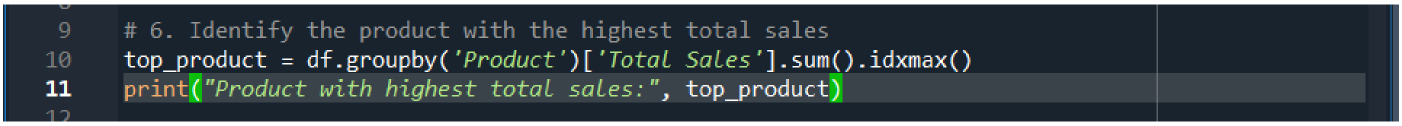


```
# 5. Find the average order value (Average of Total Sales)
print("Average order value:", df['Total Sales'].mean())
```

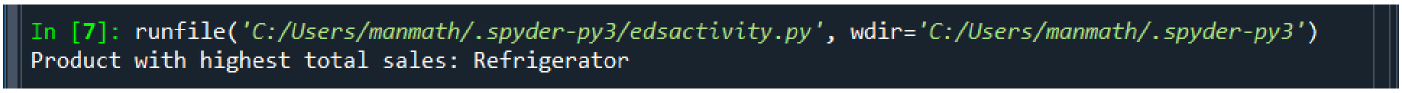


```
In [6]: runfile('C:/Users/manmath/.spyder-py3/edsactivity.py', wdir='C:/Users/manmath/.spyder-py3')
Average order value: 975.38
```

6.

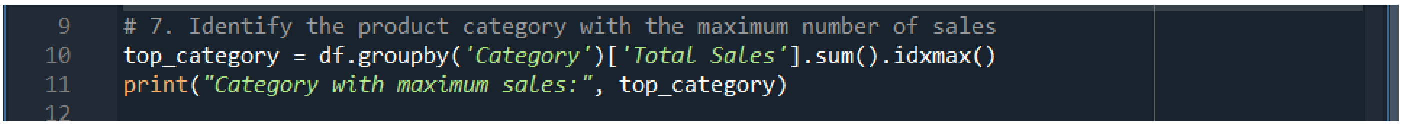


```
# 6. Identify the product with the highest total sales
top_product = df.groupby('Product')['Total Sales'].sum().idxmax()
print("Product with highest total sales:", top_product)
```

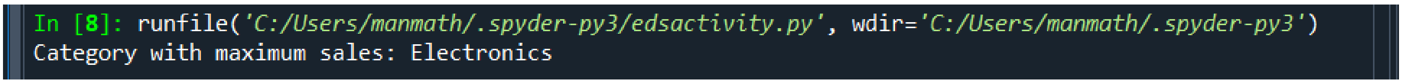


```
In [7]: runfile('C:/Users/manmath/.spyder-py3/edsactivity.py', wdir='C:/Users/manmath/.spyder-py3')
Product with highest total sales: Refrigerator
```

7.

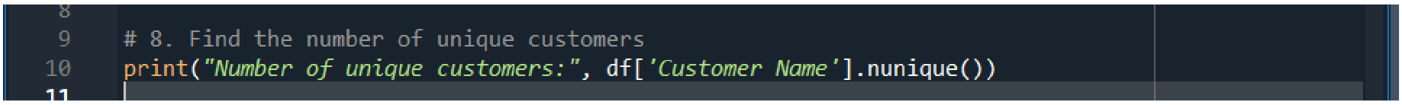


```
# 7. Identify the product category with the maximum number of sales
top_category = df.groupby('Category')['Total Sales'].sum().idxmax()
print("Category with maximum sales:", top_category)
```

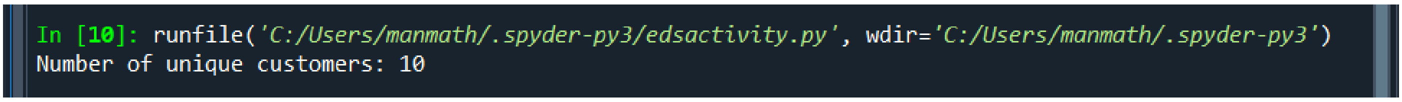


```
In [8]: runfile('C:/Users/manmath/.spyder-py3/edsactivity.py', wdir='C:/Users/manmath/.spyder-py3')
Category with maximum sales: Electronics
```

8.

