Name - Priyam Mohite
Divison - CM
Batch -CM-1.
Roll no - 23
PRN - 202401030043 .

Topic Name - Grocery

```
# Problem 1
print("\n1. Find the total number of products in the dataset.")
print("Solution:", df.shape[0])
```

```
1. Find the total number of products in the dataset.
Solution: 1757
```

```
[21] # Problem 2
print("\n2. Find the number of unique sub-categories.")
print("Solution:", df['Sub Category'].nunique())
```

```
2. Find the number of unique sub-categories.
Solution: 19
```

```
[22] # Problem 3
print("\n3. Calculate the average product price.")
print("Solution:", df['Price'].mean())
```

```
3. Calculate the average product price.
Solution: nan
```

```
[23] # Problem 4
print("\n4. Find the minimum and maximum product price.")
print("Solution: Min =", df['Price'].min(), ", Max =", df['Price'].max())
```

```
4. Find the minimum and maximum product price.
Solution: Min = nan , Max = nan
```

```
# Problem 5
print("\n5. How many products have no discount?")
print("Solution:", df[df['Discount'] == 'No Discount'].shape[0])
```

```
5. How many products have no discount?
Solution: 1626
```

```
[25] # Problem 6
print("\n6. List products with the highest price.")
max_price = df['Price'].max()
print("Solution:", df[df['Price'] == max_price]['Title'].tolist())
```

```
6. List products with the highest price.
Solution: []
```

```
[26] # Problem 7
print("\n7. Find the top 3 most expensive products.")
print("Solution:\n", df.nlargest(3, 'Price')[['Title', 'Price']])
```

```
7. Find the top 3 most expensive products.
Solution:
                                               Title  Price
0  David’s Cookies Mile High Peanut Butter Cake, ...    NaN
1  The Cake Bake Shop 8" Round Carrot Cake (16-22...    NaN
2  St Michel Madeleine, Classic French Sponge Cak...    NaN
```

```python
# Problem 8
print("\n8. Find average price per sub-category.")
print("Solution:\n", df.groupby('Sub Category')['Price'].mean())
```

```
8. Find average price per sub-category.
Solution:
 Sub Category
Bakery & Desserts              NaN
Beverages & Water              NaN
Breakfast                      NaN
Candy                          NaN
Cleaning Supplies              NaN
Coffee                         NaN
Deli                           NaN
Floral                         NaN
Gift Baskets                   NaN
Household                      NaN
Kirkland Signature Grocery     NaN
Laundry Detergent & Supplies   NaN
Meat & Seafood                 NaN
Organic                        NaN
Pantry & Dry Goods             NaN
Paper & Plastic Products       NaN
Poultry                        NaN
Seafood                        NaN
Snacks                         NaN
Name: Price, dtype: float64
```

```python
[28] # Problem 9
print("\n9. Find the average numeric rating.")
print("Solution:", df['Numeric Rating'].mean())
```

```
9. Find the average numeric rating.
Solution: 4.340356564019449
```

```python
[29] # Problem 10
print("\n10. List products with rating greater than 4.5.")
print("Solution:", df[df['Numeric Rating'] > 4.5]['Title'].tolist())
```

```
10. List products with rating greater than 4.5.
Solution: ['The Cake Bake Shop 8" Round Carrot Cake (16-22 Servings)', "David's Cookies Butter Pecan Meltaways 32 oz, 2-pack", 'La Grande Galette French Butter Cookies, 1.3 lb, 6-pack'
```

```python
# Problem 11
print("\n11. List products that have more than 500 reviews.")
print("Solution:", df[df['Number of Reviews'] > 500]['Title'].tolist())
```

```
11. List products that have more than 500 reviews.
Solution: ["David's Cookies Butter Pecan Meltaways 32 oz, 2-pack", 'David’s Cookies Premier Chocolate Cake, 7.2 lbs (Serves 14)', 'La Grande Galette French Butter Cookies, 1.3 lb, 6-pa
```

```python
[31] # Problem 12
print("\n12. Find the sub-category with the highest average rating.")
