In [3]:
```
import pandas as pd
import numpy as np

# Load the dataset
df = pd.read_csv('Groceries_dataset.csv.zip')
```

In [4]:
```
# Display first few rows
df.head()
```

Out[4]:

	Member_number	Date	itemDescription
0	1808	21-07-2015	tropical fruit
1	2552	05-01-2015	whole milk
2	2300	19-09-2015	pip fruit
3	1187	12-12-2015	other vegetables
4	3037	01-02-2015	whole milk

In [5]:
```
# 1. Find the total number of transactions
num_transactions = df['Member_number'].nunique()
print(f"Total number of transactions: {num_transactions}")
```

```
Total number of transactions: 3898
```

In [6]:
```
# 2. Find the total number of unique items sold
unique_items = df['itemDescription'].nunique()
print(f"Total unique items sold: {unique_items}")
```

```
Total unique items sold: 167
```

In [7]:
```
# 3. Find the most popular item sold
most_popular_item = df['itemDescription'].value_counts().idxmax()
print(f"Most popular item: {most_popular_item}")
```

```
Most popular item: whole milk
```

In [8]:
```
# 4. Find the least popular item sold
least_popular_item = df['itemDescription'].value_counts().idxmin()
print(f"Least popular item: {least_popular_item}")
```

```
Least popular item: kitchen utensil
```

In [9]:
```
# 5. Find the number of sales for 'whole milk'
whole_milk_sales = df[df['itemDescription'] == 'whole milk'].shape[0]
print(f"Number of 'whole milk' sales: {whole_milk_sales}")
```

```
Number of 'whole milk' sales: 2502
```

In [10]:
```
# 6. Find the number of different items sold per transaction
different_items_per_transaction = df.groupby('Member_number')['itemDescription']
print(different_items_per_transaction.head())
```

```
Member_number
1000    11
1001     9
1002     8
1003     6
1004    16
Name: itemDescription, dtype: int64
```

In [11]:
```
# 7. Find the average number of items per transaction
average_items_per_transaction = df.groupby('Member_number').size().mean()
print(f"Average number of items per transaction: {average_items_per_transaction:
```

```
Average number of items per transaction: 9.94
```

In [12]:
```
# 8. Find the top 5 most frequently bought items
top5_items = df['itemDescription'].value_counts().head(5)
print("Top 5 most frequently bought items:")
print(top5_items)
```

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

In [13]:
```
# 9. Find the total number of transactions per month
df['Date'] = pd.to_datetime(df['Date'], dayfirst=True)
df['Month'] = df['Date'].dt.month
transactions_per_month = df.groupby('Month')['Member_number'].nunique()
print("Transactions per month:")
print(transactions_per_month)
```

```
Transactions per month:
Month
1     1106
2     1023
3     1030
4     1059
5     1146
6     1043
7     1085
8     1130
9     1024
10    1085
11    1082
12    1039
Name: Member_number, dtype: int64
```

In [14]:
```
# 10. Find the item bought by most members
item_by_most_members = df.groupby('itemDescription')['Member_number'].nunique().
print(f"Item bought by most members: {item_by_most_members}")
```

```
Item bought by most members: whole milk
```

In [15]:
```
# 11. Find members who bought 'yogurt'
yogurt_buyers = df[df['itemDescription'] == 'yogurt']['Member_number'].unique()
print(f"Members who bought yogurt: {yogurt_buyers[:5]} (showing first 5)")
```

```
Members who bought yogurt: [4056 4918 1723 2600 4040] (showing first 5)
```

In [16]:
```
# 12. Find days with the highest number of transactions
transactions_per_day = df.groupby('Date')['Member_number'].nunique()
highest_transaction_day = transactions_per_day.idxmax()
print(f"Day with highest transactions: {highest_transaction_day.date()}")
```

```
Day with highest transactions: 2014-08-28
```

In [17]:
```
# 13. Find the most popular item in December
december_items = df[df['Month'] == 12]['itemDescription'].value_counts().idxmax(
print(f"Most popular item in December: {december_items}")
```

```
Most popular item in December: whole milk
```

In [18]:
```
# 14. Find the least popular item in January
january_items = df[df['Month'] == 1]['itemDescription'].value_counts().idxmin()
print(f"Least popular item in January: {january_items}")
```

```
Least popular item in January: whisky
```

In [19]:
```
# 15. Find the total number of yogurt sold
total_yogurt_sold = df[df['itemDescription'] == 'yogurt'].shape[0]
print(f"Total yogurt sold: {total_yogurt_sold}")
```

```
Total yogurt sold: 1334
```

In [20]:
```
# 16. Find top 3 items for each month
top3_items_monthly = df.groupby(['Month', 'itemDescription']).size().groupby(lev
print("Top 3 items each month:")
print(top3_items_monthly)
```

```
Top 3 items each month:
Month  itemDescription
1      whole milk          199
       rolls/buns          162
       other vegetables    154
2      whole milk          182
       other vegetables    150
       rolls/buns          128
3      whole milk          207
       other vegetables    132
       rolls/buns          121
4      whole milk          234
       other vegetables    150
       rolls/buns          147
5      whole milk          209
       other vegetables    169
       rolls/buns          162
6      whole milk          200
       other vegetables    164
       soda                143
7      whole milk          210
       other vegetables    148
       yogurt              131
8      whole milk          236
       other vegetables    195
       rolls/buns          140
9      whole milk          213
       other vegetables    142
       rolls/buns          140
10     whole milk          195
       rolls/buns          174
       other vegetables    173
11     whole milk          228
       other vegetables    161
       rolls/buns          151
12     whole milk          189
       other vegetables    160
       rolls/buns          121
dtype: int64
```

In [21]:
```
# 17. Find members who bought more than 20 items
top_buyers = df.groupby('Member_number').size()
members_more_than_20 = top_buyers[top_buyers > 20].index.tolist()
print(f"Members who bought more than 20 items: {members_more_than_20[:5]} (showi
```

```
Members who bought more than 20 items: [1004, 1052, 1087, 1098, 1116] (showing fi
rst 5)
```

In [22]:
```
# 18. Find days with sales above average
total_sales_per_day = df.groupby('Date').size()
average_daily_sales = total_sales_per_day.mean()
days_above_average = total_sales_per_day[total_sales_per_day > average_daily_sal
print(f"Days with sales above average: {len(days_above_average)} days")
```

```
Days with sales above average: 343 days
```

In [23]:
```
# 19. Calculate the proportion of transactions involving 'root vegetables'
root_vegetables_transactions = df[df['itemDescription'] == 'root vegetables']['M
proportion_root_vegetables = root_vegetables_transactions / num_transactions
print(f"Proportion of transactions with root vegetables: {proportion_root_vegeta
```

```
Proportion of transactions with root vegetables: 23.06%
```

In [24]:
```
# 20. Find correlation between month and number of items sold
monthly_items_sold = df.groupby('Month').size()
correlation = np.corrcoef(df['Month'], monthly_items_sold.loc[df['Month']])[0, 1
print(f"Correlation between month and items sold: {correlation:.2f}")
```

```
Correlation between month and items sold: -0.08
```

In [ ]:
```

```