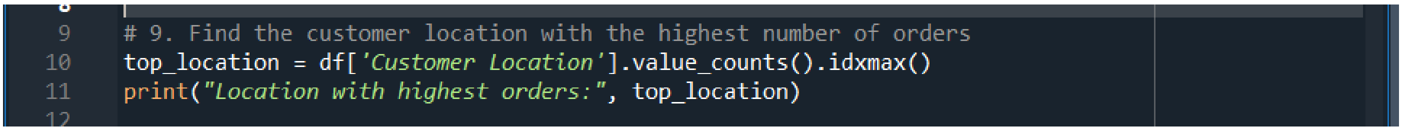


```
# 8. Find the number of unique customers
print("Number of unique customers:", df['Customer Name'].nunique())
```

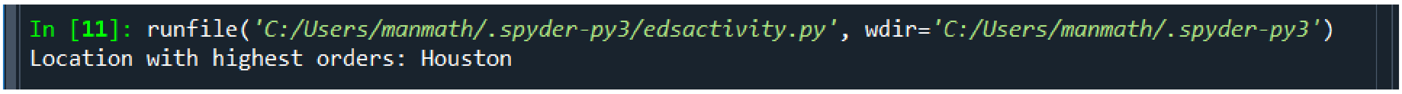


```
In [10]: runfile('C:/Users/manmath/.spyder-py3/edsactivity.py', wdir='C:/Users/manmath/.spyder-py3')
Number of unique customers: 10
```

9.

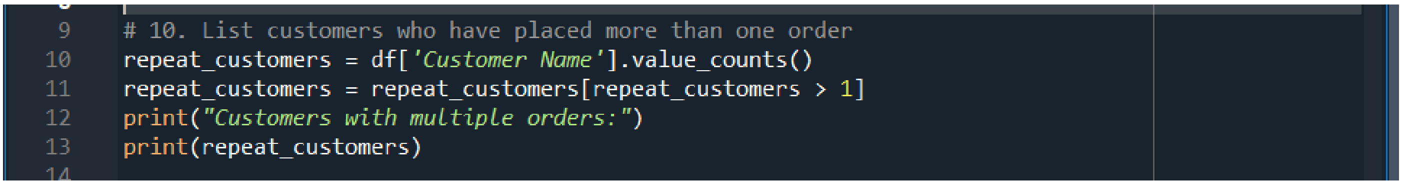


```
# 9. Find the customer location with the highest number of orders
top_location = df['Customer Location'].value_counts().idxmax()
print("Location with highest orders:", top_location)
```

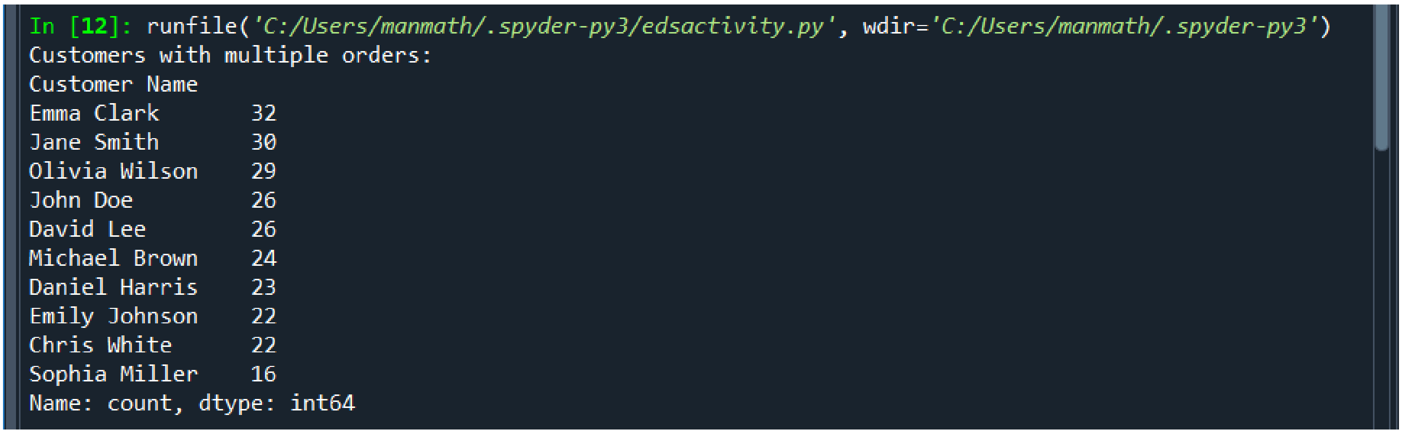


```
In [11]: runfile('C:/Users/manmath/.spyder-py3/edsactivity.py', wdir='C:/Users/manmath/.spyder-py3')
Location with highest orders: Houston
```

