EDS Theory Activity 1 Dataset :- Amazon Product

Name: Devansh Wagh

Roll no. :- CC-18

PRN: 202401050024

Division: CC

Python Code for Given Dataset:

EDS theory act.1

File Edit View Insert Runtime Tools Help

Commands + Code + Text

```
import pandas as pd
import numpy as np

# Create a sample Amazon Product dataset
data = {
    'Product_ID': [f'P{i:03d}' for i in range(1, 21)],
    'Product_Name': ['Laptop', 'Phone', 'Shoes', 'Headphones', 'Backpack',
                     'Watch', 'T-shirt', 'Camera', 'Tablet', 'Sunglasses',
                     'Book', 'Charger', 'Speaker', 'Keyboard', 'Mouse',
                     'Monitor', 'Printer', 'Router', 'Powerbank', 'TV'],
    'Category': ['Electronics', 'Electronics', 'Fashion', 'Electronics', 'Fashion',
                 'Fashion', 'Fashion', 'Electronics', 'Electronics', 'Fashion',
                 'Books', 'Electronics', 'Electronics', 'Electronics', 'Electronics',
                 'Electronics', 'Electronics', 'Electronics', 'Electronics', 'Electronics'],
    'Sub_Category': ['Laptop', 'Phone', 'Footwear', 'Audio', 'Bags',
                     'Accessories', 'Apparel', 'Camera', 'Tablet', 'Accessories',
                     'Education', 'Accessories', 'Audio', 'Accessories', 'Accessories',
                     'Monitor', 'Printer', 'Networking', 'Accessories', 'Television'],
    'Price': np.random.randint(100, 2000, size=20),
    'Discount': np.random.randint(5, 30, size=20),
    'Rating': np.round(np.random.uniform(2.5, 5.0, size=20), 1),
    'Number_of_Reviews': np.random.randint(10, 5000, size=20),
    'Stock_Available': np.random.randint(0, 100, size=20),
    'Seller_Name': ['SellerA', 'SellerB', 'SellerC', 'SellerD', 'SellerE',
                    'SellerF', 'SellerG', 'SellerH', 'SellerI', 'SellerJ',
                    'SellerK', 'SellerL', 'SellerM', 'SellerN', 'SellerO',
                    'SellerP', 'SellerQ', 'SellerR', 'SellerS', 'SellerT'],
    'Delivery_Time_days': np.random.randint(1, 10, size=20)
}

df = pd.DataFrame(data)

print("Dataset:")
print(df)
print("\n" + "="*80 + "\n")
```

Output:

```
Dataset:
   Product_ID Product_Name     Category Sub_Category  Price  Discount  Rating  \
0        P001       Laptop  Electronics       Laptop   1128        22     4.9
1        P002        Phone  Electronics        Phone    938         9     2.9
2        P003        Shoes      Fashion     Footwear   1221        26     3.8
3        P004   Headphones  Electronics        Audio   1306        21     4.9
4        P005     Backpack      Fashion         Bags    474        28     4.8
5        P006        Watch      Fashion  Accessories   1017        14     2.6
6        P007      T-shirt      Fashion      Apparel    515        23     4.3
7        P008       Camera  Electronics       Camera   1876         5     2.5
8        P009       Tablet  Electronics       Tablet    540        17     4.8
9        P010   Sunglasses      Fashion  Accessories    680         8     4.7
10       P011         Book        Books    Education    224        19     4.4
11       P012      Charger  Electronics  Accessories   1234        13     4.5
12       P013      Speaker  Electronics        Audio   1830        19     2.8
13       P014     Keyboard  Electronics  Accessories    716        25     4.8
14       P015        Mouse  Electronics  Accessories   1207        12     3.9
15       P016      Monitor  Electronics      Monitor    787        13     4.3
16       P017      Printer  Electronics      Printer   1148         9     2.6
17       P018       Router  Electronics   Networking    313        25     3.2
18       P019    Powerbank  Electronics  Accessories    455        20     3.2
19       P020           TV  Electronics   Television   1187        27     4.5
```

