Exploratory Data Analysis

Loading the necessary packages and modules

```
# Importing important pacakges and modules
import pandas as pd
import numpy as np
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
import matplotlib.pyplot as plt
```

Loading the Datasets

```
transaction_data = pd.read_excel('C:/Users/LENOVO/Downloads/QVI_transaction_data.xlsx')
```

```
print('Shape of transaction data :', transaction_data.shape)
```

```
Shape of transaction data : (264836, 8)
```

```
behave_data = pd.read_csv('C:/Users/LENOVO/Downloads/QVI_purchase_behaviour.csv')
```

```
print('Shape of customer purchase data :', behave_data.shape)
```

```
Shape of customer purchase data : (72637, 3)
```

Both the datasets have been read, now its time to clean the data

The date cell was noted and was converted to 'datetime' through excel. By searching online, we got to know that dates in excel and csv files start from 1899-12-30.

Exploring the Customer Transaction dataset

Now, we will get the basic details of the database i.e, Null Values, Unique Values and, dtypes of the columns.

```
def basic_details(df):
    b = pd.DataFrame()
    b['Missing value'] = df.isnull().sum()
    b['N unique value'] = df.nunique()
    b['dtype'] = df.dtypes
    return b
```

	Missing value	N unique value	dtype
DATE	0	364	datetime64[ns]
STORE_NBR	0	272	int64
LYLTY_CARD_NBR	0	72637	int64
TXN_ID	0	263127	int64
PROD_NBR	0	114	int64
PROD_NAME	0	114	object
PROD_QTY	0	6	int64
TOT_SALES	0	112	float64

Filtering and Cleaning the dataset

Let's filter only those rows which have 'Chips' and 'Nachos' in their product name

```
d = transaction_data.PROD_NAME.str
```

```
transaction = transaction_data[(d.contains('Chips'))|(d.contains('Chps'))|(d.contains('Nachos'))|(d.contains('chip'))|
                               (d.contains('chips'))|(d.contains('nacho'))|(d.contains('Nacho'))|(d.contains('nachos'))|
                               (d.contains('Chip'))|(d.contains('Chp'))|(d.contains('chps'))|(d.contains('chp'))]
```

Now the database has been filtered only for products with Chips and Nachos.

Lets's extract and clean the Brand Name for all the products

```
transaction['BRAND'] = transaction['PROD_NAME'].apply(lambda x: x.split()[:3])
transaction['BRAND'] = transaction['BRAND'].apply(lambda y: ' '.join(y))
```

```
transaction['BRAND'] = transaction['BRAND'].replace({'Natural Chip Compny':'Natural Chip Company', 'CCs Nacho Cheese':'CCs',
       'Smiths Chip Thinly':'Smiths', 'Kettle Tortilla ChpsHny&Jlpno':'Kettle',
       'Smiths Crinkle Chips':'Smiths', 'Doritos Corn Chip':'Doritos', 'Thins Chips Light&':'Thins Chips',
       'Thins Chips Originl':'Thins Chips', 'Natural ChipCo Hony':'Natural Chip Company', 'Dorito Corn Chp':'Doritos',
       'Thins Chips Seasonedchicken':'Thins Chips', 'Doritos Corn Chips':'Doritos',
       'Cobs Popd Swt/Chlli':'Cobs', 'Natural Chip Co':'Natural Chip Company', 'Cobs Popd Sea':'Cobs',
       'French Fries Potato':'French Fries', 'WW Original Corn':'WW', 'Thins Potato Chips':'Thins Chips',
       'Cobs Popd Sour':'Cobs', 'Smiths Crnkle Chip':'Smiths', 'WW D/Style Chip':'WW',
       'WW Original Stacked':'WW', 'Thins Chips Salt':'Thins Chips',
       'Kettle Tortilla ChpsBtroot&Ricotta':'Kettle', 'Tostitos Smoked Chipotle':'Tostitos',
       'WW Supreme Cheese':'WW', 'Kettle Tortilla ChpsFeta&Garlic':'Kettle',
       'WW Sour Cream':'WW', 'Natural ChipCo Sea':'Natural Chip Company'})
```

Finally, we have 10 brands for Chips products

```
transaction['BRAND'].unique()
```

```
y(['Natural Chip Company', 'CCs', 'Smiths', 'Kettle', 'Doritos',
   'Thins Chips', 'Cobs', 'French Fries', 'WW', 'Tostitos'],
  dtype=object)
```