10.

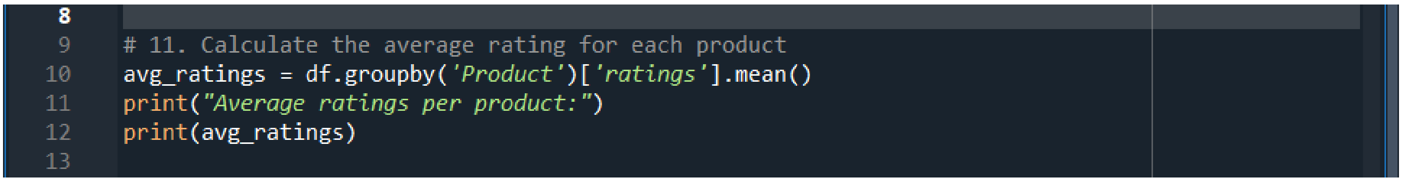


```
# 10. List customers who have placed more than one order
repeat_customers = df['Customer Name'].value_counts()
repeat_customers = repeat_customers[repeat_customers > 1]
print("Customers with multiple orders:")
print(repeat_customers)
```

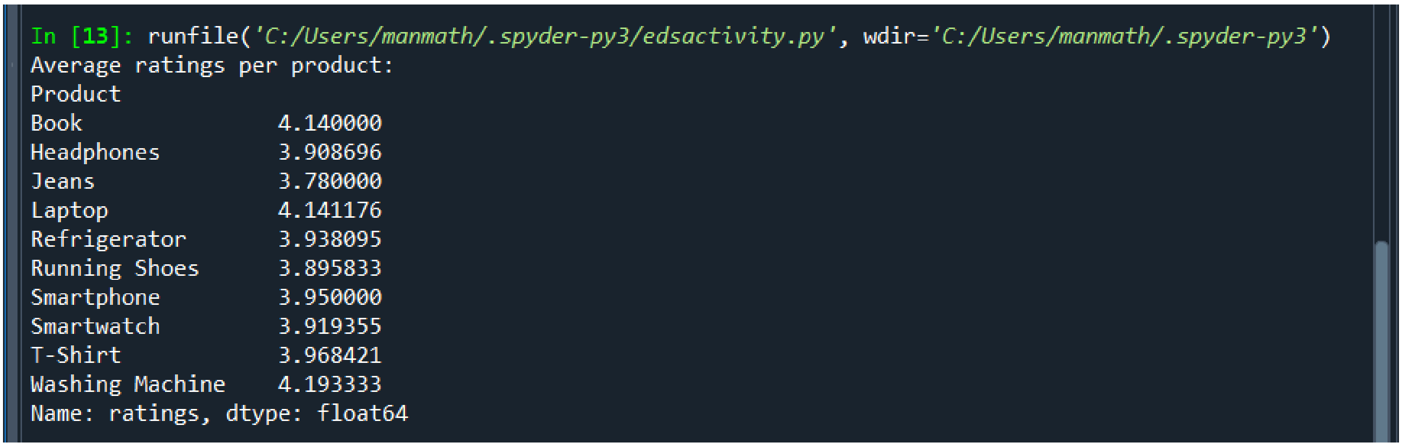


```
In [12]: runfile('C:/Users/manmath/.spyder-py3/edsactivity.py', wdir='C:/Users/manmath/.spyder-py3')
Customers with multiple orders:
Customer Name
Emma Clark        32
Jane Smith        30
Olivia Wilson     29
John Doe          26
David Lee         26
Michael Brown     24
Daniel Harris     23
Emily Johnson     22
Chris White       22
Sophia Miller     16
Name: count, dtype: int64
```

