Assignment 1

Name : Lakshya Agarwal

PRN : 202401120047

Class : CS8

Roll No : CS8-65

Subject : EDS (Essentials of Data-Science)

Topic : Grocery Dataset Analysis with Pandas & NumPy

Full Dataset

Item Name	Category	Price	Quantity	Inventory Value	Price per Unit	Revenue	Price Rank
Item1	Bakery	24	47	1128	0.5106382978723404	1128	27.0
Item2	Beverages	199	35	6965	5.685714285714286	6965	2.0
Item3	Dairy	199	14	2786	14.214285714285714	2786	2.0
Item4	Beverages	184	17	3128	10.823529411764707	3128	5.0
Item5	Beverages	199	36	7164	5.527777777777778	7164	2.0
Item6	Vegetables	60	40	2400	1.5	2400	23.5
Item7	Dairy	117	4	468	29.25	468	13.0
Item8	Dairy	64	2	128	32.0	128	21.0
Item9	Dairy	73	6	438	12.166666666666666	438	19.0
Item10	Beverages	140	42	5880	3.3333333333333335	5880	10.0
Item11	Bakery	60	4	240	15.0	240	23.5
Item12	Dairy	144	29	4176	4.9655172413793105	4176	8.0
Item13	Beverages	30	18	540	1.6666666666666667	540	25.0
Item14	Vegetables	82	26	2132	3.1538461538461537	2132	18.0
Item15	Bakery	176	44	7744	4.0	7744	7.0
Item16	Vegetables	27	34	918	0.7941176470588235	918	26.0
Item17	Bakery	141	10	1410	14.1	1410	9.0
Item18	Beverages	98	36	3528	2.7222222222222223	3528	16.0
Item19	Fruits	69	14	966	4.928571428571429	966	20.0
Item20	Bakery	23	31	713	0.7419354838709677	713	28.0
Item21	Vegetables	18	48	864	0.375	864	29.0
Item22	Beverages	99	15	1485	6.6	1485	15.0
Item23	Bakery	62	8	496	7.75	496	22.0
Item24	Fruits	139	14	1946	9.928571428571429	1946	11.0
Item25	Fruits	93	23	2139	4.043478260869565	2139	17.0

Item26	Dairy	101	40	4040	2.525	4040	14.0
Item27	Dairy	120	21	2520	5.714285714285714	2520	12.0
Item28	Vegetables	197	16	3152	12.3125	3152	4.0
Item29	Bakery	181	45	8145	4.022222222222222	8145	6.0
Item30	Bakery	17	18	306	0.9444444444444444	306	30.0

1. Calculate the total inventory value (Price × Quantity) for each item.

Code:

```
grocery_df['Inventory Value'] = grocery_df['Price'] * grocery_df['Quantity']
grocery_df[['Item Name', 'Inventory Value']]
```

Output:

Item Name	Inventory Value
Item1	1128
Item2	6965
Item3	2786
Item4	3128
Item5	7164

2. What is the average price of items in each category?

Code:

```
grocery_df.groupby('Category')['Price'].mean()
```

Output:

Category	Price
Bakery	85.5
Beverages	135.57142857142858
Dairy	116.85714285714286
Fruits	100.33333333333333

Vegetables	76.8

3. Filter out items where the quantity is above the dataset average.

Code:

```
avg_qty = grocery_df['Quantity'].mean()
grocery_df[grocery_df['Quantity'] > avg_qty]
```

Output:

Item Name	Category	Price	Quantity	Inventory Value
Item1	Bakery	24	47	1128
Item2	Beverages	199	35	6965
Item5	Beverages	199	36	7164
Item6	Vegetables	60	40	2400
Item10	Beverages	140	42	5880

4. Identify the category that has the highest total quantity in stock.

Code:

```
grocery_df.groupby('Category')['Quantity'].sum().idxmax()
```

Output:

index	Category with Highest Quantity
0	Bakery

5. Calculate and add a new column showing price-to-quantity ratio.

Code:

```
grocery_df['Price per Unit'] = grocery_df['Price'] / grocery_df['Quantity']
grocery_df[['Item Name', 'Price per Unit']]
```