Let's check for outliers in features "Product Quantity, Date, Pack Size"

```
transaction['PROD_QTY'].value_counts()
```

```
2      79236
1       9182
5        149
3        144
4        116
200        2
Name: PROD_QTY, dtype: int64
```

We can see that there are only 2 data inputs for buy-outs when the product quantity is 200.

```
transaction.loc[transaction['PROD_QTY'] == 200, :]
```

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	BRAND
69762	2018-08-19	226	226000	226201	4	Dorito Corn Chp Supreme 380g	200	650.0	Doritos
69763	2019-05-20	226	226000	226210	4	Dorito Corn Chp Supreme 380g	200	650.0	Doritos

This shows that the same customer purchased 200 packets of chips on different days from the same store which can tell us that the customer might be from a retail business. Let's see if this customer makes other purchases or not.

```
1 transaction.loc[transaction['LYLTY_CARD_NBR'] == 226000, :]
```

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	BRAND
69762	2018-08-19	226	226000	226201	4	Dorito Corn Chp Supreme 380g	200	650.0	Doritos
69763	2019-05-20	226	226000	226210	4	Dorito Corn Chp Supreme 380g	200	650.0	Doritos

Therefore, we can see that this customer hasn;t bought any other products, thus, we can drop the details for the same

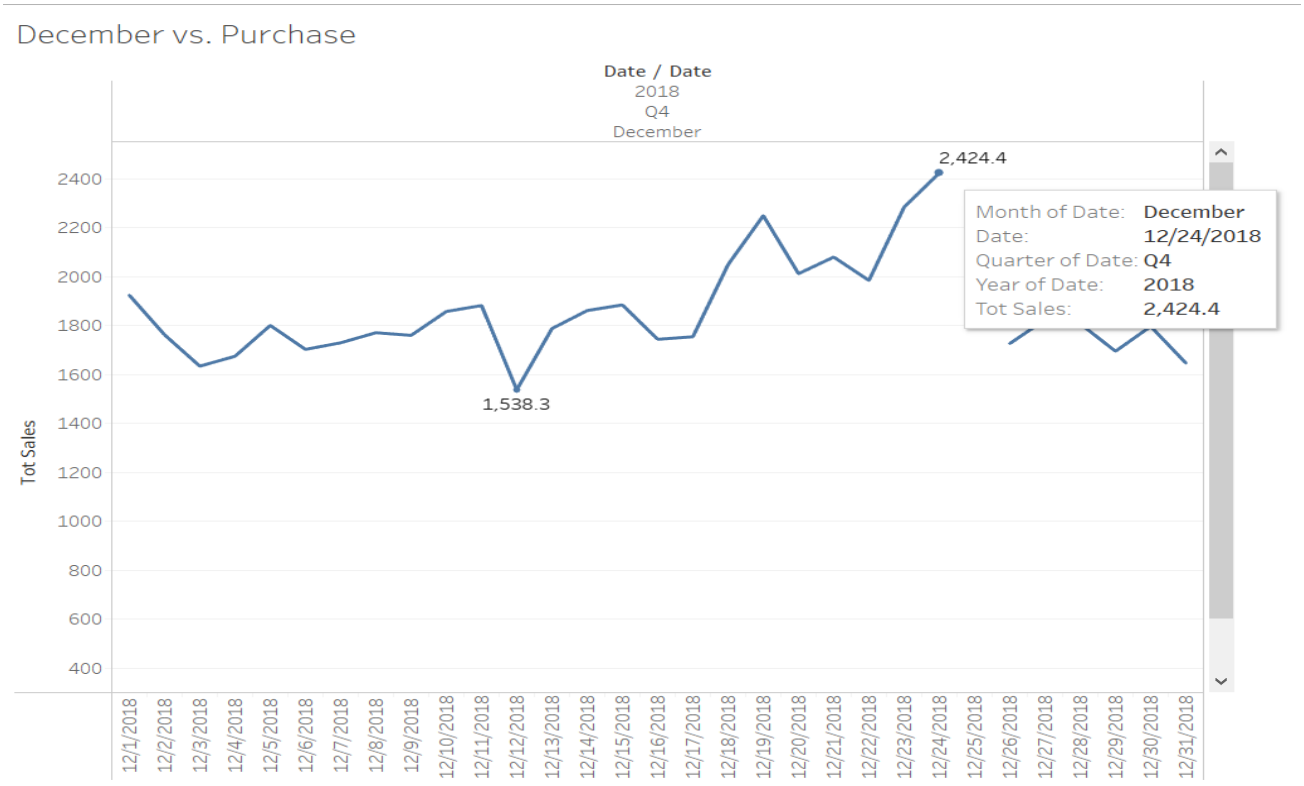
```
transaction = transaction[transaction.LYLTY_CARD_NBR != 226000]
```

As the sales are given for 1st July 2018 to 30th June 2019; a year length. Let's analyze it.

```
1 len(transaction['DATE'].unique())
```

```
364
```

Therefore, we can see that there is one day on which no purchases were made.