11.

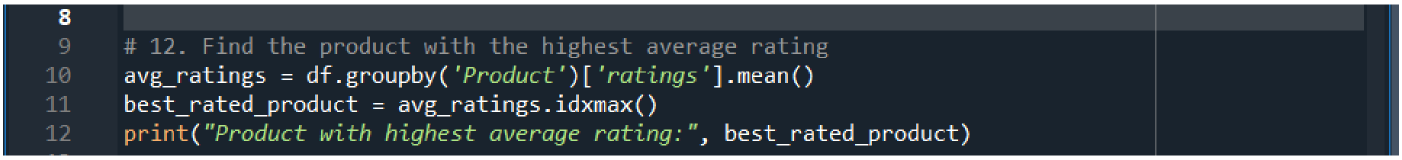


```python
# 11. Calculate the average rating for each product
avg_ratings = df.groupby('Product')['ratings'].mean()
print("Average ratings per product:")
print(avg_ratings)
```

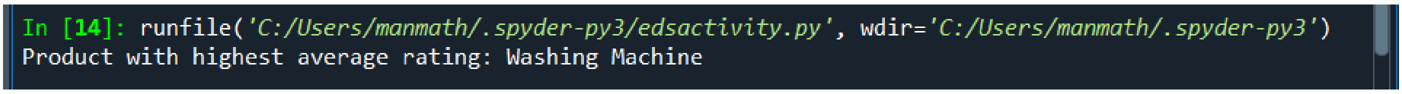


```
In [13]: runfile('C:/Users/manmath/.spyder-py3/edsactivity.py', wdir='C:/Users/manmath/.spyder-py3')
Average ratings per product:
Product
Book               4.140000
Headphones         3.908696
Jeans              3.780000
Laptop             4.141176
Refrigerator       3.938095
Running Shoes      3.895833
Smartphone         3.950000
Smartwatch         3.919355
T-Shirt            3.968421
Washing Machine    4.193333
Name: ratings, dtype: float64
```

12.

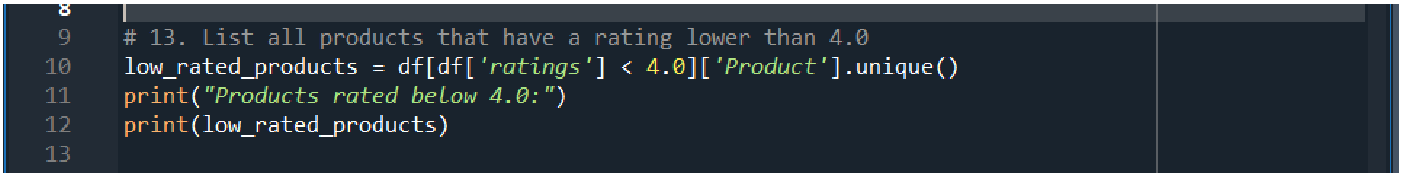


```python
# 12. Find the product with the highest average rating
avg_ratings = df.groupby('Product')['ratings'].mean()
best_rated_product = avg_ratings.idxmax()
print("Product with highest average rating:", best_rated_product)
```

