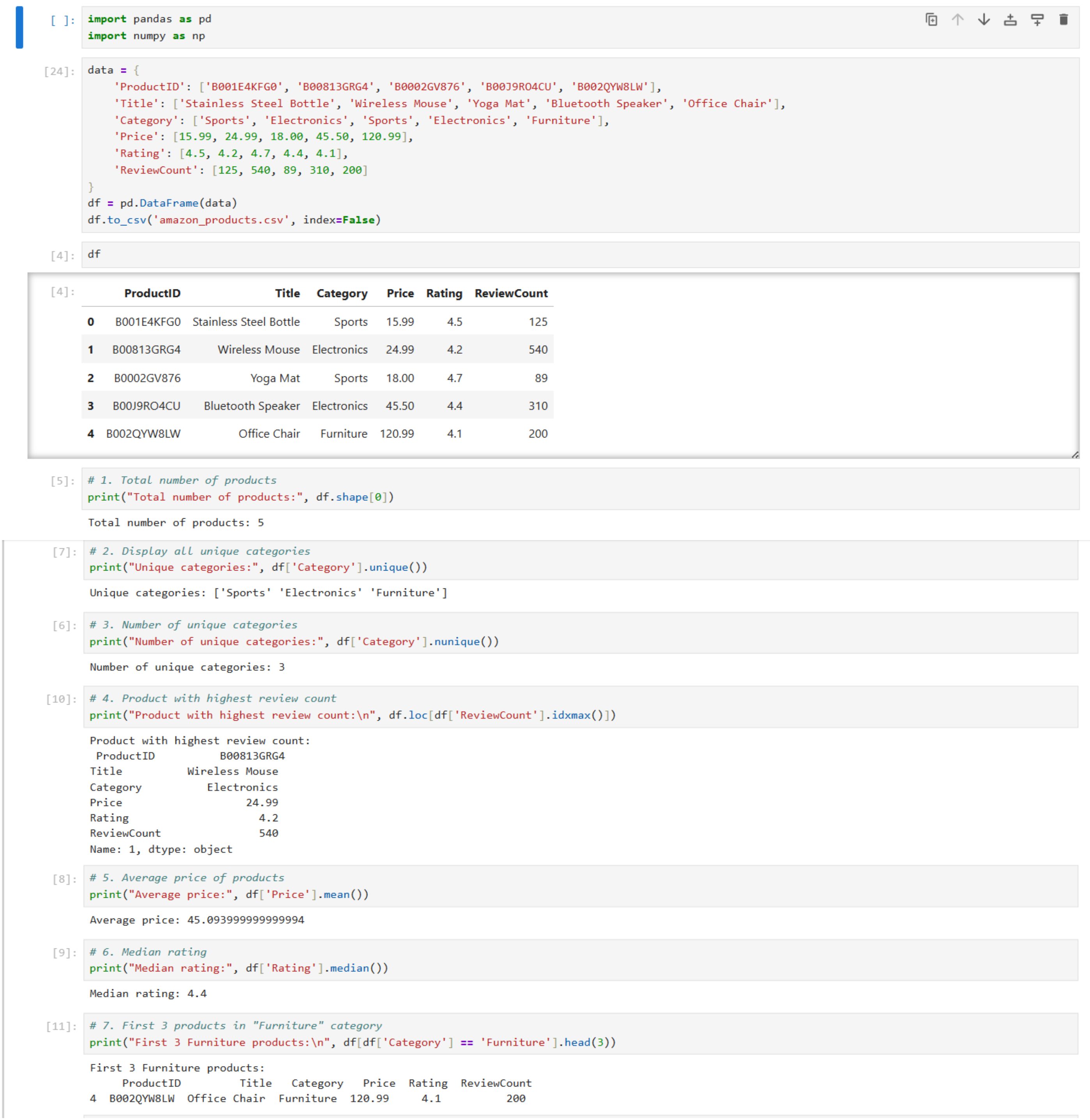
ESSENTIAL OF DATASCIENCE

Name: - Krushna Arun Donge

PRN: - 202401040225

Div: - CS2

Roll No.: - 60



[ ]:
```
import pandas as pd
import numpy as np
```

[24]:
```
data = {
    'ProductID': ['B001E4KFG0', 'B00813GRG4', 'B0002GV876', 'B00J9RO4CU', 'B002QYW8LW'],
    'Title': ['Stainless Steel Bottle', 'Wireless Mouse', 'Yoga Mat', 'Bluetooth Speaker', 'Office Chair'],
    'Category': ['Sports', 'Electronics', 'Sports', 'Electronics', 'Furniture'],
    'Price': [15.99, 24.99, 18.00, 45.50, 120.99],
    'Rating': [4.5, 4.2, 4.7, 4.4, 4.1],
    'ReviewCount': [125, 540, 89, 310, 200]
}
df = pd.DataFrame(data)
df.to_csv('amazon_products.csv', index=False)
```

[4]:
```
df
```

[4]:

	ProductID	Title	Category	Price	Rating	ReviewCount
0	B001E4KFG0	Stainless Steel Bottle	Sports	15.99	4.5	125
1	B00813GRG4	Wireless Mouse	Electronics	24.99	4.2	540
2	B0002GV876	Yoga Mat	Sports	18.00	4.7	89
3	B00J9RO4CU	Bluetooth Speaker	Electronics	45.50	4.4	310
4	B002QYW8LW	Office Chair	Furniture	120.99	4.1	200

[5]:
```
# 1. Total number of products
print("Total number of products:", df.shape[0])
```

```
Total number of products: 5
```

[7]:
```
# 2. Display all unique categories
print("Unique categories:", df['Category'].unique())
```

```
Unique categories: ['Sports' 'Electronics' 'Furniture']
```

[6]:
```
# 3. Number of unique categories
print("Number of unique categories:", df['Category'].nunique())
```

```
Number of unique categories: 3
```

[10]:
```
# 4. Product with highest review count
print("Product with highest review count:\n", df.loc[df['ReviewCount'].idxmax()])
```

```
Product with highest review count:
 ProductID          B00813GRG4
Title           Wireless Mouse
Category           Electronics
Price                    24.99
Rating                     4.2
ReviewCount                540