We can see that purchases were not made on 25th December 2018; Christmas as shops will be preferably closed.

Now, let's extract the pack sie from the product description

```
transaction['PACK_SIZE'] = transaction['PROD_NAME'].apply(lambda x: x.split()[-1:])
transaction['PACK_SIZE'] = transaction['PACK_SIZE'].apply(lambda y: ''.join(y))
```

Cleaning the packet size of the chip products

```
transaction['PACK_SIZE'].unique()
```

```
array(['175g', '170g', '150g', '330g', '380g', '110g', '200g', '160g'],
      dtype=object)
```

```
transaction['PACK_SIZE'] = transaction['PACK_SIZE'].replace({'SeaSalt175g':'175g', 'Chs&Onion170g':'170g',
                                                             'CutSalt/Vinegr175g':'175g', 'Chckn175g':'175g'})
```

The Customer Transaction dataset is now clean and we can move onto the Customer-Purchase Behaviour dataset.

Exploring the Customer-Purchase Behaviour dataset

```
basic_details(behave_data)
```

	Missing value	N unique value	dtype
LYLTY_CARD_NBR	0	72637	int64
LIFESTAGE	0	7	object
PREMIUM_CUSTOMER	0	3	object

```
for col in ['LIFESTAGE',  'PREMIUM_CUSTOMER']:
    print(behave_data[col].value_counts(),'\n')
```

```
RETIREES                14805
OLDER SINGLES/COUPLES   14609
YOUNG SINGLES/COUPLES   14441
OLDER FAMILIES           9780
YOUNG FAMILIES           9178
MIDAGE SINGLES/COUPLES   7275
NEW FAMILIES             2549
Name: LIFESTAGE, dtype: int64

Mainstream    29245
Budget        24470
Premium       18922
Name: PREMIUM_CUSTOMER, dtype: int64
```

There are no discrepancies in the dataset and there is no chance of any outliers.

Let's merge the Transaction and Purchase behavior data frames and dive to further analysis.

Data analysis on customer segments

Now that the data is ready for analysis, we can define some metrics of interest to the client:

- Who spends the most on chips (total sales), describing customers by lifestage and how premium their general purchasing behaviour is
- How many customers are in each segment
- How many chips are bought per customer by segment
- What's the average chip price by customer segment

We could also ask our data team for more information. Examples are:

Total Sales by Customer group

Lifestage	Premium Customer Budget	Mainstream	Premium
MIDAGE SINGLES/COUPLES	12,248	30,233	19,157
NEW FAMILIES	7,143	5,670	3,747
OLDER FAMILIES	55,397	35,109	26,147
OLDER SINGLES/COUPLES	45,217	44,498	43,721
RETIREES	37,648	51,106	31,696
YOUNG FAMILIES	46,246	31,258	28,182
YOUNG SINGLES/COUPLES	20,584	51,745	14,241

We can observe that the highest sales are given by "**Older Families**" who are **Budget** customers of **55,397** and "**Retirees and Young Singles/Couples**" in **Mainstream** customers of **>51,000**. These 3 lots contribute most to the sale of chips followed by Budget Young Families.

Segemnted number of Customers by Product Quantity

	Premium Customer		
Lifestage	Budget	Mainstream	Premium
MIDAGE SINGLES/COUPLES	3,315	7,713	5,150
NEW FAMILIES	1,857	1,453	975
OLDER FAMILIES	14,953	9,476	7,149
OLDER SINGLES/COUPLES	11,797	11,765	11,408
RETIREES	9,730	13,424	8,177
YOUNG FAMILIES	12,458	8,455	7,575
YOUNG SINGLES/COUPLES	5,692	12,851	3,922

Customer segments of "**Budget – Older Families**" and "**Mainstream – Retirees & Young Singles/Couples**" buy the highest number of products, **14,953** and **avg. 13,000**. This can be correlated with why these are the same customer segments which bring in the most amount of sales with each of them having the **number of customers in the range 3500 – 4000**.