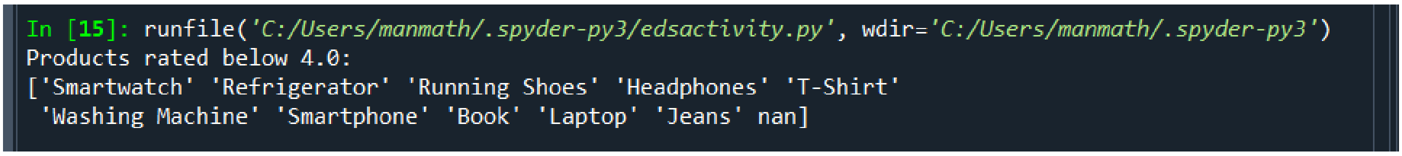


```
In [14]: runfile('C:/Users/manmath/.spyder-py3/edsactivity.py', wdir='C:/Users/manmath/.spyder-py3')
Product with highest average rating: Washing Machine
```

13.

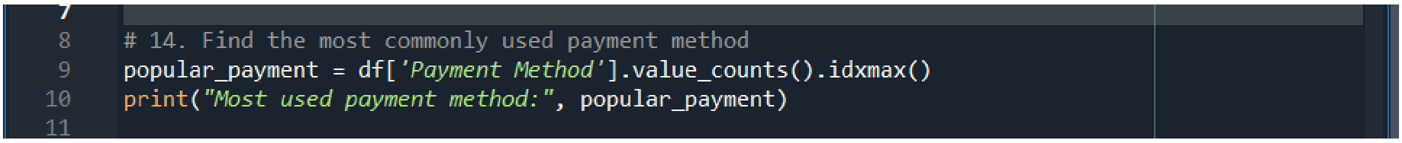


```python
# 13. List all products that have a rating lower than 4.0
low_rated_products = df[df['ratings'] < 4.0]['Product'].unique()
print("Products rated below 4.0:")
print(low_rated_products)
```

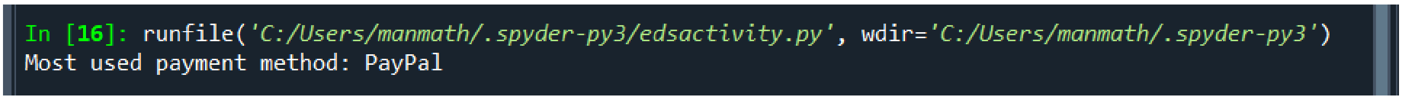


```
In [15]: runfile('C:/Users/manmath/.spyder-py3/edsactivity.py', wdir='C:/Users/manmath/.spyder-py3')
Products rated below 4.0:
['Smartwatch' 'Refrigerator' 'Running Shoes' 'Headphones' 'T-Shirt'
 'Washing Machine' 'Smartphone' 'Book' 'Laptop' 'Jeans' nan]
```

14.

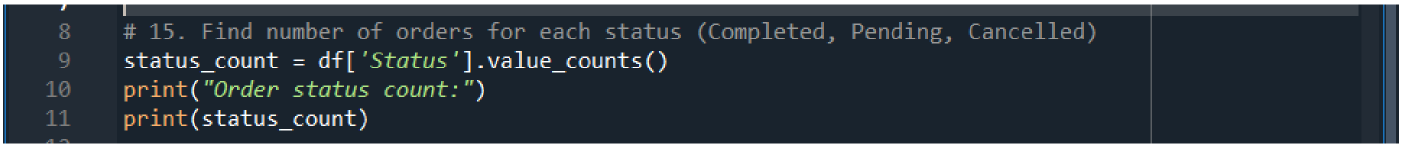


```python
# 14. Find the most commonly used payment method
popular_payment = df['Payment Method'].value_counts().idxmax()
print("Most used payment method:", popular_payment)
```