Output:

Item Name	Price per Unit

Item1	0.5106382978723404
Item2	5.685714285714286
Item3	14.214285714285714
Item4	10.823529411764707
Item5	5.527777777777778

6. Show the top 3 most expensive items.

Code:

```
grocery_df.nlargest(3, 'Price')[['Item Name', 'Price']]
```

Output:

Item Name	Price
Item2	199
Item3	199
Item5	199

7. Determine the standard deviation of prices across all items.

Code:

```
grocery_df['Price'].std()
```

Output:

index	Std Dev of Price
0.0	61.18695676499134

8. Compute total inventory value for each category.

Code:

```
grocery_df.groupby('Category')['Inventory Value'].sum()
```

Output:

Category	Inventory Value
Bakery	20182
Beverages	28690
Dairy	14556
Fruits	5051
Vegetables	9466

9. Identify item(s) that match the median price.

Code:

```
median_price = grocery_df['Price'].median()
grocery_df[grocery_df['Price'] == median_price][['Item Name', 'Price']]
```

Output:

Item Name	Price

10. Find items that have the same quantity as at least one other item.

Code:

```
grocery_df[grocery_df.duplicated('Quantity', keep=False)][['Item Name', 'Quantity']]
```

Output:

Item Name	Quantity
Item3	14
Item5	36
Item6	40
Item7	4
Item11	4
Item13	18
Item18	36
Item19	14

Item24	14
Item26	40
Item30	18

11. What is the most common quantity in the dataset?

Code:

```
grocery_df['Quantity'].mode()[0]
```

Output:

index	Most Common Quantity
0	14

12. Classify items into price brackets and count how many fall in each.

Code:

```
brackets = pd.cut(grocery_df['Price'], bins=[0, 50, 100, float('inf')], labels=['Low', 'Mid', 'High'])
brackets.value_counts()
```

Output:

index	Price
High	14
Mid	10
Low	6

13. Find the cheapest item in every category.

Code:

```
idx = grocery_df.groupby('Category')['Price'].idxmin()
grocery_df.loc[idx][['Item Name', 'Category', 'Price']]
```

Output:

Item Name	Category	Price
Item30	Bakery	17
Item13	Beverages	30
Item8	Dairy	64
Item19	Fruits	69
Item21	Vegetables	18

14. Which item has the highest potential revenue (Price × Quantity)?

Code:

```
grocery_df.loc[grocery_df['Revenue'].idxmax()][['Item Name', 'Revenue']]
```

Output:

index	28
Item Name	Item29
Revenue	8145

15. Display cumulative inventory value across items.

Code:

```
grocery_df[['Item Name', 'Cumulative Value']].head()
```

Output:

Item Name	Cumulative Value
Item1	1128
Item2	8093
Item3	10879
Item4	14007
Item5	21171

16. Are all item names unique?

Code:

```
grocery_df['Item Name'].is_unique
```

Output:

index	All Unique
0	True

17. How many categories contain more than 1 item?

Code:

```
cat_counts = grocery_df['Category'].value_counts()
(cat_counts > 1).sum()
```

Output:

index	Categories with >1 Item
0	5

18. List items whose prices are multiples of 10.

Code:

```
grocery_df[grocery_df['Price'] % 10 == 0][['Item Name', 'Price']]
```

Output:

Item Name	Price
Item6	60
Item10	140
Item11	60
Item13	30
Item27	120

19. Create a summary showing min, max, and mean quantity per category.

Code:

```
grocery_df.groupby('Category')['Quantity'].agg(['min', 'max', 'mean'])
```

Output:

min	max	mean
4.0	47.0	25.875
15.0	42.0	28.428571428571427
2.0	40.0	16.571428571428573
14.0	23.0	17.0
16.0	48.0	32.8

20. Find the item(s) with the minimum inventory value.

Code:

```
min_val = grocery_df['Inventory Value'].min()
grocery_df[grocery_df['Inventory Value'] == min_val][['Item Name',
'Inventory Value']]
```

Output:

Item Name	Inventory Value
Item8	128