	Number_of_Reviews	Stock_Available	Seller_Name	Delivery_Time_days
0	2829	44	SellerA	6
1	855	41	SellerB	5
2	3307	66	SellerC	3
3	4281	95	SellerD	2
4	1845	37	SellerE	6
5	3397	12	SellerF	9
6	2359	18	SellerG	3
7	4438	3	SellerH	4
8	3592	61	SellerI	5
9	4976	46	SellerJ	6
10	3163	19	SellerK	3
11	4217	98	SellerL	9
12	4591	53	SellerM	7
13	1873	96	SellerN	1
14	4833	76	SellerO	1
15	3033	34	SellerP	1
16	1427	54	SellerQ	2
17	4583	15	SellerR	5
18	3050	45	SellerS	4
19	4552	5	SellerT	5

==========

Problem Statements:

1 . Find the average price of all products

```
[6]  print("1. Find the average Price of all products.")
     print(df['Price'].mean())
```

```
1. Find the average Price of all products.
972.9
```

2 . Find the product with the highest Rating

```
[7]  print("\n2. Find the product with the highest Rating.")
     print(df[df['Rating'] == df['Rating'].max()][['Product_Name', 'Rating']])
```

```
2. Find the product with the highest Rating.
   Product_Name  Rating
12      Speaker     5.0
```

3 . How many products belong to Fashion Category

```
[8]  print("\n3. How many products belong to 'Fashion' Category?")
     print((df['Category'] == 'Fashion').sum())
```

```
3. How many products belong to 'Fashion' Category?
5
```

4 . List all products with price greater than 1000

```
[27] print("\n4. List all products with Price greater than 1000.")
     print(df[df['Price'] > 1000][['Product_Name', 'Price']])
```

```
4. List all products with Price greater than 1000.
   Product_Name  Price
0        Laptop   1511
1         Phone   1260
4      Backpack   1327
5         Watch   1305
7        Camera   1050
8        Tablet   1993
11      Charger   1285
12      Speaker   1189
13     Keyboard   1638
14        Mouse   1248
18    Powerbank   1032
```

5 . Find the maximum Discount offered

```
print("\n5. Find the maximum Discount offered.")
print(df['Discount'].max())
```

```
5. Find the maximum Discount offered.
29
```

6 . Calculate the final price after discount for each product

```
[11] print("\n6. Calculate the final price after discount for each product.")
     df['Final_Price'] = df['Price'] - (df['Price'] * df['Discount'] / 100)
     print(df[['Product_Name', 'Final_Price']])
```

```
6. Calculate the final price after discount for each product.
   Product_Name  Final_Price
0        Laptop      1072.81
1         Phone       932.40
2         Shoes       113.05
3    Headphones       144.40
4      Backpack       995.25
5         Watch       965.70
6       T-shirt       873.84
7        Camera       924.00
8        Tablet      1434.96
9    Sunglasses       547.36
10         Book       480.34
11      Charger       963.75
12      Speaker      1105.77
13     Keyboard      1310.40
14        Mouse       886.08
15      Monitor       256.32
16      Printer       315.35
17       Router       638.25
18    Powerbank       928.80
19           TV       533.70
```

7 . Which seller has the most products listed ?

```
[12] print("\n7. Which Seller has the most products listed?")
     print(df['Seller_Name'].value_counts().idxmax())
```

```
7. Which Seller has the most products listed?
SellerA
```

8 . Find products with rating less than 3.

```
[13] print("\n8. Find products with Rating less than 3.0")
     print(df[df['Rating'] < 3.0][['Product_Name', 'Rating']])
```

```
8. Find products with Rating less than 3.0
   Product_Name  Rating
0        Laptop     2.6
1         Phone     2.7
10         Book     2.8
19           TV     2.9
```

9 . Find the average Delivery Time.

```
[14] print("\n9. Find the average Delivery Time.")
     print(df['Delivery_Time_days'].mean())
```

```
9. Find the average Delivery Time.
5.55
```