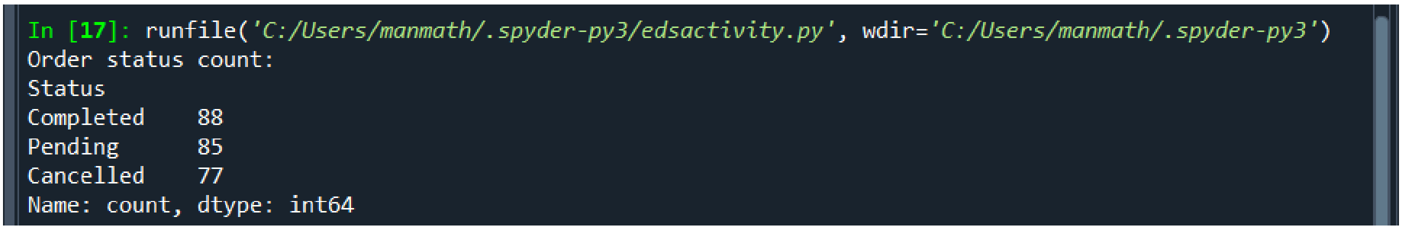


```
In [16]: runfile('C:/Users/manmath/.spyder-py3/edsactivity.py', wdir='C:/Users/manmath/.spyder-py3')
Most used payment method: PayPal
```

15.

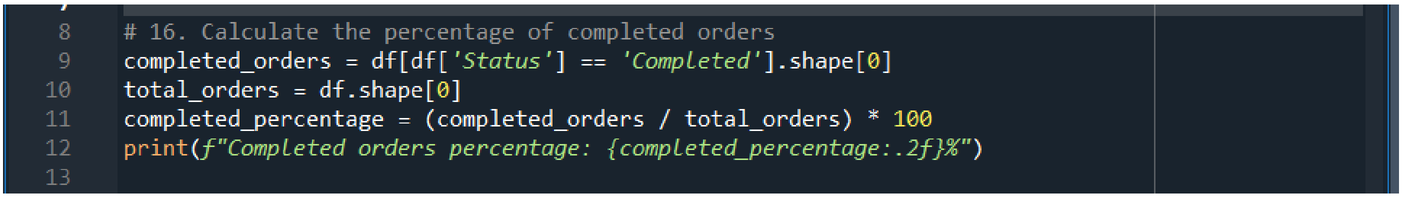


```python
# 15. Find number of orders for each status (Completed, Pending, Cancelled)
status_count = df['Status'].value_counts()
print("Order status count:")
print(status_count)
```

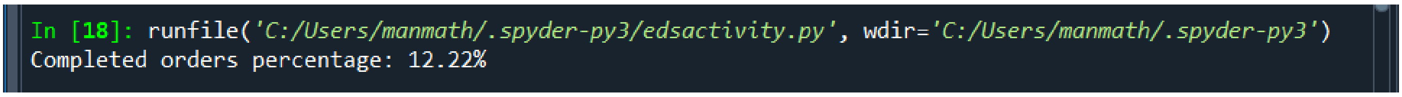


```
In [17]: runfile('C:/Users/manmath/.spyder-py3/edsactivity.py', wdir='C:/Users/manmath/.spyder-py3')
Order status count:
Status
Completed    88
Pending      85
Cancelled    77
Name: count, dtype: int64
```

16.

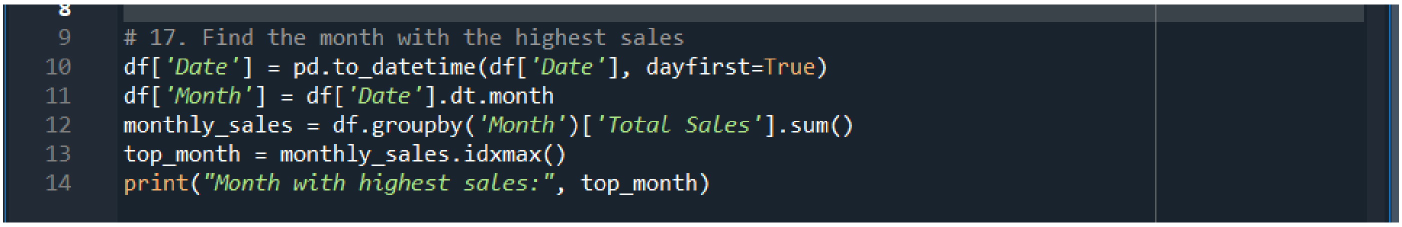


```python
# 16. Calculate the percentage of completed orders
completed_orders = df[df['Status'] == 'Completed'].shape[0]
total_orders = df.shape[0]
completed_percentage = (completed_orders / total_orders) * 100
print(f"Completed orders percentage: {completed_percentage:.2f}%")
```

