Rupesh Santosh Panda

202401040008

CS1-23

Dataset link:https://www.kaggle.com/datasets/heeraldedhia/groceries-dataset

Importing the dataset

```
import pandas as pd
import numpy as np
df=pd.read_csv('Groceries_dataset.csv')
df['Date'] = pd.to_datetime(df['Date'], format='%d-%m-%Y')#convert datecolumn to datetime
```

1.Number of unique customers

```
us = df['Member_number'].nunique()
print(f"1. Total unique customers: {us}")
```

Output:

```
PS E:\MIT fy\EDS> python -u "e:\MIT fy\EDS\panda.py"
1. Total unique customers: 3898
PS E:\MIT fy\EDS>
```

2.Number of unique items

```
ui = df['itemDescription'].nunique()
print(f"2. Total unique items: {ui}")
```

Output:

```
PS E:\MIT fy\EDS> python -u "e:\MIT fy\EDS\panda.py"
2. Total unique items: 167
PS E:\MIT fy\EDS>
```

3.Most frequently purchased item

```
mf = df['itemDescription'].value_counts().idxmax()
print(f"3. Most frequent item: {mf}")
```

Output:

```
PS E:\MIT fy\EDS> python -u "e:\MIT fy\EDS\panda.py"
3. Most frequent item: whole milk
PS E:\MIT fy\EDS>
```

4.Least frequently purchased item

```
lf = df['itemDescription'].value_counts().idxmin()
print(f"4. Least frequent item: {lf}")
```

Output:

```
PS E:\MIT fy\EDS> python -u "e:\MIT fy\EDS\panda.py"
4. Least frequent item: kitchen utensil
PS E:\MIT fy\EDS>
```

5.Most active customer id

```
mc = df['Member_number'].value_counts().idxmax()
print(f"5. Most active customer ID: {mc}")
```

Output:

```
PS E:\MIT fy\EDS> python -u "e:\MIT fy\EDS\panda.py"
5. Most active customer ID: 3180
PS E:\MIT fy\EDS>
```

6.Busiest day

```
bd = df['Date'].value_counts().idxmax()
print(f"6. Busiest day: {bd.date()}")
```

Output:

```
6. Busiest day: 2015-01-21
```

7.Avg items/customer

```
ar = df.groupby('Member_number')['itemDescription'].count().mean()
print(f"7. Avg items/customer: {ar:.2f}")
```

Output:

```
7. Avg items/customer: 9.94
```

8.Top 5 items

```
t5 = df['itemDescription'].value_counts().head(5)
print("8. Top 5 items:\n", t5)
```

Output:

```
8. Top 5 items:
 itemDescription
whole milk           2502
other vegetables     1898
rolls/buns           1716
soda                 1514
yogurt               1334
Name: count, dtype: int64
```

9.Items by most active customer

```
ia = df[df['Member_number'] == mc]['itemDescription'].unique()
print(f"9. Items by most active customer: {list(ia)}")
```

Output:

```
9. Items by most active customer: ['whole milk', 'tropical fruit', 'ham', 'root vegetables', 'bottled beer', 'pastry', 'sausage', 'other vegetables', 'citrus fruit', 'yogurt', 'rolls/buns', 'sugar', 'zwieback', 'condensed milk', 'domestic eggs', 'margarine', 'curd', 'baking powder', 'flour', 'bathroom cleaner', 'onions', 'waffles', 'canned beer', 'newspapers']
```

10.Transaction per month

```
tp = df['Date'].dt.to_period('M').value_counts().sort_index()
print("10. Transactions per month:\n", tp)
```

```
10. Transactions per month:
 Date
2014-01    1527
2014-02    1437
2014-03    1411
2014-04    1561
2014-05    1615
2014-06    1570
2014-07    1576
2014-08    1575
2014-09    1472
2014-10    1591
2014-11    1469
2014-12    1473
2015-01    1797
2015-02    1560
2015-03    1722
2015-04    1699
2015-05    1793
2015-06    1694
2015-07    1724
2015-08    1921
2015-09    1587
2015-10    1670
2015-11    1785
2015-12    1536
Freq: M, Name: count, dtype: int64
```