Average Product Ratio

Lifestage	Budget	Premium Customer Mainstream	Premium
MIDAGE SINGLES/COUPLES	3.393	3.473	3.276
NEW FAMILIES	2.822	2.794	2.762
OLDER FAMILIES	4.382	4.451	4.341
OLDER SINGLES/COUPLES	3.447	3.492	3.489
RETIREES	3.233	3.162	3.193
YOUNG FAMILIES	4.266	4.238	4.302
YOUNG SINGLES/COUPLES	2.677	2.819	2.729

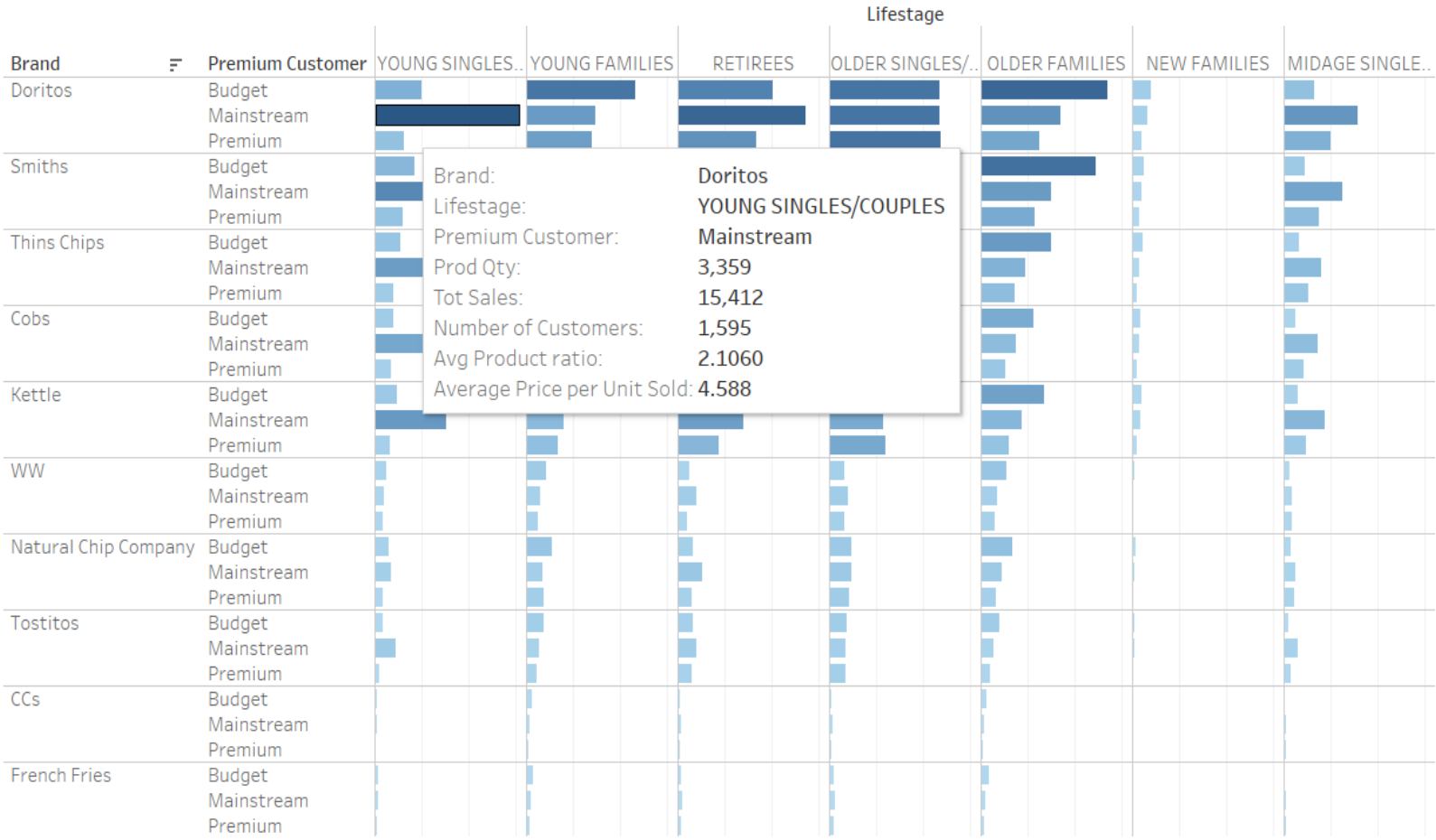
"**Older Families**" and "**Younger Families**" buy more chips products than others. Although the product quantity is low from "**Younger Families" but** their **Average product ratio per customer** is high, due to which, they **contribute 46,246 to sales** of chips.