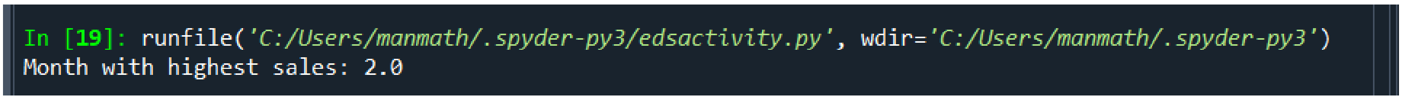


```
In [18]: runfile('C:/Users/manmath/.spyder-py3/edsactivity.py', wdir='C:/Users/manmath/.spyder-py3')
Completed orders percentage: 12.22%
```

17.

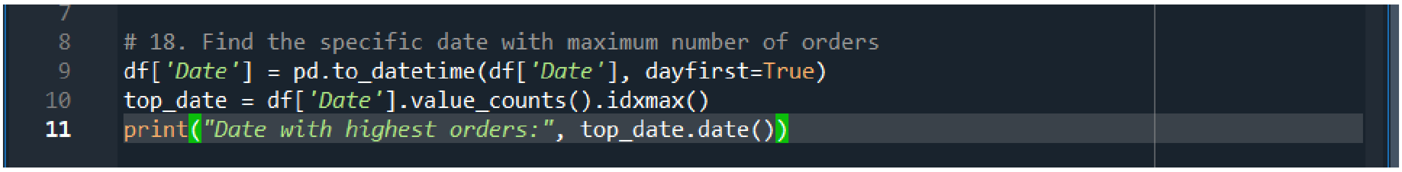


```python
# 17. Find the month with the highest sales
df['Date'] = pd.to_datetime(df['Date'], dayfirst=True)
df['Month'] = df['Date'].dt.month
monthly_sales = df.groupby('Month')['Total Sales'].sum()
top_month = monthly_sales.idxmax()
print("Month with highest sales:", top_month)
```

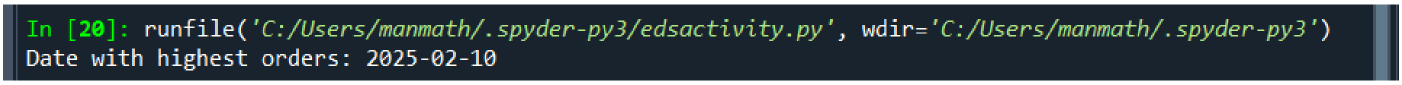


```
In [19]: runfile('C:/Users/manmath/.spyder-py3/edsactivity.py', wdir='C:/Users/manmath/.spyder-py3')
Month with highest sales: 2.0
```

18.

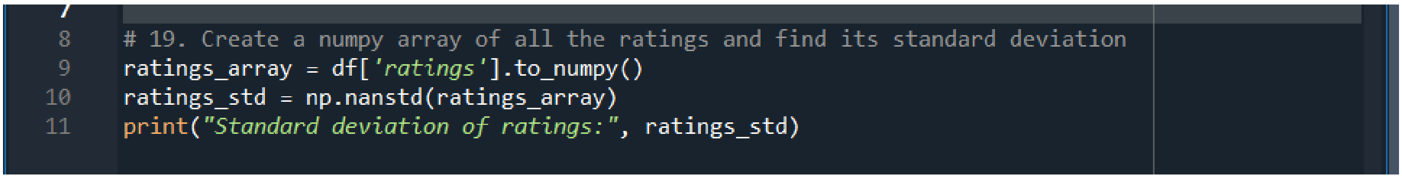


```python
# 18. Find the specific date with maximum number of orders
df['Date'] = pd.to_datetime(df['Date'], dayfirst=True)
top_date = df['Date'].value_counts().idxmax()
print("Date with highest orders:", top_date.date())
```

