EDS Activity 1

Name : Diya Rao
Roll No : CS2-66
PRN : 202401040343

20 Problems statements : Grocery dataset

1) **How many unique members are there in the dataset?**

```
[2] print("1. Unique members:", df['Member_number'].nunique())

1. Unique members: 3898
```

2) **How many unique items are sold?**

```
[3] print("2. Unique items:", df['itemDescription'].nunique())

2. Unique items: 167
```

3) **What are the top 10 most frequently bought items?**

```
[4] print("3. Top 10 most frequently bought items:")
    print(df['itemDescription'].value_counts().head(10))

3. Top 10 most frequently bought items:
itemDescription
whole milk          2502
other vegetables    1898
rolls/buns          1716
soda                1514
yogurt              1334
root vegetables     1071
tropical fruit      1032
bottled water        933
sausage              924
citrus fruit         812
Name: count, dtype: int64
```

4) **How many total transactions occurred each day?**

```
[5] print("4. Total transactions per day:")
    print(df.groupby('Date').size())

4. Total transactions per day:
Date
01-01-2014    48
01-01-2015    48
01-02-2014    62
01-02-2015    61
01-03-2014    54
              ..
31-07-2015    63
31-08-2014    47
31-08-2015    47
31-10-2014    57
31-10-2015    46
Length: 728, dtype: int64
```

5) **Which member made the most purchases?**

```
[6] print("5. Member with most purchases:", df['Member_number'].value_counts().idxmax())

5. Member with most purchases: 3180
```

6) **Which day had the highest number of transactions?**

```
[7] print("6. Day with most transactions:", df.groupby('Date').size().idxmax())

6. Day with most transactions: 21-01-2015
```

7) **Which item was bought the most on a single day?**

```
[8] print("7. Most bought item on a single day:", df.groupby(['Date', 'itemDescription']).size().idxmax())

7. Most bought item on a single day: ('01-04-2015', 'whole milk')
```

8) **How many purchases were made each month?**

yaml Copy Edit

```
2014-01    1527
2014-02    1437
2014-03    1411
2014-04    1561
2014-05    1615
```

9) **Which items were bought by the most number of unique members?**

```
[10] print("9. Items bought by most unique members:")
     print(df.groupby('itemDescription')['Member_number'].nunique().sort_values(ascending=False).head(10))

9. Items bought by most unique members:
itemDescription
whole milk          1786
other vegetables    1468
rolls/buns          1363
soda                1222
yogurt              1103
tropical fruit       911
root vegetables      899
bottled water        833
sausage              803
citrus fruit         723
Name: Member_number, dtype: int64
```

10) **Which members bought the largest variety of items?**

```
[11] print("10. Members who bought the largest variety of items:")
     print(df.groupby('Member_number')['itemDescription'].nunique().sort_values(ascending=False).head(10))

10. Members who bought the largest variety of items:
Member_number
2051    26
1379    26
4875    25
3050    25
3308    24
3100    24
1410    24
2433    24
3737    24
1052    23
Name: itemDescription, dtype: int64
```

11) **Find the average number of items bought per transaction.**

```
[12] print("11. Average number of items bought per transaction:",
           df.groupby(['Member_number', 'Date']).size().mean())

11. Average number of items bought per transaction: 2.590723785337165
```

12) **What are the top 5 item pairs most frequently bought together?**

```
[13] from itertools import combinations
     from collections import Counter

     print("12. Top 5 most frequently bought item pairs:")
     baskets = df.groupby(['Member_number', 'Date'])['itemDescription'].apply(set)
     pair_counter = Counter()
     for basket in baskets:
         for pair in combinations(sorted(basket), 2):
             pair_counter[pair] += 1
     print(pair_counter.most_common(5))

12. Top 5 most frequently bought item pairs:
[(('other vegetables', 'whole milk'), 222), (('rolls/buns', 'whole milk'), 209), (('soda', 'whole milk'), 174), (('whole milk', 'yogurt'), 167), (('other vegetables', 'rolls/buns'), 158)]
```

13) **What is the distribution of the number of items per transaction?**

```
[14] # 13. Distribution of items per transaction
     print("13. Distribution of items per transaction:")
     print(df.groupby(['Member_number', 'Date']).size().describe())

13. Distribution of items per transaction:
count    14963.000000
mean         2.590724
std          1.117469
min          2.000000
25%          2.000000
50%          2.000000
75%          3.000000
max         11.000000
dtype: float64
```

14) **How many single-item transactions are there?**

```
[15] print("14. Number of single-item transactions:",
           (df.groupby(['Member_number', 'Date']).size() == 1).sum())

14. Number of single-item transactions: 0
```

15) **Which item is most frequently part of large transactions (>=5 items)?**

```
[16] # 15. Most common item in large transactions (>=5 items)
     print("15. Most common item in large transactions:")
     large_tx = df.groupby(['Member_number', 'Date']).filter(lambda x: len(x) >= 5)
     print(large_tx['itemDescription'].value_counts().head(5))

15. Most common item in large transactions:
itemDescription
whole milk          330
other vegetables    226
rolls/buns          207
soda                169
yogurt              165
Name: count, dtype: int64
```

16) **What is the trend of total purchases over the year (monthly)?**

```
yaml                                                    Copy    Edit

2014-01    1527
2014-02    1437
2014-03    1411
2014-04    1561
2014-05    1615
```

17) **What is the average number of purchases per member?**

```
[18] # 17. Average number of purchases per member
     print("17. Average number of purchases per member:",
           df.groupby('Member_number').size().mean())

17. Average number of purchases per member: 9.944843509492047
```

18) **Which items were only bought once in the entire dataset?**

```
[19] # 18. Items bought only once
     print("18. Items bought only once:")
     print(df['itemDescription'].value_counts()[lambda x: x == 1])

18. Items bought only once:
itemDescription
kitchen utensil         1
preservation products    1
Name: count, dtype: int64
```

19) **What are the first and last purchase dates for each member?**

```
[20] # 19. First and last purchase date per member
     print("19. First and last purchase date per member:")
     print(df.groupby('Member_number')['Date'].agg(['min', 'max']))

19. First and last purchase date per member:
                    min         max
Member_number
1000            15-03-2015  27-05-2015
1001            02-05-2015  20-01-2015
1002            09-02-2014  30-08-2015
1003            10-02-2015  27-02-2014
1004            01-05-2014  19-08-2014
...                    ...         ...
4996            20-02-2015  24-11-2015
4997            05-01-2014  27-12-2015
4998            14-10-2015  14-10-2015
4999            09-04-2014  26-12-2015
5000            09-03-2014  16-11-2014

[3898 rows x 2 columns]
```

20) **Which item has the highest customer retention (repeated purchases by same members)?**

```
# 20. Item with the highest customer retention (repeat purchases)
print("20. Items with the highest customer retention:")
repeat_buyers = df.groupby(['itemDescription', 'Member_number']).size()
repeat_counts = repeat_buyers[repeat_buyers > 1].groupby(level=0).count()
print(repeat_counts.sort_values(ascending=False).head(5))

20. Items with the highest customer retention:
itemDescription
whole milk          534
other vegetables    357
rolls/buns          293
soda                252
yogurt              196
dtype: int64
```