Output:

11. Unique items in Jan 2015

```
Jan15 = df[df['Date'].dt.to_period('M') == '2015-
01']['itemDescription'].nunique()
print(f"11. Unique items in Jan 2015: {Jan15}")
```

Output:

```
11. Unique items in Jan 2015: 137
```

12. Avg daily transactions

```
at = df.groupby('Date').size().mean()
print(f"12. Avg daily transactions: {at:.2f}")
```

Output:

```
12. Avg daily transactions: 53.25
```

13. Customer with most diverse purchases

```
dc = df.groupby('Member_number')['itemDescription'].nunique().idxmax()
print(f"13. Customer with most diverse purchases: {dc}")
```

Output:

```
13. Customer with most diverse purchases: 1379
```

14. Weekend transactions

```
wt = df[df['Date'].dt.dayofweek.isin([5,6])].shape[0]
print(f"14. Weekend transactions: {wt}")
```

Output:

```
14. Weekend transactions: 11081
```

15. Items purchased once

```
temp_counts = df['itemDescription'].value_counts()
i1 = temp_counts[temp_counts == 1].index.tolist()
print(f"15. Items purchased once: {i1}")
```

Output:

```
15. Items purchased once: ['kitchen utensil', 'preservation products']
```

16. Median transactions/customer

```
mt = df.groupby('Member_number').size().median()
print(f"16. Median transactions/customer: {mt}")
```

Output:

```
16. Median transactions/customer: 9.0
```

17. Most popular item each month

```
pem = df.groupby(df['Date'].dt.to_period('M'))['itemDescription'] \
                        .agg(lambda x: x.value_counts().idxmax())
print("17. Most popular item each month:\n", pem)
```

Output:

```
17. Most popular item each month:
 Date
2014-01          rolls/buns
2014-02    other vegetables
2014-03          whole milk
2014-04          whole milk
2014-05          rolls/buns
2014-06          whole milk
2014-07          whole milk
2014-08    other vegetables
2014-09          whole milk
2014-10          rolls/buns
2014-11          whole milk
2014-12          whole milk
2015-01          whole milk
2015-02          whole milk
2015-03          whole milk
2015-04          whole milk
2015-05          whole milk
2015-06          whole milk
2015-07          whole milk
2015-08          whole milk
2015-09          whole milk
2015-10          whole milk
2015-11          whole milk
2015-12          whole milk
Freq: M, Name: itemDescription, dtype: object
```

18. Avg daily items per customer

```
dc = df.groupby(['Member_number','Date']).size()
adpc = dc.groupby('Member_number').mean()
print("18. Avg daily items per customer:\n", adpc)
```

Output:

```
18. Avg daily items per customer:
 Member_number
1000    2.600000
1001    2.400000
1002    2.000000
1003    2.000000
1004    2.625000
           ...
4996    3.333333
4997    3.000000
4998    2.000000
4999    2.666667
5000    2.333333
Length: 3898, dtype: float64
```

19. Date w/ most variety

```
dv = df.groupby('Date')['itemDescription'].nunique().idxmax()
print(f"19. Date w/ most variety: {dv.date()}")
```

Output:

```
19. Date w/ most variety: 2015-08-08
```

20. Unique items per month

```
um = df.groupby(df['Date'].dt.to_period('M'))['itemDescription'].nunique()
print("20. Unique items per month:\n", um)
```

Output:

```
20. Unique items per month:
 Date
2014-01    139
2014-02    131
2014-03    138
2014-04    149
2014-05    139
2014-06    140
2014-07    142
2014-08    145
2014-09    137
2014-10    144
2014-11    135
2014-12    137
2015-01    137
2015-02    132
2015-03    138
2015-04    136
2015-05    135
2015-06    135
2015-07    139
2015-08    141
2015-09    130
2015-10    135
2015-11    133
2015-12    134
Freq: M, Name: itemDescription, dtype: int64
```