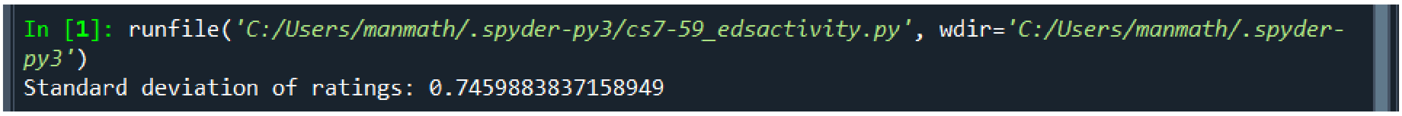


```
In [20]: runfile('C:/Users/manmath/.spyder-py3/edsactivity.py', wdir='C:/Users/manmath/.spyder-py3')
Date with highest orders: 2025-02-10
```

19.

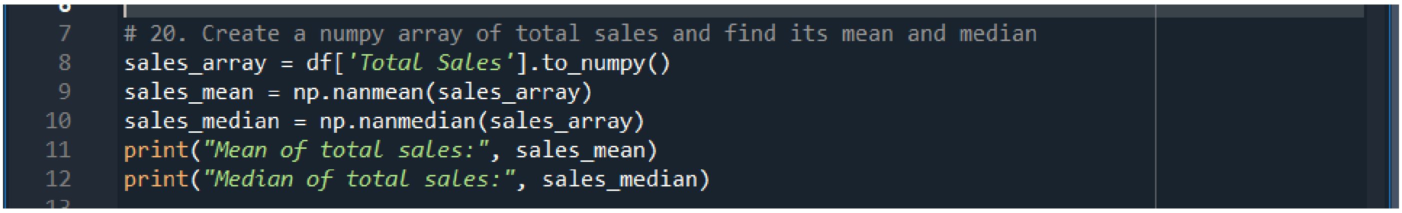


```python
# 19. Create a numpy array of all the ratings and find its standard deviation
ratings_array = df['ratings'].to_numpy()
ratings_std = np.nanstd(ratings_array)
print("Standard deviation of ratings:", ratings_std)
```

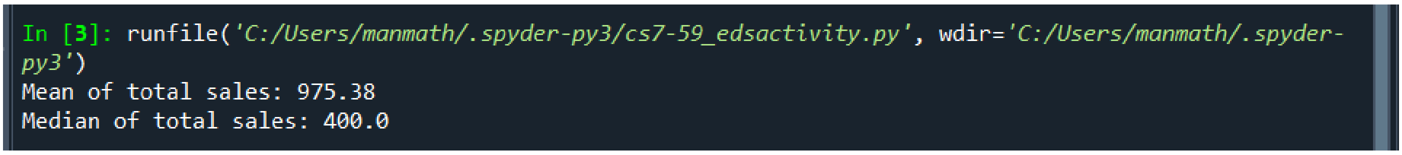


```
In [1]: runfile('C:/Users/manmath/.spyder-py3/cs7-59_edsactivity.py', wdir='C:/Users/manmath/.spyder-
py3')
Standard deviation of ratings: 0.7459883837158949
```

20.



```python
# 20. Create a numpy array of total sales and find its mean and median
sales_array = df['Total Sales'].to_numpy()
sales_mean = np.nanmean(sales_array)
sales_median = np.nanmedian(sales_array)
print("Mean of total sales:", sales_mean)
print("Median of total sales:", sales_median)
```



```
In [3]: runfile('C:/Users/manmath/.spyder-py3/cs7-59_edsactivity.py', wdir='C:/Users/manmath/.spyder-
py3')
Mean of total sales: 975.38
Median of total sales: 400.0
```