EDS THEORY ASSIGNMENT

NAME: Omkar Vairagkar

PRN NO.: 202401040045

ROLL NO: CS3-24

```
#mount the google datset
from google.colab import drive
drive.mount('/content/drive')

# load the dataset
!pip install numpy pandas
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

path='/content/drive/MyDrive/Groceries_dataset.csv'
df=pd.read_csv(path)

df.info()
```

```
Mounted at /content/drive
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (2.0.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38765 entries, 0 to 38764
Data columns (total 3 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Member_number    38765 non-null  int64
 1   Date             38765 non-null  object
 2   itemDescription  38765 non-null  object
dtypes: int64(1), object(2)
memory usage: 908.7+ KB
```

```
import pandas as pd
import numpy as np

# Convert 'Date' to datetime
df['Date'] = pd.to_datetime(df['Date'], dayfirst=True)
```

Problem statements

1. Find total number of unique customers.

```
import pandas as pd
import numpy as np

# Unique customers
unique_customers = df['Member_number'].nunique()
print("Total unique customers:", unique_customers)
```

```
Total unique customers: 3898
```

2.Find the total number of unique items sold.

```
import pandas as pd
import numpy as np

unique_items = df['itemDescription'].nunique()
print("Total unique items:", unique_items)
```

```
Total unique items: 167
```

3.Find the most commonly purchased item.

```
import pandas as pd
import numpy as np
```

```
most_common_item = df['itemDescription'].mode()[0]
print("Most commonly purchased item:", most_common_item)
```

```
Most commonly purchased item: whole milk
```

4.Find the least commonly purchased item.

```
import pandas as pd
import numpy as np

least_common_items = df['itemDescription'].value_counts().idxmin()
print("Least commonly purchased item:", least_common_items)
```

```
Least commonly purchased item: kitchen utensil
```

5.Find the total number of transactions made each day.

```
import pandas as pd
import numpy as np

transactions_per_day = df.groupby('Date').size()
print(transactions_per_day.head())
```

```
Date
01-01-2014    48
01-01-2015    48
01-02-2014    62
01-02-2015    61
01-03-2014    54
dtype: int64
```

6.Find the top 5 items bought most often.

```
import pandas as pd
import numpy as np

top_5_items = df['itemDescription'].value_counts().head(5)
print(top_5_items)
```

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

7.Find the number of transactions done by the most active customer.

```
import pandas as pd
import numpy as np

most_active_customer = df['Member_number'].value_counts().idxmax()
transactions_most_active = df['Member_number'].value_counts().max()
print(f"Most active customer (ID {most_active_customer}) made {transactions_most_active} transactions.")
```

```
Most active customer (ID 3180) made 36 transactions.
```

8.Find the top 10 customers who made the most purchases.

```
import pandas as pd
import numpy as np

top_10_customers = df['Member_number'].value_counts().head(10)
print(top_10_customers)
```

```
Member_number
3180    36
3737    33
3050    33
2051    33
3915    31
2433    31
2271    31
2625    31
3872    30
```

```
4875    29
Name: count, dtype: int64
```

9.Find how many times "whole milk" was bought.

```
import pandas as pd
import numpy as np

whole_milk_count = df[df['itemDescription'] == 'whole milk'].shape[0]
print("Whole milk was bought", whole_milk_count, "times.")
```

```
Whole milk was bought 2502 times.
```

10.Find how many unique customers bought "whole milk".

```
import pandas as pd
import numpy as np

customers_whole_milk = df[df['itemDescription'] == 'whole milk']['Member_number'].nunique()
print("Unique customers who bought whole milk:", customers_whole_milk)
```

```
Unique customers who bought whole milk: 1786
```

11.Find the total number of transactions in 2015.

```
import pandas as pd
import numpy as np

transactions_2015 = df[df['Date'].dt.year == 2015].shape[0]
print("Transactions in 2015:", transactions_2015)
```

```
Transactions in 2015: 20488
```

12.Find how many items were sold in July 2015.

```
import pandas as pd
import numpy as np

july_sales = df[(df['Date'].dt.month == 7) & (df['Date'].dt.year == 2015)].shape[0]
print("Items sold in July 2015:", july_sales)
```

```
Items sold in July 2015: 1724
```

13.Find the customer who bought the most different types of items.

```
import pandas as pd
import numpy as np

most_variety_customer = df.groupby('Member_number')['itemDescription'].nunique().idxmax()
print("Customer who bought the most different types of items:", most_variety_customer)
```

```
Customer who bought the most different types of items: 1379
```

14.Find the number of unique items bought per customer.

```
import pandas as pd
import numpy as np

items_per_customer = df.groupby('Member_number')['itemDescription'].nunique()
print(items_per_customer.head())
```

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

15.Find the month with the highest number of sales.

```
import pandas as pd
import numpy as np
```

```
df['Month'] = df['Date'].dt.month
month_sales = df['Month'].value_counts().idxmax()
print("Month with highest sales:", month_sales)
```

```
Month with highest sales: 8
```

16.List all items bought by Member_number = 1808.

```
import pandas as pd
import numpy as np

items_1808 = df[df['Member_number'] == 1808]['itemDescription'].unique()
print("Items bought by Member 1808:", items_1808)
```

```
Items bought by Member 1808: ['tropical fruit' 'long life bakery product' 'meat' 'sugar' 'rolls/buns'
 'semi-finished bread' 'whole milk' 'citrus fruit' 'candy' 'napkins']
```

17.Check for missing values.

```
import pandas as pd
import numpy as np

missing_values = df.isnull().sum()
print("Missing values in each column:\n", missing_values)
```

```
Missing values in each column:
 Member_number      0
Date               0
itemDescription    0
Month              0
dtype: int64
```

18.Find how many transactions happened per month.

```
import pandas as pd
import numpy as np

transactions_per_month = df.groupby(df['Date'].dt.month).size()
print(transactions_per_month)
```

```
Date
1     3324
2     2997
3     3133
4     3260
5     3408
6     3264
7     3300
8     3496
9     3059
10    3261
11    3254
12    3009
dtype: int64
```

19.Find the top 3 months where "whole milk" was sold most.

```
import pandas as pd
import numpy as np

whole_milk_months = df[df['itemDescription'] == 'whole milk'].groupby(df['Date'].dt.month).size().sort_values(ascending=False).head(3)
print(whole_milk_months)
```

```
Date
8     236
4     234
11    228
dtype: int64
```

20.Create a new column 'Month' from 'Date' and find the busiest month.

```
import pandas as pd
import numpy as np

busiest_month = df['Month'].value_counts().idxmax()
print("Busiest month:", busiest_month)
```

```
Busiest month: 8
```

## Thank You