Quantium Data Analytics Virtual Internship : - Task 2

Importing Libraries

In [2]:

```
# Data Manipulation Library
import numpy as np
import pandas as pd

# Data Visualization Library
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as pe
```

In [3]:

```
df = pd.read_csv("E:\Virtual Internship Data Analytics\Quantium Virtual Internship\Task 2\Task 2 Dataset\QVI_da
```

In [6]:

```
df.head()
```

Out[6]:

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

In [ ]:

Checking Data Types and changing Data types according to fields

In [7]:

```
df.dtypes
```

Out[7]:

```
LYLTY_CARD_NBR        int64
DATE                 object
STORE_NBR             int64
TXN_ID                int64
PROD_NBR              int64
PROD_NAME            object
PROD_QTY              int64
TOT_SALES           float64
PACK_SIZE             int64
BRAND                object
LIFESTAGE            object
PREMIUM_CUSTOMER     object
dtype: object
```

In [13]:

```
df['DATE'] = pd.to_datetime(df['DATE'])
```

```
In [15]: df.dtypes
```

```
Out[15]: LYLTY_CARD_NBR                int64
         DATE                 datetime64[ns]
         STORE_NBR                     int64
         TXN_ID                        int64
         PROD_NBR                      int64
         PROD_NAME                    object
         PROD_QTY                      int64
         TOT_SALES                   float64
         PACK_SIZE                     int64
         BRAND                        object
         LIFESTAGE                    object
         PREMIUM_CUSTOMER             object
         dtype: object
```

```
In [16]: df.head()
```

Out[16]:

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

Checking for Null Values

```
In [5]: df.isnull().sum()
```

```
Out[5]: LYLTY_CARD_NBR      0
        DATE                0
        STORE_NBR           0
        TXN_ID              0
        PROD_NBR            0
        PROD_NAME           0
        PROD_QTY            0
        TOT_SALES           0
        PACK_SIZE           0
        BRAND               0
        LIFESTAGE           0
        PREMIUM_CUSTOMER    0
        dtype: int64
```

```
In [ ]:
```

Checking for Outliers

In [18]:
```
df.select_dtypes(["int","float"])
```

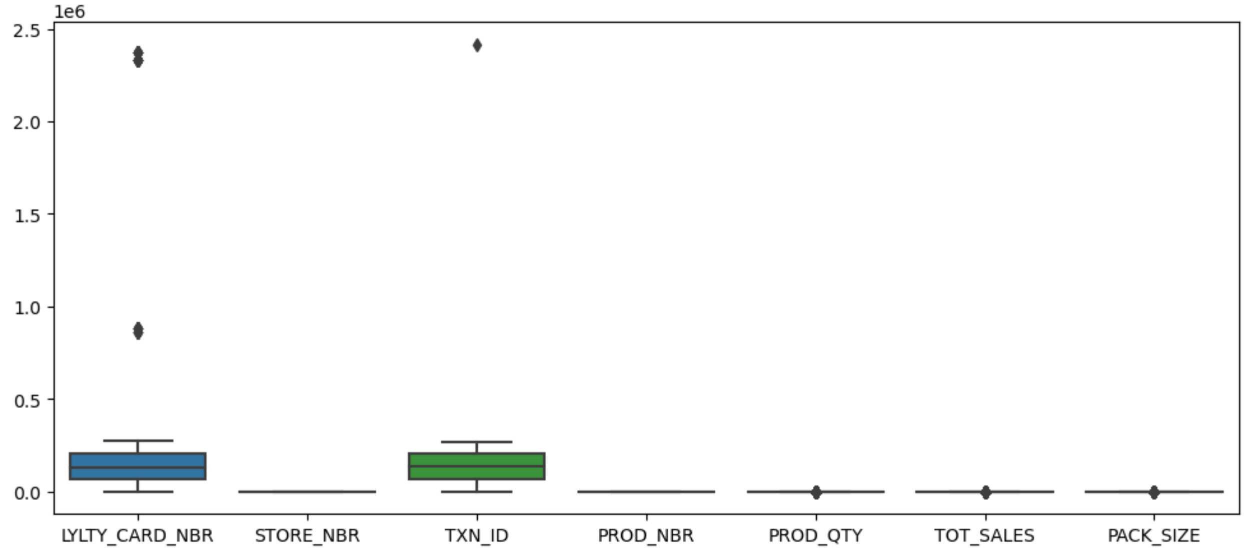
Out[18]:

	LYLTY_CARD_NBR	STORE_NBR	TXN_ID	PROD_NBR	PROD_QTY	TOT_SALES	PACK_SIZE
0	1000	1	1	5	2	6.0	175
1	1002	1	2	58	1	2.7	150
2	1003	1	3	52	1	3.6	210
3	1003	1	4	106	1	3.0	175
4	1004	1	5	96	1	1.9	160
...	...	...	...	...	...	...	...
264829	2370701	88	240378	24	2	7.2	210
264830	2370751	88	240394	60	2	9.2	150
264831	2370961	88	240480	70	2	8.4	165
264832	2370961	88	240481	65	2	10.2	300
264833	2373711	88	241815	16	2	11.4	330

264834 rows × 7 columns

In [25]:
```
plt.figure(figsize = (12,5))
sns.boxplot(df.select_dtypes(["int","float"]))
```

Out[25]:
```
<Axes: >
```



In [ ]:
```

```

Total Sales Revenue

```
In [30]: df.head()
```

Out[30]:

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

```
In [37]: round(df.TOT_SALES.sum(),2)

Out[37]: 1933115
```

```
In [48]: print("\033[1m  Total Sales {} $  \033[1m ".format(round(df.TOT_SALES.sum(),2)) )

  Total Sales 1933115.0 $
```

```
In [ ]:
```

Total Number Of Customers

```
In [49]: df.head()
```

Out[49]:

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

```
In [ ]:
```

```
In [53]: df["LYLTY_CARD_NBR"].nunique()

Out[53]: 72636
```

```
In [55]: print("\033[1m Total Numbers of Customers {}. \033[0m".format(df["LYLTY_CARD_NBR"].nunique()))

 Total Numbers of Customers 72636.
```

In [ ]:

Average Number Of Transactions Per Customer

In [57]:
```
df.head()
```

Out[57]:

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

In [94]:
```
avg_cust = df[["TOT_SALES"]].groupby(df["LYLTY_CARD_NBR"]).mean().sort_values("TOT_SALES",ascending= False)
avg_cust
```

Out[94]:

	TOT_SALES
LYLTY_CARD_NBR	
44350	29.5
154199	28.5
66014	28.5
60198	27.0
149267	27.0
...	...
186138	1.5
176370	1.5
254180	1.5
41089	1.5
35191	1.5

72636 rows × 1 columns

In [89]:
```
df[['TOT_SALES']].groupby(df['LYLTY_CARD_NBR']).count().mean()
```

Out[89]:
```
TOT_SALES    3.646043
dtype: float64
```

In [93]:
```
print("\033[1m Average transactions per customer: {} \033[0m".format(round(average_transactions_per_customer))
```

```
 Average transactions per customer: 4
```

In [ ]:

In [ ]: