EDS ASSIGNMENT

Data Set : Amazon Product Dataset

Name : Snehal Shripati Mundhe
Div : CS5
Batch : C53
Roll No. : 59
PRN : 202401100096

```
import pandas as pd
```

```
from google.colab import files
uploaded = files.upload()
```

Choose files amazon_pr..._dataset.csv

- **amazon_products_400_dataset.csv**(text/csv) - 30009 bytes, last modified: 28/04/2025 - 100% done

```
path="/content/amazon_products_400_dataset.csv"
df=pd.read_csv(path)
df.head(400)
```

	ProductID	Title	Category	Subcategory	Price	Rating	Number_of_Reviews	Availability	Date_Added	Discount_Percentag
0	P0001	Laptop Stand	Home	Monitors	139.21	4.6	4887	Yes	2023-01-01	4
1	P0002	Robot Vacuum	Accessories	Furniture	140.58	4.7	2758	Yes	2023-01-02	1
2	P0003	Action Camera	Accessories	Phone Accessories	368.77	4.3	950	No	2023-01-03	
3	P0004	LED Desk Lamp	Wearables	Cameras	132.86	4.3	3134	Yes	2023-01-04	1
4	P0005	USB-C Hub	Accessories	Audio	320.22	4.3	2059	Yes	2023-01-05	1
...	...	...	...	...	...	...	...	...	...	..
395	P0396	Noise Cancelling Headphones	Electronics	Health Devices	95.35	4.1	3098	Yes	2024-01-31	1
396	P0397	Smartwatch	Electronics	Watches	480.78	4.9	2155	Yes	2024-02-01	1

Next steps: Generate code with df View recommended plots New interactive sheet

```
import numpy as np
avg_price = np.mean(df['Price'])
avg_price
```

```
np.float64(259.59725)
```

```
price_std = np.std(df['Price'])
price_std
```

```
146.5361043444158
```

```
high_rating_count = np.sum(df['Rating'] > 4.5)
high_rating_count
```

```
np.int64(98)
```

```
max_reviews = np.max(df['Number_of_Reviews'])
max_reviews
```

```
4998
```

```
min_discount = np.min(df['Discount_Percentage'])
min_discount
```

```
0
```

```
total_price_sum = np.sum(df['Price'])
total_price_sum
```

```
np.float64(103838.9)
```

```
avg_reviews = np.mean(df['Number_of_Reviews'])
avg_reviews
```

```
np.float64(2476.65)
```

```
high_price_count = np.sum(df['Price'] > 300)
high_price_count
```

```
np.int64(172)
```

```
median_rating = np.median(df['Rating'])
median_rating
```

```
np.float64(4.3)
```

```
available_count = np.sum(df['Availability'].values == 'Yes')
available_count
```

```
np.int64(321)
```

```
summary_table = np.array([
    ['Mean Price', np.mean(df['Price'])],
    ['Max Price', np.max(df['Price'])],
    ['Min Price', np.min(df['Price'])],
    ['Mean Rating', np.mean(df['Rating'])],
    ['Max Rating', np.max(df['Rating'])],
    ['Min Rating', np.min(df['Rating'])]
])
summary_table
```

```
array([['Mean Price', '259.59725'],
       ['Max Price', '499.68'],
       ['Min Price', '10.12'],
       ['Mean Rating', '4.2357499999999995'],
       ['Max Rating', '5.0'],
       ['Min Rating', '3.5']], dtype='<U32')
```

```
price_rating_reviews_table = np.column_stack((
    df['Price'].values[:10],
    df['Rating'].values[:10],
    df['Number_of_Reviews'].values[:10]
))
price_rating_reviews_table
```

```
array([[1.3921e+02, 4.6000e+00, 4.8870e+03],
       [1.4058e+02, 4.7000e+00, 2.7580e+03],
       [3.6877e+02, 4.3000e+00, 9.5000e+02],
       [1.3286e+02, 4.3000e+00, 3.1340e+03],
       [3.2022e+02, 4.3000e+00, 2.0590e+03],
       [2.5219e+02, 4.4000e+00, 2.9100e+02],
       [2.9091e+02, 3.5000e+00, 2.4350e+03],
       [4.2067e+02, 4.1000e+00, 4.3180e+03],
       [2.0815e+02, 4.4000e+00, 7.2100e+02],
       [3.8775e+02, 4.4000e+00, 2.1300e+02]])
```

```
min_price_per_category = df.groupby('Category')['Price'].min()
min_price_per_category
```

	Price
Category	
Accessories	11.15
Electronics	12.52
Home	10.60
Office	10.12
Wearables	12.57

```
df['Date_Added'] = pd.to_datetime(df['Date_Added'])
products_after_june = df[df['Date_Added'] > '2023-06-01']
products_after_june
```

	ProductID	Title	Category	Subcategory	Price	Rating	Number_of_Reviews	Availability	Date_Added	Discount_Percentag
152	P0153	Noise Cancelling Headphones	Wearables	Cameras	414.84	4.3	4915	No	2023-06-02	2
153	P0154	Wireless Mouse	Office	Computer Accessories	402.38	4.9	4358	Yes	2023-06-03	3
154	P0155	Smart TV	Office	Watches	318.59	4.8	4608	Yes	2023-06-04	1
155	P0156	Smartwatch	Accessories	Health Devices	116.00	4.9	332	No	2023-06-05	2
156	P0157	Fitness Tracker	Home	Computer Accessories	263.95	4.7	3728	Yes	2023-06-06	3
...	...	...	...	...	...	...	...	...	...	..
395	P0396	Noise Cancelling Headphones	Electronics	Health Devices	95.35	4.1	3098	Yes	2024-01-31	1
396	P0397	Smartwatch	Electronics	Watches	480.78	4.9	2155	Yes	2024-02-01	1

Next steps: Generate code with products_after_june View recommended plots New interactive sheet

```
avg_discount_by_subcategory = df.groupby('Subcategory')['Discount_Percentage'].mean()
avg_discount_by_subcategory
```

	Discount_Percentage
Subcategory	
Audio	23.972222
Cameras	19.812500
Computer Accessories	25.159091
Furniture	23.228571
Health Devices	27.447368
Lighting	24.808511
Monitors	26.350000
Phone Accessories	22.275000
Storage	25.521739
Watches	23.761905

```
median_price_accessories = df[df['Category'] == 'Accessories']['Price'].median()
median_price_accessories
```

```
254.72500000000002
```

```
pivot_table = pd.pivot_table(df, index='Category', values=['Rating', 'Price'], aggfunc='mean')
pivot_table
```

	Price	Rating
Category		
Accessories	240.065500	4.271250
Electronics	307.817647	4.137647
Home	237.386389	4.236111
Office	246.416222	4.256667
Wearables	263.012055	4.284932

Next steps: Generate code with pivot_table View recommended plots New interactive sheet

```
df['Date_Added'] = pd.to_datetime(df['Date_Added'])
products_added_monthly = df['Date_Added'].dt.to_period('M').value_counts().sort_index().to_frame(name='Products_Added')
products_added_monthly
```

Date_Added	Products_Added
2023-01	31
2023-02	28
2023-03	31
2023-04	30
2023-05	31
2023-06	30
2023-07	31
2023-08	31
2023-09	30
2023-10	31
2023-11	30
2023-12	31
2024-01	31
2024-02	4

Next steps: Generate code with products_added_monthly | View recommended plots | New interactive sheet

```
monthly_products = df['Date_Added'].dt.to_period('M').value_counts().sort_index()

for month, count in monthly_products.items():
    print(f"{month}: {'-' * (count // 2)} ({count})")
```

```
2023-01: --------------- (31)
2023-02---------------- (28)
2023-03: --------------- (31)
2023-04: --------------- (30)
2023-05: --------------- (31)
2023-06: --------------- (30)
2023-07: --------------- (31)
2023-08: --------------- (31)
2023-09: --------------- (30)
2023-10: --------------- (31)
2023-11: --------------- (30)
2023-12: --------------- (31)
2024-01: --------------- (31)
2024-02: -- (4)
```

```
avg_discount_subcategory = df.groupby('Subcategory')[['Discount_Percentage']].mean()
avg_discount_subcategory
```

Subcategory	Discount_Percentage
Audio	23.972222
Cameras	19.812500
Computer Accessories	25.159091
Furniture	23.228571
Health Devices	27.447368
Lighting	24.808511
Monitors	26.350000
Phone Accessories	22.275000
Storage	25.521739
Watches	23.761905

Next steps: Generate code with `avg_discount_subcategory` □ View recommended plots New interactive sheet