10 . List products with stock less than 10.

```
[15] print("\n10. List products with Stock less than 10.")
     print(df[df['Stock_Available'] < 10][['Product_Name', 'Stock_Available']])
```

```
10. List products with Stock less than 10.
   Product_Name  Stock_Available
3    Headphones                3
4      Backpack                9
16      Printer                0
```

11 . Find total number of reviews for Electronics category.

```
[16] print("\n11. Find total number of Reviews for Electronics category.")
     print(df[df['Category'] == 'Electronics']['Number_of_Reviews'].sum())
```

```
11. Find total number of Reviews for Electronics category.
29889
```

12 . List the top 5 products with highest number of reviews.

```
[17] print("\n12. List the top 5 products with highest number of reviews.")
     print(df[['Product_Name', 'Number_of_Reviews']].sort_values(by='Number_of_Reviews', ascending=False).head(5))
```

```
12. List the top 5 products with highest number of reviews.
   Product_Name  Number_of_Reviews
19           TV               4957
2         Shoes               4700
16      Printer               4591
6       T-shirt               4172
11      Charger               4101
```

13 . Count how many products have Rating >=4.5

```
[18] print("\n13. Count how many products have Rating >= 4.5")
     print((df['Rating'] >= 4.5).sum())
```

```
13. Count how many products have Rating >= 4.5
5
```

14 . List products ordered by price descending

```
[19] print("\n14. List products ordered by Price descending.")
     print(df[['Product_Name', 'Price']].sort_values(by='Price', ascending=False))
```

```
14. List products ordered by Price descending.
   Product_Name  Price
8        Tablet   1993
13     Keyboard   1638
0        Laptop   1511
4      Backpack   1327
5         Watch   1305
11      Charger   1285
1         Phone   1260
14        Mouse   1248
12      Speaker   1189
7        Camera   1050
18    Powerbank   1032
6       T-shirt    993
17       Router    851
9    Sunglasses    622
19           TV    593
10         Book    511
16      Printer    371
15      Monitor    356
3    Headphones    190
2         Shoes    133
```

15 . Find the average price for each Category

```
[20] print("\n15. Find the average Price for each Category.")
     print(df.groupby('Category')['Price'].mean())
```

```
15. Find the average Price for each Category.
Category
Books           511.0
Electronics    1040.5
Fashion         876.0
Name: Price, dtype: float64
```

16 . Find products with final price <500 after discount

```
[21] print("\n16. Find products with Final Price < 500 after discount.")
     print(df[df['Final_Price'] < 500][['Product_Name', 'Final_Price']])
```

```
16. Find products with Final Price < 500 after discount.
   Product_Name  Final_Price
2         Shoes       113.05
3    Headphones       144.40
10         Book       480.34
15      Monitor       256.32
16      Printer       315.35
```

17 . What is the minimum stock available across products

```
[22] print("\n17. What is the minimum Stock_Available across products?")
     print(df['Stock_Available'].min())
```

```
17. What is the minimum Stock_Available across products?
0
```

18 . How many unique sellers are there ?

```
[23] print("\n18. How many unique Sellers are there?")
     print(df['Seller_Name'].nunique())
```

```
18. How many unique Sellers are there?
20
```

19 . List products with delivery time more than 5 days.

```
[24] print("\n19. List products with delivery time more than 5 days.")
     print(df[df['Delivery_Time_days'] > 5][['Product_Name', 'Delivery_Time_days']])
```

```
19. List products with delivery time more than 5 days.
   Product_Name  Delivery_Time_days
0        Laptop                   7
1         Phone                   9
2         Shoes                   6
4      Backpack                   9
7        Camera                   6
9    Sunglasses                   6
12      Speaker                   9
13     Keyboard                   6
14        Mouse                   9
16      Printer                   8
```

20 . Find correlation between price and number of reviews

```
[25] print("\n20. Find correlation between Price and Number_of_Reviews.")
     print(df[['Price', 'Number_of_Reviews']].corr())
```

```
20. Find correlation between Price and Number_of_Reviews.
                      Price  Number_of_Reviews
Price              1.000000          -0.452272
Number_of_Reviews -0.452272           1.000000
```