Name: 1, dtype: object
```

[8]:
```
# 5. Average price of products
print("Average price:", df['Price'].mean())
```

```
Average price: 45.093999999999994
```

[9]:
```
# 6. Median rating
print("Median rating:", df['Rating'].median())
```

```
Median rating: 4.4
```

[11]:
```
# 7. First 3 products in "Furniture" category
print("First 3 Furniture products:\n", df[df['Category'] == 'Furniture'].head(3))
```

```
First 3 Furniture products:
      ProductID         Title   Category   Price  Rating  ReviewCount
4  B002QYW8LW  Office Chair  Furniture  120.99     4.1          200
```

[12]:

```python
# 8. Standard deviation of prices
print("Standard deviation of prices:", df['Price'].std())
```

```
Standard deviation of prices: 44.004733040890045
```

[13]:

```python
# 9. Minimum rating
print("Minimum rating:", df['Rating'].min())
```

```
Minimum rating: 4.1
```

[14]:

```python
# 10. Maximum price
print("Maximum price:", df['Price'].max())
```

```
Maximum price: 120.99
```

[15]:

```python
# 11. Check if any product has price = 0
print("Any product with price = 0:", (df['Price'] == 0).any())
```

```
Any product with price = 0: False
```

[16]:

```python
# 12. Count of products with review > 100
print("Products with review count > 100:", (df['ReviewCount'] > 100).sum())
```

```
Products with review count > 100: 4
```

[18]:

```python
# 13. Sort products by price descending
print("Products sorted by price descending:\n")
df.sort_values(by='Price', ascending=False)
```

```
Products sorted by price descending:
```

[18]:

	ProductID	Title	Category	Price	Rating	ReviewCount
4	B002QYW8LW	Office Chair	Furniture	120.99	4.1	200
3	B00J9RO4CU	Bluetooth Speaker	Electronics	45.50	4.4	310
1	B00813GRG4	Wireless Mouse	Electronics	24.99	4.2	540
2	B0002GV876	Yoga Mat	Sports	18.00	4.7	89
0	B001E4KFG0	Stainless Steel Bottle	Sports	15.99	4.5	125