print("Solution:", df.groupby('Sub Category')['Numeric Rating'].mean().idxmax())
```

```
12. Find the sub-category with the highest average rating.
Solution: Breakfast
```

```python
[32] # Problem 13
print("\n13. Find the most common word in product titles.")
words = ' '.join(df['Title'].dropna()).lower().split()
word_counts = Counter(words)
print("Solution:", word_counts.most_common(1)[0][0])
```

```
13. Find the most common word in product titles.
Solution: oz,
```

```python
# Problem 14
print("\n14. List products whose title contains the word 'Chocolate'.")
print("Solution:", df[df['Title'].str.contains('Chocolate', case=False, na=False)]['Title'].tolist())
```

```
14. List products whose title contains the word 'Chocolate'.
Solution: ['David’s Cookies Premier Chocolate Cake, 7.2 lbs (Serves 14)', 'The Cake Bake Shop 8" Round Chocolate Cake (16-22 Servings)', "David's Cookies 90-piece Gourmet Chocolate Chu
```

```python
# Problem 14
print("\n14. List products whose title contains the word 'Chocolate'.")
print("Solution:", df[df['Title'].str.contains('Chocolate', case=False, na=False)]['Title'].tolist())
```

```
14. List products whose title contains the word 'Chocolate'.
Solution: ['David's Cookies Premier Chocolate Cake, 7.2 lbs (Serves 14)', 'The Cake Bake Shop 8" Round Chocolate Cake (16-22 Servings)', "David's Cookies 90-piece Gourmet Chocolate Chu
```

```python
# Problem 15
print("\n15. List products mentioning 'Kosher' in features.")
print("Solution:", df[df['Feature'].str.contains('Kosher', case=False, na=False)]['Title'].tolist())
```

```
15. List products mentioning 'Kosher' in features.
Solution: ['David's Cookies Mile High Peanut Butter Cake, 6.8 lbs (14 Servings)', "David's Cookies Butter Pecan Meltaways 32 oz, 2-pack", 'David's Cookies Premier Chocolate Cake, 7.2 ]
```

```python
# Problem 16
print("\n16. Find the correlation between product price and rating.")
print("Solution:\n", df[['Price', 'Numeric Rating']].corr())
```

```
16. Find the correlation between product price and rating.
Solution:
                Price  Numeric Rating
Price             NaN             NaN
Numeric Rating    NaN             1.0
```

```python
# Problem 17
print("\n17. List products priced above the average price.")
avg_price = df['Price'].mean()
print("Solution:", df[df['Price'] > avg_price]['Title'].tolist())
```

```
17. List products priced above the average price.
Solution: []
```

```python
# Problem 18
print("\n18. Find the number of products with any discount (other than 'No Discount').")
print("Solution:", df[df['Discount'] != 'No Discount'].shape[0])
```

```
18. Find the number of products with any discount (other than 'No Discount').
Solution: 131
```

```python
# Problem 19
print("\n19. List products mentioning 'Butter' in the product description.")
print("Solution:", df[df['Product Description'].str.contains('Butter', case=False, na=False)]['Title'].tolist())
```

```
19. List products mentioning 'Butter' in the product description.
Solution: ['David's Cookies Mile High Peanut Butter Cake, 6.8 lbs (14 Servings)', 'St Michel Madeleine, Classic French Sponge Cake 100 - count', "David's Cookies Butter Pecan Meltaways
```

```python
# Problem 20
print("\n20. Create a summary report including total products, average price, average rating, and percentage with no discount.")
summary = {
    'Total Products': df.shape[0],
    'Average Price': round(df['Price'].mean(), 2),
    'Average Rating': round(df['Numeric Rating'].mean(), 2),
    '% Products with No Discount': round((df[df['Discount'] == 'No Discount'].shape[0] / df.shape[0]) * 100, 2)
}
```

```
20. Create a summary report including total products, average price, average rating, and percentage with no discount.
```