Average Price per Unit by Cust Segment

Lifestage	Budget	Premium Customer Mainstream	Premium
MIDAGE SINGLES/COUPLES	3.6948	3.9197	3.7199
NEW FAMILIES	3.8467	3.9021	3.8435
OLDER FAMILIES	3.7047	3.7050	3.6574
OLDER SINGLES/COUPLES	3.8329	3.7823	3.8325
RETIREES	3.8693	3.8071	3.8762
YOUNG FAMILIES	3.7122	3.6970	3.7204
YOUNG SINGLES/COUPLES	3.6163	4.0266	3.6310

"**Mainstream Mid-Age**" and "**Young Singles/Couples**" are willing to pay more per packet of chips compared to their budget and premium counterparts. This may be due to premium shoppers being more likely to buy healthy snacks and when they buy chips, this is mainly for entertainment purposes rather than their consumption.

Brand Overview



When we look into brand and customer segmentation, the most sought out brand is "**Doritos**" which is popular among Mainstream customers and "**Budget – Older Families**". We can see that "**New Families**" is the customer segment that buys "chips" the **least**. This could be the case because they are more health-conscious than their counterparts due to several reasons.

Number of Customers

Lifestage	Premium Customer: Budget	Mainstream	Premium
MIDAGE SINGLES/COUPLES	977	2,221	1,572
NEW FAMILIES	658	520	353
OLDER FAMILIES	3,412	2,129	1,647
OLDER SINGLES/COUPLES	3,422	3,369	3,270
RETIREES	3,010	4,245	2,561
YOUNG FAMILIES	2,920	1,995	1,761
YOUNG SINGLES/COUPLES	2,126	4,558	1,437

"**Mainstream – Retirees**" and "**Young Singles/Couples**", and "**Budget – Older Singles/Couples**" and "**Older Families**" are the major customers when it comes to buying chips.

Pack Size

Lifestage	Premium Custo..	Pack Size: 110g	150g	160g	170g	175g	200g	330g	380g
YOUNG SINGLES/COUPLES	Budget	528	860	261	1,010	2,159	370	170	334
	Mainstream	1,617	2,722	232	2,259	3,949	325	582	1,165
	Premium	446	604	201	635	1,403	254	113	266
YOUNG FAMILIES	Budget	1,185	2,143	438	2,233	4,459	715	439	846
	Mainstream	896	1,460	318	1,500	2,978	479	275	549
	Premium	851	1,237	257	1,297	2,738	435	258	502
RETIREES	Budget	1,143	1,916	277	1,676	3,205	397	389	727
	Mainstream	1,488	2,530	408	2,347	4,626	621	462	942
	Premium	971	1,522	206	1,373	2,785	330	288	702
OLDER SINGLES/COUPLES	Budget	1,337	2,245	350	2,017	4,009	524	439	876
	Mainstream	1,347	2,093	404	2,075	3,954	624	390	878
	Premium	1,214	2,205	355	1,923	3,931	502	431	847
OLDER FAMILIES	Budget	1,489	2,455	594	2,687	5,313	863	559	993
	Mainstream	981	1,582	327	1,615	3,428	576	333	634
	Premium	695	1,113	313	1,243	2,579	490	257	459
NEW FAMILIES	Budget	231	344	56	325	626	74	55	146
	Mainstream	184	320	40	231	463	50	39	126
	Premium	127	185	27	158	301	65	39	73
MIDAGE SINGLES/COUPLES	Budget	345	588	135	589	1,157	193	88	220
	Mainstream	951	1,549	192	1,306	2,487	256	369	603
	Premium	545	890	213	860	1,846	280	131	385

Ways to Increase Productivity and Proficiency

The category manager may want to increase the category's performance by off-locating some Doritos and smaller packs of chips in the discretionary area near segments where Young Singles/Couples frequent more often to increase visibility and impulse behavior.

Quantium can help the Category Manager with recommendations of where these segments are and further help them with measuring the impact of the changed placement.