Quantium Task 2

In [1]:

```
## Importng Necessary Libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
#load the dataset sang hu tere
dataset = pd.read_csv(r"C:\Users\user\Downloads\Quantium\QVI_data.csv")
dataset.head()
```

Out[2]:

	LYLTY_CARD_NBR	DATE	STORE_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	PACK_SIZE	BRAND	LIFESTAGE	PREMIUM_CUSTOMER
0	1000	17-10-2018	1	1	5	Natural Chip Compny SeaSalt175g	2	6.0	175	NATURAL	YOUNG SINGLES/COUPLES	Premium
1	1002	16-09-2018	1	2	58	Red Rock Deli Chikn&Garlic Aioli 150g	1	2.7	150	RRD	YOUNG SINGLES/COUPLES	Mainstream
2	1003	07-03-2019	1	3	52	Grain Waves Sour Cream&Chives 210G	1	3.6	210	GRNWVES	YOUNG FAMILIES	Budget
3	1003	08-03-2019	1	4	106	Natural ChipCo Hony Soy Chckn175g	1	3.0	175	NATURAL	YOUNG FAMILIES	Budget
4	1004	02-11-2018	1	5	96	WW Original Stacked Chips 160g	1	1.9	160	WOOLWORTHS	OLDER SINGLES/COUPLES	Mainstream

Information about dataset

In [23]:

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 264834 entries, 0 to 264833
Data columns (total 12 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   LYLTY_CARD_NBR    264834 non-null  int64  
 1   DATE              264834 non-null  object 
 2   STORE_NBR         264834 non-null  int64  
 3   TXN_ID            264834 non-null  int64  
 4   PROD_NBR          264834 non-null  int64  
 5   PROD_NAME         264834 non-null  object 
 6   PROD_QTY          264834 non-null  int64  
 7   TOT_SALES         264834 non-null  float64
 8   PACK_SIZE         264834 non-null  int64  
 9   BRAND             264834 non-null  object 
 10  LIFESTAGE         264834 non-null  object 
 11  PREMIUM_CUSTOMER  264834 non-null  object 
dtypes: float64(1), int64(6), object(5)
memory usage: 24.2+ MB
```

In [32]:

```
# Calculate total sales
Total_sales_revenue = dataset['TOT_SALES'].sum()
print(f"Total Sales Revenue: {Total_sales_revenue}")
```

```
Total Sales Revenue: 1933114.9999999998
```

In [10]:

```
dataset.describe()
```

Out[10]:

	LYLTY_CARD_NBR	STORE_NBR	TXN_ID	PROD_NBR	PROD_QTY	TOT_SALES	PACK_SIZE
count	2.648340e+05	264834.000000	2.648340e+05	264834.000000	264834.000000	264834.000000	264834.000000
mean	1.355488e+05	135.079423	1.351576e+05	56.583554	1.905813	7.299346	182.425512
std	8.057990e+04	76.784063	7.813292e+04	32.826444	0.343436	2.527241	64.325148
min	1.000000e+03	1.000000	1.000000e+00	1.000000	1.000000	1.500000	70.000000
25%	7.002100e+04	70.000000	6.760050e+04	28.000000	2.000000	5.400000	150.000000
50%	1.303570e+05	130.000000	1.351365e+05	56.000000	2.000000	7.400000	170.000000
75%	2.030940e+05	203.000000	2.026998e+05	85.000000	2.000000	9.200000	175.000000
max	2.373711e+06	272.000000	2.415841e+06	114.000000	5.000000	29.500000	380.000000

- As there is nothing related to customer details like customer_id to calculate total number of customers, we can select 'LYLTY_CARD_NBR' to count total no. of customers.

In [6]:

```
dataset.nunique()
```

Out[6]:

```
LYLTY_CARD_NBR       72636
DATE                   364
STORE_NBR              272
TXN_ID              263125
PROD_NBR               114
PROD_NAME              114
PROD_QTY                 5
TOT_SALES              111
PACK_SIZE               21
BRAND                   21
LIFESTAGE                7
PREMIUM_CUSTOMER         3
dtype: int64
```

In [7]:

```
# total number of customers
Total_Customers = dataset['LYLTY_CARD_NBR'].nunique()
print(f"total number of customers: {Total_Customers}")
```

```
total number of customers: 72636
```

In [9]:

```
# total number of transaction
Total_transaction = dataset['TXN_ID'].count()
print(f"total number of transaction: {Total_transaction}")
```

```
total number of transaction: 264834
```

In [10]:

```
# average number of transactions per customer
transactions_per_customers = Total_transaction/Total_Customers
print(f"average number of transactions per customer: {transactions_per_customers}")
```

```
average number of transactions per customer: 3.646043284321824
```