[23]:

```python
# 14. Add 'Price_After_Tax' (18% tax added)
print("DataFrame with Price After Tax:\n")
df['Price_After_Tax'] = df['Price'] * 1.18
df
```

```
DataFrame with Price After Tax:
```

[23]:

	ProductID	Title	Category	Price	Rating	ReviewCount	Price_After_Tax
0	B001E4KFG0	Stainless Steel Bottle	Sports	15.99	4.5	125	18.8682
1	B00813GRG4	Wireless Mouse	Electronics	24.99	4.2	540	29.4882
2	B0002GV876	Yoga Mat	Sports	18.00	4.7	89	21.2400
3	B00J9RO4CU	Bluetooth Speaker	Electronics	45.50	4.4	310	53.6900
4	B002QYW8LW	Office Chair	Furniture	120.99	4.1	200	142.7682

[20]:

```python
# 15. Count of products per category
print("Product count per category:\n", df['Category'].value_counts())
```

```
Product count per category:
 Category
Sports         2
Electronics    2
Furniture      1
Name: count, dtype: int64
```

[21]:

```python
# 16. Products where title contains "Wireless"
print("Products with 'Wireless' in title:\n", df[df['Title'].str.contains('Wireless')])
```

```
Products with 'Wireless' in title:
     ProductID           Title     Category  Price  Rating  ReviewCount  \
1  B00813GRG4  Wireless Mouse  Electronics  24.99     4.2          540   

   Price_After_Tax  
1          29.4882  
```

[22]:

```
# 17. Replace missing ratings with category-wise mean
# Artificially create missing value for demonstration
df.loc[1, 'Rating'] = np.nan
df['Rating'] = df['Rating'].fillna(df.groupby('Category')['Rating'].transform('mean'))
print("DataFrame after filling missing ratings:\n")
df
```

```
DataFrame after filling missing ratings:
```

[22]:

	ProductID	Title	Category	Price	Rating	ReviewCount	Price_After_Tax
0	B001E4KFG0	Stainless Steel Bottle	Sports	15.99	4.5	125	18.8682
1	B00813GRG4	Wireless Mouse	Electronics	24.99	4.4	540	29.4882
2	B0002GV876	Yoga Mat	Sports	18.00	4.7	89	21.2400
3	B00J9RO4CU	Bluetooth Speaker	Electronics	45.50	4.4	310	53.6900
4	B002QYW8LW	Office Chair	Furniture	120.99	4.1	200	142.7682

[23]:

```
# 18. 25th and 75th percentile of prices
print("25th and 75th percentile of prices:\n", df['Price'].quantile([0.25, 0.75]))
```

```
25th and 75th percentile of prices:
 0.25    18.0
0.75    45.5
Name: Price, dtype: float64
```

[24]:

```
# 19. Pivot table: total reviews by category
pivot = pd.pivot_table(df, values='ReviewCount', index='Category', aggfunc=np.sum)
print("Pivot table (total reviews by category):\n", pivot)
```

```
Pivot table (total reviews by category):
               ReviewCount
Category
Electronics          850
Furniture            200
Sports               214
```

```
C:\Users\donge\AppData\Local\Temp\ipykernel_21856\2109664222.py:2: FutureWarning: The provided callable <function sum at 0x00000274FE367060> is current
ly using DataFrameGroupBy.sum. In a future version of pandas, the provided callable will be used directly. To keep current behavior pass the string "su
m" instead.
  pivot = pd.pivot_table(df, values='ReviewCount', index='Category', aggfunc=np.sum)
```

[25]:

```
# 20. Normalize ReviewCount (Min-Max Normalization)
df['ReviewCount_Normalized'] = (df['ReviewCount'] - df['ReviewCount'].min()) / (df['ReviewCount'].max() - df['ReviewCount'].min())
print("DataFrame with normalized ReviewCount:\n", df[['ReviewCount', 'ReviewCount_Normalized']])
```

```
DataFrame with normalized ReviewCount:
    ReviewCount  ReviewCount_Normalized
0          125                0.079823
1          540                1.000000
2           89                0.000000
3          310                0.490022
4          200                0.246120
```